

Document Title	Specification of a Transport Layer for SAE J1939
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	425
Document Classification	Standard

Document Version	1.2.0
Document Status	Final
Part of Release	4.0
Revision	3

Document Change History			
Date	Version	Changed by	Change Description
30.09.2011	1.2.0	AUTOSAR Administration	API changes: CancelTransmit, CancelReceive, and ChangeParameter were changed to synchronous behavior
15.10.2010	1.1.0	AUTOSAR Administration	<ul style="list-style-type: none">Fixed service API IDsRemoved duplicate requirement J1939TP0099Clarified requirements J1939TP0125 and J1939TP0189
30.11.2009	1.0.0	AUTOSAR Administration	Initial release

Disclaimer

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

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.

For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

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

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

Table of Contents

1	Introduction and Functional Overview	6
2	Glossary, Acronyms, and Abbreviations	8
3	Related Documentation	10
3.1	Input Documents	10
3.2	Related Standard Documents	11
4	Constraints and Assumptions	12
4.1	Limitations	12
4.2	Applicability to Automotive Domains	12
5	Module Architecture	13
5.1	Dependencies on Other Modules	13
5.2	File Hierarchy	14
6	Requirements Traceability	17
7	Functional Specification	21
7.1	Basic Principles of SAE J1939	21
7.2	Basic Functionality of J1939Tp	22
7.3	Services Provided to the Upper Layer	23
7.3.1	Initialization and Shutdown	24
7.3.2	Communication Services	24
7.4	Services Provided to the Lower Layer	25
7.4.1	Transmission Confirmation	25
7.4.2	Reception Indication	25
7.5	Internal Behavior	25
7.5.1	Session Handling	25
7.5.1.1	Close Transmission	25
7.5.1.2	Abort Transmission	26
7.5.1.3	Close Reception	26
7.5.1.4	Abort Reception	26
7.5.1.5	Abort CMDT Connection	26
7.5.2	N-SDU Reception	26
7.5.3	N-SDU Transmission	27
7.5.4	Data Flow on the CAN Bus	28
7.5.4.1	Data Flow using Direct Transmission	28
7.5.4.2	Data Flow using CMDT	29
7.5.4.3	Data Flow using BAM	30
7.5.5	Interface between J1939Tp and PduR	31
7.5.5.1	Direct Transmission	32
7.5.5.2	Direct Reception	32
7.5.5.3	Segmented Transmission	33
7.5.5.4	Segmented Reception	34
7.5.6	Interface between J1939Tp and CanIf	34
7.5.7	Relationship between N-SDU and N-PDU in J1939Tp	37

7.5.8	Concurrent Connections	38
7.5.9	N-PDU Padding.....	38
7.5.10	Handling of Unexpected N-PDU Arrivals.....	39
7.6	Error Classification	39
7.7	Error Detection	40
7.8	Error Notification.....	40
7.9	Debugging of J1939Tp	42
8	API Specification	43
8.1	Imported Types	43
8.2	Type Definitions.....	43
8.3	Function Definitions.....	44
8.3.1	J1939Tp_Init	44
8.3.2	J1939Tp_Shutdown	45
8.3.3	J1939Tp_GetVersionInfo	46
8.3.4	J1939Tp_Transmit	46
8.3.5	J1939Tp_CancelTransmit	47
8.3.6	J1939Tp_CancelReceive	48
8.3.7	J1939Tp_ChangeParameter	48
8.4	Callback Notifications	49
8.4.1	J1939Tp_RxIndication	49
8.4.2	J1939Tp_TxConfirmation	50
8.5	Scheduled Functions.....	50
8.5.1	J1939Tp_MainFunction.....	50
8.6	Expected Interfaces.....	51
8.6.1	Mandatory Interfaces	51
8.6.2	Optional Interfaces	52
9	Sequence Diagrams	53
10	Configuration Specification	54
10.1	How to Read this Chapter	54
10.1.1	Configuration and Configuration Parameters	54
10.1.2	Variants.....	54
10.1.3	Containers.....	55
10.1.4	Specification Template for Configuration Parameters	55
10.2	Containers and Configuration Parameters	56
10.2.1	J1939Tp	61
10.2.2	J1939TpConfiguration	61
10.2.3	J1939TpDemEventParameterRefs	61
10.2.4	J1939TpRxChannel	62
10.2.5	J1939TpRxCmNPdu	63
10.2.6	J1939TpRxDtNPdu	64
10.2.7	J1939TpRxPg	64
10.2.8	J1939TpRxDirectNPdu	65
10.2.9	J1939TpRxNSdu.....	66
10.2.10	J1939TpTxFcNPdu.....	66
10.2.11	J1939TpTxChannel	67
10.2.12	J1939TpRxFcNPdu	68
10.2.13	J1939TpTxCmNPdu	69
10.2.14	J1939TpTxDtNPdu	70

10.2.15	J1939TpTxPg	71
10.2.16	J1939TpTxDirectNPdu	71
10.2.17	J1939TpTxNSdu.....	72
10.2.18	J1939TpGeneral.....	73
10.3	Published Information.....	75
11	Not applicable requirements	76

1 Introduction and Functional Overview

This specification describes the functionality, the API, and the configuration of the AUTOSAR Basic Software module J1939Tp, which implements an SAE J1939 compatible transport layer for AUTOSAR. The terms J1939Tp and J1939 transport layer are used synonymously in this document.

SAE J1939 has a broad acceptance in the truck domain, and consists of several documents describing the layers of the communication protocol from the physical layer to diagnostics and the application layer. SAE J1939-21 describes the data link and transport layer, which includes two transport protocol variants:

- BAM (Broadcast Announce Message) for broadcast messages, and
- CMDT (Connection Mode Data Transfer) for point-to-point connections.

AUTOSAR encourages the implementation of existing standards. Therefore, the AUTOSAR J1939 Transport Layer specification is tightly based on the SAE J1939 Data Link Layer standard document J1939-21 from December 2006. It is assumed that the reader has basic knowledge of that document.

The module J1939Tp resides functionally between the PDU Router and the CAN Interface, as shown in the following figure:

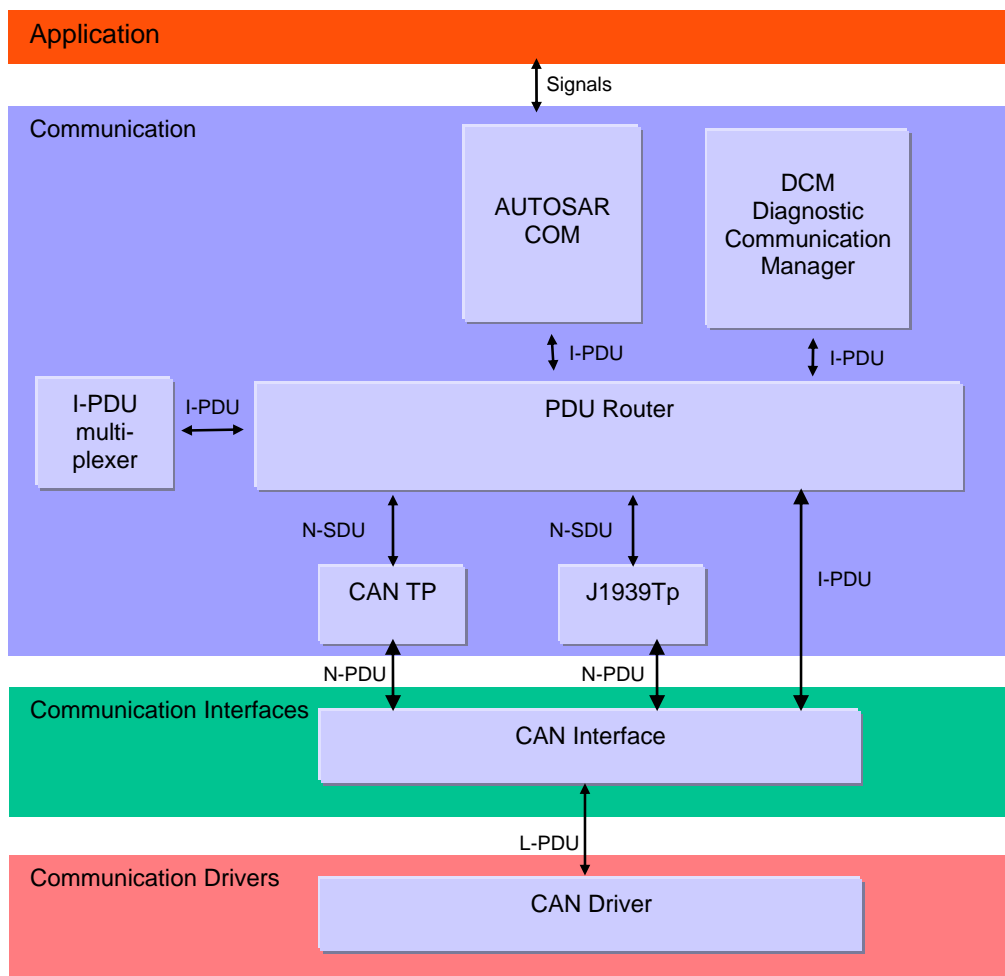


Figure 1-1 : AUTOSAR Communication Stack

The main purpose of J1939Tp is to segment and reassemble J1939 PGNs (represented by I-PDUs) that are longer than 8 bytes. Depending on the configuration, either the broadcast variant BAM or the point-to-point variant CMDT of the SAE J1939 transport protocol is used. The segmented messages are sent and received via the CAN Interface.

The PDU-Router (PduR) deploys AUTOSAR COM and DCM I-PDUs onto different communication protocols (e.g. CAN or J1939, segmented via transport layer, or directly). At runtime, the decision where to route I-PDUs is based on the I-PDU identifier. In the same way, the CanIf uses the N-PDU identifier to decide whether a received message must be processed by one of the available transport layer modules or may be forwarded directly to the PduR.

J1939Tp supports fixed and variable size I-PDUs with more than 8 bytes. I-PDUs that do not exceed 8 bytes are exchanged directly between PduR and CanIf.

Fixed size I-PDUs are always segmented, while variable size I-PDUs are only segmented when they exceed 8 bytes. J1939Tp forwards variable size I-PDUs with an actual size of 8bytes or less and a configured maximum size of more than 8 bytes directly to the CanIf.

J1939Tp provides the following functionality:

- Segmentation and direct transmission of data in transmit direction.
- Reassembling and direct reception of data in receive direction.
- Control of data flow.
- Detection of errors during segmentation or reassembly.

2 Glossary, Acronyms, and Abbreviations

The following table presents a glossary of J1939 specific terms. For all other terms, please check the AUTOSAR Glossary.

Glossary Term	Explanation
Transport Protocol	The SAE J1939 transport protocol is used for the segmented transmission of messages with more than 8 bytes of data. The transport protocol is defined in the network layer standard document (SAE J1939/21).
Parameter	A parameter is a signal of the SAE J1939 application layer. Parameters are uniquely identified by the SPN.
Parameter Group	A parameter group is a message of the SAE J1939 application layer. Each parameter group contains several parameters (signals), and is uniquely identified by the PGN.
Address Claiming	Address Claiming forms the network management of SAE J1939 defined in the standard document SAE J1939/81. Address claiming assigns a temporary 8-bit identifier to each ECU connected to one J1939 network. Within this network, the 8-bit identifier is unique. The 8-bit identifier is used as source and target address of parameter groups (messages) transferred via the J1939 network. The address claiming procedure is based on the exchange of AddressClaimed messages (PGN 00EE00).
Request Handling	Request and response handling is defined in the SAE J1939 network layer (SAE J1939/21). It uses a dedicated request message (PGN 00EA00) to trigger transmission of certain messages, or execution of certain actions. The response is either the requested parameter group, or a dedicated acknowledge message (PGN 00E800).
J1939 Diagnostics	The SAE J1939 diagnostic layer is defined in the standard document SAE J1939/73. The J1939 diagnostics is functionally similar to the UDS diagnostics, and has recently been extended to support OBD for emission relevant values.

The following table lists acronyms and abbreviations that are specific to the J1939 transport layer. For all other abbreviations, please check the AUTOSAR Glossary or the AUTOSAR BSW Module List.

Acronym / Abbreviation	Description
BAM	Broadcast Announce Message, broadcast variant of SAE J1939 transport protocol
CMDT	Connection Mode Data Transfer, peer-to-peer variant of SAE J1939 transport protocol
DA	Destination Address, part of the 29 bit identifier of SAE J1939 messages
DMx	Diagnostic messages of the SAE J1939 diagnostics layer
NAME	Unique 64 bit identifier of each ECU connected to an SAE J1939 network
PDUF	PDU Format, part of the 29 bit identifier of SAE J1939 messages which identifies the message and determines the layout of the 29 bit identifier
PDUS	PDU Specific, part of the 29 bit identifier of SAE J1939 messages which identifies broadcast messages which do not have a destination address
PG	Parameter Group, SAE J1939 term for a specific message layout
PGN	Parameter Group Number, unique identifier of an SAE J1939 parameter group
SA	Source Address, part of the 29 bit identifier of SAE J1939 messages
SPN	Suspect Parameter Number, unique identifier of an SAE J1939 parameter
TP.CM	Connection Management message (PGN 00EC00) used by SAE J1939 transport protocol
TP.CM/Abort	Connection Abort, terminates a CMDT transmission via SAE J1939 transport protocol
TP.CM/BAM	Broadcast Announce Message, initiates a BAM transmission via SAE J1939 transport protocol
TP.CM/CTS	Connection Mode Clear To Send, acknowledges and controls a CMDT transmission via SAE J1939 transport protocol
TP.CM/EOMAck	End Of Message Acknowledge, finalizes a CMDT transmission via SAE J1939 transport protocol
TP.CM/RTS	Connection Mode Request To Send, initiates a CMDT transmission via SAE J1939 transport protocol
TP.DT	Data Transfer message (PGN 00EB00) used by SAE J1939 transport protocol

3 Related Documentation

3.1 Input Documents

- [1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf
- [2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf
- [3] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf
- [4] Specification of COM
AUTOSAR_SWS_COM.pdf
- [5] Requirements on CAN
AUTOSAR_SRS_CAN.pdf
- [6] Specification of CAN Interface
AUTOSAR_SWS_CANInterface.pdf
- [7] Requirements on a Transport Layer for SAE J1939
AUTOSAR_SRS_SAEJ1939TransportLayer.pdf
- [8] Specification of PDU Router
AUTOSAR_SWS_PDURouter.pdf
- [9] Specification of BSW Scheduler
AUTOSAR_SWS_Scheduler.pdf
- [10] Specification of Diagnostic Event Manager
AUTOSAR_SWS_DiagnosticEventManager.pdf
- [11] Specification of Development Error Tracer
AUTOSAR_SWS_DevelopmentErrorTracer.pdf
- [12] Basic Software Module Description Template
AUTOSAR_SRS_BSWGeneral.pdf
- [13] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf
- [14] Specification of System Template
AUTOSAR_TPS_SystemTemplate.pdf
- [15] Specification of Memory Mapping
AUTOSAR_SWS_MemoryMapping.pdf

3.2 Related Standard Documents

- [16] SAE J1939-21(2006-12), Data Link Layer
- [17] SAE J1939-7x(2006-xx), Application Layer

4 Constraints and Assumptions

4.1 Limitations

The AUTOSAR architecture contains several communication system specific transport layers (J1939Tp, CanTp, FlexRayTp, etc.). All of these modules need to have identical APIs, with the exception of API functions for which the PduR has separate configuration abilities.

The J1939Tp module does not implement the TriggerTransmit API, because it is only needed for time triggered bus architectures.

J1939Tp provides the API functions for cancellation of active transmission (see 8.3.5) and reception (see 8.3.6), and to change parameters (see 8.3.7), but these are only implemented by dummy functions, because the corresponding functionality is currently not required by any use case for AUTOSAR J1939 ECUs.

The J1939Tp implements the transport layer functionality of J1939 with the limitations described in section 3.3 of [7]. They mostly concern handling of:

- Message Request
- Network Management
- Diagnostic Messages

4.2 Applicability to Automotive Domains

The J1939 transport layer supports the implementation of ECUs that shall be able to communicate in a J1939 network.

5 Module Architecture

5.1 Dependencies on Other Modules

This section shows the relations between the J1939Tp and other AUTOSAR basic software modules. Besides the standard modules DET, DEM, EcuM, and SchM, which have interfaces to all BSW modules, J1939Tp only interacts with the PduR and the CanIf. The interfaces of J1939Tp are thus identical to the interfaces of CanTp.

The figure below shows the interactions between J1939Tp, PduR, and CanIf.

