| Document Title | Specification of IEEE1722 Transport Protocol Module |
|---|---|
| **Document Category** | Software specification (SWS) |
| **Document UID** | 1093 |
| **Document Status** | published |
| **Part of AUTOSAR Standard** | Classic Platform |
| **Part of Standard Release** | R26-11 |

| Document Change History | | | |
|---|---|---|---|
| **Date** | **Release** | **Changed by** | **Description** |
| 2026-11-26 | R26-11 | AUTOSAR Release Management | • API cleanup<br>• RS dependency removed<br>• Editorial changes |
| 2025-11-25 | R25-11 | AUTOSAR Release Management | • Consider format dependencies with "IEC 61883" and "IIDC" in ECUC parameter description.<br>• Typo's corrected. IEC68133 -> IEC61883 for related entities (e.g. datatypes) |
| 2024-11-27 | R24-11 | AUTOSAR Release Management | • Update module for a fully integration with the communication layers by LSduR |
| 2023-11-23 | R23-11 | AUTOSAR Release Management | • Initial release |

**Disclaimer**

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

# Table of Contents

1 Introduction and functional overview 9

2 Acronyms, Abbreviations and Definitions 10
2.1 Acronyms and abbreviations . . . 10
2.2 Definitions . . . 10
2.2.1 Stream . . . 10
2.2.2 IEEE1722 stream . . . 10
2.2.3 IEEE1722 stream id . . . 11
2.2.4 AVTP stream . . . 11
2.2.5 AVTPDU . . . 11
2.2.6 AVTPDU-header . . . 11
2.2.7 AVTPDU-common-header . . . 11
2.2.8 AVTPDU-common-stream-header . . . 12
2.2.9 AVTPDU-common-control-header . . . 12
2.2.10 AVTPDU-alternative-header . . . 12
2.2.11 AVTPDU-payload . . . 12
2.2.12 Stream data producer . . . 12
2.2.13 Stream data consumer . . . 12
2.2.14 AVTP presentation time . . . 13
2.2.15 Max transit time . . . 13
2.2.16 Media clock . . . 13
2.2.17 Media clock provider . . . 13
2.2.18 Media clock consumer . . . 13
2.2.19 ACF-stream . . . 14
2.2.20 ACF-message . . . 14
2.2.21 ACF-message-header . . . 14
2.2.22 ACF-message-payload . . . 14

3 Related documentation 15
3.1 Input documents & related standards and norms . . . 15
3.2 Related specification . . . 15

4 Constraints and assumptions 16
4.1 Limitations . . . 16
4.2 Applicability to car domains . . . 16

5 Dependencies to other modules 17
5.1 L-SDU router . . . 17
5.2 StbM . . . 17
5.3 EthIf . . . 17

6 Requirements Tracing 18

7 Functional specification 19
7.1 Overview . . . 19
7.2 Module Handling . . . 20
7.2.1 Initialization . . . 20
7.3 State handling . . . 21
7.3.1 State handling of streams . . . 21
7.3.2 State handling of PDUs . . . 23
7.4 Global time related handling . . . 24
7.5 Communication . . . 25
7.5.1 Transmission requests . . . 26
7.5.2 Internal transmission request processing . . . 28
7.5.3 Reception indications . . . 32
7.5.4 Internal reception indication processing . . . 33
7.5.5 Transmission confirmation . . . 36
7.5.6 Release reception buffer . . . 36
7.5.7 Mainfunction processing . . . 36
7.5.8 Buffer handling . . . 37
7.5.9 ACF-stream handling . . . 38
7.5.9.1 ACF-message collection and transmission . . . 39
7.5.9.2 L-SDU handling and forwarding . . . 43
7.5.10 AVTPDU-header creation . . . 46
7.5.10.1 Treatment of shared AVTPDU-header fields . . . 47
7.5.10.2 AVTPDU-common-header fields . . . 48
7.5.10.3 AVTPDU-common-stream-header fields . . . 49
7.5.10.4 AVTPDU-alternative-header fields . . . 51
7.5.10.5 61883_IIDC-header fields . . . 51
7.5.10.6 AAF-header fields . . . 53
7.5.10.7 ACF-header fields . . . 55
7.5.10.8 ACF-message creation . . . 56
7.5.10.9 CRF-header fields . . . 59
7.5.10.10 RVF-header fields . . . 61
7.5.11 AVTPDU-header inspection . . . 62
7.6 Meta data handling . . . 64
7.6.1 Meta data item types . . . 64
7.6.1.1 IEEE1722 common stream header . . . 64
7.6.1.2 IEEE1722 IEC61883/IIDC specific stream header . . . 65
7.6.1.3 IEEE1722 AAF specific stream header . . . 66
7.6.1.4 IEEE1722 RVF stream header . . . 67
7.6.1.5 IEEE1722 CRF stream header . . . 68
7.6.1.6 IEEE1722 ACF stream (NTSCF/TSCF) header . . . 69
7.6.1.7 IEEE1722 stream interaction with lower layer . . . 72
7.7 Error Classification . . . 73
7.7.1 Development Errors . . . 73

7.7.2 Runtime Errors . . . 74
7.7.3 Production Errors . . . 74
7.8 Security Events . . . 74
8 API specification 75
8.1 Imported types . . . 75
8.2 Type definitions . . . 75
8.2.1 IEEE1722Tp_ConfigType . . . 75
8.2.2 IEEE1722Tp_StreamIndexType . . . 76
8.2.3 IEEE1722Tp_StreamStateType . . . 76
8.2.4 IEEE1722Tp_CommonStreamHeaderType . . . 77
8.2.5 IEEE1722Tp_Txlec61883IidcType . . . 80
8.2.6 IEEE1722Tp_Rxlec61883IidcType . . . 81
8.2.7 IEEE1722Tp_Txlec61883Type . . . 82
8.2.8 IEEE1722Tp_Rxlec61883Type . . . 84
8.2.9 IEEE1722Tp_Txlec61883CipNoSphType . . . 88
8.2.10 IEEE1722Tp_Rxlec61883CipNoSphType . . . 90
8.2.11 IEEE1722Tp_Txlec61883CipWithSphType . . . 95
8.2.12 IEEE1722Tp_Rxlec61883CipWithSphType . . . 97
8.2.13 IEEE1722Tp_TxAafPcmType . . . 102
8.2.14 IEEE1722Tp_RxAafPcmType . . . 103
8.2.15 IEEE1722Tp_TxAafAes3Type . . . 105
8.2.16 IEEE1722Tp_RxAafAes3Type . . . 106
8.2.17 IEEE1722Tp_TxRvfType . . . 109
8.2.18 IEEE1722Tp_RxRvfType . . . 112
8.2.19 IEEE1722Tp_TxCrfType . . . 117
8.2.20 IEEE1722Tp_RxCrfType . . . 119
8.3 Function definitions . . . 122
8.3.1 Generic Functions . . . 122
8.3.1.1 IEEE1722Tp_Init . . . 122
8.3.1.2 IEEE1722Tp_DeInit . . . 123
8.3.1.3 IEEE1722Tp_Transmit . . . 124
8.3.1.4 IEEE1722Tp_TxConfirmation . . . 124
8.3.1.5 IEEE1722Tp_RxIndication . . . 125
8.3.1.6 IEEE1722Tp_ReleaseRxBuffer . . . 126
8.3.1.7 IEEE1722Tp_ActivateStream . . . 126
8.3.1.8 IEEE1722Tp_DeactivateStream . . . 127
8.3.2 API Parameter Checking . . . 127
8.4 Scheduled functions . . . 128
8.4.1 IEEE1722Tp_MainFunctionTx . . . 128
8.4.2 IEEE1722Tp_MainFunctionRx . . . 129
8.5 Expected interfaces . . . 129
8.5.1 Mandatory interfaces . . . 129
8.5.2 Optional interfaces . . . 130

8.6 Service Interfaces . . . 130
9 Sequence diagrams 131
9.1 IEEE1722 stream transmission . . . 131
9.2 IEEE1722 stream reception . . . 132
10 Configuration specification 133
10.1 Containers and configuration parameters . . . 133
10.1.1 IEEE1722Tp . . . 133
10.1.2 IEEE1722TpGeneral . . . 134
10.1.3 IEEE1722TpConfig . . . 138
10.1.3.1 IEEE1722Tp Lower Layer Pdu configuration . . . 139
10.1.4 IEEE1722Tp stream configuration . . . 144
10.1.4.1 IEEE1722Tp Upper Layer Pdu configuration . . . 149
10.1.4.2 IEEE1722TpStreamTxQueue . . . 155
10.1.4.3 IEEE1722TpStreamRxQueue . . . 156
10.1.4.4 IEEE1722TpStreamSubtype . . . 158
10.1.4.5 IEEE1722TpStreamCRF . . . 160
10.1.4.6 IEEE1722TpStreamAAF . . . 164
10.1.4.7 IEEE1722TpStreamIIDC . . . 174
10.1.4.8 IEEE1722TpStreamRVF . . . 179
10.1.4.9 IEEE1722TpStreamACF . . . 186
10.1.4.10 IEEE1722TpStreamAcfCan . . . 190
10.1.4.11 IEEE1722TpStreamAcfLin . . . 201
A Not applicable requirements 205
B Change history of AUTOSAR traceable items 206
B.1 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 206
B.1.1 Added Constraints in R24-11 . . . 206
B.1.2 Changed Constraints in R24-11 . . . 206
B.1.3 Deleted Constraints in R24-11 . . . 206
B.1.4 Added Specification Items in R24-11 . . . 207
B.1.5 Changed Specification Items in R24-11 . . . 207
B.1.6 Deleted Specification Items in R24-11 . . . 207
B.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 207
B.2.1 Added Constraints in R25-11 . . . 207
B.2.2 Changed Constraints in R25-11 . . . 207
B.2.3 Deleted Constraints in R25-11 . . . 207
B.2.4 Added Specification Items in R25-11 . . . 207
B.2.5 Changed Specification Items in R25-11 . . . 208
B.2.6 Deleted Specification Items in R25-11 . . . 208
B.3 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 208

B.3.1 Added Constraints in R26-11 . . . . . 208
B.3.2 Changed Constraints in R26-11 . . . . . 208
B.3.3 Deleted Constraints in R26-11 . . . . . 208
B.3.4 Added Specification Items in R26-11 . . . . . 208
B.3.5 Changed Specification Items in R26-11 . . . . . 208
B.3.6 Deleted Specification Items in R26-11 . . . . . 209

# Known Limitations

- IEEE1722Tp streams used for audio, video streaming, interaction with a media clock (clock reference format) and ACF-messages for CAN and LIN interact only with CDDs which act as stream data producer or stream data consumer. Exchange of data with AUTOSAR standardized BSW software modules (e.g. COM) is supported - RTE is not supported. This may be introduced after R24-11.

# 1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the AUTOSAR Basic Software module CP_SWS_1722Tp.

The purpose of the 1722Tp module is to provide the possibility to use the [1, IEEE1722] standardized transport protocol for time-sensitive applications in a bridged local area network on classic platform in AUTOSAR. The transport protocol defines so-called "AVTP streams" to exchange data between time-sensitive applications. Basically, the source of a stream is called "stream data producer" (or "talker") and the destination is called "stream data consumer" (or "listener"). Most likely one end node in an Ethernet network hosts a stream data producer and one or multiple other end node(s) host a stream data consumer. AVTP streams carry an AVTPDU-header where a so-called "AVTP presentation time" is available. The AVTP presentation time enables the possibility to handle data synchronously across a local area network at multiple stream data consumers. The IEEE1722 transport protocol defines several AVTP subtypes. The IEEE1722Tp module supports a subset of the AVTP subtypes to cover the following use cases:

- audio and video streaming
- distribution of a generated clock rate of a so-called media clock
- encapsulation of bus frames (e.g. CAN frames, LIN frames) and transport via an AVTP stream across the network

The 1722Tp module provide the possibility to configure, to transmit and receive AVTP streams.

# 2 Acronyms, Abbreviations and Definitions

The glossary below includes acronyms and abbreviations relevant to the IEEE1722Tp module that are not included in the [2, AUTOSAR glossary].

## 2.1 Acronyms and abbreviations

| Abbreviation / Acronym: | Description: |
|---|---|
| TSN | Time-Sensitve Networking |
| TSpec | Traffic Specification as defined by [3, IEEE802.1Q] |
| AVTP | Audio/Video Transport Protocol as defined by [1, IEEE1722] |
| AVTPDU | Audio/Video Transport Protocol Data Unit as defined by [1, IEEE1722] |
| IIDC | Instrumentation and Industrial Digital Camera as defined by [1, IEEE1722] |
| 61883_IIDC | IEC 61883/IIDC format as defined by [1, IEEE1722] |
| AAF | AVTP Audio Format as defined by [1, IEEE1722] |
| RVF | Raw Video Format as defined by [1, IEEE1722] |
| CRF | Control Reference Format as defined by [1, IEEE1722] |
| TSCF | Time Sensitive Control Format as defined by [1, IEEE1722] |
| NTSCF | None Time Sensitive Control Format as defined by [1, IEEE1722] |
| ACF | AVTP Control Format as defined by [1, IEEE1722] |
| ACF_CAN | Controller Area Network (CAN)/CAN with Flexible Data-Rate (CAN FD) message as defined by [1, IEEE1722] |
| ACF_CAN_BRIEF | Abbreviated CAN/CAN FD message as defined by [1, IEEE1722] |
| ACF_LIN | LIN® message as defined by [1, IEEE1722] |
| gPTP | generalized Precision Time Protocol [4, IEEE Std 802.1AS] |
| LL-PDU | Lower Layer PDU, which is used to interact with a lower layer module |
| UL-PDU | Upper Layer PDU, which is used to interact with an uper layer module |
| MTU | Maximum transmission unit |

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

## 2.2 Definitions

### 2.2.1 Stream

**Definition:** A "stream" represent multiple Ethernet frames which are grouped by similar frame attributes (e.g. MAC source address).

### 2.2.2 IEEE1722 stream

**Definition:** An "IEEE1722 stream" represents multiple Ethernet frames which have EtherType set to AVTP Ethertype (0x22F0) (see [1, IEEE1722]) and carry an AVTPDU. A single IEEE1722 stream carries a system-wide unique IEEE1722 stream id.

### 2.2.3 IEEE1722 stream id

**Definition:** An "IEEE1722 stream id" represents a system-wide unique IEEE1722 stream id (see [1, IEEE1722]) to identify a single stream. The stream id comprises a 48 bit MAC address and a 16 bit unsigned integer unique id. The unique id could be used to address specific TSN applications at a destination node.

### 2.2.4 AVTP stream

**Definition:** An "AVTP stream" has the same definition as IEEE1722 stream. It is just another term with the same meaning.

### 2.2.5 AVTPDU

**Definition:** An "AVTPDU" is defined by [1, IEEE1722] and represents the data which is transported within an IEEE1722 tream. The AVTPDU consists of an AVTPDU-header and the AVTPDU-payload.

### 2.2.6 AVTPDU-header

**Definition:** An "AVTPDU-header" is defined by [1, IEEE1722] and represents the first part of an AVTPDU. The first byte of the AVTPDU-header encode the format of the AVTPDU. Serveral formats are specified by [1, IEEE1722] and called "AVTP stream data subtype" (e.g. AAF (AVTP Audio Format)). All AVTP stream data subtypes share the same layout according the first 12 bits. The first 12 bits are defined as AVTPDU-common-header. The remaing AVTPDU-header information could differ per subtype. A subset of the stream data subtypes share the same format for the remaing AVTP-header information. Therefore [1, IEEE1722] define 3 different header formats which represent the remaing AVTP-header information: AVTPDU-common-stream-header, AVTPDU-common-control-header and AVTPDU-alternative-header. For example, an AAF-header denotes an AVTPDU of subtype "AVTP Audio format". The AAF-header uses the header fields of the AVTPDU-common-header and AVTPDU-common-stream-header.

### 2.2.7 AVTPDU-common-header

**Definition:** An "AVTPDU-common-header" is defined by [1, IEEE1722] and represents the first 12 bits (subtype (8 bits), header specific (1 bit), version (3 bits) of a AVTPDU-header. The AVTPDU-common-header contains the basic fields that all formats of AVTP stream data subtypes share.

### 2.2.8 AVTPDU-common-stream-header

**Definition:** An "AVTPDU-common-stream-header" is defined by [1, IEEE1722] and expands the AVTPDU-common-header used by a subset of AVTP stream data subtypes (e.g. 61883_IIDC, AAF, TSCF, RVF)

### 2.2.9 AVTPDU-common-control-header

**Definition:** An "AVTPDU-common-stream-header" is defined by [1, IEEE1722] and expands the AVTPDU-common-header used by a subset of AVTP stream data subtypes (e.g. ADP)

### 2.2.10 AVTPDU-alternative-header

**Definition:** An "AVTPDU-alternative-header" is defined by [1, IEEE1722] and used for AVTP stream data subtypes that do not exhibit the commonalities shared between formats that use the AVTPDU-common-stream-header or AVTPDU-common-control-headers. For example, CRF and NTSCF subtypes uses the AVTPDU-alternative-header.

### 2.2.11 AVTPDU-payload

**Definition:** An "AVTPDU-payload" is defined by [1, IEEE1722] and represents the second part of an AVTPDU. The AVTPDU-payload carry data of subtype encoded in the AVTPDU-header. For example, an AAF-payload carry audio data (i.e. audio samples).

### 2.2.12 Stream data producer

**Definition:** A "stream data producer" represent an end node in an Ethernet network which produces (continously) data. The data is transmitted via a stream and received by 1 or multiple end nodes (stream data consumer).
**Note:** The term "talker" is synonymous with "stream data producer".

### 2.2.13 Stream data consumer

**Definition:** A "stream data consumer" represent an end node in an Ethernet network which consumes (continously) data. The data is received via a stream.
**Note:** The term "listener" is synonymous with "stream data consumer".

### 2.2.14 AVTP presentation time

**Definition:** The "AVTP presentation time" is defined by [1, IEEE1722] and represents the gPTP time at which designated data within an AVTPDU payload is transferred to a time-sensitive application of an stream data consumer. An AVTPDU-header of header format AVTPDU-common-stream-header carries the presentation time as "avtp_timestamp" according to [1, IEEE1722]. AVTP presentation time is calculated as "TavtpPresentationTime" = "TcurrentGlobalTime" + "TmaxTransitTime". Please note: presentation time does not cover format conversion time and processing time of the receiving time-sensitive application (see [1, IEEE1722] figure 6 "Figure 6 - AVTP Timing Reference Planes").

### 2.2.15 Max transit time

**Definition:** "Max transit time" is defined by [1, IEEE1722]. The basic method to calculate an appropriate "Max Transit Time" is to take the worst-case transit time from the stream data producer (talker) to a stream data consumer (listener) and choose a max transit time that is greater than or equal to the largest worst-case transit time.

### 2.2.16 Media clock

**Definition:** "Media clock" is defined by [1, IEEE1722] and represents an entity which generate a rate (e.g. precise hardware clock with an constant rate (e.g. 48kHz). The media clock is hosted by the media clock provider.

### 2.2.17 Media clock provider

**Definition:** A "media clock provider" is an end node in the network which hosts an media clock. The media clock provider transmit an IEEE1722 stream to 1 or multiple media clock consumer. The IEEE1722 stream is of subtype CRF (clock reference format) and contain several presentation timestamps which correlates to the media clock rate.

### 2.2.18 Media clock consumer

**Definition:** A "media clock consumer" is an end node in the network which receive a IEEE1722 stream of subtype CRF (clock reference format) from a media clock provider. The media clock consumer perform a recovery of its media clock (e.g. PLL) based on the received encapsulated data from the media clock provider.

### 2.2.19 ACF-stream

**Definition:** An "ACF-stream" represents an IEEE1722 stream of subtype TSCF (time-synchronous control format) or NTSCF(non-time-synchronous control format) which transport encapsulated bus frames as ACF-messages (e.g. ACF_CAN) within its AVTPDU-payload (ACF-payload).

### 2.2.20 ACF-message

**Definition:** An "ACF-message" is defined by [1, IEEE1722]) and represents an encapsulated bus frame of a certain kind of bus type (e.g. CAN). The bus frame is encapsulate with an corresponding ACF-message type (e.g. ACF_CAN). The ACF-message consist of an ACF-message-header and an ACF-message-payload. Multipe ACF-messages of different ACF-message types could form an ACF-payload which is transported within the same ACF-stream.

### 2.2.21 ACF-message-header

**Definition:** An "ACF-message-header" represents the first part of an ACF-message. The first 7 bits represents the ACF-message type. Several ACF-message types are specified by [1, IEEE1722] (e.g. ACF_CAN, ACF_LIN). The following 9 bits represents the lenght of the subsequential ACF-message-payload.

### 2.2.22 ACF-message-payload

**Definition:** An "ACF-message-payload" is defined by [1, IEEE1722] and represents the second part of an ACF-message. The ACF-message-payload carry data of ACF-message type encoded in the ACF-message-header. For example, an ACF_CAN-payload carry an CAN2.0 frame.

# 3 Related documentation

## 3.1 Input documents & related standards and norms

[1] IEEE Standard 1722-2016 - IEEE Standard for a Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks

[2] Glossary
AUTOSAR_FO_TR_Glossary

[3] IEEE 802.1Q-2022 - IEEE Standard for Local and Metropolitan Area Network - Bridges and Bridged Networks
https://ieeexplore.ieee.org/

[4] IEEE 802.1AS-2011 - IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, Rev. 2011

[5] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[6] Specification of Linklayer Sdu Routing Module
AUTOSAR_CP_SWS_LSduRouter

[7] System Template
AUTOSAR_CP_TPS_SystemTemplate

## 3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [5, SWS BSW General], which is also valid for IEEE1722Tp.

Thus, the specification SWS BSW General shall be considered as additional and required specification for the IEEE1722Tp module.