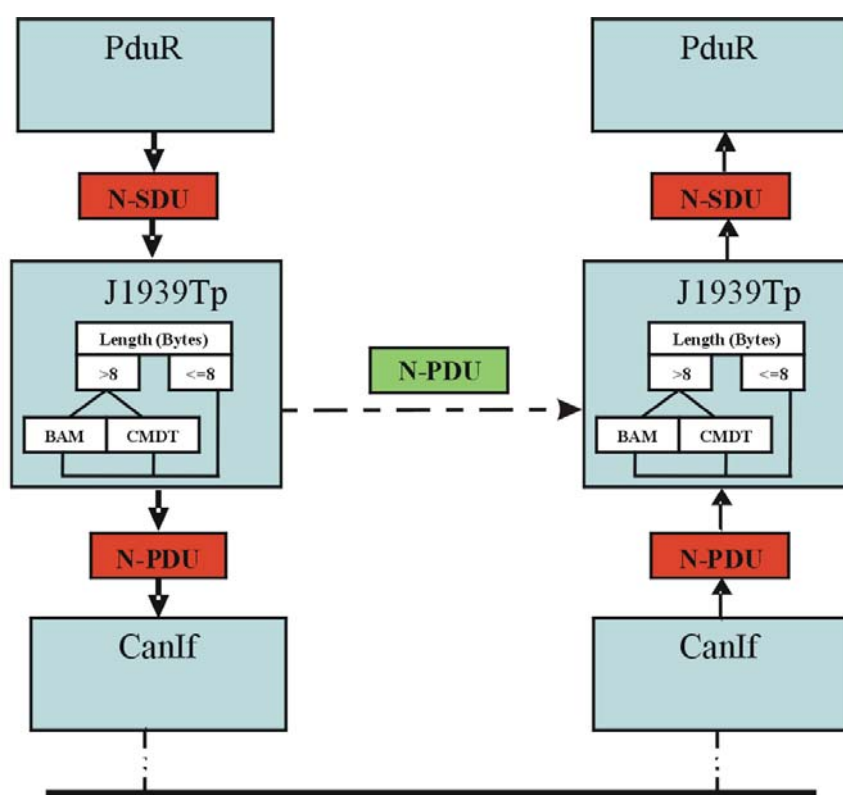


Figure 5-1: J1939 Transport Layer interactions

The J1939Tp's upper interface offers the PduR module access to transmitted and received N-SDUs corresponding to J1939 PGNs with a maximum length of more than 8 bytes of data.

J1939 PGNs with a maximum length of no more than 8 bytes are exchanged directly between PduR and CanIf.

5.2 File Hierarchy

AUTOSAR specifies that an ECU can be created from modules provided as object code, source code (generated or static), or both.

The decision to provide a module as object code or source code is based on a compromise between IP protection, test coverage, code efficiency and configurability at system generation time. Depending on the configurability requirements of the OEM, suppliers may deliver the J1939Tp module as object code or source code.

The file hierarchy defined in this section, which is shown in the following picture, allows the separation of platform, compiler, and implementation specific definitions and declarations from general definitions, as well as the separation of source code and configuration.

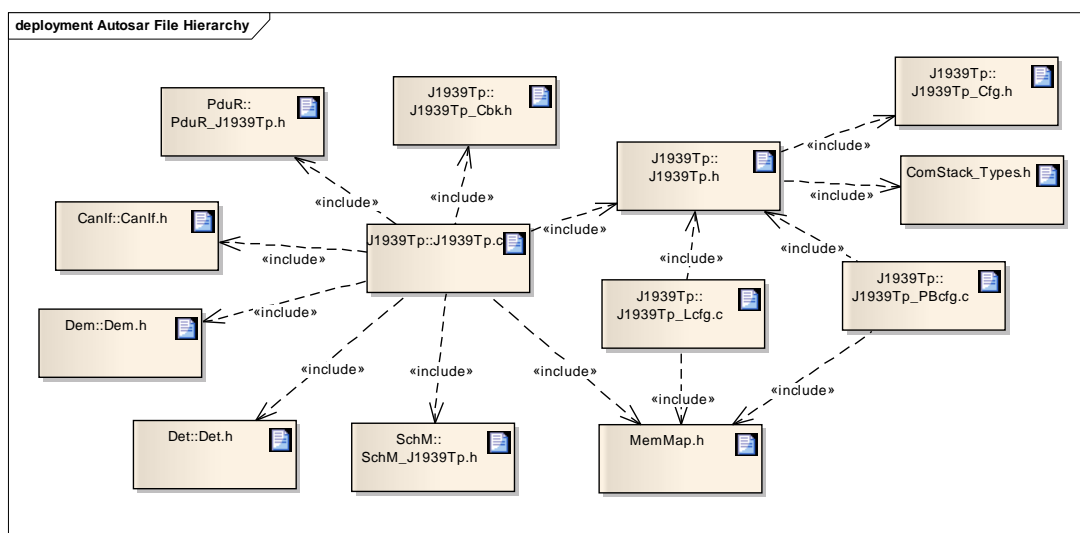


Figure 5-2: File Structure

[J1939TP0184] [

The implementation of the complete API including all service and callback functions of J1939Tp is placed in *J1939Tp.c*. Internal functionality like BAM and CMDT handling may be moved to additional implementation files.] ()

[J1939TP0007] [

Internally used data types and functions shall be defined locally in *J1939Tp.c*. This prevents visibility of these symbols outside of the J1939Tp module.] ()

[J1939TP0185] [

J1939Tp_Lcfg.c shall contain configuration dependent RAM and ROM tables that are defined at link time. This file may be generated.] (BSW00380)

[J1939TP0186] [

J1939Tp_PBcfg.c shall contain configuration dependent ROM tables that are defined at post-build time, respectively. This file may be generated.] (BSW00380)

[J1939TP0014] [

The J1939Tp module implementation files (*J1939Tp.c*, *J1939Tp_Lcfg.c*, and *J1939Tp_PBcfg.c*) shall include *MemMap.h* to implement memory abstraction according to [15].] (BSW00436)

[J1939TP0003] [

The module implementation files shall include the global header file *J1939Tp.h*.] ()

[J1939TP0005] [

J1939Tp.h shall contain all externally visible variable declarations, type definitions, function prototypes, and macro definitions. No internal symbols shall be exported via this include file.] ()

[J1939TP0012] [

J1939Tp.h shall export module and version identification information according to **J1939TP0196**.] ()

[J1939TP0016] [

J1939Tp.h shall include *ComStack_Types.h*, which in turn includes *Compiler.h* and *Std_Types.h* (not shown in Figure 5-2).] (BSW00348, BSW00353, BSW00361)

[J1939TP0009] [

J1939Tp.h shall include *J1939Tp_Cfg.h*.] ()

[J1939TP0008] [

J1939Tp_Cfg.h shall contain all configuration dependent type definitions, variable declarations, function prototypes, and macro definitions that are known at pre-compile time. This file may be generated.] (BSW00345, BSW00381, BSW00412)

[J1939TP0013] [

J1939Tp.c shall include *J1939Tp_Cbk.h*.] ()

[J1939TP0102] [

J1939Tp_Cbk.h shall provide the function prototypes of all callback functions of section 8.4.] (BSW00370)

[J1939TP0002] [

J1939Tp.c shall include *Dem.h*, which makes the service *Dem_ReportErrorStatus* available to J1939Tp. See also section 8.6.1.] (BSW00409)

[J1939TP0193] [

J1939Tp.c shall include *Det.h* to import the development error notification API. This API is optional; the header is included depending on **J1939TP0042_Conf** :. See also section 8.6.2.] ()

[J1939TP0010] [

J1939Tp.c shall include *SchM_J1939Tp.h* to implement module specific exclusive areas.] (BSW00435)

[J1939TP0015] [

J1939Tp.c shall include *PduR_J1939Tp.h*, which contains the callbacks functions of the PduR module that are used by the J1939Tp module. See also section 8.6.1.] ()

[J1939TP0172] [

J1939Tp.c shall include *CanIf.h*, which contains the service functions of the CanIf module that are used by the J1939Tp module. See also section 8.6.1.] ()

6 Requirements Traceability

Requirement	Satisfied by
-	J1939TP0012
-	J1939TP0193
-	J1939TP0040
-	J1939TP0165
-	J1939TP0152
-	J1939TP0174
-	J1939TP0098
-	J1939TP0159
-	J1939TP0194
-	J1939TP0190
-	J1939TP0192
-	J1939TP0072
-	J1939TP0100
-	J1939TP0032
-	J1939TP0118
-	J1939TP0020
-	J1939TP0163
-	J1939TP0045
-	J1939TP0195
-	J1939TP0079
-	J1939TP0021
-	J1939TP0176
-	J1939TP0007
-	J1939TP0180
-	J1939TP0119
-	J1939TP0181
-	J1939TP0048
-	J1939TP0038
-	J1939TP0120
-	J1939TP0178
-	J1939TP0179
-	J1939TP0022
-	J1939TP0005
-	J1939TP0076
-	J1939TP0173
-	J1939TP0161
-	J1939TP0097
-	J1939TP0191

-	J1939TP0043
-	J1939TP0015
-	J1939TP0009
-	J1939TP0094
-	J1939TP0116
-	J1939TP0075
-	J1939TP0047
-	J1939TP0126
-	J1939TP0124
-	J1939TP0085
-	J1939TP0035
-	J1939TP0031
-	J1939TP0175
-	J1939TP0177
-	J1939TP0073
-	J1939TP0057
-	J1939TP0172
-	J1939TP0013
-	J1939TP0154
-	J1939TP0039
-	J1939TP0046
-	J1939TP0106
-	J1939TP0101
-	J1939TP0036
-	J1939TP0162
-	J1939TP0030
-	J1939TP0026
-	J1939TP0189
-	J1939TP0071
-	J1939TP0003
-	J1939TP0060
-	J1939TP0184
-	J1939TP0095
-	J1939TP0160
-	J1939TP0041
BSW00305	J1939TP0086
BSW00314	J1939TP0197
BSW00323	J1939TP0017, J1939TP0188
BSW00325	J1939TP0197
BSW00326	J1939TP0197
BSW00327	J1939TP0115
BSW00333	J1939TP0110, J1939TP0114

BSW00335	J1939TP0019
BSW00336	J1939TP0093
BSW00337	J1939TP0115
BSW00338	J1939TP0077
BSW00339	J1939TP0078
BSW00341	J1939TP0197
BSW00345	J1939TP0008
BSW00347	J1939TP0197
BSW00348	J1939TP0016
BSW00353	J1939TP0016
BSW00357	J1939TP0096
BSW00358	J1939TP0087
BSW00359	J1939TP0112, J1939TP0108
BSW00360	J1939TP0112, J1939TP0108
BSW00361	J1939TP0016
BSW00370	J1939TP0102
BSW00373	J1939TP0104
BSW00375	J1939TP0197
BSW00376	J1939TP0104
BSW00377	J1939TP0197
BSW00380	J1939TP0186, J1939TP0185
BSW00381	J1939TP0008
BSW00385	J1939TP0115
BSW00387	J1939TP0197
BSW00400	J1939TP0187
BSW00405	J1939TP0187
BSW00406	J1939TP0023
BSW00407	J1939TP0092, J1939TP0090, J1939TP0089, J1939TP0025
BSW00409	J1939TP0002
BSW00412	J1939TP0008
BSW00413	J1939TP0197
BSW00414	J1939TP0187
BSW00415	J1939TP0197
BSW00416	J1939TP0197
BSW00417	J1939TP0197
BSW00419	J1939TP0197
BSW00422	J1939TP0082, J1939TP0083
BSW00423	J1939TP0197
BSW00427	J1939TP0197
BSW00433	J1939TP0197
BSW00435	J1939TP0010
BSW00436	J1939TP0014

BSW00437	J1939TP0197
BSW00438	J1939TP0187
BSW00439	J1939TP0197
BSW00440	J1939TP0197
BSW00441	J1939TP0019, J1939TP0115
BSW00442	J1939TP0088, J1939TP0166, J1939TP0167, J1939TP0164
BSW00443	J1939TP0197
BSW00444	J1939TP0197
BSW00445	J1939TP0197
BSW00446	J1939TP0197
BSW00447	J1939TP0197
BSW00449	J1939TP0197
BSW00450	J1939TP0023
BSW00453	J1939TP0197
BSW00455	J1939TP0197
BSW005	J1939TP0197
BSW101	J1939TP0087
BSW161	J1939TP0197
BSW162	J1939TP0197
BSW164	J1939TP0197
BSW168	J1939TP0197
BSW170	J1939TP0197
BSW171	J1939TP0090, J1939TP0074, J1939TP0125
BSW42400001	J1939TP0125
BSW42400002	J1939TP0153
BSW42400003	J1939TP0067, J1939TP0055, J1939TP0056, J1939TP0054
BSW42400010	J1939TP0024
BSW42400011	J1939TP0023, J1939TP0025
BSW42400018	J1939TP0058
BSW42400019	J1939TP0155
BSW42400020	J1939TP0155
BSW42400021	J1939TP0113, J1939TP0109
BSW42400022	J1939TP0018, J1939TP0121, J1939TP0123
BSW42400023	J1939TP0018, J1939TP0064
BSW42400024	J1939TP0068
BSW42400025	J1939TP0062, J1939TP0058
BSW42400040	J1939TP0018
BSW42400042	J1939TP0156, J1939TP0157

7 Functional Specification

This chapter describes the functionality of the AUTOSAR J1939 Transport Layer. It explains the services provided to the upper and lower layers and the internal behavior of J1939Tp.

7.1 Basic Principles of SAE J1939

The SAE J1939 standard defines a set of Parameter Groups (PGs), each containing signals with defined content and semantics. The following information is provided for each PG:

- Payload length type: maximal number of bytes, fixed or variable size.
- Parameter Group Number (PGN): 18 bit value containing the following fields:
 - o 2 bit data page information (DP and EDP)
 - o 8 bit PDU-Format (PF)
 - o 8 bit PDU-Specific (PS)

PGNs with PDU-Format < 240 (format 1) identify point-to-point messages, while PGNs with PDU-Format >= 240 (format 2) identify broadcast messages. The PDU-Specific field is only relevant for broadcast messages (format 2); it is always zero for point-to-point messages (format 1).

J1939 uses 29-bit CAN identifiers to identify each message. The CAN identifier contains a 3-bit-priority, the PGN, the Source Address (SA), and the Destination Address (DA, only for point-to-point messages).

Usually, an ECU has just one node address, which is used as DA in received messages and as SA in transmitted messages. However, a single ECU can also implement several different J1939 nodes at once, each with its own node address. In this case, the ECU accepts any of these node addresses as DA and sends with any of the defined node addresses as SA.

[J1939TP0152] [

The J1939Tp shall respect the definition of DA and SA according to [16] and [17] .

This definition is static and cannot change at runtime.] ()

The SAE J1939 transport layer uses two special point-to-point messages identified by PGNs of format 1 to transport segmented messages, both with a fixed length of 8 bytes. These messages are called transport frames in the context of this document.

- TP.CM is used for connection management. The first byte of the payload identifies its role, which may be one of the following:
 - o TP.CM/BAM is used to initiate a BAM transfer.
 - o TP.CM/RTS is transmitted to initiate a CMDT transfer.
 - o TP.CM/CTS is used for flow control during a CMDT transfer.
 - o TP.CM/EOMAck indicates the end of a CMDT transfer.
 - o TP.CM/Abort indicates an error and terminates the CMDT transfer.
- TP.DT contains a sequence number in the first byte and 7 bytes of data.

A single TP.CM or TP.DT frame, identified by a certain CAN Identifier, is used for different PGs. The PGN of the transported PG is contained in the payload of the TP.CM frames as specified in [16].

The destination address (DA) of CMDT related transport frames contains a legal node address and thus allows a point-to-point connection, while the DA of BAM related transport frames is always set to FF₁₆ to create a broadcast connection.

[J1939TP0018] [

The J1939Tp module shall follow the recommendations of SAE J1939-21 [16] if they are not explicitly excluded in this document.] (BSW42400022, BSW42400023, BSW42400040)

7.2 Basic Functionality of J1939Tp

This section describes aspects of the functionality of J1939Tp that are not related to neighboring modules.

[J1939TP0036] [

To assure a unique PDU router handling of all J1939 PGs which is independent of the Payload Length type (variable or fixed), the J1939Tp shall be used for the transmission of all Parameter Groups that are longer than 8 Bytes, independent of their length being fixed or variable.

This means that PGs with variable length that have a configured maximum size of more than 8 bytes but do not exceed 8 bytes at runtime shall be transported using J1939Tp even though no segmentation is necessary and a direct transmission from PDU router to CAN Interface would have been possible.] ()

[J1939TP0155] [

The J1939 transport layer shall implement the following two J1939 transport protocol variants defined in [16]:

- BAM for broadcast transmission
- CMDT for point-to-point transmission] (BSW42400019, BSW42400020)

[J1939TP0125] [

The J1939 Transport Layer shall be configurable to either use both BAM and CMDT transport protocols, or only BAM, or only CMDT.] (BSW171, BSW42400001)

[J1939TP0174] [

The transport protocol variant used to transfer a large PG shall be exclusively determined by the PDU-Format (PF) field of the PGN:

- If the value of PF is less than 240 and DA is a specific node address, then the payload of the PG must be transported using the CMDT protocol.
- If the value of PF is at least 240 or DA is the broadcast address (FF₁₆), then the payload of the PG must be transported using the BAM protocol.] ()

This requirement contradicts [16], in which the chosen transport protocol variant is also determined by the request protocol.

[J1939TP0041] [

J1939Tp shall send the transport frames TP.CM and TP.DT and direct frames using the service function `CanIf_Transmit()`.] ()

[J1939TP0189] [

In contradiction with [16], parallel transmission via transport protocol and directly of a specific message (identical PGN, SA, and DA) with variable length shall not be supported. The same applies for parallel reception via transport protocol and directly.] ()

The reason for this restriction is that COM has only one buffer per I-PDU. This buffer is locked during transmission via TP, so that there is no possibility to change the PDU. In addition, during reception via TP the COM buffer is locked, so that there is no place to store the same PDU when received in parallel via a direct frame. Parallel transmission and reception would require additional buffers in COM with the possibility of parallel access.