# 4 Constraints and assumptions

## 4.1 Limitations

The IEEE1722Tp module support a subset of the AVTP stream data subtypes specified by [1, IEEE1722]:

- audio and video streaming: AAF, RVF, 61883_IIDC
- distribution of a generated clock rate provided by a media clock: CRF
- transport of encapsulated bus frames (ACF_CAN, ACF_CAN_BRIEF and ACF_LIN) via an ACF-stream, where the time-synchronous TSCF or the non-time-synchronous NTSCF AVTP stream data subtype is used in the ACF-header.

The IEEE1722Tp module is responsible to forward 1722Tp streams from the lower layers to stream data consumers, and from stream data producer to the lower layers. The time synchronous handling of the transported data with respect to the given AVTP presentation time is in responsibility of the according stream data consumer. Thus, the IEEE1722Tp module cannot ensure time synchronous handling with the accuracy of the AVTP presentation time in units of nanoseconds.

An ACF-stream with ACF-header set to TSCF (time-synchronous control format) carries an AVTP presentation time. The AVTP presentation time is given in units of nanoseconds. The IEEE1722Tp module can only perform a forwarding of bus frames with a resolution accuracy of the main function period (e.g. 5 ms). Please note, synchronicity of forwarded bus frames across multiple bus cluster highly depends on the surrounding infrastructur and software implementation, e.g. internal data processing, accuracy of the synchronized global time, busload.

## 4.2 Applicability to car domains

The IEEE1722Tp module can be used in all kinds of vehicles that feature Ethernet network and use IEEE1722 streams.

# 5 Dependencies to other modules

This section describes the relations to other modules and files within the AUTOSAR basic software architecture. It contains brief descriptions of configuration information, which are required by the IEEE1722Tp module from other modules.

## 5.1 L-SDU router

The IEEE1722Tp module uses APIs of the L-SDU router [6] to interchange data (L-SDUs).

## 5.2 StbM

[1, IEEE1722] specifies an AVTP presentation time which is available in each of the defined AVTP stream data subtypes, except for `NTSCF`. Upon a transmission request for a IEEE1722 stream the AVTP presentation time is calculate by the IEEE1722Tp based on the current synchronized global time and the configured max transit time. The StbM_GetCurrentTime() API could be used to access the current synchronized global time

## 5.3 EthIf

[1, IEEE1722] specifies an AVTP presentation time which is available in each of the defined AVTP stream data subtypes, except for `NTSCF`. The IEEE1722Tp module need access to the current synchronized time as described in Chapter 5.2. An alternative to call `StbM_GetCurrentTime` API is to call `EthIf_GetCurrentTimeTuple`, which may support more accurate time, since it directly gets the current synchronized global time directly from the according hardware clock.

# 6 Requirements Tracing

Empty section

| Requirement | Description | Satisfied by |
|---|---|---|
| **[SRS_BSW_00101]** | The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function | [CP_SWS_IEEE1722Tp_91022] [CP_SWS_IEEE1722Tp_91023] |
| **[SRS_BSW_00310]** | API naming convention | [CP_SWS_IEEE1722Tp_91022] [CP_SWS_IEEE1722Tp_91023] |
| **[SRS_BSW_00350]** | All AUTOSAR Basic Software Modules shall allow the enabling/ disabling of detection and reporting of development errors. | [CP_SWS_IEEE1722Tp_00004] [CP_SWS_IEEE1722Tp_00012] [CP_SWS_IEEE1722Tp_00013] [CP_SWS_IEEE1722Tp_00014] |
| **[SRS_BSW_00358]** | The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void | [CP_SWS_IEEE1722Tp_91022] |
| **[SRS_BSW_00385]** | List possible error notifications | [CP_SWS_IEEE1722Tp_91020] [CP_SWS_IEEE1722Tp_91021] |
| **[SRS_BSW_00386]** | The BSW shall specify the configuration and conditions for detecting an error | [CP_SWS_IEEE1722Tp_00004] [CP_SWS_IEEE1722Tp_00012] [CP_SWS_IEEE1722Tp_00013] [CP_SWS_IEEE1722Tp_00014] [CP_SWS_IEEE1722Tp_00226] [CP_SWS_IEEE1722Tp_00227] |
| **[SRS_BSW_00404]** | BSW Modules shall support post-build configuration | [CP_SWS_IEEE1722Tp_91001] |
| **[SRS_BSW_00414]** | Init functions shall have a pointer to a configuration structure as single parameter | [CP_SWS_IEEE1722Tp_91022] |
| **[SRS_BSW_00441]** | Naming convention for type, macro and function | [CP_SWS_IEEE1722Tp_91001] [CP_SWS_IEEE1722Tp_91002] [CP_SWS_IEEE1722Tp_91003] |

**Table 6.1: Requirements Tracing**

# 7 Functional specification

This chapter defines the behavior of the IEEE1722Tp module. The API of the module is defined in Chapter 8, while the configuration is defined in Chapter 10.

## 7.1 Overview

The task of the IEEE1722Tp module is to process transmit requests and receive indications of IEEE1722Tp-related streams and forward particular AVTPDU-header information and payload via the LSduR to the according destination module(s). Additionally, the IEEE1722Tp module is able to tunnel different protocols via an ACF-message (e.g. CAN, LIN). Therefore a concatenation of multiple ACF-messages into one ACF-payload is possible. The ACF-payload shall be transmitted as an AVTP stream. A received ACF-stream is inspected by the IEEE1722Tp module and the concatinated ACF-messages of the ACF-payload are unpacked and forwarded via the LSduR to the according destination module(s).

The following figure shows how the IEEE1722Tp module is integrated in the AUTOSAR BSW communication stack:

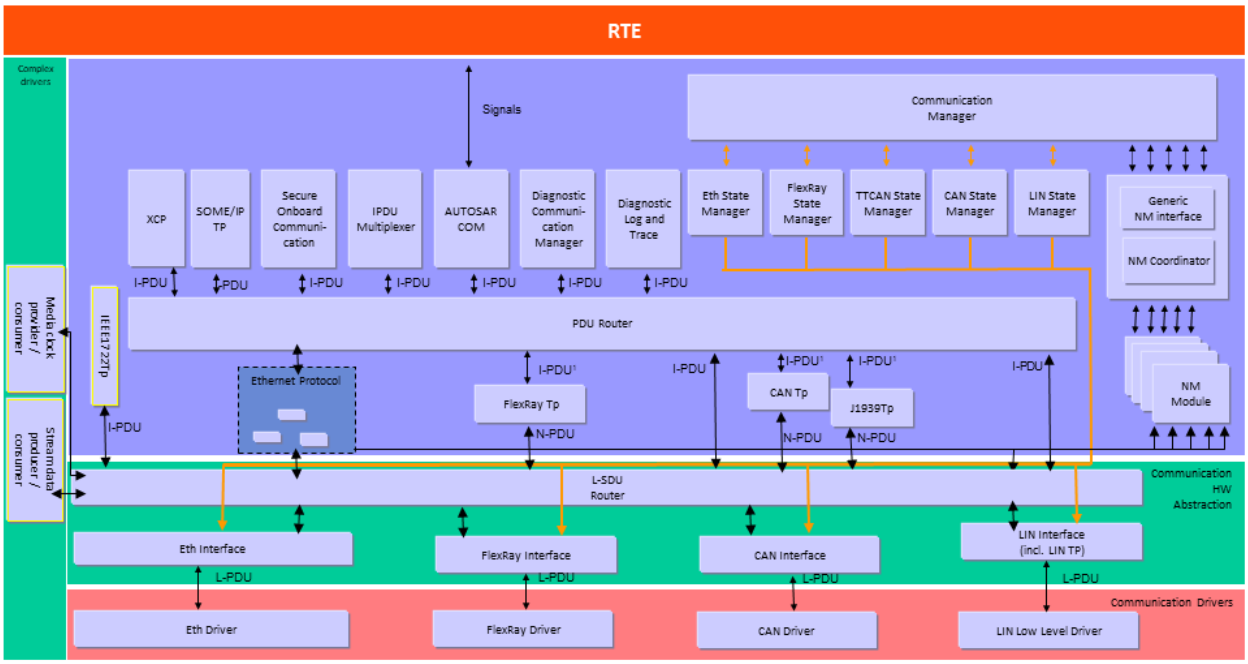


**Figure 7.1: AUTOSAR BSW architecture showing the IEEE1722Tp module**

The following scenarios are supported by the IEEE1722Tp module:

- Processing of transmission requests via the LSduR for configured IEEE1722 streams of AVTP stream data subtypes `AAF`, `61883_IIDC` and `RVF` initiated by an stream data producer.

- Processing of reception indication for configured IEEE1722 streams of AVTP stream data subtypes AAF, 61883_IIDC and RVF. Including forwarding via the LSduR to the corresponding local stream data consumer.
- Processing of transmission request for configured IEEE1722 streams of AVTP stream data subtypes CRF initiated by a media clock provider
- Processing of reception indication for configured IEEE1722 streams of AVTP stream data subtype CRF. Including forwarding via the LSduR to the corresponding local media clock consumer.
- Processing of transmission requests via LSduR for L-SDUs of type CAN and LIN. Including encapsulation to ACF-messages with corresponding ACF-message-type (ACF_CAN, ACF_CAN_BRIEF or ACF_LIN), adding ACF-message to configured ACF-stream and perform a transmission request based on the transmission trigger conditions.
- Processing of reception indication for configured IEEE1722 streams of AVTP stream data subtype ACF. Including inspection of the ACF-message-payload, unpacking of ACF-messages, may queueing L-SDUs and forwarding the L-SDUs via the LSduR to the corresponding destination module(s)

## 7.2 Module Handling

This section contains description of auxiliary functionality of the IEEE1722Tp module.

### 7.2.1 Initialization

The IEEE1722Tp module is initialized via IEEE1722Tp_Init, and de-initialized via IEEE1722Tp_DeInit. Except for IEEE1722Tp_Init, the API functions of the IEEE1722Tp module may only be called after the module has been properly initialized.

**[CP_SWS_IEEE1722Tp_00001]**

*Status:* DRAFT

⌈A call to IEEE1722Tp_Init shall perform the following actions:

- Initializes all internal variables.
- Flush all internal queues.
- Set all PDUs of all configured IEEE1722TpLowerLayerPduPools, IEEE1722TpUpperLayerTxPduPoolEntrys, and IEEE1722TpUpperLayerRxPduPoolEntrys to state PDU_AVAILABLE.
- Set all configured IEEE1722TpStreams to state IEEE1722TP_STREAM_DEACTIVATED

- Set the sequence number of each configured IEEE1722TpStream that carries a sequence number in its AVTPDU-header format to $00^{16}$.
- Set the IEEE1722Tp module to initialized state.

⌋

Note for [CP_SWS_IEEE1722Tp_00001]: Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields.

**[CP_SWS_IEEE1722Tp_00002]**

*Status:* DRAFT

⌈A call to IEEE1722Tp_DeInit sets the IEEE1722Tp module back to the uninitialized state.⌋

**[CP_SWS_IEEE1722Tp_00003]**

*Status:* DRAFT

⌈If development error reporting is enabled via IEEE1722TpDevErrorDetect, the IEEE1722Tp module shall call Det_ReportError with the error code IEEE1722TP_E_UNINIT when any API other than IEEE1722Tp_Init is called in uninitialized state.⌋

**[CP_SWS_IEEE1722Tp_00004]**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00350, SRS_BSW_00386

⌈When IEEE1722Tp_Init is called in initialized state, the IEEE1722Tp module shall not re-initialize its internal variables, flush its internal queues or change the state of PDUs or streams. It shall instead call Det_ReportError with the error code IEEE1722TP_E_REINIT if development error reporting is enabled (see IEEE1722TpDevErrorDetect).⌋

## 7.3 State handling

### 7.3.1 State handling of streams

**[CP_SWS_IEEE1722Tp_00005]**

*Status:* DRAFT

⌈IEEE1722Tp module shall maintain for each stream of all configured IEEE1722Tp-Streams two states: state IEEE1722TP_STREAM_ACTIVATED and state IEEE1722TP_STREAM_DEACTIVATED⌋

**[CP_SWS_IEEE1722Tp_00006]**

*Status:* DRAFT

⌈If IEEE1722Tp_ActivateStream is called and the given StreamIndex refer to a IEEE1722TpStream which already is in state IEEE1722TP_STREAM_ACTIVATED, then the IEEE1722Tp module shall ignore the call and return with E_OK.⌋

**[CP_SWS_IEEE1722Tp_00007]**

*Status:* DRAFT

⌈If IEEE1722Tp_ActivateStream is called and the given StreamIndex refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_DEACTIVATED, then the IEEE1722Tp module shall set the state of this IEEE1722TpStream to IEEE1722TP_STREAM_ACTIVATED and return with E_OK.⌋

**[CP_SWS_IEEE1722Tp_00008]**

*Status:* DRAFT

⌈If IEEE1722Tp_DeactivateStream is called and the given StreamIndex refer to a IEEE1722TpStream which already in state IEEE1722TP_STREAM_DEACTIVATED, then the IEEE1722Tp module shall ignore the call and return with E_OK.⌋

**[CP_SWS_IEEE1722Tp_00009]**

*Status:* DRAFT

⌈If IEEE1722Tp_DeactivateStream is called and the given StreamIndex refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED, then the IEEE1722Tp module shall set the state of this IEEE1722TpStream to IEEE1722TP_STREAM_DEACTIVATED and return with E_OK.⌋

**[CP_SWS_IEEE1722Tp_00010]**

*Status:* DRAFT

⌈If the state of a IEEE1722TpStream transit from IEEE1722TP_STREAM_ACTIVATED to IEEE1722TP_STREAM_DEACTIVATED and this IEEE1722TpStream has a IEEE1722TpStreamRxQueue configured, then the IEEE1722Tp module call for each entry in the queue LSduR_IEEE1722TpReleaseRxBuffer with the RxPduId set to stored PduId and flush the queue.⌋

**[CP_SWS_IEEE1722Tp_00011]**

*Status:* DRAFT

⌈If the state of a IEEE1722TpStream transit from IEEE1722TP_STREAM_ACTIVATED to IEEE1722TP_STREAM_DEACTIVATED and this IEEE1722TpStream has a IEEE1722TpStreamTxQueue configured, then the IEEE1722Tp module shall call for each entry in the queue LSduR_IEEE1722TpTxConfirmation with TxPduId set to stored PduId and result set to E_NOT_OK, and flush the queue.⌋

### 7.3.2 State handling of PDUs

PDUs are used to transfer data across the layers in the AUTOSAR communication stack. The IEEEE1722Tp module provide the possibility to configure so-called "PDU-pools". Each IEEE1722TpStream references an IEEE1722TpLowerLayerPduPool to interchange data with the lower layers, and optionally it could aggregate either a set of IEEE1722TpUpperLayerTxPduPoolEntry or a set of IEEE1722TpUpperLayerRxPduPoolEntry to interchange data with the upper layers. The IEEE1722Tp module act as a pass-through module and is requested to transmit data via PDUs and indicated to receive data via PDUs. Independent of the interaction direction (either interaction with the upper layer or with the lower layer), the IEEE1722Tp module has to maintain the usage-state of PDUs from the according PDU-pool. Therefore PDUs have two states PDU_IN_USE or PDU_AVAILABLE.

Note: The definition of PDU_IN_USE or PDU_AVAILABLE represent only the functional behavior, but not the implementation, since the state of a PDU is kept locally and is not propagated to other modules. Therefore, no type definition for the PDU state is specified.

**[CP_SWS_IEEE1722Tp_00012]**
*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00350, SRS_BSW_00386

⌈IEEE1722Tp module shall maintain for each PDU of all configured IEEE1722TpLowerLayerPduPools, IEEE1722TpUpperLayerTxPduPoolEntrys and IEEE1722TpUpperLayerRxPduPoolEntrys two states: state PDU_AVAILABLE and state PDU_IN_USE⌋

**[CP_SWS_IEEE1722Tp_00013]**
*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00350, SRS_BSW_00386

⌈If the IEEE1722Tp module is for a certain PDU configured with KeepLocalPduBuffer set to TRUE and is requested to transmit data or is indicated to receive data, or if transmission confirmation or release reception buffer is indicated, then the IEEE1722Tp module shall check the state of the PDU according the given PDU-ID:

- If the PDU of the given PDU-ID is in state PDU_AVAILABLE and requested to be transmitted or indicated to be received, then the IEEE1722Tp module shall set the state of this PDU to PDU_IN_USE. Otherwise the IEEE1722Tp module shall abort further handling, report a runtime error IEEE1722TP_E_PDU_STATE_TRANSITION_FAILED and, if possible return with E_NOT_OK.
- If the PDU of the given PDU-ID is in state PDU_IN_USE and transmission confirmation or release reception buffer is indicated, then the IEEE1722Tp module shall set the state of this PDU to PDU_AVAILABLE. Otherwise the IEEE1722Tp module shall abort further handling, report an runtime error IEEE1722TP_E_PDU_STATE_TRANSITION_FAILED and return.

⌋

**[CP_SWS_IEEE1722Tp_00014]**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00350, SRS_BSW_00386

⌈If the IEEE1722Tp module is requested to transmit data and the function call return with E_NOT_OK, then the IEEE1722Tp module shall set the state of the affected PDU to PDU_AVAILABLE.⌋

## 7.4 Global time related handling

The IEEE1722Tp module need to know the current synchronized global time and its state. This is needed for several scenario, e.g.:

- determine the avtp timestamp for transmission
- determine the message timestamp for ACF-messages
- evaluate presentation time for ACF-messages received via an ACF-stream with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS
- determine the state of the current synchronized time to set the tu (time uncertain) headerfield value

Note: The IEEE1722Tp module could retrieve the current synchronized global time either via a call of EthIf_GetCurrentTimeTuple or StbM_GetCurrentTime. See also Chapter 5.2 and Chapter 5.3

Each IEEE1722TpStream need to know from which global time source the current global synchronized time is retrieved. An IEEE1722TpStream could reference an StbMSynchronizedTimeBase via IEEE1722TpStbMSynchronizedTimeBaseRef. With this reference the IEEE1722Tp module derive the StbMSynchronizedTimeBaseIdentifier for call of StbM_GetCurrentTime or StbM_GetTimeBaseStatus.

**[CP_SWS_IEEE1722Tp_00015]**

*Status:* DRAFT

⌈If an IEEE1722TpStream reference a StbMSynchronizedTimeBase via IEEE1722TpStbMSynchronizedTimeBaseRef and StbM_GetCurrentTime or StbM_GetTimeBaseStatus need to be called, then the IEEE1722Tp module shall derive the StbMSynchronizedTimeBaseIdentifier from the referenced StbMSynchronizedTimeBase and use it a timeBaseId for the function call.⌋

Additionally an IEEE1722TpStream could reference an EthIfClkUnit via IEEE1722TpEthIfClkUnitRef. This could be used, if a function call towards the StbM, to retrieve the current global synchronized time, impact the accuray such that the system timing requirement are violated. Therefore a call of EthIf_GetCurrentTimeTuple may support a better accuracy.

**[CP_SWS_IEEE1722Tp_CONSTR_00025] IEEE1722TpEthIfClkUnitRef optional**

*Status:* DRAFT

⌈If an IEEE1722TpEthIfClkUnitRef of a particular IEEE1722TpStream is configured, then IEEE1722TpEthIfClkUnitRef shall reference the EthIfClkUnit, which corresponds to the global time domain of this IEEE1722TpStream referenced via IEEE1722TpStbMSynchronizedTimeBaseRef.⌋

**[CP_SWS_IEEE1722Tp_00016]**

*Status:* DRAFT

⌈If the IEEE1722Tp module reference EthIfClkUnit via IEEE1722TpEthIfClkUnitRef and EthIf_GetCurrentTimeTuple need to be called, then the IEEE1722Tp module shall derive the EthIfClkUnitIdx from the referenced EthIfClkUnit and use it as ClkUnitIdx, and derive the EthIfPhysControllerIdx from the EthIfPhysController where the EthIfClkUnit belongs to and use it as CtrlIdx for the function call.⌋

## 7.5 Communication

The IEEE1722Tp module communicate via transmit and reception APIs with other BSW modules. The communication processing is devided in transmission requests and reception indication. A simplified communication processing works as follow:

### Transmission

- Arrival of transmission request
- Evaluate transmission request and internal states
- If transmission request evaluation is successful, perform internal communication processing and forward to LSduR
- If transmission request evaluation fails, abort internal communication processing and return with E_NOT_OK

### Reception

- Arrival of reception indication
- Evaluate reception indication and internal states
- If reception evaluation is successful, perform internal communication processing and forward to LSduR
- If reception evaluation evaluation fail, abort internal communication processing and return

The IEEE1722Tp module uses meta data as specified in [5, CP-SWS-BSWGeneral]. Basically, the IEEE1722Tp module act as intermediate layer to transfer provided data to IEEE1722 streams, and to extract data from received IEEE1722 streams and to forward the extracted data to the upper layer (e.g. IEEE1722 related applications). The following communication scenarios have to be considered:

- UpperLayer-To-Stream-TxData: upper layer (e.g. audio CDD) forward data transmission via LSduR to IEEE1722Tp module. The IEEE1722Tp module create an IEEE1722 stream and forward transmssion request via LSduR to EthIf
- LowerLayer-To-Stream-TxData: lower layer <Bus>-frames (e.g. CAN) forward reception indication via LSduR to IEEE1722Tp module. IEEE1722Tp module create an ACF-message and add tjos ACF-message to an IEEE1722 stream. IEEE1722Tp module forward transmission request via LSduR to EthIf
- Stream-To-UpperLayer-RxData: EthIf forward reception indication of an IEEE1722 stream via LSduR to IEEE1722Tp module. IEEE1722Tp inspect the received IEEE1722 stream, extract data and forward data via LSduR to upper layer (e.g. audio CDD)
- Stream-To-UpperLowerLayer-RxTxData: EthIf forward reception indication of an IEEE1722 stream via LSduR to IEEE1722Tp module. IEEE1722Tp inspect the received IEEE1722 stream, extract <Bus>-frames (e.g. CAN) and forward data via LSduR to upper / lower layer (e.g. PduR / CanIf)

In each communication scenarios the IEEE1722Tp module could consume and/or produce meta data. IEEE1722Tp consume meta data in the role of an callee. IEEE1722Tp produce meta data in the role of an caller. Independent if acting in the role of an callee or caller, the IEEE1722Tp need to know the meta data type. For each communication scenario meta data types are specified to be consumed and to be produced (see Chapter 7.6.1).

The following sub chapters describe in detail the expected behaviour of the communication processing. Please note, API parameter checks are described in chapter Chapter 8.3.2

### 7.5.1 Transmission requests

**[CP_SWS_IEEE1722Tp_00017]**

*Status:* DRAFT

⌈If a transmission request is indicated with a call of `IEEE1722Tp_Transmit` and the given `TxPduId` refer to a `IEEE1722TpStream` which is in state `IEEE1722TP_STREAM_DEACTIVATED`, then the IEEE1722Tp module shall reject the transmission request by returning with `E_NOT_OK`⌋

**[CP_SWS_IEEE1722Tp_00018]**

*Status:* DRAFT

⌈If a transmission request is indicated with a call of IEEE1722Tp_Transmit and the following conditions are true:

- the given TxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED
- the given TxPduId refer to a PDU with KeepLocalPduBuffer set to TRUE
- this IEEE1722TpStream has a IEEE1722TpStreamTxQueue configured and the queue has space to store a transmission request

then the IEEE1722Tp module shall store the transmission request (including TxPduId, PduInfoPtr) and return with E_OK. Otherwise report runtime error IEEE1722TP_E_TX_QUEUE_OVERRUN and return with E_NOT_OK.⌋

**[CP_SWS_IEEE1722Tp_00019]**

*Status:* DRAFT

⌈If a transmission request is indicated with a call of IEEE1722Tp_Transmit and the following conditions are true:

- the given TxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED
- the given TxPduId refer to a PDU with KeepLocalPduBuffer set to FALSE
- this IEEE1722TpStream has a IEEE1722TpStreamTxQueue configured and the queue has space to store a transmission request

then the IEEE1722Tp module shall store the transmission request (including TxPduId, PDU payload given with PduInfoPtr.SduDataPtr and PduInfoPtr.SduLength, and, if available, meta data given with PduInfoPtr.MetaDataPtr) and return with E_OK. Otherwise the IEEE1722Tp module shall report a runtime error IEEE1722TP_E_TX_QUEUE_OVERRUN and return with E_NOT_OK⌋

Note:

- Transmission requests which are available in a IEEE1722TpStreamTxQueue are processed in context of IEEE1722Tp_MainFunctionTx.
- Usage of IEEE1722TpStreamTxQueue in combination with a IEEE1722TpStream of sub type IEEE1722TpStreamACF may impact performance and waste resources, since the transmission request is stored in the transmission request queue and within the internal processing, in an ACF-stream for transmission, again.

**[CP_SWS_IEEE1722Tp_00020]**

*Status:* DRAFT

⌈If a transmission request is indicated with a call of IEEE1722Tp_Transmit, the given TxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED and no IEEE1722TpStreamTxQueue is configured, then this transmission request shall immediately be forwarded to the internal transmission request processing and processed.⌋

Note to [CP_SWS_IEEE1722Tp_00020]:

- Transmission requests for IEEE1722TpStream where no IEEE1722TpStreamTxQueue is available are processed immediately.
- Refer to Chapter 7.5.2 for details on internal transmission request processing.

### 7.5.2 Internal transmission request processing

A transmission request is handled in the internal transmission request processing. The internal transmission request processing include the creation of an AVTPDU-header of the corresponding IEEE1722TpStreamSubtype. AVTPDU-header creation can be found in Chapter 7.5.10

The handling depends slightly if the internal transmission request processing is triggered immediately in the contex of IEEE1722Tp_Transmit or deferred in context of the IEEE1722Tp_MainFunctionTx.

**[CP_SWS_IEEE1722Tp_00021]**

*Status:* DRAFT

⌈If a transmission request is forwarded to the internal transmission request processing in context of IEEE1722Tp_Transmit and the processing is aborted, then the IEEE1722Tp module shall report runtime error IEEE1722TP_E_TX_INTERNAL_PROCESSING_FAILED and return with E_NOT_OK.⌋

**[CP_SWS_IEEE1722Tp_00022]**

*Status:* DRAFT

⌈If the state of a IEEE1722TpStream transit from IEEE1722TP_STREAM_ACTIVATED to IEEE1722TP_STREAM_DEACTIVATED and this IEEE1722TpStream has a IEEE1722TpStreamTxQueue configured, then the IEEE1722Tp module shall call for each entry in the queue LSduR_IEEE1722TpTxConfirmation with TxPduId set to stored PduId and result set to E_NOT_OK, and flush the queue.

If a transmission request is forwarded to the internal transmission request processing in context of IEEE1722Tp_MainFunctionTx and the internal processing is aborted, then the IEEE1722Tp module shall call LSduR_IEEE1722TpTxConfirmation with processed TxPduId and result set to E_NOT_OK, report runtime error IEEE1722TP_E_TX_INTERNAL_PROCESSING_FAILED and return.⌋

**[CP_SWS_IEEE1722Tp_00023]**

*Status:* DRAFT

⌈If a transmission request is forwarded to the internal transmission request processing in context of IEEE1722Tp_MainFunctionTx and the internal processing is aborted or successfully finished, then the IEEE1722Tp module shall remove the transmission request from the corresponding IEEE1722TpStreamTxQueue.⌋

Note: No transmission retry is perfomed by the IEEE1722Tp module.

**[CP_SWS_IEEE1722Tp_00024]**

*Status:* DRAFT

⌈If a transmission request is forwarded to the internal transmission request processing and the given TxPduId refer to a IEEE1722TpStream configured to a IEEE1722Tp-StreamSubtype other than IEEE1722TpStreamACF, then following actions shall be performed:

- Consume meta data items which relate to the IEEE1722TpStreamSubtype and evaluate if consumed values are valid. Valid values shall be used, the remaining meta data items shall be taken from the corresponding configuration parameters of this IEEE1722TpStream.
- Create an AVTPDU-header with respect to the configured IEEE1722TpStream-Subtype. If creation of an AVTPDU-header was successful, check for an available LL-PDU from the referenced IEEE1722TpLowerLayerPduPool. Otherwise abort internal transmission request processing.
- If LL-PDU is available, produce the following meta data. Otherwise abort internal transmission request processing:
  - create a list-element-struct of type ListElemStructType according to [CP_SWS_IEEE1722Tp_00025] and set LISTELEM_PTR to the address of the created list-element-struct
  - transfer MAC address determined with first processing step (see above) to ETHERNET_MAC_64 and set the remaing bits to zero.
- Update PduInfoPtr.SduDataPtr and PduInfoPtr.SduLength of LL-PDU with PduInfoPtr.SduDataPtr and PduInfoPtr.SduLength given with TxPduId of the transmission request.
- Call LSduR_IEEE1722TpTransmit with TxPduId set to PduId of LL-PDU and PduInfoPtr set to PduInfoPtr of LL-PDU.

⌋

Note for [CP_SWS_IEEE1722Tp_00024]: Refer to Chapter 7.6.1 for details on meta data item types and to Chapter 7.5.10 for details on AVTPDU-header creation.

**[CP_SWS_IEEE1722Tp_00025]**

*Status:* DRAFT

⌈If IEEE1722Tp has to create a list-element-struct of type `ListElemStructType` due to internal transmission request processing and the creation of the AVTPDU-header was successful, then IEEE1722Tp module shall consider the following points:

- create an instance of type `ListElemStructType` and set `NextListElemPtr` to `NUL_PTR`
- set `DataPtr` to address of the available AVTPDU-header and `DataLength` to the length of the availalbe AVTP-header

⌋

**[CP_SWS_IEEE1722Tp_00026]**

*Status:* DRAFT

⌈If a transmission request is forwarded to the internal transmission request processing, the given `TxPduId` refer to a `IEEE1722TpStream` configured to a `IEEE1722Tp-StreamSubtype` set to `IEEE1722TpStreamACF` and the given `TxPduId` refer to a PDU where meta data are configured, then the following actions shall be performed:

- if given `TxPduId` refer to a configured `IEEE1722TpStreamAcfPayload` which is set to `IEEE1722TpStreamAcfCan`, then the internal transmission request processing shall proceed with the following steps:
  - Consume meta data type item `CAN_ID_32` and extract CAN-ID
  - If given `TxPduId` corresponds to an `IEEE1722TpStreamAcfCan` which is configured to `IEEE1722TpStreamAcfCanPduFilter` configured and the extracted CAN-ID pass the `IEEE1722TpStreamAcfCanPduFilter`, then the internal transmission request processing shall proceed. Otherwise report an runtime error with `IEEE1722TP_E_CAN_FILTER_DROPPED_TX_CAN_FRAME` and abort the internal transmission request processing.
  - If the given `TxPduId` corresponds to a `IEEE1722TpStreamAcfCan` which has an `IEEE1722TpStreamAcfCanPduFilter` configured and pass the filter, or `IEEE1722TpStreamAcfCanPduFilter` is not avialable, then the internal transmission request shall create an ACF-message and proceed with ACF-stream handling
- if given `TxPduId` corresponds to a `IEEE1722TpStreamAcfPayload` which is configured to `IEEE1722TpStreamAcfLin`, then the internal transmission request processing shall proceed with the following steps:
  - Consume `LIN_NAD_8`
  - Create an ACF-message and proceed with ACF-stream handling

⌋

Note for [CP_SWS_IEEE1722Tp_00027]: Refer to Chapter 7.5.10.8 for details on ACF-message creation and to Chapter 7.5.9 for details on ACF-stream handling.

**[CP_SWS_IEEE1722Tp_00027]**

*Status:* DRAFT

⌈If a transmission request is forwarded to the internal transmission request processing, the given `TxPduId` refer to a `IEEE1722TpStream` configured to a `IEEE1722TpStreamSubtype` set to `IEEE1722TpStreamACF` and the given `TxPduId` refer to a PDU where meta data is not configured, then the following actions shall be performed:

- If given `TxPduId` is associated with a configured `IEEE1722TpStreamAcfPayload` set to `IEEE1722TpStreamAcfCan`, then the internal transmission request processing shall create an ACF-message of ACF-message type ACF_CAN and proceed with ACF-stream handling
- If configured `IEEE1722TpStreamAcfPayload` is set to `IEEE1722TpStreamAcfLin`, then the internal transmission request processing shall create an ACF-message of ACF-message type ACF_LIN and proceed with ACF-stream handling

⌋

Note for [CP_SWS_IEEE1722Tp_00027]: Refer to Chapter 7.5.10.8 for details on ACF-message creation and to Chapter 7.5.9 for details on ACF-stream handling.

**[CP_SWS_IEEE1722Tp_CONSTR_00001]**

*Status:* DRAFT

⌈An `IEEE1722TpStreamAcfCanPduFilter` shall either have `IEEE1722TpStreamAcfCanIdMask` or `IEEE1722TpStreamAcfCanIdRange` configured. Otherwise, the configuration shall be rejected as invalid.⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00002]**

*Status:* DRAFT

⌈An `IEEE1722TpStreamAcfCanPdu` which aggregate an `IEEE1722TpStreamAcfPdu` where meta data is configured, shall not have an `IEEE1722TpStreamAcfCanId` configured. Such an configuration shall be rejected as invalid.⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00003]**

*Status:* DRAFT

⌈An `IEEE1722TpStreamAcfCanPdu` which aggregate an `IEEE1722TpStreamAcfCanId`, shall not have an `IEEE1722TpStreamAcfCanPduFilter` configured. Such an configuration shall be rejected as invalid.⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00004]**

*Status:* DRAFT

⌈An `IEEE1722TpStream` configured to a `IEEE1722TpStreamSubtype` set to `IEEE1722TpStreamACF` and where `IEEE1722TpStreamDirection` is set to

IEEE1722TpStreamTx, shall not have IEEE1722TpUpperLayerTxPduPoolEntrys configured. Such an configuration shall be rejected as invalid.⌋

### 7.5.3 Reception indications

**[CP_SWS_IEEE1722Tp_00028]**

*Status:* DRAFT

⌈If a reception is indicated with a call of IEEE1722Tp_RxIndication and the given RxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_DEACTIVATED, then the IEEE1722Tp module shall discard the reception indication silently.⌋

**[CP_SWS_IEEE1722Tp_00029]**

*Status:* DRAFT

⌈If a reception is indicated with a call of IEEE1722Tp_RxIndication and the given RxPduId is not available in the configuration, then the IEEE1722Tp module shall shall discard the reception indication.⌋

**[CP_SWS_IEEE1722Tp_00030]**

*Status:* DRAFT

⌈If a reception is indicated with a call of IEEE1722Tp_RxIndication and the following conditions are true:

- the given RxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED
- the given RxPduId refer to a PDU with KeepLocalPduBuffer set to TRUE
- this IEEE1722TpStream has an IEEE1722TpStreamRxQueue configured and the queue has space to store a reception indication

then the IEEE1722Tp module shall store the reception indication (including RxPduId and PduInfoPtr) and return. Otherwise the IEEE1722Tp module shall report a runtime error IEEE1722TP_E_RX_QUEUE_OVERRUN, call LSduR_IEEE1722TpReleaseRxBuffer with given RxPduId and return.⌋

**[CP_SWS_IEEE1722Tp_00031]**

*Status:* DRAFT

⌈If a reception is indicated with a call of IEEE1722Tp_RxIndication and the following conditions are true:

- the given RxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED
- the given RxPduId refer to a PDU with KeepLocalPduBuffer set to FALSE

- this IEEE1722TpStream has an IEEE1722TpStreamRxQueue configured and the queue has space to store a reception indication

then the IEEE1722Tp module shall store the reception indication (including RxPduId, PDU payload given with PduInfoPtr.SduDataPtr and PduInfoPtr.SduLength, and, if available, meta data given with PduInfoPtr.MetaDataPtr and return. Otherwise the IEEE1722Tp module shall report a runtime error IEEE1722TP_E_RX_QUEUE_OVERRUN and return.⌋

Note:

- Reception indications which are available in a IEEE1722TpStreamRxQueue are processed in context of IEEE1722Tp_MainFunctionRx
- Usage of IEEE1722TpStreamRxQueue in combination with a IEEE1722TpStream of sub type IEEE1722TpStreamACF may impact performance and waste resources, since the reception indication is stored in the reception indication queue and within the internal processing again.

**[CP_SWS_IEEE1722Tp_00032]**

*Status:* DRAFT

⌈If a reception is indicated with a call of IEEE1722Tp_RxIndication, the given RxPduId refer to a IEEE1722TpStream which is in state IEEE1722TP_STREAM_ACTIVATED and no IEEE1722TpStreamRxQueue is configured, then this reception indication shall immediately be forwarded to the internal reception indication processing and proceed.⌋

Note to [CP_SWS_IEEE1722Tp_00032]:

- Reception indication for IEEE1722TpStream where no IEEE1722TpStreamRxQueue is available are processed immediately.
- Refer to Chapter 7.5.4 for details on internal reception indication processing

### 7.5.4 Internal reception indication processing

A reception indication is handled in the internal reception indication processing. The internal reception indication processing include the inspection of an AVTPDU-header of the corresponding IEEE1722TpStreamSubtype and the handling of the AVTPDU-payload. The inspection can be found in Chapter 7.5.11

**[CP_SWS_IEEE1722Tp_00033]**

*Status:* DRAFT

⌈If a reception indication is forwarded to the internal reception indication processing and the processing is aborted, then the IEEE1722Tp module shall report runtime error IEEE1722TP_E_RX_INTERNAL_PROCESSING_FAILED and if given RxPduId is

associated with a PDU where KeepLocalPduBuffer is set to TRUE, call LSduR_IEEE1722TpReleaseRxBuffer with RxPduId set to the given RxPduId and return.⌋

Note: LSduR_IEEE1722TpReleaseRxBuffer could be called in context of the IEEE1722Tp_RxIndication

**[CP_SWS_IEEE1722Tp_00034]**

*Status:* DRAFT

⌈If a reception indication is forwarded to the internal reception indication processing in context of IEEE1722Tp_MainFunctionRx and the internal processing is aborted or has successfully finished, then the IEEE1722Tp module shall remove the reception indication from the corresponding IEEE1722TpStreamRxQueue.⌋

**[CP_SWS_IEEE1722Tp_00035]**

*Status:* DRAFT

⌈If a reception indication is forwarded to the internal reception indication processing and the given RxPduId refer to a IEEE1722TpStream configured to a IEEE1722TpStreamSubtype other than IEEE1722TpStreamACF, then the following actions shall be performed:

- Inspect the AVTPDU-header with respect to the configured IEEE1722TpStreamSubtype. If inspection of an AVTPDU-header was successful, check for an available UL-PDU from the according IEEE1722TpUpperLayerRxPduPoolEntry. Otherwise abort internal reception indication processing.
- If UL-PDU is available then consider the following points. Otherwise abort internal reception indication processing:
  - If available, consume MetaDataItem TIMETUPLE_TYPE_PTR.
  - If configured at UL-PDU, produce an MetaDataItem TIMETUPLE_TYPE_PTR and transfer the value of the consumed TIMETUPLE_TYPE_PTR to the produced TIMETUPLE_TYPE_PTR
  - Produce an instance of the MetaDataItem according to the IEEE1722TpStreamACF and the corresponding defined type. Transfer AVTP-header field values of the received IEEE1722TpStream to corresponding type elements.
  - Identify the start address of the AVTPDU-payload according the IEEE1722 stream format of the processed IEEE1722TpStreamSubtype
  - Set the PduInfoPtr.SduDataPtr of the UL-PDU to the identified start address of AVTP-payload and set the PduInfoPtr.SduLength of the UL-PDU to the value of the identified length of the identified AVTP-payload. Set the PduInfoPtr.MetaDataPtr of UL-PDU to the memory start address of the produced and serialized meta data items (see above).

- Call LSduR_IEEE1722TpRxIndication with RxPduId set to PduId of UL-PDU and PduInfoPtr set to PduInfoPtr of UL-PDU.

⌋

Note for [CP_SWS_IEEE1722Tp_00035]: Refer to Chapter 7.5.11 for details on AVTPDU-header inspection. Refer to Chapter 7.6.1 for details on meta data item types and to Chapter 8.2 for the corresponding defined types.

**[CP_SWS_IEEE1722Tp_00036]**

*Status:* DRAFT

⌈If a reception indication is forwarded to the internal reception indication processing and the given TxPduId refer to a IEEE1722TpStream configured to a IEEE1722TpStreamSubtype set to IEEE1722TpStreamACF, then the following actions shall be performed:

- If available, consume MetaDataItem TIMETUPLE_TYPE_PTR.
- Inspect the AVTPDU-header with respect to the configured IEEE1722TpStreamSubtype. If inspection of an AVTPDU-header was successful, proceed with the next step. Otherwise abort internal reception indication processing.
- Identify the start address of the AVTPDU-payload according the IEEE1722 stream format of the processed IEEE1722TpStreamSubtype and iterate over the received AVTPDU-payload with respect to the identified length of the AVTPDU-payload:
  - if an ACF-message of type ACF_CAN is identified, a CAN-ID is identified that corresponds to a IEEE1722TpStreamAcfCan where the bus id and message type match the configured values of IEEE1722TpStreamAcfBusId and IEEE1722TpStreamAcfCanMessageType and this IEEE1722TpStreamAcfCan has an IEEE1722TpStreamAcfCanPduFilter configured where the received CAN-ID pass the IEEE1722TpStreamAcfCanPduFilter or this IEEE1722TpStreamAcfCan has no IEEE1722TpStreamAcfCanPduFilter configured, then the internal transmission request processing shall proceed with ACF-stream handling. Otherwise report an runtime error with IEEE1722TP_E_DROPPED_RX_CAN_FRAME, skip this ACF-message and proceed with the iteration.
  - if an ACF-message of type ACF_LIN is identified, a LIN-ID is identified that corresponds to a IEEE1722TpStreamAcfLin where the bus id and message type match the configured values of IEEE1722TpStreamAcfBusId, then the internal transmission request processing shall proceed with ACF-stream handling. Otherwise report an runtime error with IEEE1722TP_E_DROPPED_RX_LIN_FRAME, skip this ACF-message and proceed with the iteration.

⌋

Note for [CP_SWS_IEEE1722Tp_00036]: Refer to Chapter 7.5.11 for details on AVTPDU-header inspection and to Chapter 7.5.9 for details on ACF-stream handling.