[J1939TP0190] [

Contradictory to [16], the J1939Tp shall abort a connection when a CTS frame is received that requests already sent data.] ()

It would be possible to implement retransmission of already sent frames by using the `RetryInfoPtr` of `PduR_J1939TpTxCopyData`, but the management overhead in J1939Tp would be considerable.

[J1939TP0192] [

J1939Tp shall abort connections after a sequence error to avoid the management overhead required to request already sent data again. This approach harmonizes with **J1939TP0190**.] ()

7.3 Services Provided to the Upper Layer

The service interface of the J1939Tp module can be divided into the following main categories:

- Initialization and shutdown
- Communication services

[J1939TP0156] [

All service interfaces provided to upper layers shall be independent of the internal communication configuration and implementation of the J1939Tp.] (BSW42400042)

The following paragraphs describe the functionality of each services category.

7.3.1 Initialization and Shutdown

The following figure summarizes all the requirements concerning initialization and shut down:

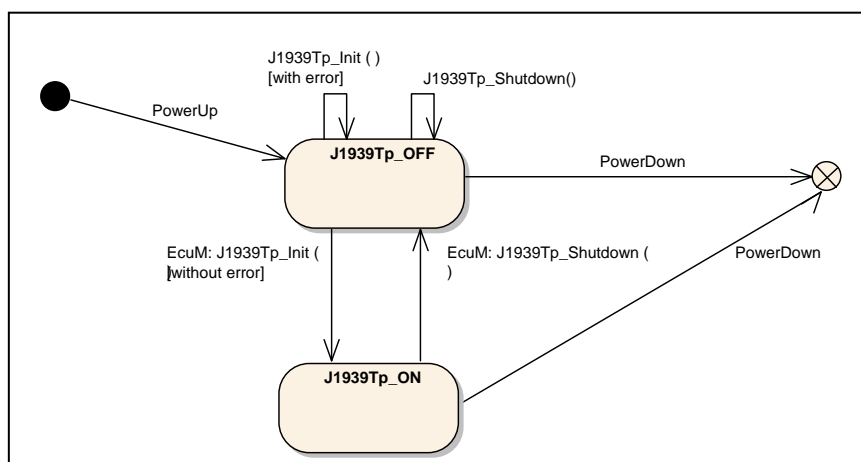


Figure 7-1: J1939 Transport Layer life cycle

[J1939TP0019] [

The J1939Tp module shall have two global states, J1939TP_ON and J1939TP_OFF.] (BSW00335, BSW00441)

[J1939TP0020] [

The J1939Tp module shall be in the J1939TP_OFF state after power up.] ()

[J1939TP0021] [

In the state J1939TP_OFF, the J1939Tp shall allow an update of the post build configuration.] ()

[J1939TP0023] [

The J1939Tp module shall perform segmentation and reassembly tasks only when the J1939Tp is in the J1939TP_ON state.] (BSW00406, BSW00450, BSW42400011)

7.3.2 Communication Services

[J1939TP0126] [

A transmit operation, J1939Tp_Transmit(), shall allow upper layers to ask for data transfer using J1939 Transport Layer. The function J1939Tp_Transmit() shall be asynchronous.] ()

[J1939TP0030] [

The function `J1939Tp_Transmit()` shall reject the transmit request and return the status value `E_NOT_OK` if the channel is occupied (see also 7.5.8). **]** ()

7.4 Services Provided to the Lower Layer

[J1939TP0157] [

All service interfaces provided to the lower layer shall be independent of the internal communication configuration and implementation of the J1939Tp. **]** (BSW42400042)

7.4.1 Transmission Confirmation

The transmission confirmation `J1939Tp_TxConfirmation()` is called by CanIf to notify J1939 TP of successful transmission of an N-PDU.

[J1939TP0035] [

After a `CanIf_Transmit()` has been issued and no TxConfirmation is received from the CanIf within a maximum time (defined by `J1939TpTxConfTimeout`, see **J1939TP0006_Conf** :), the J1939Tp module shall abort the failed transmission as defined in **J1939TP0032**, and any late confirmation shall be ignored. **]** ()

7.4.2 Reception Indication

The J1939Tp module shall provide a `J1939Tp_RxIndication()` API to allow the CanIf to notify that a new N-PDU has been received (see **J1939TP0109**).

CanIf shall perform the Reception Indication according to its configuration (i.e. in ISR context if configured so).

7.5 Internal Behavior

7.5.1 Session Handling

7.5.1.1 Close Transmission

[J1939TP0119] [

When the transport transmission session is successfully completed, the J1939Tp module shall call a notification service of the upper layer, `PduR_J1939TpTxConfirmation`, with the result `NTFRSLT_OK`, to notify that the N-SDU transfer is successfully processed. DEM shall be notified as described in **J1939TP0082**. **]** ()

7.5.1.2 Abort Transmission

[J1939TP0032] [

An Abort Transmission feature shall indicate the upper layer that the transmission of a given N-SDU has been aborted. It uses the callback function `PduR_J1939TpTxConfirmation` with `NTFRSLT_E_NOT_OK` and reports to DET with the value `J1939TP_E_TRANSMISSION_ABORTED`. DEM shall be notified as described in **J1939TP0083**.] ()

7.5.1.3 Close Reception

[J1939TP0118] [

A close connection feature shall indicate the upper layer that the transport reception session is completed. It uses the `PduR_J1939TpRxIndication()` with `NTFRSLT_E_OK` to indicate that the N-SDU reception is successfully processed. DEM shall be notified as described in **J1939TP0082**.] ()

7.5.1.4 Abort Reception

[J1939TP0031] [

An Abort Reception feature shall indicate the upper layer that the reception of a given N-SDU has been aborted. It uses the `PduR_J1939TpRxIndication()` with state `NTFRSLT_E_NOT_OK` and report to DET with the value `J1939TP_E_RECEPTION_ABORTED`. DEM shall be notified as described in **J1939TP0083**.] ()

7.5.1.5 Abort CMTD Connection

[J1939TP0097] [

An Abort CMTD Connection feature shall indicate to the other ECU participating in the concerned connection that the current CMTD session (transmission or reception) cannot be completed successfully. A `TP.CM/Abort` abort message shall be transmitted to the other ECU via CAN as specified by [16]. This is necessary every time an error happens after `TP.CM/CTS` has been successfully sent or received.] ()

7.5.2 N-SDU Reception

The J1939 Transport Layer has no buffering capacity.

[J1939TP0043] [

Depending on the control byte of the initializing `TP.CM` frame (`BAM` or `RTS`), the J1939Tp module shall use `BAM` or `CMTD` as transport protocol to handle the data reception.] ()

[J1939TP0038] [

When receiving an N-PDU containing a TP.CM/BAM or a TP.CM/RTS the J1939Tp module shall first notify the upper layer (PDU Router) before processing the frame reassembly. It uses the `PduR_J1939TpStartOfReception` function with the following parameters:

- the Identifier of the corresponding N-SDU,
- the total Data Length (after reassembly) and
- a pointer to a location where the upper layer stores its currently available buffer size.] ()

[J1939TP0162] [

After the reception of an N-PDU containing a TP.DT frame, the function `PduR_J1939TpCopyRxData` shall be called with the following parameters:

- the Identifier of the corresponding N-SDU,
- `PduInfoPtr` with max. 7 bytes of data,
- a pointer to a location where the upper layer stores its currently available buffer size. The value returned in this parameter shall not be evaluated by J1939TP.] ()

[J1939TP0173] [

The J1939Tp module shall abort the reception silently if any of the following conditions occurs:

- The `PduR_J1939TpStartOfReception` function returns `BUFREQ_E_NOT_OK` or `BUFREQ_E_OVFL` or `BUFREQ_E_BUSY` or
- The protocol chosen in **J1939TP0043** does not match the transport protocol configured for the transported PGN (see **J1939TP0029_Conf** :).

In case BAM has been used, no further activity is needed.

In Case of CMDT, a CMDT Connection Abort shall be executed as described in **J1939TP0097**.] ()

[J1939TP0040] [

The J1939Tp module shall abort the reception like indicated in **J1939TP0031** if any of the following conditions occurs:

- The value returned by `PduR_J1939TpStartOfReception` via `RxBufferSizePtr` is smaller than the total data length of the N-SDU (after reassembly) or
- The `PduR_J1939TpCopyRxData` function returns `BUFREQ_E_NOT_OK`

In case BAM has been used, no further activity is needed.

In Case of CMDT, a CMDT Connection Abort shall be executed as described in **J1939TP0097**.] ()

7.5.3 N-SDU Transmission

As described in section 7.3.2, the upper layer (PDU Router) asks for the transmission of an N-SDU by calling `J1939Tp_Transmit()`. The parameters of `J1939Tp_Transmit()` describe the Identifier of the N-SDU (`NSduId`) and a reference to a `PduInfoType` that indicates the full length of the N-SDU to transmit (full Tx N-SDU data length) and a pointer to the payload N-SDU.

[J1939TP0039] [

The J1939Tp module shall depending on the PGN use one of the module implementing CMDT or BAM as transport protocol to handle the data transmission.

See also **J1939TP0137_Conf** :.] ()

[J1939TP0045] [

The function J1939Tp_Transmit shall use the NSduId and the SduLength provided in the PduInfoType structure. It shall not use the pointer to the payload of the N-SDU.] ()

[J1939TP0047] [

After a transmission request from the upper layer, the J1939Tp module shall initiate the transmission by sending:

- For CMDT: a TP.CM/RTS frame
- For BAM: a TP.CM/BAM frame] ()

[J1939TP0046] [

For each TP.DT frame to be sent, the J1939Tp module shall previously call PduR_J1939TpCopyTxData with the following parameters:

- the Identifier of the corresponding N-SDU,
- PduInfoPtr with max. 7 bytes as Data Length,
- A NULL_PTR for the RetryInfoPtr (unused feature) and
- a pointer to a location where the upper layer stores its currently available data.

The value returned in this parameter shall not be evaluated by J1939TP.] ()

[J1939TP0048] [

If the upper Layer returns the PduR_J1939TpCopyTxData function call with BUFREQ_E_NOT_OK or BUFREQ_E_BUSY, or when a timeout occurs while waiting for TP.CM/CTS, the J1939Tp module shall abort the transmission session like specified in **J1939TP0032**. In Case of CMDT, a CMDT Connection Abort shall be executed as described in **J1939TP0097**.] ()

7.5.4 Data Flow on the CAN Bus

7.5.4.1 Data Flow using Direct Transmission

The following figure shows an example of direct message transmission between two ECUs using J1939Tp. This is the only case of transmission of a J1939Tp N-SDU using no TP.CM or TP.DT frame. The SA is always included in the CAN identifier. Depending on the PDU-Format of the concerned PG, the CAN Identifier might contain the DA.

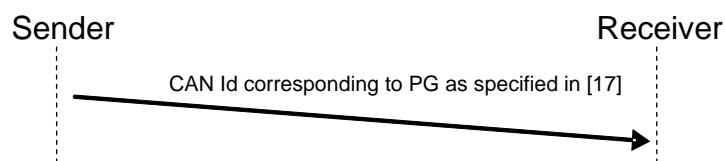


Figure 7-2: Example of direct data flow for PGs of variable length ≤ 8 Bytes

7.5.4.2 Data Flow using CMTD

The following figure shows an example of segmented message transmission between two ECUs using CMTD as transport protocol. The usage of the CMTD transport protocol is a peer-to-peer communication (i.e. 1 to 1 communication, like physical addressing in diagnostics). In the example, the transmitted PG has a total length of 16 bytes, which corresponds to 3 blocks of 7 bytes.

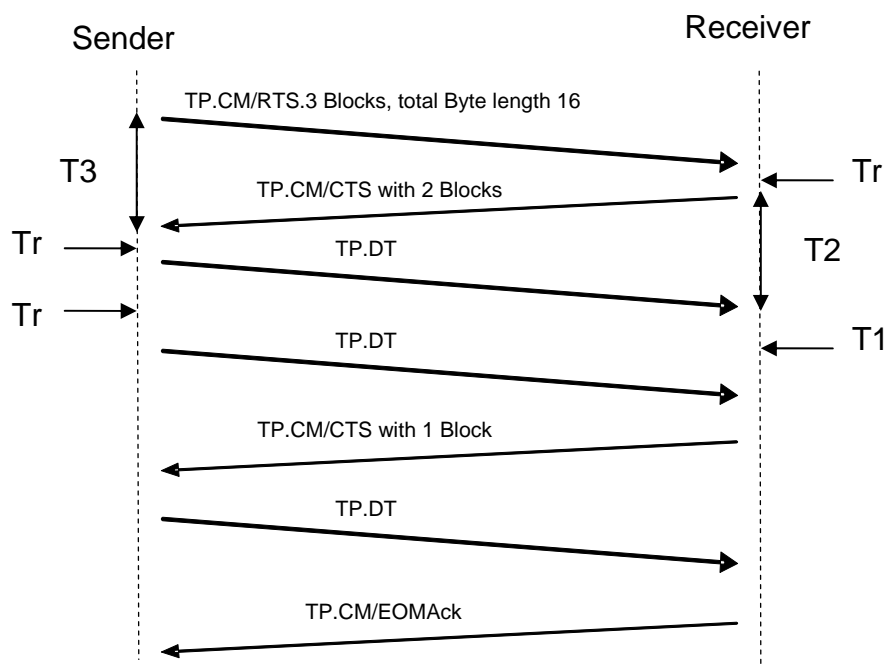


Figure 7-3: Example of data flow without error using CMTD as transport protocol

[J1939TP0123] [

J1939Tp shall implement all CMTD related timing constraints (Tr, T1, T2, T3, T4) as described in [16]. They supervise the CMTD Dataflow.] (BSW42400022)

[J1939TP0165] [

The initial sequence of RTS and CTS shall be used to adjust the sender to the capabilities of the receiver:

- The sender specifies a maximal block size using a TP.CM/RTS frame (see **J1939TP0122_Conf** :),

- The receiver answers with the block size that shall be used for this session using a TP.CM/CTS (see **J1939TP0061_Conf** :).] ()

[J1939TP0195] [

If a TP.CM/CTS wait frame as specified by [16] is received, the J1939Tp shall wait for another TP.CM/CTS frame.] ()

[J1939TP0194] [

If a TP.CM/CTS frame requesting already sent data is received, the J1939Tp may abort the transmission (in contradiction with [16]) using the mechanisms described by **J1939TP0031** and **J1939TP0097**.] ()

[J1939TP0100] [

If a timeout occurs by the CMDT transmission supervision then the J1939Tp module shall abort the transmission session

- as specified in **J1939TP0031** and
- as specified in **J1939TP0097** when the timeout occurred after successful transmission of the TP.CM/RTS frame.] ()

[J1939TP0159] [

If a timeout occurs by the reception supervision like described in [16] then the J1939Tp module shall abort the reception session

- as specified in **J1939TP0032** and
- as specified in **J1939TP0097**.] ()

[J1939TP0098] [

If a TP.CM /Abort frame is received after a TP.CM/RTS frame has been successfully sent by the ECU, the transmission shall be aborted like indicated in **J1939TP0032**.

No transmission of TP.CM/Abort is necessary (**J1939TP0097**).] ()

[J1939TP0163] [

If a TP.CM/Abort frame is received after a TP.CM/RTS frame has been received, the reception shall be aborted like indicated in **J1939TP0031**. No transmission of TP.CM/Abort is necessary (**J1939TP0097**).] ()

7.5.4.3 Data Flow using BAM

[J1939TP0121] [

J1939Tp shall implement the BAM related Timing constraints (T1) like described in [16]. They supervise the BAM Dataflow.] (BSW42400022)

[J1939TP0160] [

If a timeout occurs during the BAM reception, the J1939Tp module shall abort the reception session as specified in **J1939TP0031**.] ()

The following figure shows an example of segmented message transmission between two ECUs using BAM as transport protocol according to [16]. The usage of the BAM transport protocol is a broadcast communication (i.e. 1 to n communication, like functional addressing in diagnostics). In the example, the transmitted PG has a total length of 3 blocks of 7 bytes that have to be successively transmitted.

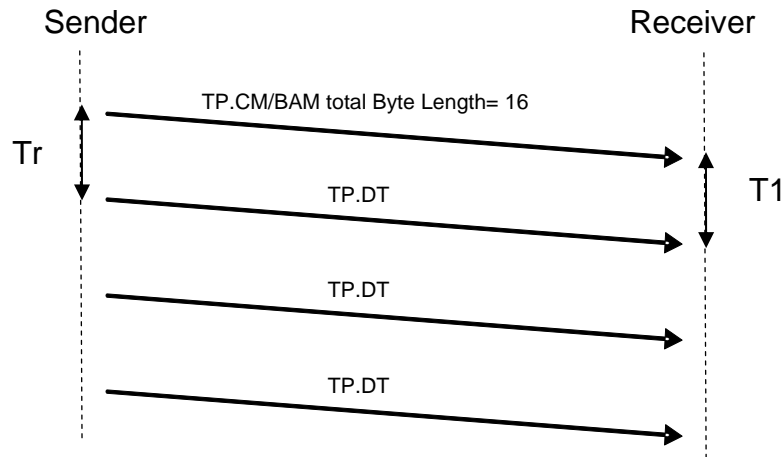


Figure 7-4: Example of data flow using BAM as transport protocol

7.5.5 Interface between J1939Tp and PduR

J1939Tp shall have no internal PDU buffers. It requests data for sending directly from the upper layers via `PduR_J1939TpCopyTxData()` and provides received data directly to the upper layers via `PduR_J1939TpCopyRxData()`.