**[CP_SWS_IEEE1722Tp_CONSTR_00005]**

*Status:* DRAFT

⌈An IEEE1722TpStream configured to a IEEE1722TpStreamSubtype set to IEEE1722TpStreamACF and where IEEE1722TpStreamDirection is set to IEEE1722TpStreamRx, shall not have IEEE1722TpUpperLayerRxPduPoolEntrys configured. Such an configuration shall be rejected as invalid.⌋

### 7.5.5 Transmission confirmation

**[CP_SWS_IEEE1722Tp_00037]**

*Status:* DRAFT

⌈If IEEE1722Tp_TxConfirmation is called with a TxPduId which is configured, then the IEEE1722Tp module shall forward the transmission confirmation to the upper layer by calling LSduR_IEEE1722TpTxConfirmation with the corresponding TxPduId and result set to the value received by IEEE1722Tp_TxConfirmation. Otherwise the IEEE1722Tp module shall silently discard the transmission confirmation and return.⌋

### 7.5.6 Release reception buffer

**[CP_SWS_IEEE1722Tp_00038]**

*Status:* DRAFT

⌈If IEEE1722Tp_ReleaseRxBuffer is called with a RxPduId which is configured, then the IEEE1722Tp module shall forward the release reception buffer function call to the lower layer by calling LSduR_IEEE1722TpReleaseRxBuffer with the corresponding RxPduId. Otherwise the IEEE1722Tp module shall silently discard the release reception buffer function call and return.⌋

### 7.5.7 Mainfunction processing

**[CP_SWS_IEEE1722Tp_00039]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall handle all configured IEEE1722TpStreamTxQueues per IEEE1722TpStream in the IEEE1722Tp_MainFunctionTx.⌋

**[CP_SWS_IEEE1722Tp_00040]**

*Status:* DRAFT

⌈If a IEEE1722TpStreamTxQueue is processed in context of IEEE1722Tp_MainFunctionTx and transmission requests available, then the IEEE1722Tp module shall forward the available transmission requests to the internal transmission request processing by starting with the oldest transmission request and proceed in ascending order.⌋

Note for [CP_SWS_IEEE1722Tp_00040]: Refer to Chapter 7.5.2 for details on internal transmission request processing.

**[CP_SWS_IEEE1722Tp_00041]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall handle all configured IEEE1722TpStreamRxQueues per IEEE1722TpStream in the IEEE1722Tp_MainFunctionRx.⌋

**[CP_SWS_IEEE1722Tp_00042]**

*Status:* DRAFT

⌈If a IEEE1722TpStreamRxQueue is processed in context of IEEE1722Tp_MainFunctionRx and reception indications available, then the IEEE1722Tp module shall forward the available reception indications to the internal reception indication processing by starting with the oldest reception indication and proceed in ascending order.⌋

Note for [CP_SWS_IEEE1722Tp_00042]: Refer to Chapter 7.5.4 for details on internal reception indication processing.

### 7.5.8 Buffer handling

Modules which handle PDUs with attribute KeepLocalPduBuffer set to TRUE, keep the local produced data until a call of TxConfirmation or ReleaseRxBuffer function call is received, or if the initiating function call (e.g. Transmit) returns with E_NOT_OK. If KeepLocalPduBuffer set to FALSE, then the local produced data is released after the initiating function call (e.g. Transmit or RxIndication) returns.

**[CP_SWS_IEEE1722Tp_00043]**

*Status:* DRAFT

⌈If the IEEE1722Tp module calls LSduR_IEEE1722TpTransmit with an TxPduId that refer to a global PDU with KeepLocalPduBuffer set to TRUE and the function return with E_OK, then the IEEE1722Tp module shall keep the buffer with local produced data (e.g. meta data) for this TxPduId. In all other cases, where IEEE1722Tp calls LSduR_IEEE1722TpTransmit, the buffer for local produced data shall be released.⌋

**[CP_SWS_IEEE1722Tp_00044]**

*Status:* DRAFT

⌈If the IEEE1722Tp module calls `LSduR_IEEE1722TpRxIndication` with an `RxPduId` that refer to a global PDU with `KeepLocalPduBuffer` set to `TRUE` and the function return , then the IEEE1722Tp module shall keep the buffer with local produced data (e.g. meta data) for this `RxPduId` . Otherwise the buffer for local produced data shall be released after the function return.⌋

**[CP_SWS_IEEE1722Tp_00045]**

*Status:* DRAFT

⌈If an internal transmission request processing or an internal reception indication processing is aborted for a PDU, then the buffer of produced local data (e.g. meta data) for this PDU shall be released.⌋

Note for [CP_SWS_IEEE1722Tp_00045]: Refer to Chapter 7.5.2 for details on internal transmission request processing and to Chapter 7.5.4 for details on internal reception indication processing.

**[CP_SWS_IEEE1722Tp_00046]**

*Status:* DRAFT

⌈If `IEEE1722Tp_TxConfirmation` is called with a `TxPduId` which is configured, the PDU of the given `TxPduId` is in state `PDU_IN_USE` and this `TxPduId` refer to a global PDU with `KeepLocalPduBuffer` set to `TRUE`, then the IEEE1722Tp module shall release the buffer for local produced data (e.g. meta data).⌋

**[CP_SWS_IEEE1722Tp_00047]**

*Status:* DRAFT

⌈If `IEEE1722Tp_ReleaseRxBuffer` is called with a `RxPduId` which is configured, the PDU of the given `RxPduId` is in state `PDU_IN_USE` and this `RxPduId` refer to a global PDU with `KeepLocalPduBuffer` set to `TRUE`, then the IEEE1722Tp module shall release the buffer for local produced data (e.g. meta data).⌋

### 7.5.9 ACF-stream handling

ACF-stream handling specifies collection of created ACF-messages and triggering for transmission requests of the according ACF-stream based on the transmission trigger condition:

- `IEEE1722TpAcfCollectionTimeout` per `IEEE1722TpStreamACF`
- `IEEE1722TpAcfCollectionThreshold` per `IEEE1722TpStreamACF`
- `TRIGGER_ALWAYS` or `TRIGGER_NEVER` per ACF-message

Addtionally, the ACF-stream handling specifies to take over unpacked ACF-messages (L-SDUs) with extracted <bus>-specific information (meta data) and forward the L-SDUs with meta data to the LSduR.

ACF-stream handling is performed in context of the internal transmission request processing (see Chapter 7.5.2) and internal reception indication processing (see Chapter 7.5.4). Each processing has its own responsibility.

ACF-handling in context of internal transmission request processing is responsible for the following points:

- collection of ACF-messages
- evaluation of transmission trigger condition for `IEEE1722TpStreamACF` with `IEEE1722TpStreamDirection` set to `IEEE1722TpStreamTx`. Evaluation is performed within each transmission request and periodically in the `IEEE1722Tp_MainFunctionTx`

ACF-handling in context of internal reception indication processing is responsible for the following points:

- take over unpacked ACF-messages (L-SDUs) with extracted <bus>-specific information (meta data)
- evaluation of forwarding condition for the unpacked L-SDUs in dependency to the properties of the transporting ACF-stream:
  - direct forwarding of L-SDUs to the LSduR which where transported by an `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `NON_TIME_SYNCHRONOUS`
  - time dependend forwarding of L-SDUs to the LSduR which where transported by an `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `TIME_SYNCHRONOUS` by considering the AVTP presentation time of this stream. Evaluation is performed within each reception indidication and periodically in the `IEEE1722Tp_MainFunctionRx`

#### 7.5.9.1 ACF-message collection and transmission

**[CP_SWS_IEEE1722Tp_00048]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall provide for each configured `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfMixedBusTypeCollection` is set to `TRUE` and `IEEE1722TpStreamDirection` set to `IEEE1722TpStreamTx` one internal ACF-transmission-queue to collect ACF-messages.⌋

**[CP_SWS_IEEE1722Tp_00049]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall provide for each configured IEEE1722TpStream-ACF with IEEE1722TpStreamAcfMixedBusTypeCollection is set to FALSE and IEEE1722TpStreamDirection set to IEEE1722TpStreamTx one internal ACF-transmission-queue per configured ACF-message type to collect ACF-messages.⌋

Note: A ACF-message type (e.g. ACF_CAN) is represented as parameter of an IEEE1722TpStreamAcfPayload, e.g. IEEE1722TpStreamAcfCan.

**[CP_SWS_IEEE1722Tp_00050]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall consider timeout timer for each configured IEEE1722 TpStreamACF with IEEE1722TpStreamDirection set to IEEE1722TpStreamTx and where an IEEE1722TpAcfCollectionTimeout is configured.⌋

**[CP_SWS_IEEE1722Tp_00051]**

*Status:* DRAFT

⌈If an ACF-message is forwarded to the ACF-stream handling in context of the internal transmission request processing, then the IEEE1722Tp module shall identify the internal ACF-transmission-queue by considering the IEEE1722TpStreamACF configuration:

- an ACF-message of type ACF_CAN, shall select the internal ACF-transmission-queue which is associated with the IEEE1722TpStreamACF where the configured IEEE1722TpStreamAcfCanId match to CAN-ID of the ACF-message
- an ACF-message of type ACF_LIN, shall select the internal ACF-transmission-queue which is associated with the IEEE1722TpStreamACF where the configured IEEE1722TpStreamAcfLinId match to LIN-ID of the ACF-message

⌋

**[CP_SWS_IEEE1722Tp_00052]**

*Status:* DRAFT

⌈If an ACF-message is forwarded to the ACF-stream handling in context of the internal transmission request processing and the evaluation of the transmission trigger conditions qualifies to collect the ACF-message, then the IEEE1722Tp shall enqueue this ACF-message in the corresponding ACF-transmission-queue.⌋

**[CP_SWS_IEEE1722Tp_00053]**

*Status:* DRAFT

⌈If an ACF-message is forwarded to the ACF-stream handling in context of the internal transmission request processing and the evaluation of the transmission trigger conditions qualifies to transmit the ACF-message, then the IEEE1722Tp shall trigger a transmission according to [CP_SWS_IEEE1722Tp_00054].⌋

**[CP_SWS_IEEE1722Tp_00054]**

*Status:* DRAFT

⌈If the evaluation of the transmission trigger conditions qualifies to transmit an ACF-transmission-queue, then the IEEE1722Tp shall perform the following actions:

- create an AVTP-header with respect to the configured `IEEE1722TpStream-AcfHeaderType` (either `NON_TIME_SYNCHRONOUS` or `TIME_SYNCHRONOUS`) of the corresponding `IEEE1722TpStreamACF`. If creation of an AVTPDU-header was successfull, check for an available LL-PDU from the referenced `IEEE1722TpLowerLayerPduPool` of the affected `IEEE1722TpStreamACF`. Otherwise abort ACF-stream handling.
- If LL-PDU is available, produce the following meta data. Otherwise abort ACF-stream handling:
  - produce `ETHERNET_MAC_64` and transfer the configured MAC address `IEEE1722TpStreamIdMacAddress` of the corresponding `IEEE1722Tp-Stream` to `ETHERNET_MAC_64` and set the remaing bits to zero.
- create an ACF-payload by considering all ACF-messages which are enqueued in corresponding ACF-transmission-queue and, if available, the ACF-message which triggered the transmission
- concatinate ACF-header and ACF-payload to form an AVTPDU
- Update `PduInfoPtr.SduDataPtr` of LL-PDU with memory start address of the created AVTPDU and `PduInfoPtr.SduLength` of LL-PDU with length of the created AVTPDU
- Call `LSduR_IEEE1722TpTransmit` with `TxPduId` set to `PduId` of LL-PDU and `PduInfoPtr` set to `PduInfoPtr` of LL-PDU.

⌋

Note to [CP_SWS_IEEE1722Tp_00054]: Refer to Chapter 7.5.10 for details of AVTP-header creation

**[CP_SWS_IEEE1722Tp_00055]**

*Status:* DRAFT

⌈If the evaluation of the transmission trigger conditions qualifies to transmit an ACF-transmission-queue and the transmission process is aborted, or transmission is successful or unsuccessful finalized, then the IEEE1722Tp module shall flush the affected ACF-transmission-queue.⌋

**[CP_SWS_IEEE1722Tp_00056]**

*Status:* DRAFT

⌈If an ACF-message is forwarded to the ACF-stream handling in context of the internal transmission request processing, then the IEEE1722Tp module shall evaluate the

transmission trigger conditions, by considering the corresponding ACF-transmission-queue and the configuration of its associated IEEE1722TpStreamACF, and the transmission trigger configuration of the corresponding IEEE1722TpStreamAcfTxPdu TRIGGER_ALWAYS or TRIGGER_NEVER.⌋

**[CP_SWS_IEEE1722Tp_00057]**

*Status:* DRAFT

⌈If an ACF-message is handled in the evaluation of transmission trigger conditions and the corresponding IEEE1722TpStreamAcfTxPdu has IEEE1722TpStreamAcfTriggerMode configured with TRIGGER_ALWAYS, then the evaluation shall qualify to trigger a transmission.⌋

**[CP_SWS_IEEE1722Tp_00058]**

*Status:* DRAFT

⌈If an ACF-message is handled in the evaluation of transmission trigger conditions, the corresponding IEEE1722TpStreamAcfTxPdu has IEEE1722TpStreamAcfTriggerMode configured with TRIGGER_NEVER, the associated IEEE1722TpStreamACF has an IEEE1722TpAcfCollectionThreshold configured and enqeueing of this ACF-message in the corresponding ACF-transmission-queue would exceed the fill-size configured with IEEE1722TpAcfCollectionThreshold of this queue, then the evaluation shall qualify to trigger a transmission.⌋

**[CP_SWS_IEEE1722Tp_00059]**

*Status:* DRAFT

⌈If an ACF-message is enqueued in the corresponding ACF-transmission-queue and this ACF-transmission queue is empty, and the associated IEEE1722TpStreamACF has an IEEE1722TpAcfCollectionTimeout configured, then the timeout timer of this ACF-transmission-queue shall be started.⌋

**[CP_SWS_IEEE1722Tp_00060]**

*Status:* DRAFT

⌈If an ACF-transmission-queue is flushed and the associated IEEE1722TpStreamACF has an IEEE1722TpAcfCollectionTimeout configured, then the timeout timer of this ACF-transmission-queue shall be stopped.⌋

**[CP_SWS_IEEE1722Tp_00061]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall perform timeout timer handling in context of IEEE1722Tp_MainFunctionTx for all ACF-transmission-queues which are associated with an IEEE1722TpStreamACF where an IEEE1722TpAcfCollectionTimeout is configured.⌋

**[CP_SWS_IEEE1722Tp_00062]**

*Status:* DRAFT

⌈If the IEEE1722Tp detect that an timeout timer expired of an ACF-transmission-queue, than the IEEE1722Tp module shall trigger to transmit this ACF-transmission-queue according to [CP_SWS_IEEE1722Tp_00054]⌋

**[CP_SWS_IEEE1722Tp_00063]**

*Status:* DRAFT

⌈If an ACF-message is handled in the evaluation of transmission trigger conditions and no trigger for transmission could be identified, then the evaluation shall qualify to collect the ACF-message.⌋

**[CP_SWS_IEEE1722Tp_00064] Mixing of ACF-message types**

*Status:* DRAFT

⌈In case `IEEE1722TpStreamAcfMixedBusTypeCollection` is set to `FALSE` for an `IEEE1722TpStreamACF`, then the IEEE1722Tp module shall collect ACF-messages separately for each ACF-message type for that `IEEE1722TpStreamACF`.⌋

Note: Setting `IEEE1722TpStreamAcfMixedBusTypeCollection` to `FALSE` still allows mixing ACF-messages with different ACF-message type in one ACF-stream, but each frame transmitted on the network for that ACF-stream carries only ACF-messages with the same ACF-message type.

#### 7.5.9.2 L-SDU handling and forwarding

As described in the superordinate chapter, arrived L-SDUs, which where transported by an `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `TIME_SYNCHRONOUS`, need a time-dependent forwarding. Such L-SDUs are forwarded in dependency to the given AVTP presentation time, which is encoded in AVTPDU header of an IEEE1722 stream. Therefore the IEEE1722Tp need to provide an internal space where arrived L-SDUs are temporally stored, togehter with additional individual information (e.g. ACF-message timestamp). This space is called "L-SDU-waiting-area". If the AVTP presentation time expires of an L-SDU, then this L-SDU is forwarded to the LSduR together with individual information as meta data. The L-SDU will stay in the L-SDU-waiting-area until `LSduR_IEEE1722TpRxIndication` returns or `IEEE1722Tp_ReleaseRxBuffer` (see Chapter 7.5.8 for detailed information) is called. A L-SDU which was forwarded to LSduR is considered as transmitted and therefore excluded from the timeout supervision of ATVP presentation time.

**[CP_SWS_IEEE1722Tp_00065]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall provide an internal memory space (L-SDU-waiting-area) for a temporally stay of arrived L-SDUs, if `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `TIME_SYNCHRONOUS` are configured.⌋

**[CP_SWS_IEEE1722Tp_00066]**

*Status:* DRAFT

⌈If an L-SDU is forwarded to the ACF-stream handling in context of the internal reception indication processing which where transported by an IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS and the L-SDU-waiting-area has space left over to store this L-SDU together with individual information (e.g. ACF-message timestamp), then the IEEE1722Tp shall add this L-SDU together with its individual information to the L-SDU-waiting-area. Otherwise abort the ACF-stream handling for this L-SDU and, if development error detection is enabled (IEEE1722TpDevErrorDetect set to TRUE), report development error IEEE1722TP_E_LSDU_WAITING_AREA_OVERRUN.⌋

**[CP_SWS_IEEE1722Tp_00067]**

*Status:* DRAFT

⌈If an ACF-stream handling is aborted for a L-SDU, then the IEEE1722Tp module shall release local produced data and remove the L-SDU from further handling.⌋

**[CP_SWS_IEEE1722Tp_00068]**

*Status:* DRAFT

⌈If an L-SDU is added to the L-SDU-waiting-area, then the IEEE1722Tp shall start the timeout supervision for the according AVTP presentation time of this L-SDU.⌋

**[CP_SWS_IEEE1722Tp_00069]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall perform timeout supervision of the AVTPDU presentation time for L-SDUs which resides in the L-SDU-waiting-area in context of IEEE1722Tp_MainFunctionRx.⌋

**[CP_SWS_IEEE1722Tp_00070]**

*Status:* DRAFT

⌈If the IEEE1722Tp module detects that an AVTP presentation time exceeds of an L-SDU before the next call of IEEE1722Tp_MainFunctionRx, then the L-SDU handling shall qualify to forward the L-SDU to LSduR.⌋

Note: The AVTP presentation time timeout supervision is handled with the resolution of the configured IEEE1722TpMainFunctionRxPeriod

**[CP_SWS_IEEE1722Tp_00071]**

*Status:* DRAFT

⌈If an L-SDU was forwarded to the LSduR (see [CP_SWS_IEEE1722Tp_00073]), then the IEEE1722Tp shall stop the timeout supervision for the according AVTP presentation time of this L-SDU and consider this L-SDU as forwarded.⌋

Note: A L-SDU will be removed from the L-SDU-waiting-area in dependency of the corresponding global PDU configuration: `KeepLocalPduBuffer`. Please refer to Chapter 7.5.8 for detailed information.

**[CP_SWS_IEEE1722Tp_00072]**

*Status:* DRAFT

⌈If an L-SDU is forwarded to the ACF-stream handling in context of the internal reception indication processing which where transported by an `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `NON_TIME_SYNCHRONOUS`, then the L-SDU handling shall qualify to forward the L-SDU to the LSduR.⌋

**[CP_SWS_IEEE1722Tp_00073]**

*Status:* DRAFT

⌈If the L-SDU handling qualifies to forward an L-SDU to the LSduR, then the IEEE1722Tp shall perform the following actions:

- Identiy the frame-type id (e.g. CAN-ID) associated with the L-SDU and select the PDU-ID of the global PDU which is referenced via `IEEE1722TpStreamAcfRxPdu` and associated with the frame-type id, i.e. has the same frame-type id configured (e.g. `IEEE1722TpStreamAcfCanId`)
- Produce meta data according the configuration of the global PDU:
  - If `IEEE1722TP_COMMON_STREAM_HEADER_PTR` is configured, then produce an instance of `IEEE1722Tp_CommonStreamHeaderType` and transfer the AVTPDU presentation time associated with this L-SDU to IEEE1722Tp_CommonStreamHeaderType.avtp_timestamp. Set the remaining fields to "value not provided". Use the pointer to the produced instance of `IEEE1722Tp_CommonStreamHeaderType` as meta data
  - If `MESSAGE_TIMESTAMP_64` is configured, then produce an instance of 64 bit type and transfer the message_timestamp (64bit) (see [1, IEEE1722] chapter "9.4 ACF messages") associated with this L-SDU to the produced instance.
  - If `MESSAGE_TIMESTAMP_VALID_8` is configured, then produce an instance of an 8 bit type and transfer the mtv (message_timestamp valid: 1 bit) (see [1, IEEE1722] chapter "9.4 ACF messages") associated with this L-SDU to the least signification bit of the produced instance, and set the remaing bits to 0
  - If `CAN_ID_32` is configured, then produce an instance of an 32 bit type and transfer the following header field values associated with the L-SDU to the produced instance:
    - transfer can_identifier (29 bits) (see [1, IEEE1722] chapter "9.4.3 CAN/-CAN FD message") to bit position 0 to 28

    - transfer fdf (CAN Flexible Data-rate (FD) format) (1 bit) (see [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message") to bit position 30
    - transfer eff (extended frame format) (1 bit) (see [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message") to bit position 31
  - If `CAN_ID_PROPS_8` is configured, then produce an instance of an 8 bit type and transfer the following header field values associated with the L-SDU to the produced instance:
    - transfer rtr (remote transmission request) (1 bit) (see [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message") to bit position 0
    - transfer brs (bit rate switch) (1 bit) (see [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message") to bit position 1
    - transfer esi (error state indicator) (1 bit) (see [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message") to bit position 2
  - If `LIN_NAD_8` is configured, then produce an instance of an 8 bit type and transfer the lin_identifier (8 bits) (see [1, IEEE1722] chapter "9.4.5 LIN message") associated with the L-SDU to the produced instance.
- Update `PduInfoPtr.SduDataPtr` of identified global PDU with memory start address of L-SDU and `PduInfoPtr.SduLength` of identified global PDU with length of L-SDU, and, if available, update `PduInfoPtr.MetaDataPtr` with memory start address of produced and serialized meta data items
- Call `LSduR_IEEE1722TpRxIndication` with `RxPduId` set to `PduId` of identified global PDU and `PduInfoPtr` set to `PduInfoPtr` of identified global PDU.

⌋

Note for [CP_SWS_IEEE1722Tp_00073]: Refer to Chapter 8.2 for details on `IEEE1722Tp_CommonStreamHeaderType`

### 7.5.10 AVTPDU-header creation

The creation of AVTPDU-header is based on the AVTPDU-header format specified by [1, IEEE1722]. [1, IEEE1722] specify 4 different formats of the AVTPDU-header format: AVTPDU-common-header, AVTPDU-common-stream-header, AVTPDU-common-control-header and AVTPDU-alternative-header. Some of the header fields, which need a specific treatment and shared between the different formats (e.g. stream id), are embraced in sub-chapter Chapter 7.5.10.1. The subsequential sub-chapters describe how to set the header field values of AVTPDU-common-header, AVTPDU-common-stream-header, AVTPDU-alternative-header and the AVTP subtype specific format. Please note, AVTPDU-common-control-header fields are not considered, since the supported `IEEE1722TpStream`s in AUTOSAR do not use the AVTPDU-common-control-header format.

#### 7.5.10.1 Treatment of shared AVTPDU-header fields

**[CP_SWS_IEEE1722Tp_00074]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall maintain for each configured IEEE1722TpStream with either the following AVTPDU-stream properties:

- IEEE1722TpStream with AVTP common-stream-header format
- IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamCRF
- IEEE1722TpStream AVTP stream data subtype IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to NON_TIME_SYNCHRONOUS

a separate sequence number and consider the following points:

- The sequence number of an particular IEEE1722TpStream shall be increased with $01_{16}$ on each request for header creation
- If the sequence number reaches the maximum value, then it should re-start with value $00_{16}$

⌋

**[CP_SWS_IEEE1722Tp_00075]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream with either the following AVTPDU-stream properties:

- IEEE1722TpStream with AVTP common-stream-header format
- IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamCRF
- IEEE1722TpStream AVTP stream data subtype IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to NON_TIME_SYNCHRONOUS

then the IEEE1722Tp module shall consider the following points for creation of the IEEE1722 stream id:

- If the MAC address is provided via meta data and qualified as valid, then the provided MAC address shall be used. Otherwise the IEEE1722Tp module shall use the configured MAC address (IEEE1722TpStreamIdMacAddress) of the processed IEEE1722TpStream.
- If the unique id is provided via meta data and qualified as valid, then the provided unique id shall be used. Otherwise the IEEE1722Tp module shall use the configured unique id (IEEE1722TpStreamIdUniquePart) of the processed IEEE1722TpStream.

⌋

**[CP_SWS_IEEE1722Tp_00076]**

*Status:* DRAFT

⌈The IEEE1722Tp module shall call `StbM_GetTimeBaseStatus` as specified with [CP_SWS_IEEE1722Tp_00015] and determine the tu header field value with respect to the following rules:

- if `StbM_GetTimeBaseStatus` return with `E_NOT_OK`, then tu (time uncertain) header field shall be set to 1
- if `StbM_GetTimeBaseStatus` return with `E_OK`, then IEEE1722Tp module shall evaluate the retrieven status bits and set tu (time uncertain) header field to 1 if (GLOBAL_TIME_BASE is set) AND ((RATE_CORRECTED not set) and at least one the remaining bits is set)), or GLOBAL_TIME_BASE is not set. Otherwise the IEEE1722Tp module shall set the tu (time uncertain) header field value to 0.

⌋

**[CP_SWS_IEEE1722Tp_00077]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` and the accumulated length of AVTPDU-header and AVTP-payload exceed the MTU (Maximum Transmission Unit) of the underlying physical transport layer, then the IEEE1722Tp module shall abort the AVTPDU-header creation, and, if development error detection is enabled (`IEEE1722TpDevErrorDetect` set to `TRUE`), report development error `IEEE1722TP_E_IEEE1722_STREAM_EXCEED_MTU`.⌋

Note: MTU is configured in Ethernet Interface module at each `EthIfController` with `EthIfCtrlMtu`.

#### 7.5.10.2 AVTPDU-common-header fields

The AVTPDU-common-header format is shared between all AVTP stream data subtypes. This chapter describe how to create and set values of the AVTPDU-common-header fields according to [1, IEEE1722] chapter "4.4.3 AVTPDU common header format".

**[CP_SWS_IEEE1722Tp_00078]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream`, then the subtype field shall be set with the AVTP stream data subtype according to [1, IEEE1722] chapter "4.4.3 AVTPDU common header format":

- If the processed `IEEE1722TpStream` has `IEEE1722TpStreamCRF` configured, then the subtype field shall be set to AVTP stream data subtype value $04_{16}$

- If the processed IEEE1722TpStream has IEEE1722TpStreamAAF configured, then the subtype field shall be set to AVTP stream subtype value $02_{16}$
- If the processed IEEE1722TpStream has IEEE1722TpStreamIIDC configured, then the subtype field shall be set to AVTP stream subtype value $00_{16}$
- If the processed IEEE1722TpStream has IEEE1722TpStreamRVF configured, then the subtype field shall be set to AVTP stream subtype value $07_{16}$
- If the processed IEEE1722TpStream has IEEE1722TpStreamACF configured with IEEE1722TpStreamAcfHeaderType set to NON_TIME_SYNCHRONOUS, then the subtype field shall be set to AVTP stream subtype value $82_{16}$
- If the processed IEEE1722TpStream has IEEE1722TpStreamACF configured with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS, then the subtype field shall be set to AVTP stream subtype value $05_{16}$

⌋

**[CP_SWS_IEEE1722Tp_00079]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream, then the IEEE1722Tp module shall set the version field to the configured value of IEEE1722TpStreamVersion (see [1, IEEE1722] chapter "4.4.3 AVTPDU common header format")⌋

Note: The value for the h (header specific) field is specified in chapter Chapter 7.5.10.3 and chapter Chapter 7.5.10.4

### 7.5.10.3 AVTPDU-common-stream-header fields

**[CP_SWS_IEEE1722Tp_00080]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream with AVTPDU-common-stream-header format and media clock restart value is available via meta data and qualified as valid, then the IEEE1722Tp module shall set the mr (media clock restart) field to the given value (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header"). Otherwise set this header field to zero.⌋

**[CP_SWS_IEEE1722Tp_00081]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream with AVTPDU-common-stream-header format, then the IEEE1722Tp module shall set sv (stream_id valid) field to 1 (see [1, IEEE1722] chapter "4.4.4.2 sv (stream_id valid) field")⌋

**[CP_SWS_IEEE1722Tp_00082]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` with AVTPDU-common-stream-header format and the avtp_timestamp valid value is available via meta data and qualified as valid, then the IEEE1722Tp module shall set the tv (avtp_timestamp valid) field to the given value (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header"). Otherwise set this header field to zero.⌋

**[CP_SWS_IEEE1722Tp_00083]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` with AVTPDU-common-stream-header format, then the IEEE1722Tp module shall set the sequence_num (sequence number) field to the current value with respect to (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header") and [CP_SWS_IEEE1722Tp_00074]⌋

**[CP_SWS_IEEE1722Tp_00084]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` with AVTPDU-common-stream-header format, then the IEEE1722Tp module shall set the tu (timestamp uncertain) field (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header") to the determined value as specified with [CP_SWS_IEEE1722Tp_00076].⌋

**[CP_SWS_IEEE1722Tp_00085]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` with AVTPDU-common-stream-header format, then the IEEE1722Tp module shall construct a stream id with respect to [CP_SWS_IEEE1722Tp_00075] and set the stream_id (stream id) field of the AVTPDU-header (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header") to the constructed stream id.⌋

**[CP_SWS_IEEE1722Tp_00086]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` with AVTPDU-common-stream-header format and the avtp timestamp value is available via meta data and qualified as valid, then the IEEE1722Tp module shall set the avtp_timestamp (avtp timestamp) field of the AVTPDU-header (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header") to available avtp timestamp value. Otherwise the IEEE1722Tp module shall calculate avtp timestamp according the following equation and set the avtp_timestamp (avtp

timestamp) field of the AVTPDU-headerfield to the calcuated presentation time:

$$T_{\text{presentation_time}} = T_{\text{current_synchronized_globaltime}} + T_{\texttt{IEEE1722TpStreamMaxTransitTime}} \quad (7.1)$$

⌋

**[CP_SWS_IEEE1722Tp_00087]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` with AVTPDU-common-stream-header format, then the IEEE1722Tp module shall set stream_data_length (stream data length) field of the AVTP-payload (see [1, IEEE1722] chapter "4.4.4 AVTPDU common stream header") to length in bytes given with `PduInfoPtr` of the processed transmission request.⌋

#### 7.5.10.4 AVTPDU-alternative-header fields

The AVTPDU-alternative-header fields are AVTP stream data subtype specific and described in the according subchapters for `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamCRF` and `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `NON_TIME_SYNCHRONOUS`.

#### 7.5.10.5 61883_IIDC-header fields

This chapter describe how to create values which are specific for AVTP stream data subtype "61883_IIDC" (IEC 61883/IIDC format) according to [1, IEEE1722] chapter "5. IEC 61883/IIDC Format".

**[CP_SWS_IEEE1722Tp_00088]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamIIDC` and `IEEE1722TpStreamIidcTag` is set to 0, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "5.2 IEC 61883/IIDC stream data encapsulation" in addition to the AVTPDU-common-header fields and AVTPDU-common-stream-header fields:

- Set gv (gateway_info valid) field to zero.
- Set gateway_info field to zero.
- Set tag field to configured value of `IEEE1722TpStreamIidcTag`.
- Set channel field to configured value of `IEEE1722TpStreamIidcChannel`.
- Set tcode (type code) field to configured value of `IEEE1722TpStreamIidcTCode`.

- Set sy field to configured value of IEEE1722TpStreamIidcSy.

⌋

Note for [CP_SWS_IEEE1722Tp_00088]: Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields and to Chapter 7.5.10.3 for details on AVTPDU-common-stream-header fields.

**[CP_SWS_IEEE1722Tp_00089]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamIIDC and IEEE1722TpStreamIidcTag is set to 1, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "5.4.3 IEC 61883 CIP header encapsulation" in addition to [CP_SWS_IEEE1722Tp_00088]:

- Set qi_1 (quadlet indicator) field to $00_2$.
- Set SID (source identifier) field to $63_{10}$.
- Set DBS (data block size) field to configured value of IEEE1722TpStreamIidc-DataBlockSize.
- Set FN (fraction number) field to configured value of IEEE1722TpStreamIidc-FractionNumber.
- Set QPC (quadlet padding count) field to value provided via meta data. If value is not available or invalid, set the value to 0.
- Set SPH (source packet header) field to configured value of IEEE1722Tp-StreamIidcSourcePacketHeader.
- Set DBC (data block count) field to value provided via meta data. If value is not available or invalid, set the value to 0.
- Set qi_2 (quadlet indicator) field to $10_2$
- Set FMT (stream format) field to configured value of IEEE1722TpStreamIidc-StreamFormat.

⌋

Note: AUTOSAR do not support AVTP gateway function

**[CP_SWS_IEEE1722Tp_00090]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamIIDC, IEEE1722TpStreamIidcTag is set to 1 and IEEE1722TpStreamIidcSource-PacketHeader is set to 0, then the IEEE1722Tp module shall set the values for the

specific header fields to the following values according to [1, IEEE1722] chapter "5.4.4 IEC 61883 (SPH = 0) encapsulation" in addition to [CP_SWS_IEEE1722Tp_00089]:

- Set tv (avtp_timestamp valid) field to value provided via meta data. If value is not available or invalid, set the value to 0.
- Set avtp_timestamp field according to [CP_SWS_IEEE1722Tp_00086].
- Set FDF (format dependent field) field to value provided via meta data. If value is not available or invalid, set the value to 0.
- Set SYT (synchronization timing) field to $FFFF_{16}$
- Set cip_no_sph_payload field to data given with `PduInfoPtr` (SduDataPtr).

⌋

**[CP_SWS_IEEE1722Tp_00091]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamIIDC`, `IEEE1722TpStreamIidcTag` is set to 1 and `IEEE1722TpStreamIidcSourcePacketHeader` is set to 1, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "5.4.4 IEC 61883 (SPH = 1) encapsulation" in addition to [CP_SWS_IEEE1722Tp_00089]:

- Set tv (avtp_timestamp valid) field to value 0.
- Set FDF (format dependent field) field to value provided via meta data. If value is not available or invalid, set the value to 0.
- Set cip_with_sph_payload field to data given with `PduInfoPtr` (SduDataPtr).

⌋

Note: The avtp_source_packet_header_timestamp field is included in the cip_with_sph_payload. The cip_with_sph_payload could include multiple source packets.

#### 7.5.10.6 AAF-header fields

This chapter describe how to create values which are specific for AVTP stream data subtype "AAF" (AVTP Audio Format) according to [1, IEEE1722] chapter "7. AVTP Audio Format".

**[CP_SWS_IEEE1722Tp_00092]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamAAF`,

then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "7.2 AAF common stream data encapsulation" in addition to the AVTPDU-common-header fields and AVTPDU-common-stream-header fields:

- Set format field to value of IEEE1722TpStreamAafFormat.
- Set sp (sparse timestamp) field to value of IEEE1722TpStreamAafSparseTimestamp.
- Set evt field to configured value of IEEE1722TpStreamAafEventDefault.

⌋

Note for [CP_SWS_IEEE1722Tp_00092]: Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields and to Chapter 7.5.10.3 for details on AVTPDU-common-stream-header fields.

**[CP_SWS_IEEE1722Tp_00093]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamAAF and IEEE1722TpStreamAafFormat is set to value that indicates AAF AVTP format PCM, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "7.3 AAF PCM stream data encapsulation" additional to [CP_SWS_IEEE1722Tp_00092]:

- Set nsr (nominal sample rate) field to value of IEEE1722TpStreamAafPcmNominalSampleRate.
- Set channels_per_frame field to value of IEEE1722TpStreamAafPcmChannelsPerFrame.
- Set bit_depth field to configured value of IEEE1722TpStreamAafPcmBitDepth.
- Set pcm_data_payload field to data given with PduInfoPtr (SduDataPtr).

⌋

**[CP_SWS_IEEE1722Tp_00094]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamAAF and IEEE1722TpStreamAafFormat is set to value that indicates AAF AVTP format AES3, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "7.3 AAF PCM stream data encapsulation" additional to [CP_SWS_IEEE1722Tp_00092]:

- Set nfr (nominal frame rate) field to value of IEEE1722TpStreamAafAes3NominalFrameRate.
- Set streams_per_frame field to value of IEEE1722TpStreamAafAes3StreamsPerFrame.
- Set aes3_data_type_h field to configured value of IEEE1722TpStreamAafAes3DataTypeH.
- Set aes3_dt_ref field to configured value of IEEE1722TpStreamAafAes3DataTypeRef.
- Set aes3_data_type_l field to configured value of IEEE1722TpStreamAafAes3DataTypeL.

⌋

#### 7.5.10.7 ACF-header fields

This chapter describe how to create values which are specific for AVTP stream data subtype "ACF" (AVTP Control Format) according to [1, IEEE1722] chapter "9. AVTP Control Format".

Note: AUTOSAR do not support stream reservation protocol (SRP), but due to [1, IEEE1722] chapter "4.4.4.2 sv (stream_id valid) field" the sv field is always set to 1 (see [CP_SWS_IEEE1722Tp_00081])

**[CP_SWS_IEEE1722Tp_00095]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to NON_TIME_SYNCHRONOUS, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "9.2 Non-Time-Synchronous Control Format header" in addition to the AVTPDU-common-header fields:

- Set sv (stream_id valid) field to $01_{16}$ (see [CP_SWS_IEEE1722Tp_00081]).
- Set ntscf_data_length field to the accumlated length of all ACF-messages transmitted as AVTPDU-payload of this IEEE1722TpStream.
- Set acf_payload_data field to the data of the concatenated ACF-messages which belong to the transmission request for this IEEE1722TpStream.

⌋

Note for [CP_SWS_IEEE1722Tp_00095]: Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields.

**[CP_SWS_IEEE1722Tp_00096]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an IEEE1722TpStream of AVTP stream data subtype IEEE1722TpStream-ACF with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "9.3 Time-Synchronous Control Format header" in addition to the AVTPDU-common-header fields and AVTPDU-common-stream-header fields:

- Set stream_data_length to the accumlated length of all ACF-messages transmitted as AVTPDU-payload of this IEEE1722TpStream.
- Set acf_payload_data field to the data of the concatenated ACF-messages which belong to the transmission request for this IEEE1722TpStream.

⌋

Note for [CP_SWS_IEEE1722Tp_00096]: Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields and to Chapter 7.5.10.3 for details on AVTPDU-common-stream-header fields.

### 7.5.10.8 ACF-message creation

An ACF-message is transported as ACF-message-payload of a NTSCF (non time synchronous control format) or TSCF (time synchronous control format). The ACF-messsage-payload can carry one or more arbitrary ACF-messages. This chapter describe how to create values which are common for ACF-messages according to [1, IEEE1722] chapter "9.4 ACF messages"

**[CP_SWS_IEEE1722Tp_00097]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an ACF-message creation for an IEEE1722TpStream, then the acf message field shall be set according to [1, IEEE1722] chapter "9.4.1.2 acf_msg_type field":

- If the processed IEEE1722TpStream has IEEE1722TpStreamACF configured with IEEE1722TpStreamAcfPayload set to IEEE1722TpStreamAcf-Can and IEEE1722TpStreamAcfCanMessageType is set to CAN, then the acf_msg_type field shall be set to ACF-message type value $01_{16}$
- If the processed IEEE1722TpStream has IEEE1722TpStreamACF configured with IEEE1722TpStreamAcfPayload set to IEEE1722TpStreamAcf-Can and IEEE1722TpStreamAcfCanMessageType is set to CAN_BRIEF, then the acf_msg_type field field shall be set to ACF-message type value $02_{16}$

- If the processed IEEE1722TpStream has IEEE1722TpStreamACF configured with IEEE1722TpStreamAcfPayload set to IEEE1722TpStreamAcfLin, then the acf_msg_type field shall be set to ACF-message type value $03_{16}$

⌋

**[CP_SWS_IEEE1722Tp_00098]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an ACF-message creation for an IEEE1722TpStreamACF configured with IEEE1722TpStreamAcfPayload, then the acf_msg_length field shall be set to the accumlated length of specific ACF-message-header (e.g. ACF_CAN (including acf_msg_subtype and acf_msg_length) and length of the corresponding ACF-message-payload given with PduInfoPtr.SduLength of the processed transmission request (see [1, IEEE1722] chapter "9.4.1.3 acf_msg_length field")⌋

**[CP_SWS_IEEE1722Tp_00099]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an ACF-message creation for an IEEE1722TpStream with IEEE1722TpStreamAcfCan set to CAN or CAN_BRIEF, then the acf_msg_payload field shall contain the payload data given with PduInfoPtr (SduDataPtr) of the processed transmission request (see [1, IEEE1722] chapter "9.4.1.4 acf_msg_payload")⌋

#### 7.5.10.8.1 ACF_CAN message fields

This chapter describe how to create values which are specific for ACF_CAN (encapsulation of CAN/CAN-FD frames) according to [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message".

**[CP_SWS_IEEE1722Tp_00100]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an ACF_CAN message creation for an IEEE1722TpStream with IEEE1722TpStreamAcfCanMessageType set to CAN, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "9.4.3 CAN/CAN FD message" in addition to [CP_SWS_IEEE1722Tp_00098] and [CP_SWS_IEEE1722Tp_00099]:

- Set pad (padding length) field to value of accumlated length of padding bytes at the and of the payload to align the payload to 32 bit boundary.
- Set mtv (message_timestamp_valid) field to 1 if message time stamp (see below "message_timestamp") is valid, otherwise to 0.
- Set rtr (remote_transmission_request) field to the value provided via meta data.

- Set eff (extended_frame_format) field to 0 if CAN frame has an 11-bit CAN identifier, otherwise to 1 for 29-bit CAN identifier.
- Set brs (bit_rate_switch) field to value provided via meta data.
- Set fdf (CAN Flexible Data-rate [FD] Format) to value provided via meta data. (Note: 0 == frame is an CAN frame with at most 8 byte payload, 1 == CAN-FD frame with at most 64 byte payload).
- Set esi (error_state_indicator) field to value provided via meta data.
- Set can_bus_id field to configured value of IEEE1722TpStreamAcfBusId of the IEEE1722TpStreamAcfCan associated with the processed IEEE1722TpStream, if available. Otherwise set the value to 0.
- Set message_timestamp field with the value of the current synchronized global time.
- Set can_identifier field to the CAN identifier (either re-constructed via meta data or configured as IEEE1722TpStreamAcfCanId of the IEEE1722TpStreamAcfCan associated with the processed IEEE1722TpStream).
- Set can_msg_payload field to data given with PduInfoPtr (SduDataPtr).

⌋

Note for [CP_SWS_IEEE1722Tp_00100]: Refer to Chapter 7.4 for details on global time related handling.