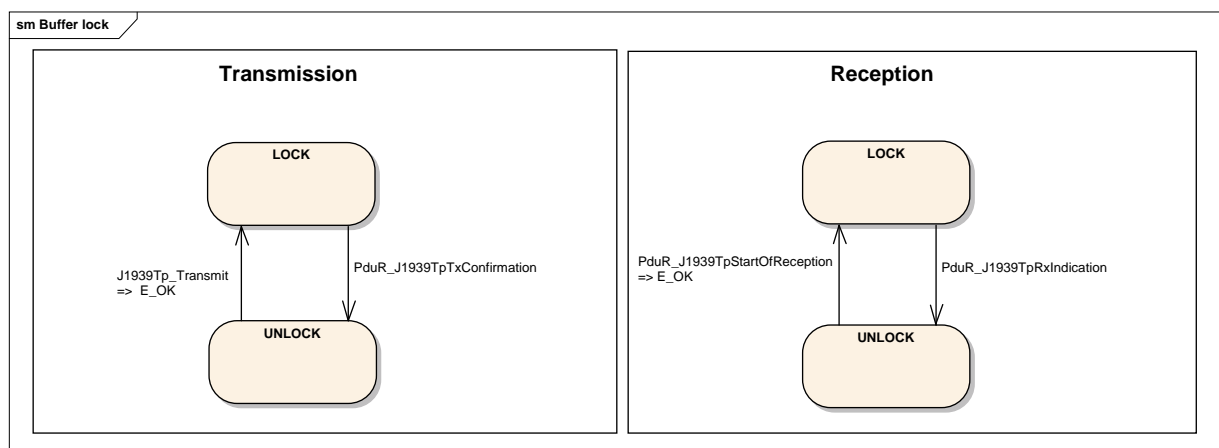


Figure 7-5: Tx and Rx Buffer locking

To guarantee data consistency, the complete buffer of the upper layers must be locked during the whole data transmission or reception. Data transmission is initialized when the PduR calls `J1939Tp_Transmit()` and is active after `J1939Tp`

executed this call successfully until J1939Tp calls PduR_J1939TpTxConfirmation(). Data reception is initialized when J1939Tp calls PduR_J1939TpStartOfReception() and is active after the PduR executed this call successfully until J1939Tp calls PduR_J1939TpRxIndication().

[J1939TP0161] [

J1939Tp shall not support delayed reception. If the available buffer in the upper layers is not sufficiently large for the whole message, the reception shall be aborted. J1939Tp does not require any dynamic buffer management in the upper layers.] ()

7.5.5.1 Direct Transmission

The following figure shows the process of sending a variable length message with at most 8 bytes of data.

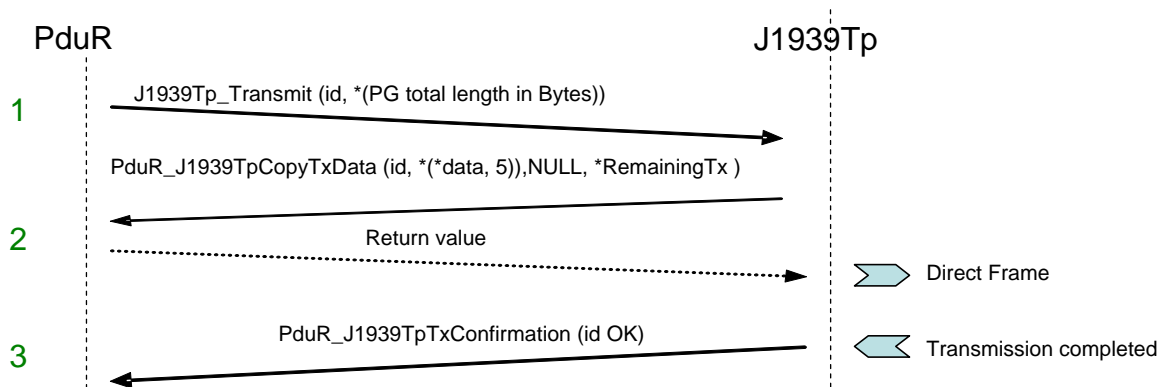


Figure 7-6: Direct transmission process between PduR and J1939Tp

- 1:** The PduR initiates the transmission of an N-SDU using J1939Tp_Transmit with the corresponding Id.
- 2:** The J1939Tp requests the message data from the PduR via PduR_J1939TpCopyTxData, and sends the direct frame containing these bytes.
- 3:** After completion of the direct transmission, the J1939Tp sends a confirmation to the PduR using PduR_J1939TpTxConfirmation.

7.5.5.2 Direct Reception

The following figure shows the process of receiving a variable length message with at most 8 bytes of data.

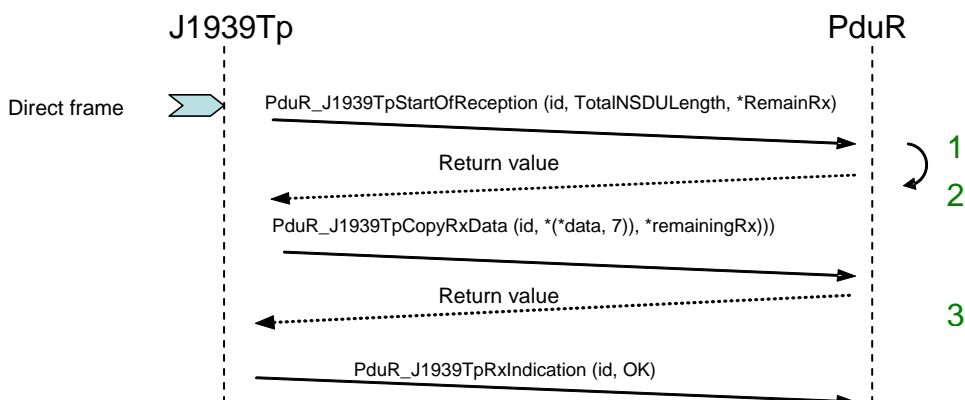


Figure 7-7: Direct reception process between J1939Tp and PduR

- 1: After receiving an indication for the direct frame from the CanIf, the J1939Tp asks the PduR via `PduR_J1939TpStartOfReception` to initiate the reception of the corresponding N-SDU.
- 2: The PduR confirms the availability of the required resources by returning `E_OK`.
- 3: The J1939Tp transfers the data to the PduR using `PduR_J1939TpCopyRxData` and notifies the PduR via a call to `PduR_J1939TpRxIndication` of the successful reception of the complete N-SDU.

7.5.5.3 Segmented Transmission

The following figure shows the process of sending a segmented message. This process does not depend on the used transport protocol (CMDT or BAM).

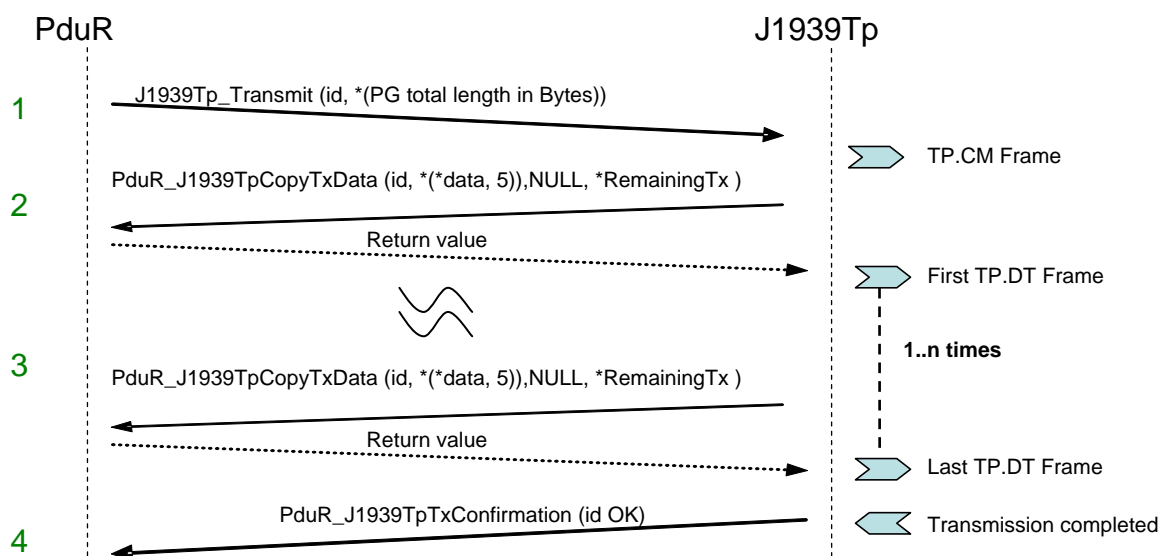


Figure 7-8: Segmented transmission process between PduR and J1939Tp

- 1: The PduR initiates the transmission of an N-SDU using `J1939Tp_Transmit` with the corresponding Id. The J1939Tp immediately sends an appropriate TP.CM frame.

2: The J1939Tp requests the first 7 bytes of data to be transmitted from the PduR via `PduR_J1939TpCopyTxData`, and sends the TP.DT frame containing these 7 bytes.

3: The transmission of TP.DT frames is repeated until the complete N-SDU data has been transmitted.

4: After completion of the segmented data transmission, the J1939Tp sends a confirmation to the PduR using `PduR_J1939TpTxConfirmation`.

7.5.5.4 Segmented Reception

The following figure shows the process of receiving a segmented message. This process is the same for CMDT and BAM.

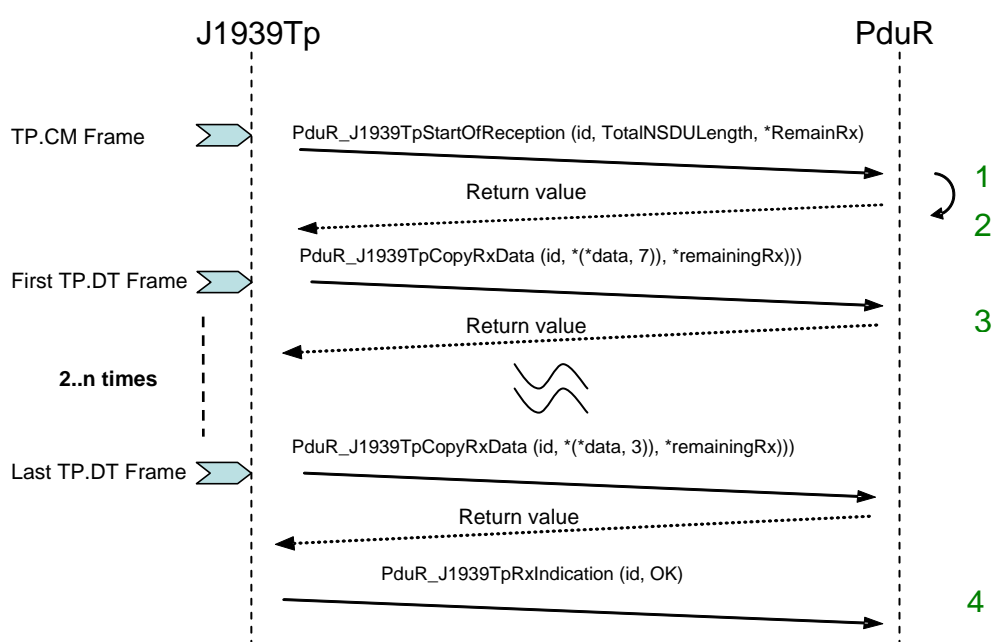


Figure 7-9: Segmented reception process between J1939Tp and PduR

1: After receiving an indication for the TP.CM frame from the CanIf, the J1939Tp asks the PduR via `PduR_J1939TpStartOfReception` to initiate the reception of a new N-SDU.

2: The PduR confirms the availability of the required resources (buffer) by returning `E_OK`.

3: After each received TP.DT Frame, the J1939Tp transfers the 7 bytes of data “on the fly” to the PduR via `PduR_J1939TpCopyRxData`.

4: After reception of the last TP.DT frame, the J1939Tp notifies the PduR via a call to `PduR_J1939TpRxIndication` of the successful reception of the complete N-SDU.

7.5.6 Interface between J1939Tp and CanIf

For the communication between CanIf and J1939Tp, the following API functions are used:

Used transport protocol

BAM, CMDT

BAM, CMDT

BAM, CMDT

API

CanIf_Transmit

J1939Tp_TxConfirmation

J1939Tp_RxIndication

The two following figures show how this API is used.

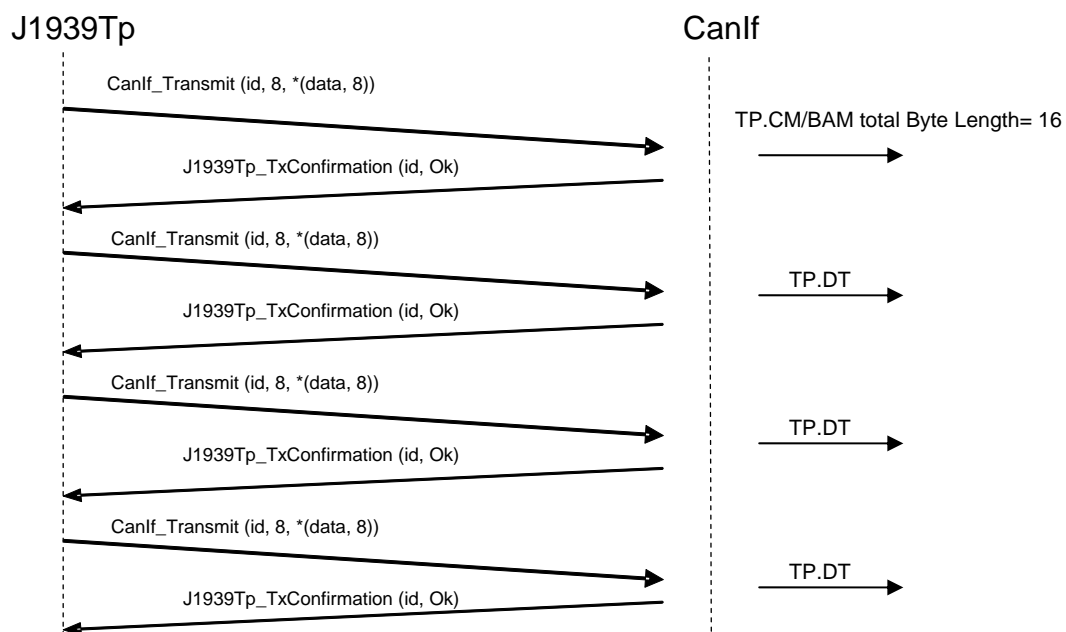


Figure 7-10: BAM transmission process between J1939Tp and CanIf

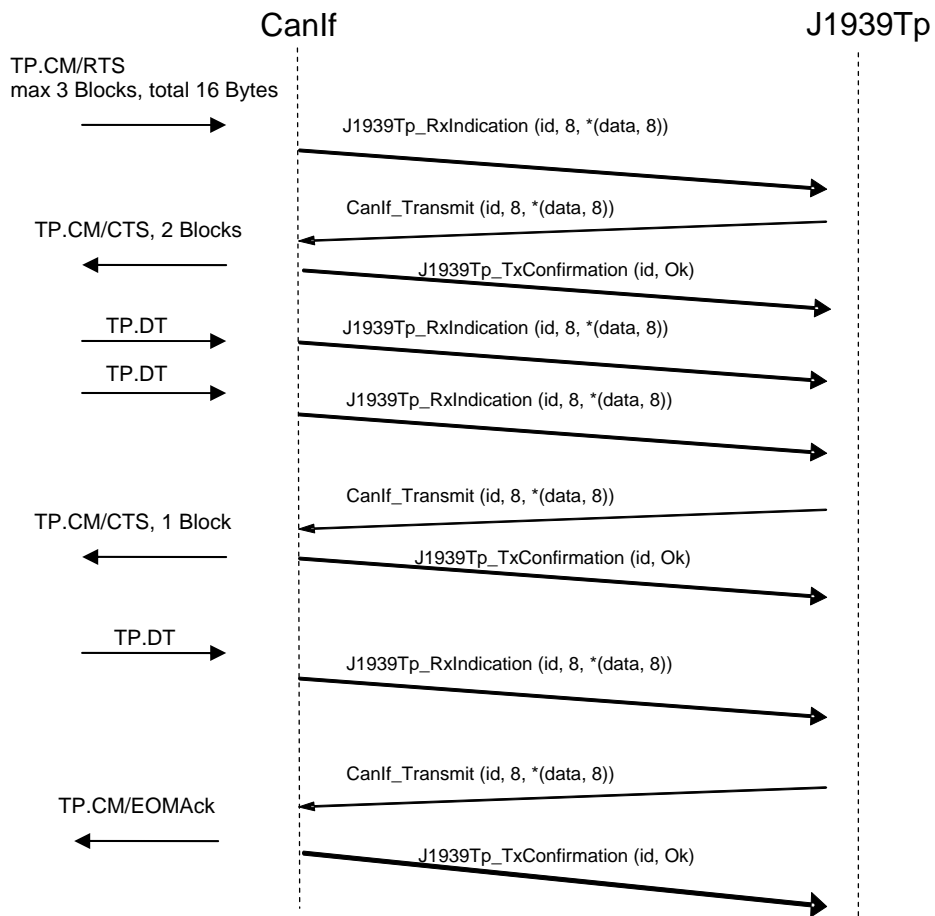


Figure 7-11: CMTD reception process between J1939Tp and CanIf

7.5.7 Relationship between N-SDU and N-PDU in J1939Tp

This section describes the relation that exists between an N-SDU and the set of N-PDUs that is required to transport the N-SDU data, as shown in the following figure.

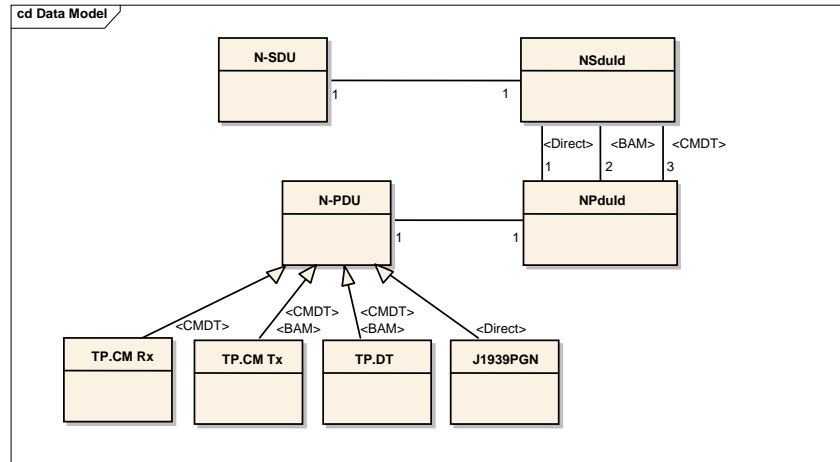


Figure 7-12: Relation between N-SDU and N-PDU

[J1939TP0153] [

Each N-SDU shall be identified by a unique ID, the NSduld. The N-SDU corresponds to a J1939 Parameter Group (PG).] (BSW42400002)

[J1939TP0056] [

An N-SDU of variable length that is configured for transmission or reception via CMDT shall be linked at configuration time to four N-PDUs:

- One for the direct frame used if the length of the N-SDU is at most 8 bytes,
- and three for TP.CM Tx, TP.CM Rx, and TP.DT frames.] (BSW42400003)

[J1939TP0067] [

An N-SDU of fixed length >8 bytes that is configured for transmission or reception via CMDT shall be linked at configuration time to three N-PDUs used for TP.CM Tx, TP.CM Rx and TP.DT frames.] (BSW42400003)

[J1939TP0055] [

An N-SDU of variable length that is configured for transmission or reception via BAM shall be linked at configuration time to three N-PDUs:

- One for the direct frame used if the length of the N-SDU is at most 8 bytes,
- and two for TP.CM/BAM and TP.DT frames used for larger N-SDUs.] (BSW42400003)

[J1939TP0054] [

An N-SDU of fixed length >8 bytes that is configured for transmission or reception via BAM shall be linked at configuration time to two J1939 N-PDUs used for TP.CM/BAM and TP.DT frames.] (BSW42400003)

[J1939TP0057] [

J1939Tp shall combine the NPduId of a TP.CM/BAM or TP.CM/RTS frame with the PGN included in the payload to identify the corresponding N-SDU.] ()

7.5.8 Concurrent Connections

Connections only concern internal J1939Tp purposes. They are transparent for the upper and lower layers of J1939Tp but influence the handling of J1939Tp.

[J1939TP0124] [

The connections are only destined for J1939Tp internal use, so they are not accessible externally.] ()

A J1939Tp connection is characterized by its direction (Receiving /Sending) and its type (BAM / CMTD / Direct). A J1939Tp connection of type BAM or CMTD uses the following transport related frames:

- control (TP.CM)
- data (TP.DT)

The CAN-Identifier corresponding to those transport specific frames is the same for all J1939 PGs longer than 8 bytes:

- sent from a given SA - in the BAM case
- sent from a given SA to a given DA - in the CMTD case

This reduces the possibility for J1939Tp to process concurrent connections.

[J1939TP0120] [

All possible connections shall be defined during configuration, even though only a few of them might be active at once. The total number of possible connections is directly configurable and will be determined during the configuration process by analyzing the available N-SDUs and N-PDUs.] ()

[J1939TP0058] [

J1939Tp shall only support concurrent connections as described in section 5.10.5 of [16]. Note that one AUTOSAR ECU can represent several J1939 nodes and thus may have more than one address (used as SA or DA) assigned to it.] (BSW42400018, BSW42400025)

[J1939TP0062] [

Each connection is independent of the other connections. This means that a connection uses its own resources, such as internal buffer, timer, or state machine.] (BSW42400025)

7.5.9 N-PDU Padding**[J1939TP0068] [**

According to [16], all unused data bytes within the last TP.DT frame or the direct frame shall be set to 0xFF.] (BSW42400024)

7.5.10 Handling of Unexpected N-PDU Arrivals

[J1939TP0064] [

J1939Tp shall follow the directions of section 5.10 of [16] when receiving unexpected but configured N-PDUs. Unexpected N-PDUs that are not configured for J1939Tp shall be ignored.] (BSW42400023)

7.6 Error Classification

The general requirements document of AUTOSAR [3] specifies that all basic software modules must distinguish (according to the product life cycle) two error types:

- Development errors: These errors should be detected and fixed during development phase. In most cases, these errors are software errors.
- Production errors: These errors are hardware errors and software exceptions that cannot be avoided and are expected to occur in the production code.

[J1939TP0071] [

On errors and exceptions, the J1939Tp module shall not modify its current module state (see Figure 7-1: J1939 Transport Layer life cycle) but shall simply report the error event.] ()

[J1939TP0072] [

The configuration tool assigns ECU dependent values to the Event Ids (see **J1939TP0183_Conf** :). The Event Ids shall be defined locally in the module J1939Tp. The definitions are ensured to be identical to the definitions used by DEM because of the reference represented by **J1939TP0183_Conf** :.] ()

[J1939TP0073] [

Development error values are of type uint8.] ()

[J1939TP0115] [J1939Tp shall use following errors:

Type or error	Relevance	Related error code	Value [hex]
API service called with wrong ID or null pointer.	Development	Could be a combination of: J1939TP_E_PARAM_ID J1939TP_E_PARAM_ADDRESS	0x01 0x02
API service used in state J1939TP_OFF.	Development	J1939TP_E_UNINIT	0x20
J1939Tp_Init() called in state J1939TP_ON.	Development	J1939TP_E_REINIT	0x21
Reception or transmission has been aborted due to	Development	J1939TP_E_RECEPTION_ABORTED	0x30 0x31

communication or API call errors.		J1939TP_E_TRANSMISSION_ABORTED	
Used for reporting the transmission or reception status to DEM (successful or unsuccessful). See J1939TP0183_Conf : for the configuration of the value.	Production	J1939TP_E_COMMUNICATION	Assigned by DEM

┘ (BSW00327, BSW00337, BSW00385, BSW00441)

7.7 Error Detection

[J1939TP0074] ┘

The detection of development errors is configurable (*ON / OFF*) at pre-compile time. The switch *J1939TpDevErrorDetect* (see chapter 10) shall activate or deactivate the detection of all development errors. ┘ (BSW171)

[J1939TP0154] ┘

If the *J1939TpDevErrorDetect* switch is enabled, API parameter checking is also enabled. The detailed description of the detected errors can be found in section 7.6 and chapter 8. ┘ ()

[J1939TP0075] ┘

The detection of production code errors cannot be switched off. ┘ ()

[J1939TP0076] ┘

The global state (see **J1939TP0019**) shall be used to check if the module has been initialized before calling an API. ┘ ()

7.8 Error Notification

[J1939TP0077] ┘

Detected development errors shall be reported to the Development Error Tracer (DET, see [11]) if the pre-processor switch *J1939TpDevErrorDetect* (see **J1939TP0042_Conf** :) is set. ┘ (BSW00338)

[J1939TP0079] ┘

The J1939Tp module shall use the following service function of the Development Error Tracer to report development errors:

```
void Det_ReportError(ModuleId, InstanceId, ApiId, ErrorId) ┘ ( )
```


The module ID 37 (25₁₆) of J1939Tp, which is used as a parameter in the Det_ReportError() call, is exported via the macro definition J1939TP_MODULE_ID in J1939Tp.h (see also **J1939TP0196**).

The Development Error Tracer module is merely an aid for BSW development and integration. The API is defined by AUTOSAR, but the functionality can be chosen and implemented according to the development needs (e.g. counting errors or sending error information via a serial interface to an external logger).

The following figure shows the interaction with the DET for bad call parameters as an example how J1939Tp uses the DET. When a development error occurs, the J1939Tp returns the value E_NOT_OK. The error ID is only reported to the DET.

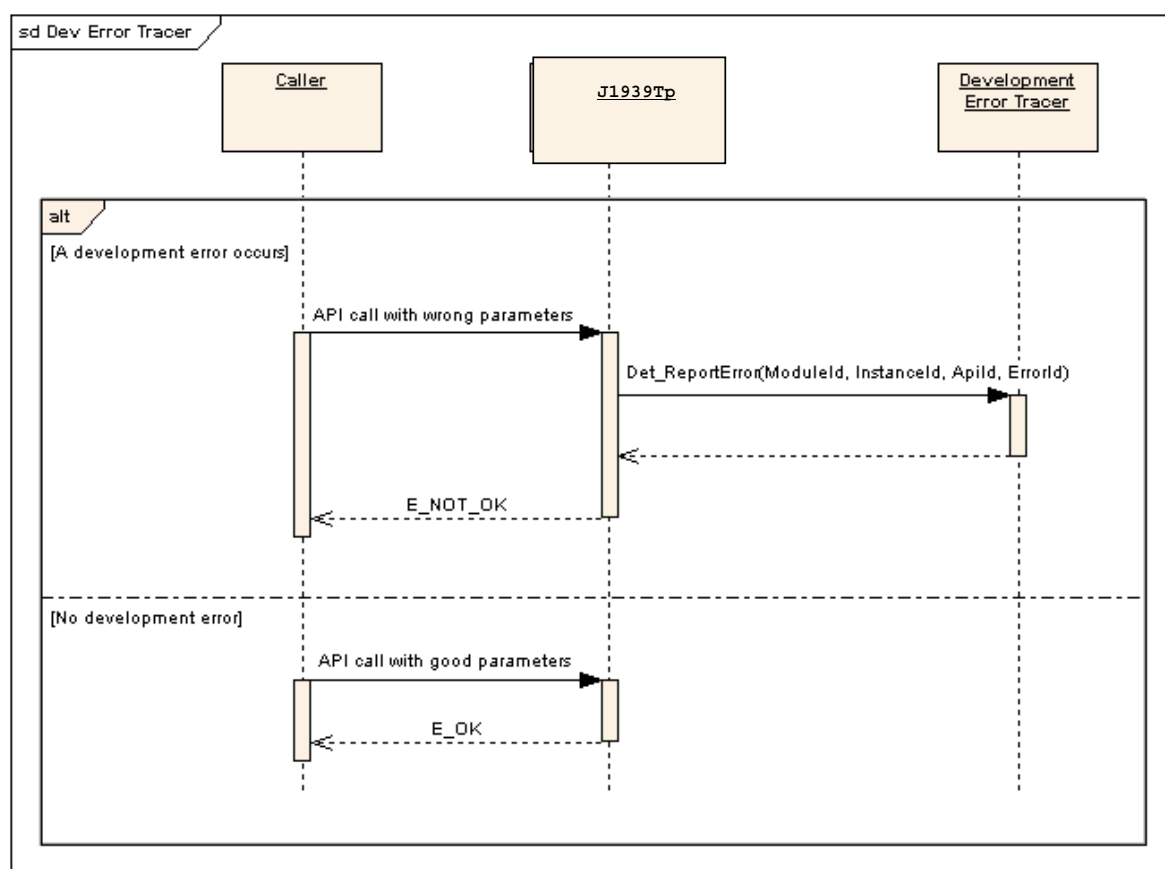


Figure 7-13: Development error reporting

[J1939TP0078] [

Production errors shall be reported to Diagnostic Event Manager (DEM, see [10]).]
(BSW00339)

The DEM module dumps the error into the “error memory” and informs the FIM module, which has a pre-defined reaction for the ECU (e.g. disabling ECU functions or switching ECU mode).

[J1939TP0191] [

J1939Tp shall use the following service function to report development errors to the Diagnostic Event Manager:

```
void Dem_ReportErrorStatus(EventId, EventStatus) ] ( )
```

[J1939TP0082] [

After a task was completed successfully by calling PduR_J1939TpTxConfirmation() or PduR_J1939TpRxIndication(), the J1939Tp module shall call DEM with EventId J1939TP_E_COMMUNICATION and EventStatus DEM_EVENT_STATUS_PREPASSED.

] (BSW00422)

[J1939TP0083] [

If the task was aborted (e.g. Tr, T1, T2, T3, T4 timeout, Abort frame received), the J1939Tp module shall call the DEM with EventId J1939TP_E_COMMUNICATION and EventStatus DEM_EVENT_STATUS_PREFAILED.] (BSW00422)

The DEM should use a debounce algorithm of type *Counter based* for J1939Tp errors.

7.9 Debugging of J1939Tp

[J1939TP0164] [

Each variable that shall be accessible by AUTOSAR debugging shall be defined as global variable.] (BSW00442)

[J1939TP0088] [

All type definitions of variables that shall be debugged shall be accessible by the header file J1939Tp.h.] (BSW00442)

[J1939TP0166] [

The declaration of variables in the header file shall be such that it is possible to calculate the size of the variables by using the “sizeof” operator of C.] (BSW00442)

[J1939TP0167] [

Variables available for debugging shall be described in the respective Basic Software Module Description] (BSW00442)

8 API Specification

8.1 Imported Types

This section lists all externally defined types that are used by J1939Tp. These types are included from the headers corresponding to the module names listed in the table below.

Module	Imported Type
ComStack_Types	BufReq_ReturnType
	NotifResultType
	PduIdType
	PduInfoType
	PduLengthType
	RetryInfoType
	TPParameterType
Dem	Dem_EventIdType
	Dem_EventStatusType
Std_Types	Std_ReturnType
	Std_VersionInfoType

[J1939TP0085] [

J1939Tp shall use the imported types without any type redefinition.] ()

8.2 Type Definitions

This section lists the types defined by J1939Tp.

[J1939TP0175] [

Name:	J1939Tp_ConfigType	
Type:	Structure	
Range:	implementation specific	The content of the initialization data structure is implementation specific.
Description:	Data structure containing post-build configuration data of J1939-TP.	

The J1939Tp_ConfigType defines a structure that contains configuration parameters J1939Tp uses at run time. It is provided as an argument to J1939Tp_Init().] ()

[J1939TP0086] [

If, for implementation reasons, some additional types are required, they shall adhere to the following naming scheme: J1939Tp_<TypeName>Type, where <TypeName> is the name of this type with the following restrictions:

- No underscore usage
- First letter of each word upper case, consecutive letters lower case.

The implementation shall ensure that these types are not visible outside of the J1939Tp module.] (BSW00305)

8.3 Function Definitions

This section defines a list of functions provided for upper layer modules. All these APIs shall provide the following DET Errors:

[J1939TP0017] [

If development error detection for the J1939Tp is enabled, all APIs using a pointer as parameter shall check the input pointer for being valid and raise the development error: E_PARAM_ADDRESS in case a pointer address NULL is received.] (BSW00323)

[J1939TP0188] [

If development error detection for the J1939Tp is enabled, all APIs using a SDU- or PDU-Identifier shall check the input Identifier and raise the development error: E_PARAM_ID in case the API has been called for a not configured PDU or SDU.] (BSW00323)

[J1939TP0025] [

If development error detection (DET) for the J1939Tp module is enabled, the J1939Tp module shall raise an error (J1939TP_E_UNINIT) when any function excepting J1939Tp_GetVersionInfo is called before the function J1939Tp_Init() has been called.] (BSW00407, BSW42400011)

8.3.1 J1939Tp_Init

[J1939TP0087] [

Service name:	J1939Tp_Init
Syntax:	void J1939Tp_Init(const J1939Tp_ConfigType* ConfigPtr)
Service ID[hex]:	0x01
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	ConfigPtr Pointer to configuration data structure.
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	This function initializes the J1939Tp module.

After power up, J1939Tp is in a state called J1939TP_OFF. In this state J1939Tp is not yet configured, and therefore cannot perform any communication task. The J1939Tp module's environment (usually the ECU Manager) will call J1939Tp_Init() before using the J1939Tp module for further processing.] (BSW101, BSW00358)

[J1939TP0024] [

The function `J1939Tp_Init()` shall initialize all global variables of the module and reset all transport protocol connections.] (BSW42400010)

[J1939TP0022] [

`J1939Tp_Init()` shall change to the internal state `J1939TP_ON` after successful initialization.] ()

[J1939TP0026] [

If called when the J1939Tp module is in the global state `J1939TP_ON`, the function `J1939Tp_Init()` shall raise the DET error `J1939TP_E_REINIT` and do nothing.] ()

[J1939TP0187] [

The provided `ConfigPtr` shall only be used, when post-build configuration is enabled, or when different configuration variants must be supported. Otherwise, the parameters should be accessed directly to avoid indirection via the `ConfigPtr`.] (BSW00400, BSW00405, BSW00414, BSW00438)

The structure of type `J1939Tp_ConfigType` pointed to by the `ConfigPtr` contains post-build parameters of the J1939Tp module. In link time or pre-compile configured environments, the structure may contain a set of post-build parameters that differ between several configuration variants. It is expected that link time and pre-compile parameters may not change for different configuration variants.

`J1939Tp_Init()` has no return value because configuration data errors should be detected during configuration time (e.g. by the configuration tools). Furthermore, if a hardware error occurs, it will be reported via the error manager modules.