#### 7.5.10.8.2 ACF_CAN_BRIEF message fields

This chapter describe how to create values which are specific for ACF_CAN_BRIEF (encapsulation of CAN/CAN-FD frames) according to [1, IEEE1722] chapter "9.4.4 Abbreviated CAN/CAN FD message".

**[CP_SWS_IEEE1722Tp_00101]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an ACF_CAN_BRIEF message creation for an IEEE1722TpStream with IEEE1722TpStreamAcfCanMessageType set to CAN_BRIEF, then the IEEE1722Tp module shall set the values for the specific header fields as described in [CP_SWS_IEEE1722Tp_00100], but skip the message_timestamp field⌋

Note: ACF_CAN_BRIEF has no timestamp field defined

#### 7.5.10.8.3 ACF_LIN message fields

This chapter describe how to create values which are specific for ACF_LIN (encapsulation of LIN frames) according to [1, IEEE1722] chapter "9.4.5 LIN® message".

**[CP_SWS_IEEE1722Tp_00102]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an ACF_LIN message creation for an `IEEE1722TpStream` with `IEEE1722TpStreamAcfPayload` set to `IEEE1722TpStreamAcfLin`, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "9.4.5 LIN® message" in addition to [CP_SWS_IEEE1722Tp_00098] and [CP_SWS_IEEE1722Tp_00099]:

- Set pad (padding length) field to value of accumlated length of padding bytes at the and of the payload to align the payload to 32 bit boundary.
- Set mtv (message_timestamp_valid) field to 1 if message time stamp (see below "message_timestamp") is valid, otherwise to 0.
- Set lin_bus_id field to configured value of `IEEE1722TpStreamAcfBusId` of the `IEEE1722TpStreamAcfLin` associated with the processed `IEEE1722TpStream`, if available. Otherwise set the value to 0.
- Set lin_identifier field to the LIN identifier (either re-constructed via meta data or configured as `IEEE1722TpStreamAcfLinId` of the `IEEE1722TpStreamAcfLin` associated with the processed `IEEE1722TpStream`).
- Set message_timestamp field with the value of the current synchronized global time.
- Set lin_msg_payload field to data given with `PduInfoPtr` (SduDataPtr).

⌋

Note for [CP_SWS_IEEE1722Tp_00102]: Refer to Chapter 7.4 for details on global time related handling.

#### 7.5.10.9 CRF-header fields

This chapter describe how to create values which are specific for AVTP stream data subtype "CRF" (Clock Reference Format) according to [1, IEEE1722] chapter "10. Clock Reference Format".

**[CP_SWS_IEEE1722Tp_00103]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamCRF`,

then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "10.4 Clock Reference Format Data encapsulation" in addition to the AVTPDU-common-header fields:

- Set sv (stream_id valid) field to value provided via meta data and qualified as valid. Otherwise set this field to 0.
- Set mr (media clock reset) field to value provided via meta data and qualified as valid. Otherwise set this field to 0.
- Set fs (frame sync) field to value provided via meta data and qualified as valid. Otherwise set this field to 0.
- Set tu (timestamp uncertain) field to value determined as specified with [CP_SWS_IEEE1722Tp_00076]. Otherwise set this field to 1.
- Set sequence_num field to value determined as specified with [CP_SWS_IEEE1722Tp_00074]
- Set type field to value of `IEEE1722TpStreamCrfType`.
- Set stream_id field to value determined as specified with [CP_SWS_IEEE1722Tp_00075]
- Set pull field to configured value of `IEEE1722TpStreamCrfPull`.
- Set base_frequency field to configured value of `IEEE1722TpStreamCrfBaseFrequency`.
- Set crf_data_length field to length in bytes of the AVTP-payload given with `PduInfoPtr` (SduLength) of the processed transmission request.
- Set timestamp_interval field to configured value of `IEEE1722TpStreamCrfTimestampInterval`.
- Set crf_data field to data given with `PduInfoPtr` (SduDataPtr).

⌋

Note to [CP_SWS_IEEE1722Tp_00103]:

- The remaining fields specified in [1, IEEE1722] chapter "10.4.13 crf_data field" reside in the crf_data field, which is provided within the CRF-payload by a upper layer module (e.g. CDD). The following fields are out of scope for the IEEE1722Tp module: User-specified type, Audio sample type, Video frame sync type, Video line sync type, Machine cycle type
- Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields.

#### 7.5.10.10 RVF-header fields

This chapter describe how to create values which are specific for AVTP stream data subtype "RVF" (Raw Video Format) according to [1, IEEE1722] chapter "10. Clock Reference Format".

**[CP_SWS_IEEE1722Tp_00104]**

*Status:* DRAFT

⌈If the internal transmission request processing performs an AVTPDU-header creation for an `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamRVF`, then the IEEE1722Tp module shall set the values for the specific header fields to the following values according to [1, IEEE1722] chapter "12.2 Raw Video Stream data encapsulation" in addition to the AVTPDU-common-header fields and AVTPDU-common-stream-header fields:

- Set active_pixels field configured value of provided via meta data. If value is not available or invalid, set this field to 0.
- Set total_lines field to configured value of `IEEE1722TpStreamRvfTotalLines`.
- Set ap (active pixels) field to configured value of `IEEE1722TpStreamRvfActivePixels`.
- Set f (field) field to configured value provided via meta data. If value is not available or invalid, set this field to 0.
- Set ef (end frame) field to configured value provided via meta data. If value is not available or invalid, set this field to 0.
- Set evt field to configured value of `IEEE1722TpStreamRvfEventDefault`
- Set pd (pull-down) field to configured value provided via meta data. If value is not available or invalid, set this field to 0.
- Set i (interlaced) field to configured value of `IEEE1722TpStreamRvfInterlaced`.
- Set pixel_depth field to configured value of `IEEE1722TpStreamRvfPixelDepth`.
- Set pixel_format field to configured value of `IEEE1722TpStreamRvfPixelFormat`.
- Set frame_rate field to configured value of `IEEE1722TpStreamRvfFrameRate`.
- Set colorspace field to configured value of `IEEE1722TpStreamRvfColorSpace`.
- Set num_lines field to configured value of provided via meta data. If value is not available or invalid, set this field to 0.

- Set i_seq_num field to configured value of provided via meta data. If value is not available or invalid, set this field to 0.
- Set line_number field to configured value of provided via meta data. If value is not available or invalid, set this field to 0.

⌋

Note for [CP_SWS_IEEE1722Tp_00104]: Refer to Chapter 7.5.10.2 for details on AVTPDU-common-header fields and to Chapter 7.5.10.3 for details on AVTPDU-common-stream-header fields.

### 7.5.11 AVTPDU-header inspection

Inspection of the AVTPDU-header include consistency of the received AVTPDU-header fields compared to the corresponding configuration. This used if an IEEE1722Tp module receives an IEEE1722 stream. The inspection of AVTPDU-header consider the AVTPDU-header format specified by [1, IEEE1722]. [1, IEEE1722] specify 4 different formats of the AVTPDU-header format: AVTPDU-common-header, AVTPDU-common-stream-header, AVTPDU-common-control-header and AVTPDU-alternative-header. Please note, AVTPDU-common-control-header fields are not considered, since the supported `IEEE1722TpStream`s in AUTOSAR do not use the AVTPDU-common-control-header format.

**[CP_SWS_IEEE1722Tp_00105]**

*Status:* DRAFT

⌈If the internal reception indication processing performs an AVTPDU-header inspection, then the IEEE1722Tp module shall inspect the AVTPDU-common-header fields according the format specified by [1, IEEE1722] and consider the following consistency checks:

- if value of subtype field of the inspected AVTPDU-header match to one of the supported IEEE1722 stream sub types (61883_IIDC, AAF, CRF, TSCF, RVF, NTSCF), then the IEEE1722Tp module shall proceed with the AVTPDU-header inspection. Otherwise the IEEE1722Tp module shall abort the AVTPDU-header inspection, and, if development error detection is enabled (`IEEE1722TpDevErrorDetect` set to `TRUE`), report development error `IEEE1722TP_E_IEEE1722_STREAM_NOT_SUPPORTED_SUBTYPE`
- if version field of the inspected AVTPDU-header match to the corresponding `IEEE1722TpStreamVersion`, then the IEEE1722Tp module shall proceed with the AVTPDU-header inspection. Otherwise the IEEE1722Tp module shall abort the AVTPDU-header inspection, and, if development error detection is enabled (`IEEE1722TpDevErrorDetect` set to `TRUE`), report development error `IEEE1722TP_E_IEEE1722_STREAM_VERSION_MISMATCH`

⌋

**[CP_SWS_IEEE1722Tp_00106]**

*Status:* DRAFT

⌈If the internal reception indication processing performs an AVTPDU-header inspection, then the IEEE1722Tp module shall inspect the AVTPDU-common-stream-header fields according the format specified by [1, IEEE1722] and consider the following consistency checks:

- if avtp_timestamp field of the inspected AVTPDU-header represents a time value that is greater than the time value of the current synchronized global time, then AVTPDU-header inspection shall proceed. Otherwise the IEEE1722Tp module shall abort the AVTPDU-header inspection, and, if development error detection is enabled (`IEEE1722TpDevErrorDetect` set to `TRUE`), report development error `IEEE1722TP_E_IEEE1722_STREAM_PRESENTATIONTIME_OUTDATED`

⌋

**[CP_SWS_IEEE1722Tp_00107]**

*Status:* DRAFT

⌈If the internal reception indication processing performs an AVTPDU-header inspection, then the IEEE1722Tp module shall inspect for each configured `IEEE1722TpStream` with either the following AVTPDU-stream properties:

- `IEEE1722TpStream` with AVTP common-stream-header format
- `IEEE1722TpStream` of AVTP stream data subtype `IEEE1722TpStreamCRF`
- `IEEE1722TpStream` AVTP stream data subtype `IEEE1722TpStreamACF` with `IEEE1722TpStreamAcfHeaderType` set to `NON_TIME_SYNCHRONOUS`

the header fields according the format specified by [1, IEEE1722] and consider the following consistency checks:

- if sequence_num (sequence number) increase continuously and warp around at reaching maximum value, then the AVTPDU-header inspection shall proceed. Otherwise the IEEE1722Tp module shall report an runtime error with `IEEE1722TP_E_IEEE1722_STREAM_DISCONTINUOUS_SEQUENCE_NUMBER`
- if stream id field of the inspected AVTPDU-header match to the corresponding composite of `IEEE1722TpStreamIdMacAddress` and `IEEE1722TpStreamIdUniquePart`, then AVTPDU-header inspection shall proceed. Otherwise the IEEE1722Tp module shall abort the AVTPDU-header inspection, and, if development error detection is enabled (`IEEE1722TpDevErrorDetect` set to `TRUE`), report development error `IEEE1722TP_E_IEEE1722_STREAM_ID_MISMATCH`

⌋

## 7.6 Meta data handling

The IEEE1722Tp module uses meta data as specified in [5, CP-SWS-BSWGeneral].

**[CP_SWS_IEEE1722Tp_CONSTR_00006]**

*Status:* DRAFT

⌈All PDUs which belong to the same IEEE1722TpStream (either referenced via IEEE1722TpUpperLayerTxPduPoolEntry or IEEE1722TpUpperLayerRxPduPoolEntry) shall have the same set of MetaDataTypes configured.⌋

### 7.6.1 Meta data item types

This sub chapters describe the expected meta data types, which are produces or consumed by IEEE1722Tp.

#### 7.6.1.1 IEEE1722 common stream header

The following MetaDataItemType is used for transmission and reception of IEEE1722 streams which use the IEEE1722 common stream header. This MetaDataItemType represents runtime values for common stream header fields, provided as pointer to an IEEE1722Tp_CommonStreamHeaderType:

- IEEE1722TP_COMMON_STREAM_HEADER_PTR

The MetaDataItemType is produced for transmission by an IEEE1722 data stream provider and consumed by the IEEE1722Tp module within the internal transmission request processing. The MetaDataItemType is produced for reception by the IEEE1722Tp module within the internal reception indication processing and consumed by the receiving IEEE1722 data stream consumer.

**[CP_SWS_IEEE1722Tp_CONSTR_00007]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerTxPduPoolEntry or IEEE1722TpUpperLayerRxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to

- IEEE1722TpStreamAAF or
- IEEE1722TpStreamIIDC or
- IEEE1722TpStreamRVF or
- IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS

shall have MetaDataItemType IEEE1722TP_COMMON_STREAM_HEADER_PTR configured.⌋

#### 7.6.1.2 IEEE1722 IEC61883/IIDC specific stream header

The following list reflect the used MetaDataItemTypes for transmission of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamIIDC. This MetaDataItemTypes represents runtime values provided as pointer to the according IEEE1722Tp specified type:

- IEEE1722TP_TX_IEC61883_IIDC_PTR represents a pointer to an IEEE1722Tp_TxIec61883IidcType
- IEEE1722TP_TX_IEC61883_PTR represents a pointer to an IEEE1722Tp_TxIec61883Type
- IEEE1722TP_TX_IEC61883_CIP_NO_SPH_PTR represents a pointer to an IEEE1722Tp_TxIec61883CipNoSphType
- IEEE1722TP_TX_IEC61883_CIP_WITH_SPH_PTR represents a pointer to an IEEE1722Tp_TxIec61883CipWithSphType

A specific MetaDataItemType is produced for transmission by an IEEE1722 data stream provider and consumed by the IEEE1722Tp module within the internal transmission request processing.

**[CP_SWS_IEEE1722Tp_CONSTR_00008]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerTxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamIIDC shall have one of the following MetaDataItemTypes configured:

- IEEE1722TP_TX_IEC61883_IIDC_PTR
- IEEE1722TP_TX_IEC61883_PTR
- IEEE1722TP_TX_IEC61883_CIP_NO_SPH_PTR
- IEEE1722TP_TX_IEC61883_CIP_WITH_SPH_PTR

⌋

The following list reflect the used MetaDataItemTypes for reception of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamIIDC. This MetaDataItemTypes represents runtime values provided as pointer to the according IEEE1722Tp specified type:

- IEEE1722TP_RX_IEC61883_IIDC_PTR represents a pointer to an IEEE1722Tp_RxIec61883IccType
- IEEE1722TP_RX_IEC61883_PTR represents a pointer to an IEEE1722Tp_RxIec61883Type

- IEEE1722TP_RX_IEC61883_CIP_NO_SPH_PTR represents a pointer to an IEEE1722Tp_RxIec61883CipNoSphType
- IEEE1722TP_RX_IEC61883_CIP_WITH_SPH_PTR represents a pointer to an IEEE1722Tp_RxIec61883CipWithSphType

A specific MetaDataItemType is produced for reception by the IEEE1722Tp module within the internal reception indication processing and consumed by the receiving IEEE1722 data stream consumer.

**[CP_SWS_IEEE1722Tp_CONSTR_00009]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerRxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamIIDC shall have one of the following MetaDataItemTypes configured:

- IEEE1722TP_RX_IEC61883_IIDC_PTR
- IEEE1722TP_RX_IEC61883_PTR
- IEEE1722TP_RX_IEC61883_CIP_NO_SPH_PTR
- IEEE1722TP_RX_IEC61883_CIP_WITH_SPH_PTR

⌋

#### 7.6.1.3 IEEE1722 AAF specific stream header

The following list reflect the used MetaDataItemTypes for transmission of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamAAF. This MetaDataItemTypes represents runtime values provided as pointer to the according IEEE1722Tp specified type:

- IEEE1722TP_TX_AAF_PCM_PTR represents a pointer to an IEEE1722Tp_TxAafPcmType
- IEEE1722TP_TX_AAF_AES3_PTR represents a pointer to an IEEE1722Tp_TxAafAes3Type

A specific MetaDataItemType is produced for transmission by an IEEE1722 data stream provider and consumed by the IEEE1722Tp module within the internal transmission request processing.

**[CP_SWS_IEEE1722Tp_CONSTR_00010]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerTxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to

IEEE1722TpStreamAAF shall have one of the following MetaDataItemTypes configured:

- IEEE1722TP_TX_AAF_PCM_PTR represents a pointer to an IEEE1722Tp_TxAafPcmType
- IEEE1722TP_TX_AAF_AES3_PTR represents a pointer to an IEEE1722Tp_TxAafAes3Type

⌋

The following list reflect the used MetaDataItemTypes for reception of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamAAF. This MetaDataItemTypes represents runtime values provided as pointer to the according IEEE1722Tp speficied type:

- IEEE1722TP_RX_AAF_PCM_PTR represents a pointer to an IEEE1722Tp_RxAafPcmType
- IEEE1722TP_RX_AAF_AES3_PTR represents a pointer to an IEEE1722Tp_RxAafAes3Type

A specific MetaDataItemType is produced for reception by the IEEE1722Tp module within the internal reception indication processing and consumed by the receiving IEEE1722 data stream consumer.

**[CP_SWS_IEEE1722Tp_CONSTR_00011]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerRxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamAAF shall have one of the following MetaDataItemTypes configured:

- IEEE1722TP_RX_AAF_PCM_PTR
- IEEE1722TP_RX_AAF_AES3_PTR

⌋

#### 7.6.1.4 IEEE1722 RVF stream header

The following MetaDataItemType is used for transmission of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamRVF. This MetaDataItemType represents runtime values provided as pointer to the according IEEE1722Tp speficied type:

- IEEE1722TP_TX_RVF_PTR represents a pointer to an IEEE1722Tp_TxRvfType

This MetaDataItemType is produced for transmission by an IEEE1722 data stream provider and consumed by the IEEE1722Tp module within the internal transmission request processing.

**[CP_SWS_IEEE1722Tp_CONSTR_00012]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerTxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamRVF shall have the MetaDataItemType IEEE1722TP_TX_RVF_PTR configured.⌋

The following MetaDataItemType is used for reception of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamRVF. This MetaDataItemType represents runtime values provided as pointer to the according IEEE1722Tp speficied type:

- IEEE1722TP_RX_RVF_PTR represents a pointer to an IEEE1722Tp_RxRvfType

This MetaDataItemType is produced for reception by the IEEE1722Tp module within the internal reception indication processing and consumed by the receiving IEEE1722 data stream consumer.

**[CP_SWS_IEEE1722Tp_CONSTR_00013]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerRxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamRVF shall have the MetaDataItemType IEEE1722TP_RX_RVF_PTR configured.⌋

#### 7.6.1.5 IEEE1722 CRF stream header

The following MetaDataItemType is used for transmission of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamCRF. This MetaDataItemType represents runtime values provided as pointer to the according IEEE1722Tp speficied type:

- IEEE1722TP_TX_CRF_PTR represents a pointer to an IEEE1722Tp_TxCrfType

This MetaDataItemType is produced for transmission by an IEEE1722 data stream provider and consumed by the IEEE1722Tp module within the internal transmission request processing.

**[CP_SWS_IEEE1722Tp_CONSTR_00014]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerTxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamCRF shall have the MetaDataItemType IEEE1722TP_TX_CRF_PTR configured.⌋

The following MetaDataItemType is used for reception of IEEE1722 streams which have IEEE1722TpStreamSubtype set to IEEE1722TpStreamCRF. This MetaDataItemType represents runtime values provided as pointer to the according IEEE1722Tp speficied type:

- IEEE1722TP_RX_CRF_PTR represents a pointer to an IEEE1722Tp_RxCrfType

This MetaDataItemType is produced for reception by the IEEE1722Tp module within the internal reception indication processing and consumed by the receiving IEEE1722 data stream consumer.

**[CP_SWS_IEEE1722Tp_CONSTR_00015]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpUpperLayerRxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamCRF shall have the MetaDataItemType IEEE1722TP_RX_CRF_PTR configured.⌋

#### 7.6.1.6 IEEE1722 ACF stream (NTSCF/TSCF) header

For IEEE1722 streams with IEEE1722TpStreamSubtype set to IEEE1722TpStreamACF no additional MetaDataItemType specified. For handling of encapsulated <Bus>Frames as ACF-messages the following MetaDataItemType are used:

- IEEE1722TP_COMMON_STREAM_HEADER_PTR: used to forward the presentation time of an received ACF-messsage transported via an ACF-stream of type TSCF to an receiving IEEE1722 application
- MESSAGE_TIMESTAMP_64: used to forward the message timestamp from an IEEE1722 application to the IEEE1722Tp module or vice versa.
- MESSAGE_TIMESTAMP_VALID_8: used to forward the validity of a message timestamp from an IEEE1722 application to the IEEE1722Tp module or vice versa.
- CAN_ID_32: used to forward the CAN-ID from CanIf to the IEEE1722Tp module or vice versa

- CAN_ID_PROPS_8: used to forward CAN frame specific information from the CanIf to the IEEE1722Tp module or vice versa
- LIN_NAD_8: used to forward the LIN id of a LIN frame from LinIf to the IEEE1722Tp module or vice versa

**[CP_SWS_IEEE1722Tp_CONSTR_00016]**

*Status:* DRAFT

⌈A PDU which refer to an IEEE1722TpStreamAcfCan that is aggregated by an IEEE1722TpStreamACF with IEEE1722TpStreamDirection set to IEEE1722TpStreamTx shall have no other MetaDataItemType configured than:

- CAN_ID_32
- CAN_ID_PROPS_8

⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00017]**

*Status:* DRAFT

⌈A PDU which refer to an IEEE1722TpStreamAcfLin that is aggregated by an IEEE1722TpStreamACF with IEEE1722TpStreamDirection set to IEEE1722TpStreamTx shall have no other MetaDataItemType configured than:

- LIN_NAD_8

⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00018]**

*Status:* DRAFT

⌈A PDU which refer to an IEEE1722TpStreamAcfCan that is aggregated by an IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS and IEEE1722TpStreamDirection set to IEEE1722TpStreamRx shall have no other MetaDataItemType configured than:

- IEEE1722TP_COMMON_STREAM_HEADER_PTR
- MESSAGE_TIMESTAMP_64
- MESSAGE_TIMESTAMP_VALID_8
- CAN_ID_32
- CAN_ID_PROPS_8

⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00019]**

*Status:* DRAFT

⌈A PDU which refer to an IEEE1722TpStreamAcfCan that is aggregated by an IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to NON_TIME_SYNCHRONOUS and IEEE1722TpStreamDirection set to IEEE1722TpStreamRx shall have no other MetaDataItemType configured than:

- MESSAGE_TIMESTAMP_64
- MESSAGE_TIMESTAMP_VALID_8
- CAN_ID_32
- CAN_ID_PROPS_8

⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00020]**

*Status:* DRAFT

⌈A PDU which refer to an IEEE1722TpStreamAcfLin that is aggregated by an IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to TIME_SYNCHRONOUS and IEEE1722TpStreamDirection set to IEEE1722TpStreamRx shall have no other MetaDataItemType configured than:

- IEEE1722TP_COMMON_STREAM_HEADER_PTR
- MESSAGE_TIMESTAMP_64
- MESSAGE_TIMESTAMP_VALID_8
- LIN_NAD_8

⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00021]**

*Status:* DRAFT

⌈A PDU which refer to an IEEE1722TpStreamAcfLin that is aggregated by an IEEE1722TpStreamACF with IEEE1722TpStreamAcfHeaderType set to NON_TIME_SYNCHRONOUS and IEEE1722TpStreamDirection set to IEEE1722TpStreamRx shall have no other MetaDataItemType configured than:

- MESSAGE_TIMESTAMP_64
- MESSAGE_TIMESTAMP_VALID_8
- LIN_NAD_8

⌋

#### 7.6.1.7 IEEE1722 stream interaction with lower layer

The following MetaDataItemTypes are used for transmission of IEEE1722 streams towards the lower layers.