8.3.2 J1939Tp_Shutdown

[J1939TP0093] [

Service name:	J1939Tp_Shutdown
Syntax:	void J1939Tp_Shutdown(void)
Service ID[hex]:	0x02
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	This function is used to shutdown the J1939Tp module.

To close down communication, the state handling (usually the ECU Manager) calls `J1939Tp_Shutdown()`.] (BSW00336)

[J1939TP0094] [

J1939Tp_Shutdown() shall close all pending transport protocol connections, free all resources and set the J1939Tp module into the global state J1939TP_OFF state.

] ()

[J1939TP0095] [

J1939Tp_Shutdown() shall not raise a notification about the pending frame transmission or reception.] ()

8.3.3 J1939Tp_GetVersionInfo

[J1939TP0089] [

Service name:	J1939Tp_GetVersionInfo
Syntax:	void J1939Tp_GetVersionInfo(Std_VersionInfoType* VersionInfo)
Service ID[hex]:	0x03
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	VersionInfo Pointer to the location where the version information of J1939Tp shall be stored.
Return value:	None
Description:	Returns the version information of J1939Tp.

] (BSW00407)

[J1939TP0092] [

This function shall return the version and module information of the J1939Tp implementation. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers] (BSW00407)

[J1939TP0090] [

The existence of this function shall be pre-compile time configurable by the configuration parameter J1939TpVersionInfoApi (see J1939TP0051_Conf :).] (BSW00407, BSW171)

Note that the function J1939Tp_GetVersionInfo may be called in global state J1939TP_OFF, i.e. before initialization of the J1939Tp module.

8.3.4 J1939Tp_Transmit

[J1939TP0096] [

Service name:	J1939Tp_Transmit
----------------------	------------------

Syntax:	Std_ReturnType J1939Tp_Transmit(PduIdType TxSduId, const PduInfoType* TxInfoPtr)	
Service ID[hex]:	0x05	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	TxSduId	ID of the J1939Tp N-SDU to be transmitted.
	TxInfoPtr	Pointer to structure with length and content of the J1939Tp N-SDU that shall be transmitted. The content of this structure is not used, instead the content is requested dynamically via PduR_J1939TpCopyTxData().
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: The request has been accepted. E_NOT_OK: The request failed. This happens when a resource could not be allocated, e. g. when the requested transmission would use a channel that is currently active.
Description:	This service function is used to request the transfer of a J1939Tp N-SDU.	

] (BSW00357)

As described in **J1939TP0119** and **J1939TP0032**, the J1939Tp module will notify the upper layer by calling the PduR_J1939TpTxConfirmation callback when the transmit request has been completed.

[J1939TP0101] [

The function J1939Tp_Transmit shall reject a request, if the J1939Tp_Transmit service is called for an N-SDU identifier that is being used in a currently running J1939 Transport Layer session.] ()

8.3.5 J1939Tp_CancelTransmit

[J1939TP0177] [

Service name:	J1939Tp_CancelTransmit	
Syntax:	Std_ReturnType J1939Tp_CancelTransmit(PduIdType TxSduId)	
Service ID[hex]:	0x09	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	TxSduId	ID of the J1939Tp N-SDU to be canceled.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: The request has been accepted. E_NOT_OK: The request failed. This happens when the provided N-SDU is currently not transmitted.
Description:	This service function is used to cancel the ongoing transmission of a J1939Tp N-SDU.	

] ()

[J1939TP0179] [

J1939Tp shall implement the API function J1939Tp_CancelTransmit. The implementation may choose to do nothing. PduR_J1939TpTxConfirmation must be

called with NTFRSLT_E_CANCELATION_NOT_OK instead of NTFRSLT_E_OK in this case.] ()

8.3.6 J1939Tp_CancelReceive

[J1939TP0176] [

Service name:	J1939Tp_CancelReceive
Syntax:	Std_ReturnType J1939Tp_CancelReceive(PduIdType RxSduId)
Service ID[hex]:	0x0a
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	RxSduId ID of the J1939Tp N-SDU to be canceled. This ID is the I-PDU ID that is used to notify the reception of a J1939Tp N-SDU to PduR via PduR_J1939TpRxIndication.
Parameters (inout):	None
Parameters (out):	None
Return value:	Std_ReturnType E_OK: The request has been accepted. E_NOT_OK: The request failed. This happens when the provided I-PDU does not refer to an N-SDU that is currently received.
Description:	This service function is used to cancel the ongoing reception of a J1939Tp N-SDU.

] ()

[J1939TP0178] [

J1939Tp shall implement the API function J1939Tp_CancelReceive. The implementation may choose to do nothing. PduR_J1939TpTxConfirmation must be called with NTFRSLT_E_CANCELATION_NOT_OK instead of NTFRSLT_E_OK in this case.] ()

8.3.7 J1939Tp_ChangeParameter

[J1939TP0180] [

Service name:	J1939Tp_ChangeParameter
Syntax:	Std_ReturnType J1939Tp_ChangeParameter(PduIdType SduId, TPParameterType Parameter, uint16 Value)
Service ID[hex]:	0x08
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	SduId ID of the N-SDU for which parameters should be changed. For Tx parameters, this is the J1939TpTxNSduId. For Rx parameters, the SduId is the PduRSourcePduHandleId of PduR. Parameter ID of parameter that should be changed. Value New value for changed parameter.
Parameters (inout):	None
Parameters (out):	None
Return value:	Std_ReturnType E_OK: The request has been accepted. E_NOT_OK: The request failed. This happens when the provided

	parameter does not exist.
Description:	This service function is used to change reception or transmission parameters of J1939Tp for a specific N-SDU.

] ()

[J1939TP0181] [

The API J1939Tp_ChangeParameter shall only be implemented as a dummy function. It shall return immediately with E_NOT_OK to signalize that currently no parameter can be changed via this API.] ()

8.4 Callback Notifications

This is a list of functions provided for other modules. The function prototypes of the callback functions shall be provided in the file J1939Tp_Cbk.h.

8.4.1 J1939Tp_RxIndication

[J1939TP0108] [

Service name:	J1939Tp_RxIndication
Syntax:	void J1939Tp_RxIndication(PduIdType RxPduId, PduInfoType* PduInfoPtr)
Service ID[hex]:	0x42
Sync/Async:	Synchronous
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.
Parameters (in):	RxPduId ID of the received I-PDU.
	PduInfoPtr Contains the length (SduLength) of the received I-PDU and a pointer to a buffer (SduDataPtr) containing the I-PDU.
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Indication of a received I-PDU from a lower layer communication module.

] (BSW00359, BSW00360)

[J1939TP0109] [

The J1939Tp module shall provide a J1939Tp_RxIndication() API to allow the CanIf to notify that a new N-PDU has been received.] (BSW42400021)

The data will be copied by the J1939Tp via the PDU structure PduInfoType. In this case, the N-PDU buffers are not global and are therefore distributed in the corresponding J1939 Transport Layer.

[J1939TP0110] [

The function J1939Tp_RxIndication shall be callable in interrupt context (it could be called from the CAN receive interrupt).] (BSW00333)

8.4.2 J1939Tp_TxConfirmation

[J1939TP0112] [

Service name:	J1939Tp_TxConfirmation
Syntax:	void J1939Tp_TxConfirmation(PduIdType TxPduId)
Service ID[hex]:	0x40
Sync/Async:	Synchronous
Reentrancy:	Reentrant for different Pdulds. Non reentrant for the same PduId.
Parameters (in):	TxPduId ID of the I-PDU that has been transmitted.
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	The lower layer communication module confirms the transmission of an I-PDU.

] (BSW00359, BSW00360)

[J1939TP0113] [

The J1939Tp module shall implement a J1939Tp_TxConfirmation API to allow the CanIf module to confirm that a TP related frame (TP.CM, TP.DT) or direct frame has been successfully transmitted to the J1939 network.] (BSW42400021)

[J1939TP0114] [

The function J1939Tp_TxConfirmation shall be callable in interrupt context (it could be called from the CAN transmit interrupt).] (BSW00333)

8.5 Scheduled Functions

The Basic Software Scheduler directly calls the functions listed in this section. Scheduled functions shall have no return value and no parameter, and need not be reentrant.

8.5.1 J1939Tp_MainFunction

[J1939TP0104] [

Service name:	J1939Tp_MainFunction
Syntax:	void J1939Tp_MainFunction(void)
Service ID[hex]:	0x04
Timing:	FIXED_CYCLIC
Description:	Main function of the J1939Tp. Used for scheduling purposes and timeout supervision.

] (BSW00373, BSW00376)

[J1939TP0106] [

The calling frequency of the function J1939Tp_MainFunction is determined by the parameter J1939TpMainFunctionPeriod (see **J1939TP0044_Conf** :).] ()

8.6 Expected Interfaces

In this section, all interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

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

[J1939TP0116] [

API function	Description
CanIf_CancelTransmit	This is a dummy API introduced for interface compatibility.
CanIf_Transmit	This service initiates a request for transmission of the CAN L-PDU specified by the CanTxPduId and CAN related data in the L-PDU structure.
Dem_ReportErrorStatus	Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function.
PduR_J1939TpCopyRxData	This function is called when a transport protocol module has data to copy for the receiving module. Several calls may be made during one transportation of an I-PDU. The service shall provide the currently available buffer size when invoked with info.SduLength equal to 0.
PduR_J1939TpCopyTxData	This function is called by the transport protocol module to query the transmit data of an I-PDU segment. Each call to this function copies the next part of the transmit data until TpDataState indicates TP_DATARETRY. In this case the API restarts to copy the data beginning at the location indicated by TpTxDataCnt. The service shall provide the size of the remaining data when invoked with info.SduLength equal to 0.
PduR_J1939TpRxIndication	Called by the transport protocol module after an I-PDU has been received successfully or when an error occurred. It is also used to confirm cancellation of an I-PDU.
PduR_J1939TpStartOfReception	This function will be called by the transport protocol module at the start of receiving an I-PDU. The I-PDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). The service shall provide the currently available maximum buffer size when invoked with TpSduLength equal to 0.
PduR_J1939TpTxConfirmation	This function is called by a transport protocol module after the I-PDU has been transmitted on its network, the result will reveal if the transmission was successful or not.

] ()

Please note: [J1939TP0116] only lists the API service function PduR_J1939TpGetAvailableTxBuffer of the PduR interface because it is a standard part of the PduR interface for TP layer modules. This function is not needed for J1939Tp and should not be used to implement the J1939Tp.

8.6.2 Optional Interfaces

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

[J1939TP0060] [

<i>API function</i>	<i>Description</i>
Det_ReportError	Service to report development errors.

] ()

9 Sequence Diagrams

The following sequence diagram shows the transmission of a large PG (N-SDU with length >8 bytes) using the CMDT transport protocol as an example of the interaction between the J1939Tp and its neighboring modules.

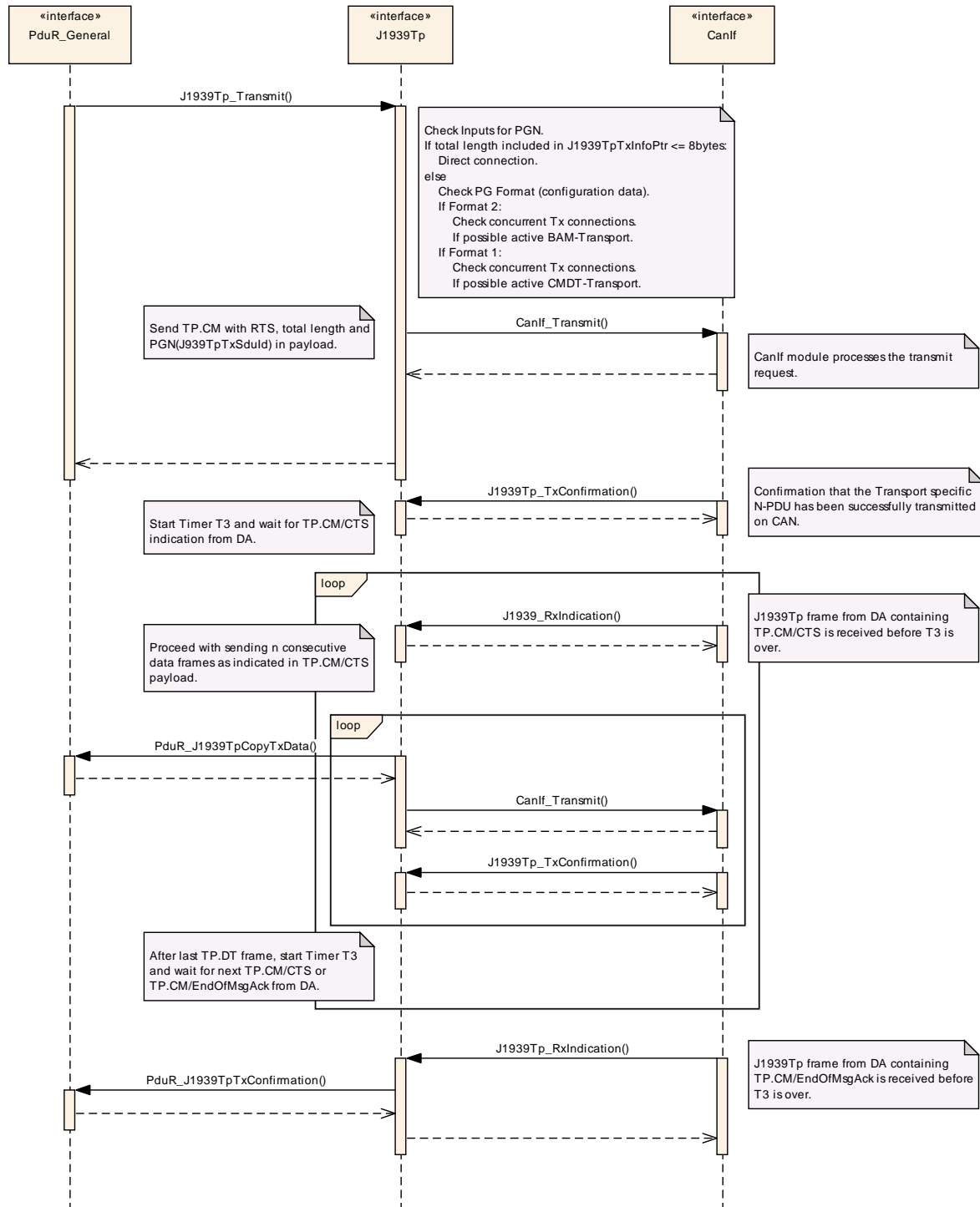


Figure 9-1: Successful N-SDU transmission

10 Configuration Specification

In general, this chapter defines configuration parameters and their clustering into containers. Section 10.1 describes fundamentals and the template used for the tables in section 10.2. Section 10.2 specifies the configuration structure (containers) and the configuration parameters of J1939Tp. Section 10.3 specifies published information of J1939Tp.

[J1939TP0049] [

The listed configuration items can be derived from a network description database, which is based on the System Description. The configuration tool should extract all relevant information to configure the J1939 Transport Protocol.] (BSW159)

[J1939TP0084] [

The consistency of the configuration must be checked by the configuration tool at configuration time. Configuration rules and constraints for plausibility checks will be performed where possible, during configuration time.] (BSW167)

10.1 How to Read this Chapter

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

- AUTOSAR Layered Software Architecture [2]
 - AUTOSAR ECU Configuration Specification [13]
- This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

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

10.1.1 Configuration and Configuration Parameters

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

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

10.1.2 Variants

Variants describe sets of configuration parameters. E.g., variant 1: only pre-compile time configuration parameters; variant 2: mix of pre-compile- and post build time-configuration parameters. In one variant a parameter can only be of one configuration class.

10.1.3 Containers

Containers structure the set of configuration parameters. This means:

- all configuration parameters are kept in containers.
- (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

10.1.4 Specification Template for Configuration Parameters

The following tables consist of three sections:

- the general section
- the configuration parameter section
- the section of included/referenced containers

SWS Item	Req-Number
Container Name	Identifies the container by a name, e.g., CanDriverConfiguration
Description	Explains the intention and the content of the container .
Configuration Parameters	

For each parameter of this container:

Name	Identifies the parameter by name. The naming convention shall follow BSW00408.>		
Description	Explains the intention of the configuration parameter.		
Type	Specify the type of the parameter (e.g., uint8..uint32) if possible or mark it "--"		
Unit	Specify the unit of the parameter (e.g., ms) if possible or mark it "--"		
Range	Specify the range (or possible values) of the parameter (e.g., 1..15, ON,OFF) if possible or mark it "--"	Describe the value(s) or ranges.	
Configuration Class	Pre-compile	see ¹	Refer here to (a) variant(s).
	Link time	see ²	Refer here to (a) variant(s).
	Post Build	see ³	Refer here to (a) variant(s).
Scope	Describe the scope of the parameter if known or mark it as "-". The scope describes the impact of the configuration parameter: Does the setting affect only one instance of the module (instance), all instances of this module (module), the ECU or a network. Possible values of scope : instance, module, ECU, network		
Dependency	Describe the dependencies with respect to the scope if known or mark it as "-".		

¹ see the explanation below this table - Pre-compile time

² see the explanation below this table - Link time

³ see the explanation below this table - Post Build

Included Containers		
Container Name	Multiplicity	Scope / Dependency
Reference a valid (sub)container by its name, e.g. CanController	Specifies the possible number of instances of the referenced container and its contained configuration parameters. Possible values: <multiplicity> <min_multiplicity..max_multiplicity>	Describe the scope of the referenced sub-container if known or mark it as "- -". The scope describes the impact of the configuration parameter: Does the setting affect only one instance of the module (instance), all instances of this module (module), the ECU or a network. Possible values of scope : instance, module, ECU, network Describe the dependencies with respect to the scope if known or mark it as "- -".

Pre-compile time - specifies whether the configuration parameter shall be of configuration class *Pre-compile time* or not

Label	Description
x	The configuration parameter shall be of configuration class <i>Pre-compile time</i> .
--	The configuration parameter shall never be of configuration class <i>Pre-compile time</i> .

Link time - specifies whether the configuration parameter shall be of configuration class *Link time* or not

Label	Description
x	The configuration parameter shall be of configuration class <i>Link time</i> .
--	The configuration parameter shall never be of configuration class <i>Link time</i> .

Post Build - specifies whether the configuration parameter shall be of configuration class *Post Build* or not

Label	Description
x	The configuration parameter shall be of configuration class <i>Post Build</i> and no specific implementation is required.
--	The configuration parameter shall never be of configuration class <i>Post Build</i> .

10.2 Containers and Configuration Parameters

The following sections summarize all configuration parameters. Additional information on the usage of these parameters can be found in chapter 7 and 8.

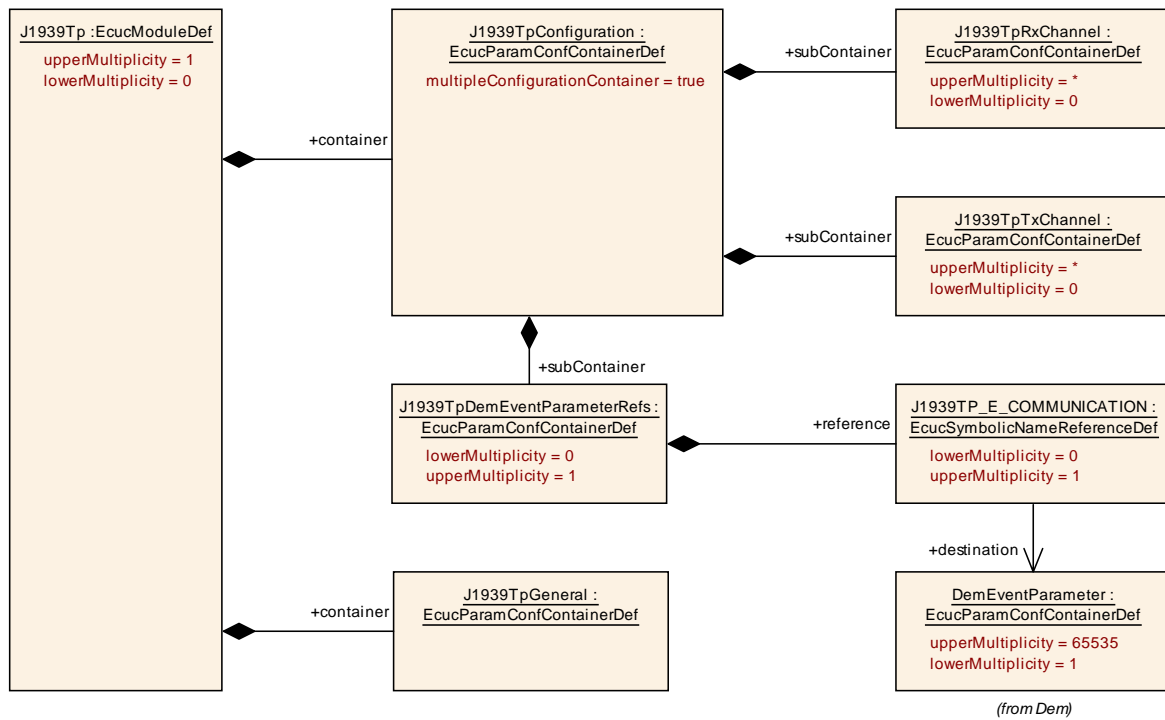


Figure 10-1: Module Configuration

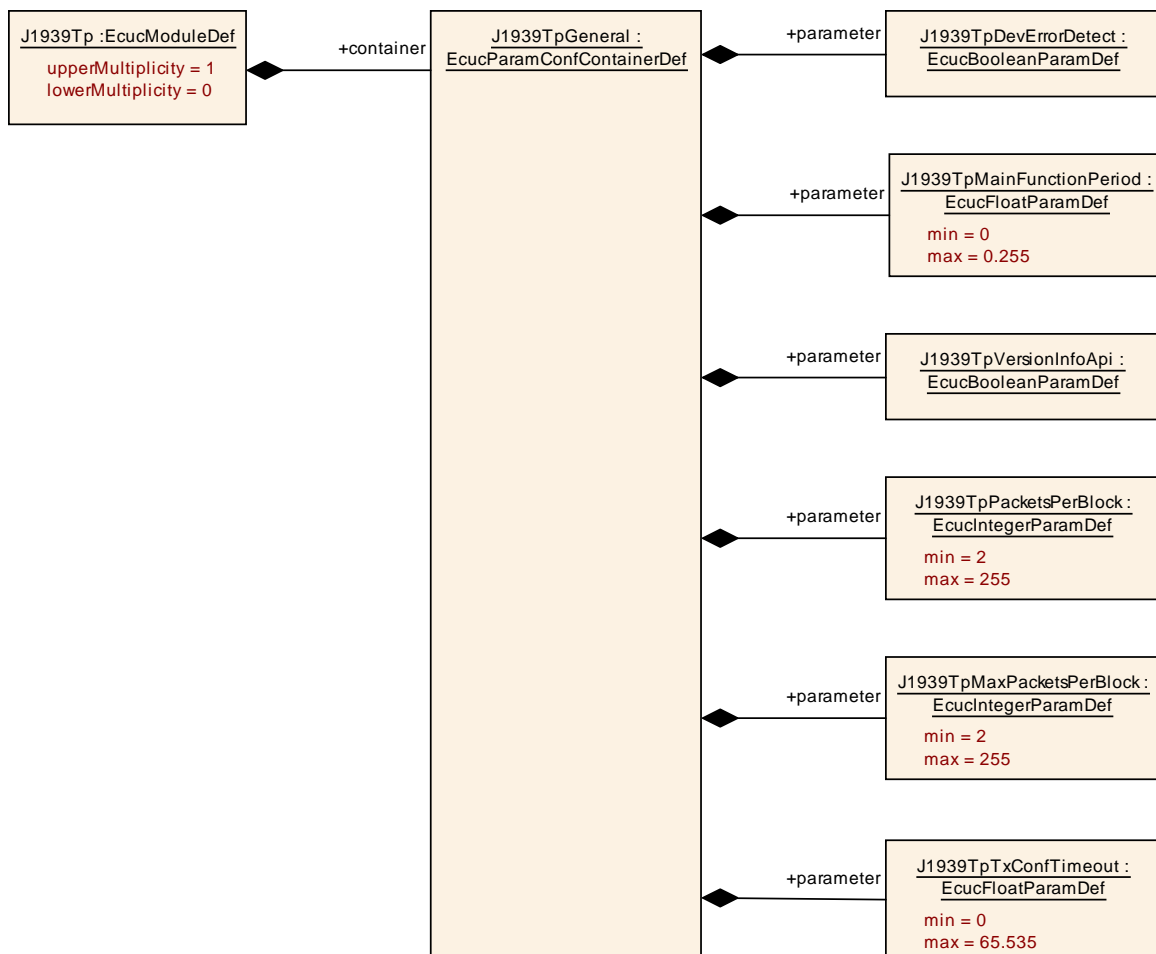


Figure 10-2: General Parameters

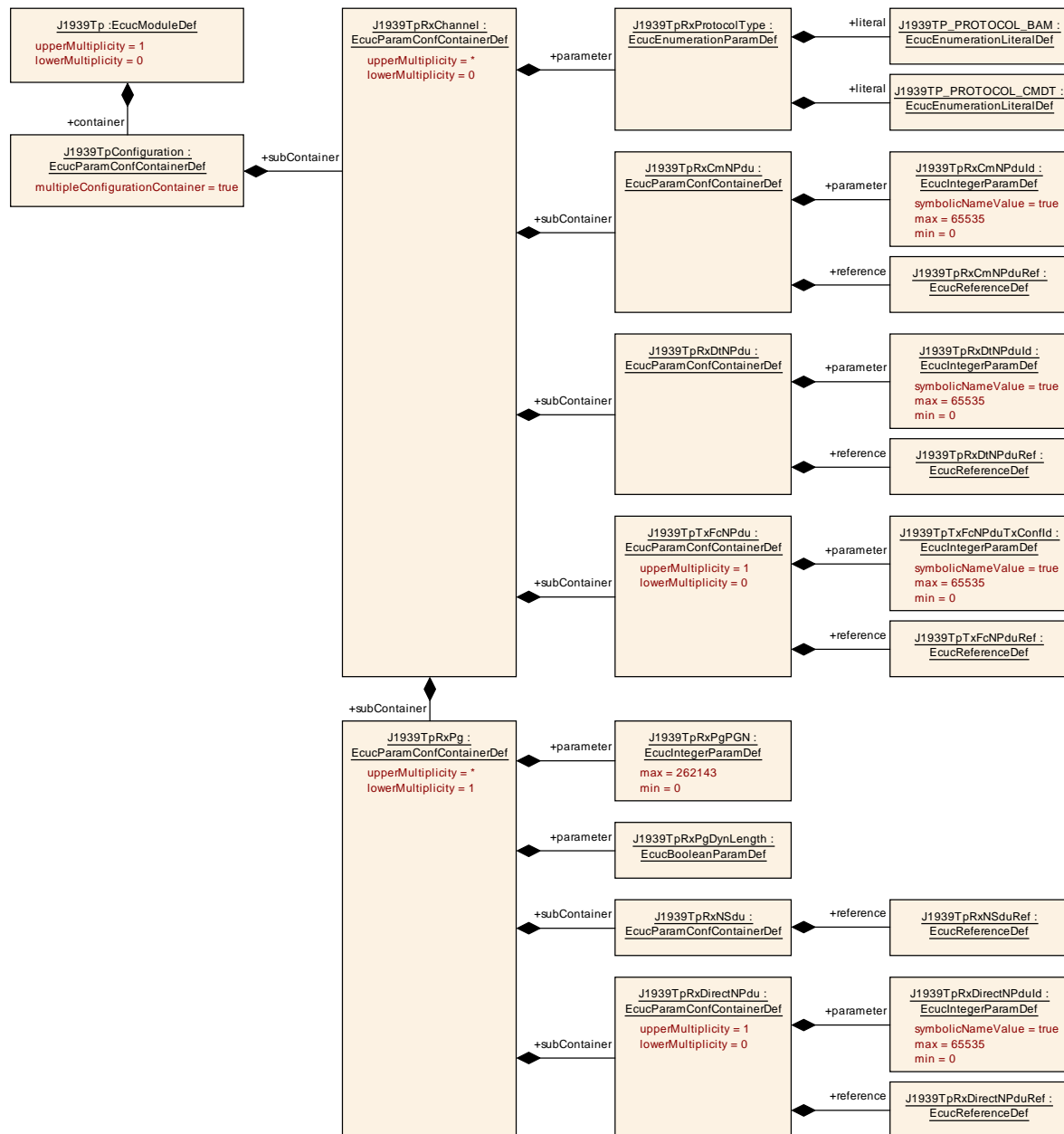


Figure 10-3: Configuration of Rx Channel

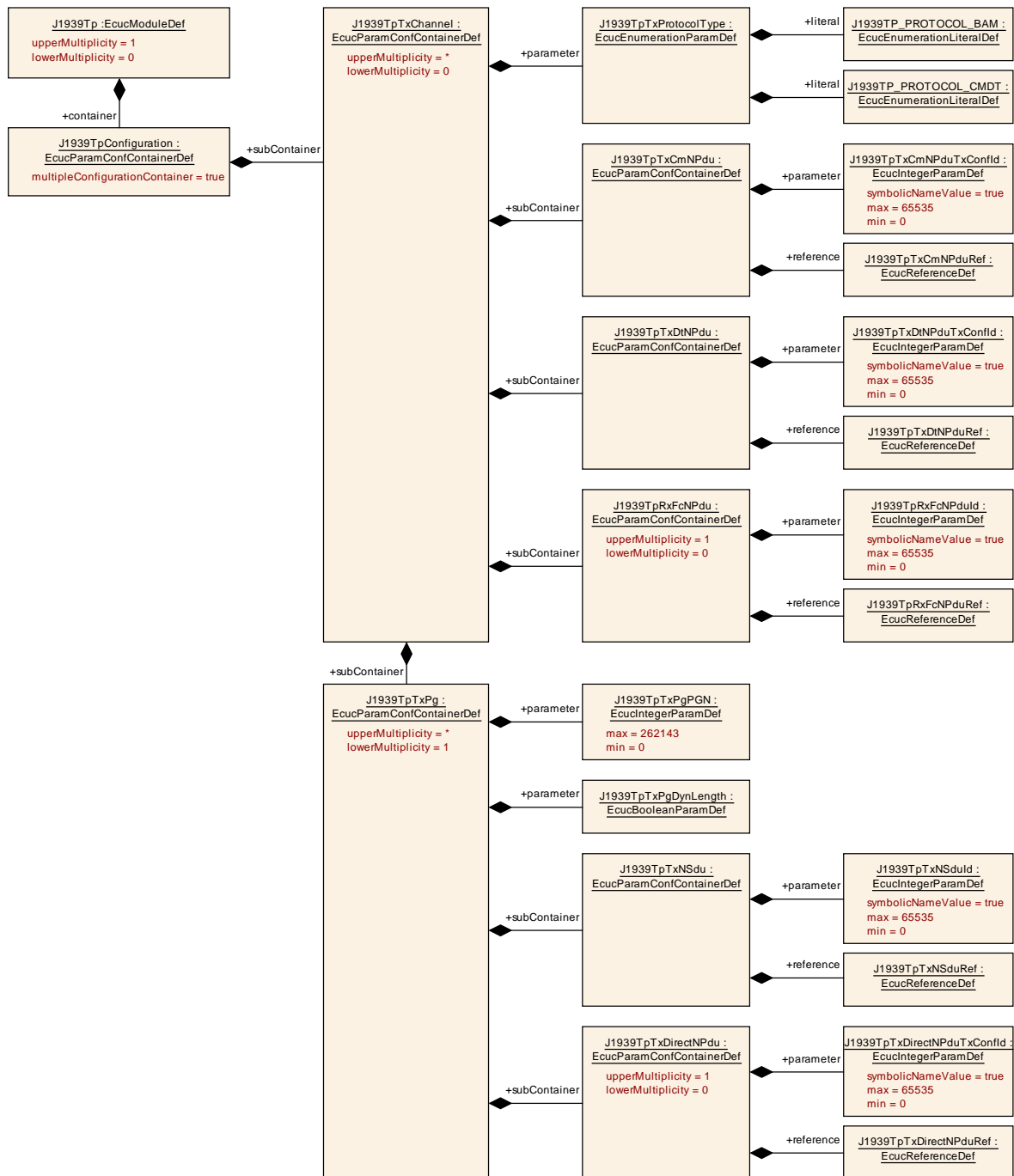


Figure 10-4: Configuration of Tx Channel

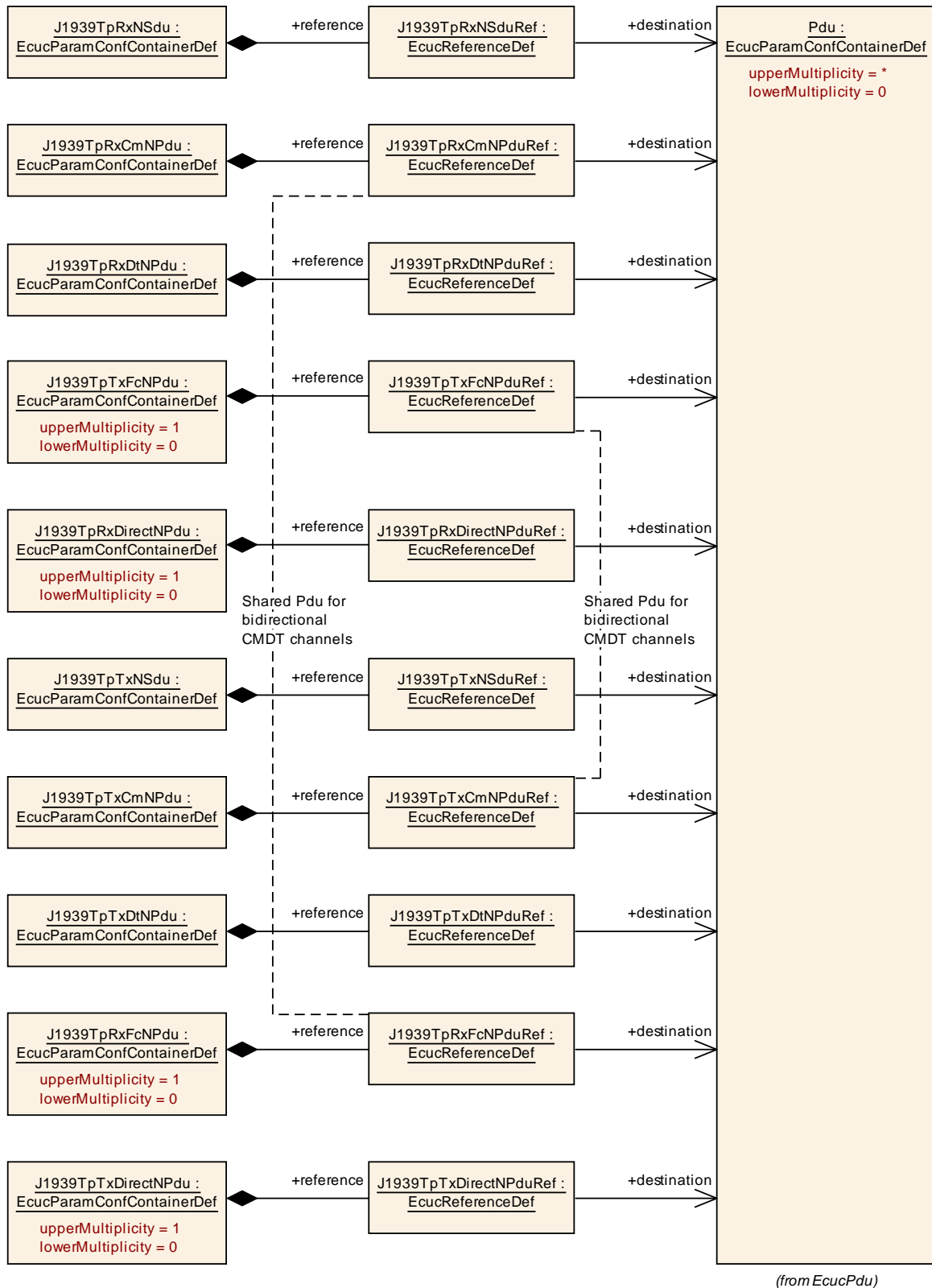


Figure 10-5: References to PDUs

10.2.1 J1939Tp

SWS Item	J1939TP0127_Conf :
Module Name	<i>J1939Tp</i>
Module Description	Configuration of the J1939Tp (J1939 Transport Protocol) module.

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939TpConfiguration	1	This container contains the configuration parameters and sub containers of the J1939Tp module that define the communication paths. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.
J1939TpGeneral	1	This container describes the general configuration parameters of the J1939Tp module.

10.2.2 J1939TpConfiguration

SWS Item	J1939TP0052_Conf :
Container Name	J1939TpConfiguration{J1939TP_CONFIGURATION} [Multi Config Container]
Description	This container contains the configuration parameters and sub containers of the J1939Tp module that define the communication paths. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939TpDemEventParameterRefs	0..1	Container for the references to DemEventParameter elements which shall be passed to the API Dem_ReportErrorStatus. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.
J1939TpRxChannel	0..*	This container describes a reception channel of the J1939Tp module. One channel is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF).
J1939TpTxChannel	0..*	This container describes a transmission channel of the J1939Tp module. One channel is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF).

10.2.3 J1939TpDemEventParameterRefs

SWS Item	J1939TP0182_Conf :
Container Name	J1939TpDemEventParameterRefs
Description	Container for the references to DemEventParameter elements which shall be passed to the API Dem_ReportErrorStatus. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized

	errors are provided in the container and can be extended by vendor specific error references.
Configuration Parameters	

SWS Item	J1939TP0183_Conf :		
Name	J1939TP_E_COMMUNICATION		
Description	Reference to the DemEventParameter which shall be issued after successful or unsuccessful communication.		
Multiplicity	0..1		
Type	Reference to [DemEventParameter]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.4 J1939TpRxChannel

SWS Item	J1939TP0053_Conf :
Container Name	J1939TpRxChannel{J1939TP_RX_CHANNEL}
Description	This container describes a reception channel of the J1939Tp module. One channel is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF).
Configuration Parameters	

SWS Item	J1939TP0029_Conf :	
Name	J1939TpRxProtocolType {J1939TP_RX_PROTOCOL_TYPE}	
Description	Protocol type of the referencing connection, which is either BAM or CMDT. Please note: The protocol type is determined at configuration time by the PDU format of the PGN of an N-SDU: If the first byte of the PGN is smaller than 0xF0, CMDT will be used, otherwise BAM.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	J1939TP_PROTOCOL_BAM	J1939 transport protocol type BAM (Broadcast Announce Message). This protocol uses two N-PDUs: The CmNPdu and the DtNPdu.
	J1939TP_PROTOCOL_CMDT	J1939 transport protocol type CMDT (Connection Mode Data Transfer). This protocol uses three N-PDUs: The CmNPdu, the DtNPdu, and the

		FcNPdu.
ConfigurationClass	Pre-compile time	X All Variants
	Link time	--
	Post-build time	--
Scope / Dependency		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939TpRxCmNPdu	1	This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection.
J1939TpRxDtNPdu	1	This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU.
J1939TpRXPg	1..*	--
J1939TpTxFcNPdu	0..1	This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT.

10.2.5 J1939TpRxCmNPdu

SWS Item	J1939TP0128_Conf :
Container Name	J1939TpRxCmNPdu{J1939TP_RX_CM_NPDU}
Description	This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection.
Configuration Parameters	

SWS Item	J1939TP0129_Conf :		
Name	J1939TpRxCmNPduId {J1939TP_RX_CM_NPDU_ID}		
Description	The N-PDU identifier used for communication with CanIf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0158_Conf :		
Name	J1939TpRxCmNPduRef {J1939TP_RX_CM_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE

	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.6 J1939TpRxDTNPdu

SWS Item	J1939TP0117_Conf :
Container Name	J1939TpRxDtNPdu{J1939TP_RX_DT_NPDU}
Description	This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU.
Configuration Parameters	

SWS Item	J1939TP0133_Conf :		
Name	J1939TpRxDTNPduId {J1939TP_RX_DT_NPDU_ID}		
Description	The N-PDU identifier used for communication with CanIf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0134_Conf :		
Name	J1939TpRxDTNPduRef {J1939TP_RX_DT_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.7 J1939TpRxPg

SWS Item	J1939TP0050_Conf :
Container Name	J1939TpRxPg{J1939TP_RX_PG}
Description	--
Configuration Parameters	

SWS Item	J1939TP0066_Conf :		
Name	J1939TpRxPgDynLength {J1939TP_RX_PG_DYNLENGTH}		
Description	This flag is set to TRUE when the N-SDU refers to a PGN with variable length. Please note: When this attribute is TRUE, the sub container J1939TpRxDirectNPdu is required.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0065_Conf :		
Name	J1939TpRxPgPGN {J1939TP_RX_PG_PGN}		
Description	Defines the PGN which is represented by the N-SDU.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 262143		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939TpRxDirectNPdu	0..1	This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less than 8 bytes. Please note: This sub container is only necessary when J1939TpRxPgDynLength is TRUE.
J1939TpRxNSdu	1	This container describes the parameters that are relevant for the reception of a specific N-SDU.

10.2.8 J1939TpRxDirectNPdu

SWS Item	J1939TP0130_Conf :
Container Name	J1939TpRxDirectNPdu{J1939TP_RX_DIRECT_NPDU}
Description	<p>This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less than 8 bytes.</p> <p>Please note: This sub container is only necessary when J1939TpRxPgDynLength is TRUE.</p>
Configuration Parameters	

SWS Item	J1939TP0131_Conf :		
Name	J1939TpRxDirectNPduId {J1939TP_RX_DIRECT_NPDU_ID}		
Description	The N-PDU identifier used for communication with CanIf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time		

	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0132_Conf :		
Name	J1939TpRxDirectNPduRef {J1939TP_RX_DIRECT_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.9 J1939TpRxNSdu

SWS Item	J1939TP0063_Conf :
Container Name	J1939TpRxNSdu{J1939TP_RX_NSDU}
Description	This container describes the parameters that are relevant for the reception of a specific N-SDU.
Configuration Parameters	

SWS Item	J1939TP0069_Conf :		
Name	J1939TpRxNSduRef {J1939TP_RX_NSdu_REF}		
Description	Reference to the Pdu object representing the N-SDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.10 J1939TpTxFcNPdu

SWS Item	J1939TP0135_Conf :		
Container Name	J1939TpTxFcNPdu{J1939TP_TX_FC_NPDU}		
Description	<p>This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection.</p> <p>Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT.</p>		

Configuration Parameters

SWS Item	J1939TP0168_Conf :		
Name	J1939TpTxFcNPduTxConfId {J1939TP_TX_FC_NPDU_TXCONF_ID}		
Description	The N-PDU identifier used for Tx confirmation from CanIf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0136_Conf :		
Name	J1939TpTxFcNPduRef {J1939TP_TX_FC_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU. Please note: When two channels have identical but exchanged source and destination addresses, the Pdu referenced by this parameter is shared with J1939TpTxCmNPduRef of the corresponding J1939TpTxChannel.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.11 J1939TpTxChannel

SWS Item	J1939TP0059_Conf :
Container Name	J1939TpTxChannel{J1939TP_TX_CHANNEL}
Description	This container describes a transmission channel of the J1939Tp module. One channel is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF).
Configuration Parameters	

SWS Item	J1939TP0137_Conf :
Name	J1939TpTxProtocolType {J1939TP_TX_PROTOCOL_TYPE}
Description	Protocol type of the referencing connection, which is either BAM or CMDT. Please note: The protocol type is determined at configuration time by the PDU format of the PGN of an N-SDU: If the first byte of the PGN is smaller than 0xF0, CMDT will be used, otherwise BAM.
Multiplicity	1
Type	EcucEnumerationParamDef

Range	J1939TP_PROTOCOL_BAM	J1939 transport protocol type BAM (Broadcast Announce Message). This protocol uses two N-PDUs: The CmNPdu and the DtNPdu.	
	J1939TP_PROTOCOL_CMDT	J1939 transport protocol type CMDT (Connection Mode Data Transfer). This protocol uses three N-PDUs: The CmNPdu, the DtNPdu, and the FcNPdu.	
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939TpRxFcNPdu	0..1	This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT.
J1939TpTxCmNPdu	1	This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection.
J1939TpTxDtNPdu	1	This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU.
J1939TpTxPg	1..*	--

10.2.12 J1939TpRxFcNPdu

SWS Item	J1939TP0144_Conf :
Container Name	J1939TpRxFcNPdu{J1939TP_RX_FC_NPDU}
Description	This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT.
Configuration Parameters	

SWS Item	J1939TP0145_Conf :
Name	J1939TpRxFcNPduId {J1939TP_RX_FC_NPDU_ID}
Description	The N-PDU identifier used for communication with CanIf.

Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0146_Conf :		
Name	J1939TpRxFcNPduRef {J1939TP_RX_FC_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU. Please note: When two channels have identical but exchanged source and destination addresses, the Pdu referenced by this parameter is shared with J1939TpRxCMNPduRef of the corresponding J1939TpRxChannel.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.13 J1939TpTxCMNPdu

SWS Item	J1939TP0138_Conf :
Container Name	J1939TpTxCMNPdu{J1939TP_TX_CM_NPDU}
Description	This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection.
Configuration Parameters	

SWS Item	J1939TP0170_Conf :		
Name	J1939TpTxCMNPduTxConfId {J1939TP_TX_CM_NPDU_TXCONF_ID}		
Description	The N-PDU identifier used for Tx confirmation from Canlf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0139_Conf :
-----------------	---------------------------

Name	J1939TpTxCMNPduRef {J1939TP_TX_CM_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.14 J1939TpTxDtNPdu

SWS Item	J1939TP0142_Conf :
Container Name	J1939TpTxDtNPdu{J1939TP_TX_DT_NPDU}
Description	This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU.
Configuration Parameters	

SWS Item	J1939TP0171_Conf :		
Name	J1939TpTxDtNPduTxConfId {J1939TP_TX_DT_NPDU_TXCONF_ID}		
Description	The N-PDU identifier used for Tx confirmation from Canlf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0143_Conf :		
Name	J1939TpTxDtNPduRef {J1939TP_TX_DT_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.15 J1939TpTxPg

SWS Item	J1939TP0070_Conf :
Container Name	J1939TpTxPg{J1939TP_TX_PG}
Description	--
Configuration Parameters	

SWS Item	J1939TP0148_Conf :		
Name	J1939TpTxPgDynLength {J1939TP_TX_PG_DYNLENGTH}		
Description	This flag is set to TRUE when the N-SDU refers to a PGN with variable length. Please note: When this attribute is TRUE, the sub container J1939TpTxDirectNPdu is required.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0150_Conf :		
Name	J1939TpTxPgPGN {J1939TP_TX_PG_PGN}		
Description	Defines the PGN which is represented by the N-SDU.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 262143		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939TpTxDirectNPdu	0..1	This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less than 8 bytes. Please note: This sub container is only necessary when J1939TpTxPgDynLength is TRUE.
J1939TpTxNSdu	1	This container describes the parameters that are relevant for the transmission of a specific N-SDU.

10.2.16 J1939TpTxDirectNPdu

SWS Item	J1939TP0140_Conf :
Container Name	J1939TpTxDirectNPdu{J1939TP_TX_DIRECT_NPDU}
Description	This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less than 8 bytes. Please note: This sub container is only necessary when J1939TpTxPgDynLength is TRUE.
Configuration Parameters	

SWS Item	J1939TP0169_Conf :
Name	J1939TpTxDirectNPduTxConfId

	{J1939TP_TX_DIRECT_NPDU_TXCONF_ID}		
Description	The N-PDU identifier used for Tx confirmation from Canlf.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0141_Conf :		
Name	J1939TpTxDirectNPduRef {J1939TP_TX_DIRECT_NPDU_REF}		
Description	Reference to the Pdu object representing the N-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.17 J1939TpTxNSdu

SWS Item	J1939TP0147_Conf :
Container Name	J1939TpTxNSdu{J1939TP_TX_NSDU}
Description	This container describes the parameters that are relevant for the transmission of a specific N-SDU.
Configuration Parameters	

SWS Item	J1939TP0149_Conf :		
Name	J1939TpTxNSduId {J1939TP_TX_NSUDU_ID}		
Description	The N-SDU identifier used for communication with PduR.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

SWS Item	J1939TP0151_Conf :		
Name	J1939TpTxNSduRef {J1939TP_TX_NSUDU_REF}		
Description	Reference to the Pdu object representing the N-SDU.		
Multiplicity	1		
Type	Reference to [Pdu]		

ConfigurationClass	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	--	
	Post-build time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
Scope / Dependency			

No Included Containers

10.2.18 J1939TpGeneral

SWS Item	J1939TP0033_Conf :
Container Name	J1939TpGeneral{J1939TP_GENERAL}
Description	This container describes the general configuration parameters of the J1939Tp module.
Configuration Parameters	

SWS Item	J1939TP0042_Conf :		
Name	J1939TpDevErrorDetect {J1939TP_DEV_ERROR_DETECT}		
Description	Switches the Development Error Detection and Notification.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0044_Conf :		
Name	J1939TpMainFunctionPeriod {J1939TP_MAIN_FUNCTION_PERIOD}		
Description	Allow to configure the time for the MainFunction (in seconds). Please note: This configuration value shall be equal to the value in the SchedulerManager module.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. 0.255		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0122_Conf :
Name	J1939TpMaxPacketsPerBlock {J1939TP_MAX_PACKETS_PER_BLOCK}
Description	Maximum number of N-PDUs the J1939Tp shall send before waiting for an authorization to continue transmission of the following TP.DT frames. This parameter is only relevant the transmission of messages via CMTD. For further details on this parameter value see SAE J1939/21.
Multiplicity	1
Type	EcucIntegerParamDef
Range	2 .. 255

Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0061_Conf :		
Name	J1939TpPacketsPerBlock {J1939TP_PACKETS_PER_BLOCK}		
Description	Number of N-PDUs the J1939Tp module allows the sender to send before waiting for an authorization to continue transmission of the following TP.DT frames. This parameter is only relevant for reception of messages via CMTD. For further details on this parameter value see SAE J1939/21.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	2 .. 255		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0006_Conf :		
Name	J1939TpTxConfTimeout {J1939TP_TX_CONF_TIMEOUT}		
Description	Timeout in seconds for the CanIf Tx confirmation. After this time the J1939Tp assumes that an N-PDU could not be transmitted. Please note: The Tx confirmation timeout should be set to a value that enabled detection of a lost Tx confirmation in time, and that ensures that normal transmission delay caused by lower message priority does not lead to an error.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. 65.535		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	J1939TP0051_Conf :		
Name	J1939TpVersionInfoApi {J1939TP_VERSION_INFO_API}		
Description	The function J1939Tp_GetVersionInfo is configurable (On/Off) by this configuration parameter.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.3 Published Information

[J1939TP0196] [

The standardized common published parameters as required by BSW00402 in the General Requirements on Basic Software Modules [3] shall be published within the header file of this module and need to be provided in the BSW Module Description. The according module abbreviation can be found in the List of Basic Software Modules [1].] (BSW003, BSW00318, BSW00371, BSW00374, BSW00402, BSW00441)

Additional module-specific published parameters are listed below if applicable.

11 Not applicable requirements

[J1939TP0197] [These requirements are not applicable to this specification.]

(BSW005, BSW161, BSW162, BSW164, BSW168, BSW170, BSW00314, BSW00325, BSW00326, BSW00341, BSW00347, BSW00375, BSW00377, BSW00387, BSW00413, BSW00415, BSW00416, BSW00417, BSW00419, BSW00423, BSW00427, BSW00433, BSW00437, BSW00439, BSW00440, BSW00443, BSW00444, BSW00445, BSW00446, BSW00447, BSW00449, BSW00453, BSW00455)