- ETHERNET_MAC_64
- LISTELEM_PTR represents a pointer to an ListElemStructType

The MetaDataItemType are produced for transmission by the IEEE1722Tp module and consumed by the EthIf within the internal transmission request processing.

**[CP_SWS_IEEE1722Tp_CONSTR_00022]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpLowerLayerTxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to

- IEEE1722TpStreamCRF or
- IEEE1722TpStreamAAF or
- IEEE1722TpStreamIIDC or
- IEEE1722TpStreamRVF

shall have the following MetaDataItemTypes configured:

- ETHERNET_MAC_64
- LISTELEM_PTR

⌋

**[CP_SWS_IEEE1722Tp_CONSTR_00023]**

*Status:* DRAFT

⌈A PDU which is referenced by an IEEE1722TpLowerLayerTxPduPoolEntry that belongs to an IEEE1722TpStream with IEEE1722TpStreamSubtype set to IEEE1722TpStreamACF shall have ETHERNET_MAC_64 configured⌋

The following MetaDataItemType is used for reception of IEEE1722 streams from the lower layers.

- TIMETUPLE_TYPE_PTR represents a pointer to an TimeTupleType

The MetaDataItemType is produced for reception by the EthIf and consumed by the IEEE1722Tp module

**[CP_SWS_IEEE1722Tp_CONSTR_00024]**

*Status:* DRAFT

⌈A PDU which is referenced by an `IEEE1722TpLowerLayerRxPduPoolEntry` that belongs to an `IEEE1722TpStream` independent of `IEEE1722TpStreamSubtype` shall have `MetaDataItemType` `TIMETUPLE_TYPE_PTR` configured.⌋

## 7.7 Error Classification

Chapter [5, General Specification of Basic Software Modules] 7.2 *"Error Handling"* describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

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

### 7.7.1 Development Errors

**[CP_SWS_IEEE1722Tp_91020] Definition of development errors in module IEEE1722Tp**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00385

⌈

| *Type of error* | *Related error code* | *Error value* |
|---|---|---|
| An API was called while the module was uninitialized | IEEE1722TP_E_UNINIT | 0x01 |
| The init API was called twice | IEEE1722TP_E_REINIT | 0x02 |
| IEEE1722Tp_Init was called with an invalid configuration pointer | IEEE1722TP_E_INIT_FAILED | 0x03 |
| An API service was called with a NULL pointer | IEEE1722TP_E_PARAM_POINTER | 0x04 |
| An API service was called with a wrong ID | IEEE1722TP_E_INVALID_PDU_SDU_ID | 0x05 |
| An API service was called with a wrong stream ID | IEEE1722TP_E_INVALID_STREAM_ID | 0x06 |
| Size of IEEE1722-based stream exceed MTU | IEEE1722TP_E_IEEE1722_STREAM_EXCEED_MTU | 0x08 |
| An IEEE1722 stream with a not supported subtye was received | IEEE1722TP_E_IEEE1722_STREAM_NOT_SUPPORTED_SUBTYPE | 0x09 |
| An IEEE1722 stream with a stream version was received, which do not match to the configured stream version | IEEE1722TP_E_IEEE1722_STREAM_VERSION_MISMATCH | 0x0A |
| An IEEE1722 stream with a stream id was received, which do not match to the configured stream id | IEEE1722TP_E_IEEE1722_STREAM_ID_MISMATCH | 0x0B |
| IEEE1722Tp module detect an outdated presentation received via an IEE1722 stream | IEEE1722TP_E_IEEE1722_STREAM_PRESENTATIONTIME_OUTDATED | 0x0C |
| IEEE1722Tp could not add an L-SDU to the L-SDU waiting area due missing space | IEEE1722TP_E_LSDU_WAITING_AREA_OVERRUN | 0x0D |

⌋

### 7.7.2 Runtime Errors

**[CP_SWS_IEEE1722Tp_91021] Definition of runtime errors in module IEEE1722 Tp**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00385

⌈

| *Type of error* | *Related error code* | *Error value* |
|---|---|---|
| A message could not be stored in the queue | IEEE1722TP_E_RX_QUEUE_OVERRUN | 0x40 |
| A message could not be transmitted | IEEE1722TP_E_TRANSMIT_FAILED | 0x41 |
| A PDU is requested to be used while it is already in use or requested to be available while it is already availalbe | IEEE1722TP_E_PDU_STATE_TRANSITION_FAILED | 0x42 |
| A transmission request was rejected, due to missing space in the queue | IEEE1722TP_E_TX_QUEUE_OVERRUN | 0x43 |
| An internal transmission request processing failed | IEEE1722TP_E_TX_INTERNAL_PROCESSING_FAILED | 0x46 |
| An internal reception indication processing failed | IEEE1722TP_E_RX_INTERNAL_PROCESSING_FAILED | 0x47 |
| CAN message dropped by CAN message filter at transmission side | IEEE1722TP_E_CAN_FILTER_DROPPED_TX_CAN_FRAME | 0x48 |
| ACF-message with ACF_CAN subtype dropped | IEEE1722TP_E_DROPPED_RX_CAN_FRAME | 0x49 |
| ACF-message with ACF_LIN subtype dropped | IEEE1722TP_E_DROPPED_RX_LIN_FRAME | 0x4A |
| IEEE1722Tp module detect a discontinuous procedure of the sequence number | IEEE1722TP_E_IEEE1722_STREAM_DISCONTINUOUS_SEQUENCE_NUMBER | 0x4B |

⌋

### 7.7.3 Production Errors

The IEEE1722Tp module does not define production errors.

## 7.8 Security Events

There are no security events.

# 8 API specification

## 8.1 Imported types

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

**[CP_SWS_IEEE1722Tp_91036] Definition of imported datatypes of module IEEE1722Tp** ⌈

| *Module* | *Header File* | *Imported Type* |
|---|---|---|
| Comtype | ComStack_Types.h | PduIdType |
| | ComStack_Types.h | PduInfoType |
| | ComStack_Types.h | PduLengthType |
| | ComStack_Types.h | TimeStampQualType (draft) |
| | ComStack_Types.h | TimeStampType (draft) |
| | ComStack_Types.h | TimeTupleType (draft) |
| StbM | Rte_StbM_Type.h | StbM_SynchronizedTimeBaseType |
| | Rte_StbM_Type.h | StbM_TimeBaseStatusType |
| | Rte_StbM_Type.h | StbM_TimeStampType |
| | Rte_StbM_Type.h | StbM_TimeTupleType |
| | Rte_StbM_Type.h | StbM_UserDataType |
| | StbM.h | StbM_VirtualLocalTimeType |
| Std | Std_Types.h | Std_ReturnType |

⌋

## 8.2 Type definitions

### 8.2.1 IEEE1722Tp_ConfigType

**[CP_SWS_IEEE1722Tp_91001] Definition of datatype IEEE1722Tp_ConfigType**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00404, SRS_BSW_00441

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_ConfigType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Implementation specific | |
| | ***Type*** | – |
| | ***Comment*** | – |

▽

△

| | |
|---|---|
| ***Description*** | This is the base type for the configuration of the IEEE1722Tp module. A pointer to an instance of this structure will be used in the initialization of the IEEE1722Tp module. The content of this structure is defined in chapter 10 Configuration specification.<br>**Tags:** atp.Status=draft |
| ***Available via*** | IEEE1722Tp.h |

⌋

### 8.2.2 IEEE1722Tp_StreamIndexType

**[CP_SWS_IEEE1722Tp_91002] Definition of datatype IEEE1722Tp_StreamIndex Type**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00441

⌈

| | | | |
|---|---|---|---|
| ***Name*** | IEEE1722Tp_StreamIndexType (draft) | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint16 | | |
| ***Range*** | 0...65535 | – | Zero-based integer number, which represents a unique stream index to address a configured stream in context of the IEEE1722Tp module. |
| ***Description*** | This type is used to address configured streams in context of the IEEE1722Tp module<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | IEEE1722Tp.h | | |

⌋

### 8.2.3 IEEE1722Tp_StreamStateType

**[CP_SWS_IEEE1722Tp_91003] Definition of datatype IEEE1722Tp_StreamState Type**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00441

⌈

| | | | |
|---|---|---|---|
| ***Name*** | IEEE1722Tp_StreamStateType (draft) | | |
| ***Kind*** | Enumeration | | |
| ***Range*** | IEEE1722TP_STREAM_ ACTIVATED | 0x00 | Indicates that a configured stream is activated. Thus, communication via this stream is enabled |

▽

△

| | | | |
|---|---|---|---|
| | IEEE1722TP_STREAM_DEACTIVATED | 0x01 | Indicates that a configured stream is de-activated. Thus, communication via this stream is disabled |
| ***Description*** | Indicates the state of IEEE1722-based stream<br>**Tags:** atp.Status=draft | | |
| ***Available via*** | IEEE1722Tp.h | | |

⌋

### 8.2.4 IEEE1722Tp_CommonStreamHeaderType

**[CP_SWS_IEEE1722Tp_91004] Definition of datatype IEEE1722Tp_CommonStreamHeaderType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_CommonStreamHeaderType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | mr | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 defined mr (media clock restart) 1 bit header field. |
| | tv | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 defined tv (avtp_timestamp valid) 1 bit header field. |
| | tu | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 defined tu (timestamp uncertain) 1 bit header field. |
| | mac_address | |
| | ***Type*** | uint64 |
| | ***Comment*** | Represents the MAC address part (48 bit) of the IEEE1722 specified stream id. |
| | unique_id | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the unique id part (16 bit) of the IEEE1722 specified stream id. |
| | avtp_timestamp | |
| | ***Type*** | uint64 |
| | ***Comment*** | Represents the IEEE1722 specified presentation time (32 bit) for an IEEE1722 stream. |
| | avtp_timestamp_provided | |
| | ***Type*** | uint8 |
| | ***Comment*** | Indicate if the avtp_timestamp was produced (0x00 not provided; 0x01 provided). |

▽

△

| | |
|---|---|
| ***Description*** | Represents the runtime values for IEEE1722 defined common stream header fields. Used for specific IEEE1722 streams (e.g. 61883_iidc, AAF, RVF, TSCF)<br>**Tags:** atp.Status=draft |
| ***Available via*** | IEEE1722Tp.h |

⌋

**[CP_SWS_IEEE1722Tp_00108] Value range definition for IEEE1722Tp_CommonStreamHeaderType.mr**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_CommonStreamHeaderType.mr shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00109] Value range definition for IEEE1722Tp_CommonStreamHeaderType.tv**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_CommonStreamHeaderType.tv shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00110] Value range definition for IEEE1722Tp_CommonStreamHeaderType.tu**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_CommonStreamHeaderType.tu shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00111] Value range definition for IEEE1722Tp_CommonStreamHeaderType.mac_address**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_CommonStreamHeaderType.mac_address shall be:

- `0x00 00 00 00 00 00 00 00` ... `0x00 00 FF FF FF FF FF FF` : valid
- `0x00 01 00 00 00 00 00 00` ... `0xFF FF FF FF FF FF FF FE` : not used
- `0xFF FF FF FF FF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00112] Value range definition for IEEE1722Tp_CommonStreamHeaderType.unique_id**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_CommonStreamHeaderType.unique_id shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00113] Value range definition for IEEE1722Tp_CommonStreamHeaderType.avtp_timestamp**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_CommonStreamHeaderType.avtp_timestamp shall be:

- `0x00 00 00 00 00 00 00 00` ... `0x00 00 00 00 FF FF FF FF` : valid
- `0x00 00 00 01 00 00 00 00` ... `0xFF FF FF FF FF FF FF FF` : not used

⌋

**[CP_SWS_IEEE1722Tp_00114] Value range definition for `IEEE1722Tp_CommonStreamHeaderType.avtp_timestamp_provided`**

*Status:* DRAFT

⌈The value range for `IEEE1722Tp_CommonStreamHeaderType.avtp_timestamp_provided` shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used

- 0xFF : value not provided

⌋

### 8.2.5 IEEE1722Tp_TxIec61883IidcType

**[CP_SWS_IEEE1722Tp_91005] Definition of datatype IEEE1722Tp_TxIec61883IidcType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxIec61883IidcType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883_iidc defined sy 4 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined IEC 61883_iidc stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00115] Value range definition for IEEE1722Tp_TxIec61883IidcType.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883IidcType.sy shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

### 8.2.6 IEEE1722Tp_RxIec61883IidcType

**[CP_SWS_IEEE1722Tp_91006] Definition of datatype IEEE1722Tp_RxIec61883IidcType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxIec61883IidcType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | tag | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883_iidc defined tag 2 bit header field. |
| | channel | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883_iidc defined channel 6 bit header field. |
| | tcode | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883_iidc defined tcode (type code) 4 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883_iidc defined sy 4 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined IEC 61883_iidc stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00116] Value range definition for IEEE1722Tp_RxIec61883IidcType.tag**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883IidcType.tag shall be:

- `0x00` ... `0x03` : valid
- `0x04` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00117] Value range definition for IEEE1722Tp_RxIec61883IidcType.channel**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883IidcType.channel shall be:

- `0x00` ... `0x3F` : valid
- `0x40` ... `0xFE` : not used

- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00118] Value range definition for IEEE1722Tp_RxIec61883IidcType.tcode**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883IidcType.tcode shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00119] Value range definition for IEEE1722Tp_RxIec61883IidcType.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883IidcType.sy shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

### 8.2.7 IEEE1722Tp_TxIec61883Type

**[CP_SWS_IEEE1722Tp_91007] Definition of datatype IEEE1722Tp_TxIec61883Type**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxIec61883Type (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | dbc | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBC (data block count) 8 bit header field. |
| | qpc | |
| | ***Type*** | uint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 IEC 61883 QPC (quadlet padding count) 3 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined sy 4 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined IEC 61883 stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00120] Value range definition for IEEE1722Tp_TxIec61883Type.dbc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883Type.dbc shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00121] Value range definition for IEEE1722Tp_TxIec61883Type.qpc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883Type.qpc shall be:

- `0x00` ... `0x07` : valid
- `0x08` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00122] Value range definition for IEEE1722Tp_TxIec61883Type.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883Type.sy shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

### 8.2.8 IEEE1722Tp_RxIec61883Type

**[CP_SWS_IEEE1722Tp_91008] Definition of datatype IEEE1722Tp_RxIec61883Type**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxIec61883Type (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | tag | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined tag 2 bit header field. |
| | channel | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined channel 6 bit header field. |
| | tcode | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined tcode (type code) 4 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined sy 4 bit header field. |
| | qi_1 | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 qi_1 (quadlet indicator) 2 bit header field. |
| | sid | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SID (source identifier) 6 bit header field. |
| | dbs | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBS (data block size) 8 bit header field. |
| | fn | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 FN (fraction number) 2 bit header field. |
| | qpc | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined QPC (quadlet padding count) 3 bit header field. |
| | sph | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SPH (source packet header) 1 bit header field. |

▽

△

| | | |
|---|---|---|
| | dpc | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBC (data block count) 8 bit header field. |
| | qi_2 | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 qi_2 (quadlet indicator) 2 bit header field. |
| | fmt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined FMT (stream format) 6 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined IEC 61883 stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00123] Value range definition for IEEE1722Tp_RxIec61883Type.tag**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.tag shall be:

- `0x00` ... `0x03` : valid
- `0x04` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00124] Value range definition for IEEE1722Tp_RxIec61883Type.channel**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.channel shall be:

- `0x00` ... `0x3F` : valid
- `0x40` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00125] Value range definition for IEEE1722Tp_RxIec61883Type.tcode**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.tcode shall be:

- `0x00` ... `0x0F` : valid

- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00126] Value range definition for IEEE1722Tp_RxIec61883Type.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.sy shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00127] Value range definition for IEEE1722Tp_RxIec61883Type.qi_1**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.qi_1 shall be:

- 0x00 ... 0x03 : valid
- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00128] Value range definition for IEEE1722Tp_RxIec61883Type.sid**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.sid shall be:

- 0x00 ... 0x3F : valid
- 0x40 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00129] Value range definition for IEEE1722Tp_RxIec61883Type.dbs**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.dbs shall be:

- 0x00 00 ... 0x00 FF : valid

- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00130] Value range definition for IEEE1722Tp_RxIec61883Type.fn**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.fn shall be:

- 0x00 ... 0x03 : valid
- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00131] Value range definition for IEEE1722Tp_RxIec61883Type.qpc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.qpc shall be:

- 0x00 ... 0x07 : valid
- 0x08 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00132] Value range definition for IEEE1722Tp_RxIec61883Type.sph**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.sph shall be:

- 0x00 ... 0x01 : valid
- 0x02 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00133] Value range definition for IEEE1722Tp_RxIec61883Type.dbc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.dbc shall be:

- 0x00 00 ... 0x00 FF : valid

- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00134] Value range definition for IEEE1722Tp_RxIec61883Type.qi_2**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.qi_2 shall be:

- 0x00 ... 0x03 : valid
- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00135] Value range definition for IEEE1722Tp_RxIec61883Type.fmt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883Type.fmt shall be:

- 0x00 ... 0x3F : valid
- 0x40 ... 0xFE : not used
- 0xFF : value not provided

⌋

### 8.2.9 IEEE1722Tp_TxIec61883CipNoSphType

**[CP_SWS_IEEE1722Tp_91009] Definition of datatype IEEE1722Tp_TxIec61883 CipNoSphType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxIec61883CipNoSphType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | dbc | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBC (data block count) 8 bit header field. |
| | qpc | |
| | ***Type*** | uint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 IEC 61883 QPC (quadlet padding count) 3 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined sy 4 bit header field. |
| | fdf | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 FDF (format dependent field) 8 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined IEC 61883 stream header fields where SPH is set to 0.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00136] Value range definition for IEEE1722Tp_TxIec61883CipNoSphType.dbc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipNoSphType.dbc shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00137] Value range definition for IEEE1722Tp_TxIec61883CipNoSphType.qpc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipNoSphType.qpc shall be:

- `0x00` ... `0x07` : valid
- `0x08` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00138] Value range definition for IEEE1722Tp_TxIec61883CipNoSphType.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipNoSphType.sy shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used

- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00139] Value range definition for IEEE1722Tp_TxIec61883CipNoSphType.fdf**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipNoSphType.fdf shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

### 8.2.10 IEEE1722Tp_RxIec61883CipNoSphType

**[CP_SWS_IEEE1722Tp_91010] Definition of datatype IEEE1722Tp_RxIec61883 CipNoSphType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxIec61883CipNoSphType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | tag | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined tag 2 bit header field. |
| | channel | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined channel 6 bit header field. |
| | tcode | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined tcode (type code) 4 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined sy 4 bit header field. |
| | qi_1 | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 qi_1 (quadlet indicator) 2 bit header field. |
| | sid | |
| | ***Type*** | uint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SID (source identifier) 6 bit header field. |
| | dbs | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBS (data block size) 8 bit header field. |
| | fn | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 FN (fraction number) 2 bit header field. |
| | qpc | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined QPC (quadlet padding count) 3 bit header field. |
| | sph | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SPH (source packet header) 1 bit header field. |
| | dbc | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBC (data block count) 8 bit header field. |
| | qi_2 | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 qi_2 (quadlet indicator) 2 bit header field. |
| | fmt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined FMT (stream format) 6 bit header field. |
| | fdf | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined FDF (format dependent field) 8 bit header field. |
| | syt | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SYT (synchronization timing) 16 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined IEC 61883 stream header fields where SPH is set to 0.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00140] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.tag**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.tag shall be:

- `0x00` ... `0x03` : valid

- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00141] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.channel**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.channel shall be:

- 0x00 ... 0x3F : valid
- 0x40 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00142] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.tcode**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.tcode shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00143] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.sy shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00144] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.qi_1**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.qi_1 shall be:

- 0x00 ... 0x03 : valid

- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00145] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.sid**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.sid shall be:

- 0x00 ... 0x3F : valid
- 0x40 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00146] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.dbs**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.dbs shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00147] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.fn**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.fn shall be:

- 0x00 ... 0x03 : valid
- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00148] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.qpc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.qpc shall be:

- 0x00 ... 0x07 : valid

- 0x08 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00149] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.sph**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.sph shall be:

- 0x00 ... 0x01 : valid
- 0x02 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00150] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.dbc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.dbc shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00151] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.qi_2**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.qi_2 shall be:

- 0x00 ... 0x03 : valid
- 0x04 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00152] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.fmt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.fmt shall be:

- 0x00 ... 0x3F : valid

- 0x40 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00153] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.fdf**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.fdf shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00154] Value range definition for IEEE1722Tp_RxIec61883CipNoSphType.syt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipNoSphType.syt shall be:

- 0x00 00 00 00 ... 0x00 00 FF FF : valid
- 0x00 00 01 00 ... 0xFF FF FF FE : not used
- 0xFF FF FF FF : value not provided

⌋

### 8.2.11 IEEE1722Tp_TxIec61883CipWithSphType

**[CP_SWS_IEEE1722Tp_91011] Definition of datatype IEEE1722Tp_TxIec61883CipWithSphType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxIec61883CipWithSphType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | dbc | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBC (data block count) 8 bit header field. |
| | qpc | |
| | ***Type*** | uint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 IEC 61883 QPC (quadlet padding count) 3 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined sy 4 bit header field. |
| | fdf | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 FDF (format dependent field) 24 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined IEC 61883 stream header fields where SPH is set to 1.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00155] Value range definition for IEEE1722Tp_TxIec61883CipWithSphType.dbc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipWithSphType.dbc shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00156] Value range definition for IEEE1722Tp_TxIec61883CipWithSphType.qpc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipWithSphType.qpc shall be:

- `0x00` ... `0x07` : valid
- `0x08` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00157] Value range definition for IEEE1722Tp_TxIec61883CipWithSphType.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipWithSphType.sy shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used

- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00158] Value range definition for IEEE1722Tp_TxIec61883CipWithSphType.fdf**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxIec61883CipWithSphType.fdf shall be:

- `0x00 00 00 00` ... `0x00 FF FF FF` : valid
- `0x01 00 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

### 8.2.12 IEEE1722Tp_RxIec61883CipWithSphType

**[CP_SWS_IEEE1722Tp_91012] Definition of datatype IEEE1722Tp_RxIec61883 CipWithSphType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxIec61883CipWithSphType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | tag | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined tag 2 bit header field. |
| | channel | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined channel 6 bit header field. |
| | tcode | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined tcode (type code) 4 bit header field. |
| | sy | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined sy 4 bit header field. |
| | qi_1 | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 qi_1 (quadlet indicator) 2 bit header field. |
| | sid | |
| | ***Type*** | uint8 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SID (source identifier) 6 bit header field. |
| | dbs | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBS (data block size) 8 bit header field. |
| | fn | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 FN (fraction number) 2 bit header field. |
| | qpc | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined QPC (quadlet padding count) 3 bit header field. |
| | sph | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined SPH (source packet header) 1 bit header field. |
| | dbc | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 DBC (data block count) 8 bit header field. |
| | qi_2 | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 qi_2 (quadlet indicator) 2 bit header field. |
| | fmt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 defined FMT (stream format) 6 bit header field. |
| | fdf | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 IEC 61883 FDF (format dependent field) 24 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined IEC 61883 stream header fields where SPH is set to 1.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00159] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.tag**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.tag shall be:

- `0x00` ... `0x03` : valid
- `0x04` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00160] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.channel**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.channel shall be:

- `0x00` ... `0x3F` : valid
- `0x40` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00161] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.tcode**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.tcode shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00162] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.sy**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.sy shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00163] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.qi_1**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.qi_1 shall be:

- `0x00` ... `0x03` : valid
- `0x04` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00164] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.sid**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.sid shall be:

- `0x00` ... `0x3F` : valid
- `0x40` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00165] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.dbs**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.dbs shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00166] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.fn**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.fn shall be:

- `0x00` ... `0x03` : valid
- `0x04` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00167] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.qpc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.qpc shall be:

- `0x00` ... `0x07` : valid
- `0x08` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00168] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.sph**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.sph shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00169] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.dbc**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.dbc shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00170] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.qi_2**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.qi_2 shall be:

- `0x00` ... `0x03` : valid
- `0x04` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00171] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.fmt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.fmt shall be:

- `0x00` ... `0x3F` : valid
- `0x40` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00172] Value range definition for IEEE1722Tp_RxIec61883CipWithSphType.fdf**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxIec61883CipWithSphType.fdf shall be:

- `0x00 00 00 00` ... `0x00 FF FF FF` : valid
- `0x01 00 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

### 8.2.13 IEEE1722Tp_TxAafPcmType

**[CP_SWS_IEEE1722Tp_91013] Definition of datatype IEEE1722Tp_TxAafPcmType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxAafPcmType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | evt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF evt 4 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined AAF stream header fields with PCM encapsulation.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00173] Value range definition for IEEE1722Tp_TxAafPcmType.evt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxAafPcmType.evt shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

### 8.2.14 IEEE1722Tp_RxAafPcmType

**[CP_SWS_IEEE1722Tp_91014] Definition of datatype IEEE1722Tp_RxAafPcmType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxAafPcmType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | format | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined format 8 bit header field. |
| | sp | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined sp (sparse timestamp) 1 bit header field. |
| | evt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined evt 4 bit header field. |
| | nsr | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined nfr (nominal sample rate) 4 bit header field. |
| | channels_per_frame | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined channels_per_frame 10 bit header field. |
| | bit_depth | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined bit_depth 8 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined AAF stream header fields with PCM encapsulation.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00174] Value range definition for IEEE1722Tp_RxAafPcmType.format**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafPcmType.format shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00175] Value range definition for IEEE1722Tp_RxAafPcmType.sp**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafPcmType.sp shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00176] Value range definition for IEEE1722Tp_RxAafPcmType.evt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafPcmType.evt shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00177] Value range definition for IEEE1722Tp_RxAafPcmType.nsr**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafPcmType.nsr shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00178] Value range definition for IEEE1722Tp_RxAafPcmType.channels_per_frame**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafPcmType.channels_per_frame shall be:

- `0x00 00` ... `0x03 FF` : valid
- `0x04 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00179] Value range definition for IEEE1722Tp_RxAafPcmType.bit_depth**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafPcmType.bit_depth shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

### 8.2.15 IEEE1722Tp_TxAafAes3Type

**[CP_SWS_IEEE1722Tp_91015] Definition of datatype IEEE1722Tp_TxAafAes3Type**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxAafAes3Type (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | evt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF evt 4 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined AAF stream header fields with AES3 encapsulation.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00180] Value range definition for IEEE1722Tp_TxAafAes3Type.evt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxAafAes3Type.evt shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

### 8.2.16 IEEE1722Tp_RxAafAes3Type

**[CP_SWS_IEEE1722Tp_91016] Definition of datatype IEEE1722Tp_RxAafAes3Type**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxAafAes3Type (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | format | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined format 8 bit header field. |
| | sp | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined sp (sparse timestamp) 1 bit header field. |
| | evt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined evt 4 bit header field. |
| | nfr | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined nfr (nominal frame rate) 4 bit header field. |
| | streams_per_frame | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined streams_per_frame 10 bit header field. |
| | aes3_data_type_h | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined aes3_data_type_h 8 bit header field. |
| | aes3_dt_ref | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 AAF defined DBS (data block size) 3 bit header field. |
| | aes3_data_type_l | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 AAF defined aes3_data_type_l 8 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined AAF stream header fields with AES3 encapsulation.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00181] Value range definition for IEEE1722Tp_RxAafAes3Type.format**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.format shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00182] Value range definition for IEEE1722Tp_RxAafAes3Type.sp**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.sp shall be:

- 0x00 ... 0x01 : valid
- 0x02 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00183] Value range definition for IEEE1722Tp_RxAafAes3Type.evt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.evt shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00184] Value range definition for IEEE1722Tp_RxAafAes3Type.nfr**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.nfr shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00185] Value range definition for IEEE1722Tp_RxAafAes3Type.streams_per_frame**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.streams_per_frame shall be:

- 0x00 00 ... 0x03 FF : valid
- 0x04 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00186] Value range definition for IEEE1722Tp_RxAafAes3Type.aes3_data_type_h**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.aes3_data_type_h shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00187] Value range definition for IEEE1722Tp_RxAafAes3Type.aes3_dt_ref**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.aes3_dt_ref shall be:

- 0x00 ... 0x07 : valid
- 0x08 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00188] Value range definition for IEEE1722Tp_RxAafAes3Type.aes3_data_type_l**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxAafAes3Type.aes3_data_type_l shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

### 8.2.17 IEEE1722Tp_TxRvfType

**[CP_SWS_IEEE1722Tp_91017] Definition of datatype IEEE1722Tp_TxRvfType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxRvfType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | ap | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined ap (active pixels) 1 bit header field. |
| | f | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined f (field) 1 bit header field. |
| | ef | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined ef (end frame) 1 bit header field. |
| | evt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined evt 4 bit header field. |
| | pd | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined pd (pull-down) 1 bit header field. |
| | num_lines | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined num_lines 4 bit header field. |
| | i_seq_num | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 RVF defined i_seq_num 8 bit header field. |
| | line_number | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 RVF defined line_number 16 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined RVF stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00189] Value range definition for IEEE1722Tp_TxRvfType.ap**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.ap shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used

- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00190] Value range definition for IEEE1722Tp_TxRvfType.f**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.f shall be:

- 0x00 ... 0x01 : valid
- 0x02 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00191] Value range definition for IEEE1722Tp_TxRvfType.ef**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.ef shall be:

- 0x00 ... 0x01 : valid
- 0x02 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00192] Value range definition for IEEE1722Tp_TxRvfType.evt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.evt shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00193] Value range definition for IEEE1722Tp_TxRvfType.pd**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.pd shall be:

- 0x00 ... 0x01 : valid
- 0x02 ... 0xFE : not used

- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00194] Value range definition for IEEE1722Tp_TxRvfType.num_lines**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.num_lines shall be:

- 0x00 ... 0x0F : valid
- 0x10 ... 0xFE : not used
- 0xFF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00195] Value range definition for IEEE1722Tp_TxRvfType.i_seq_num**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.i_seq_num shall be:

- 0x00 00 ... 0x00 FF : valid
- 0x01 00 ... 0xFF FE : not used
- 0xFF FF : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00196] Value range definition for IEEE1722Tp_TxRvfType.line_number**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxRvfType.line_number shall be:

- 0x00 00 00 00 ... 0x00 00 FF FF : valid
- 0x00 01 00 00 ... 0xFF FF FF FE : not used
- 0xFF FF FF FF : value not provided

⌋

### 8.2.18 IEEE1722Tp_RxRvfType

**[CP_SWS_IEEE1722Tp_91018] Definition of datatype IEEE1722Tp_RxRvfType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxRvfType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | active_pixels | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 RVF defined active_pixels 16 bit header field. |
| | total_lines | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 RVF defined total_lines 16 bit header field. 0xFF FF FF FF Value not set |
| | ap | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined ap (active pixels) 1 bit header field. |
| | f | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined f (field) 1 bit header field. |
| | ef | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined ef (end frame) 1 bit header field. |
| | evt | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined evt 4 bit header field. |
| | pd | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined pd (pull-down) 1 bit header field. |
| | i | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined i (interlaced) 1 bit header field. |
| | pixel_depth | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined pixel_depth 4 bit header field. |
| | pixel_format | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined pixel_format 4 bit header field. |
| | frame_rate | |
| | ***Type*** | uint16 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 RVF defined frame_rate 8 bit header field. |
| | colorspace | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined colorspace 4 bit header field. |
| | num_lines | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 RVF defined num_lines 4 bit header field. |
| | i_seq_num | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 RVF defined i_seq_num 8 bit header field. |
| | line_number | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 RVF defined line_number 16 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined RVF stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00197] Value range definition for IEEE1722Tp_RxRvfType.active_pixels**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.active_pixels shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00198] Value range definition for IEEE1722Tp_RxRvfType.total_lines**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.total_lines shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00199] Value range definition for IEEE1722Tp_RxRvfType.ap**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.ap shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00200] Value range definition for IEEE1722Tp_RxRvfType.f**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.f shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00201] Value range definition for IEEE1722Tp_RxRvfType.ef**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.ef shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00202] Value range definition for IEEE1722Tp_RxRvfType.evt**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.evt shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00203] Value range definition for IEEE1722Tp_RxRvfType.pd**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.pd shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00204] Value range definition for IEEE1722Tp_RxRvfType.i**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.i shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00205] Value range definition for IEEE1722Tp_RxRvfType.pixel_depth**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.pixel_depth shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00206] Value range definition for IEEE1722Tp_RxRvfType.pixel_format**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.pixel_format shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00207] Value range definition for IEEE1722Tp_RxRvfType.frame_rate**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.frame_rate shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x10 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00208] Value range definition for IEEE1722Tp_RxRvfType.colorspace**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.colorspace shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00209] Value range definition for IEEE1722Tp_RxRvfType.num_lines**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.num_lines shall be:

- `0x00` ... `0x0F` : valid
- `0x10` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00210] Value range definition for IEEE1722Tp_RxRvfType.i_seq_num**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.i_seq_num shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x10 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00211] Value range definition for IEEE1722Tp_RxRvfType.line_number**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxRvfType.line_number shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

### 8.2.19 IEEE1722Tp_TxCrfType

**[CP_SWS_IEEE1722Tp_91019] Definition of datatype IEEE1722Tp_TxCrfType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_TxCrfType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | mr | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF mr (media clock restart) 1 bit header field. |
| | tu | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF tu (timestamp uncertain) 1 bit header field. |
| | mac_address | |
| | ***Type*** | uint64 |
| | ***Comment*** | Represents the MAC address part (48 bit) of the IEEE1722 specified stream id. |
| | unique_id | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the unique id part (16 bit) of the IEEE1722 specified stream id. |
| | fs | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF fs (frame sync) 1 bit header field. |
| ***Description*** | Represents the Tx runtime values for IEEE1722 defined CRF stream header fields.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00212] Value range definition for IEEE1722Tp_TxCrfType.mr**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxCrfType.mr shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00213] Value range definition for IEEE1722Tp_TxCrfType.tu**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxCrfType.tu shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00214] Value range definition for IEEE1722Tp_TxCrfType.mac_address**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxCrfType.mac_address shall be:

- `0x00 00 00 00 00 00 00 00` ... `0x00 00 FF FF FF FF FF FF` : valid
- `0x00 01 00 00 00 00 00 00` ... `0xFF FF FF FF FF FF FF FE` : not used
- `0xFF FF FF FF FF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00215] Value range definition for IEEE1722Tp_TxCrfType.unique_id**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxCrfType.unique_id shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00216] Value range definition for IEEE1722Tp_TxCrfType.fs**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_TxCrfType.fs shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

### 8.2.20 IEEE1722Tp_RxCrfType

**[CP_SWS_IEEE1722Tp_91033] Definition of datatype IEEE1722Tp_RxCrfType ⌈**

| | | |
|---|---|---|
| ***Name*** | IEEE1722Tp_RxCrfType | |
| ***Kind*** | Structure | |
| ***Elements*** | mr | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF mr (media clock restart) 1 bit header field. |
| | tu | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF tu (timestamp uncertain) 1 bit header field. |
| | mac_address | |
| | ***Type*** | uint64 |
| | ***Comment*** | Represents the MAC address part (48 bit) of the IEEE1722 specified stream id. |
| | unique_id | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the unique id part (16 bit) of the IEEE1722 specified stream id. |
| | fs | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF fs (frame sync) 1 bit header field. |
| | type | |
| | ***Type*** | uint16 |
| | ***Comment*** | Represents the IEEE1722 CRF type 8 bit header field. |
| | pull | |
| | ***Type*** | uint8 |
| | ***Comment*** | Represents the IEEE1722 CRF pull 3 bit header field. |
| | base_frequency | |
| | ***Type*** | uint32 |

▽

△

| | | |
|---|---|---|
| | ***Comment*** | Represents the IEEE1722 CRF defined base_frequency 29 bit header field. |
| | timestamp_interval | |
| | ***Type*** | uint32 |
| | ***Comment*** | Represents the IEEE1722 CRF defined timestamp_interval 16 bit header field. |
| ***Description*** | Represents the Rx runtime values for IEEE1722 defined CRF stream header fields. | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

**[CP_SWS_IEEE1722Tp_00217] Value range definition for IEEE1722Tp_RxCrfType.mr**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.mr shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00218] Value range definition for IEEE1722Tp_RxCrfType.tu**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.tu shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00219] Value range definition for IEEE1722Tp_RxCrfType.mac_address**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.mac_address shall be:

- `0x00 00 00 00 00 00 00 00` ... `0x00 00 FF FF FF FF FF FF` : valid
- `0x00 01 00 00 00 00 00 00` ... `0xFF FF FF FF FF FF FF FE` : not used
- `0xFF FF FF FF FF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00220] Value range definition for IEEE1722Tp_RxCrfType.unique_id**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.unique_id shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00221] Value range definition for IEEE1722Tp_RxCrfType.fs**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.fs shall be:

- `0x00` ... `0x01` : valid
- `0x02` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00222] Value range definition for IEEE1722Tp_RxCrfType.type**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.type shall be:

- `0x00 00` ... `0x00 FF` : valid
- `0x01 00` ... `0xFF FE` : not used
- `0xFF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00223] Value range definition for IEEE1722Tp_RxCrfType.pull**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.pull shall be:

- `0x00` ... `0x07` : valid
- `0x08` ... `0xFE` : not used
- `0xFF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00224] Value range definition for IEEE1722Tp_RxCrfType.base_frequency**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.base_frequency shall be:

- `0x00 00 00 00` ... `0x1F FF FF FF` : valid
- `0x20 00 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

**[CP_SWS_IEEE1722Tp_00225] Value range definition for IEEE1722Tp_RxCrfType.timestamp_interval**

*Status:* DRAFT

⌈The value range for IEEE1722Tp_RxCrfType.timestamp_interval shall be:

- `0x00 00 00 00` ... `0x00 00 FF FF` : valid
- `0x00 01 00 00` ... `0xFF FF FF FE` : not used
- `0xFF FF FF FF` : value not provided

⌋

## 8.3 Function definitions

This chapter contains all APIs provided by the module

### 8.3.1 Generic Functions

#### 8.3.1.1 IEEE1722Tp_Init

**[CP_SWS_IEEE1722Tp_91022] Definition of API function IEEE1722Tp_Init**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00101, SRS_BSW_00358, SRS_BSW_00414, SRS_BSW_00310

⌈

| ***Service Name*** | IEEE1722Tp_Init (draft) |
|---|---|
| ***Syntax*** | `void IEEE1722Tp_Init (`<br>`  const IEEE1722Tp_ConfigType* ConfigPtr`<br>`)` |
| ***Service ID [hex]*** | 0x01 |

▽

△

| | | |
|---|---|---|
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | ConfigPtr | Pointer to post build configuration |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This function initializes the IEEE1722Tp module. In configurations, in which IEEE1722Tp is assigned to more than one partition (i.e. IEEE1722Tp_Main Functions are mapped to partitions), IEEE1722Tp may provide one init function per partition.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

### 8.3.1.2 IEEE1722Tp_DeInit

**[CP_SWS_IEEE1722Tp_91023] Definition of API function IEEE1722Tp_DeInit**

*Status:* DRAFT

*Upstream requirements:* SRS_BSW_00101, SRS_BSW_00310

⌈

| | |
|---|---|
| ***Service Name*** | IEEE1722Tp_DeInit (draft) |
| ***Syntax*** | `void IEEE1722Tp_DeInit (`<br>`  void`<br>`)` |
| ***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 resets the IEEE1722Tp module to the uninitialized state.<br>**Tags:** atp.Status=draft |
| ***Available via*** | IEEE1722Tp.h |

⌋

#### 8.3.1.3 IEEE1722Tp_Transmit

### [CP_SWS_IEEE1722Tp_91025] Definition of API function IEEE1722Tp_Transmit

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | IEEE1722Tp_Transmit (draft) | |
| ***Syntax*** | `Std_ReturnType IEEE1722Tp_Transmit ( PduIdType TxPduId, const PduInfoType* PduInfoPtr )` | |
| ***Service ID [hex]*** | 0x04 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in)*** | TxPduId | Identifier of the PDU to be transmitted |
| | PduInfoPtr | Length of and pointer to the PDU data and pointer to MetaData. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | `E_OK`: Transmit request has been accepted.<br>`E_NOT_OK`:Transmit request has been rejected. |
| ***Description*** | Requests transmission of a PDU.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

#### 8.3.1.4 IEEE1722Tp_TxConfirmation

### [CP_SWS_IEEE1722Tp_91029] Definition of callback function IEEE1722Tp_Tx Confirmation

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | IEEE1722Tp_TxConfirmation (draft) | |
| ***Syntax*** | `void IEEE1722Tp_TxConfirmation ( PduIdType TxPduId, Std_ReturnType result )` | |
| ***Service ID [hex]*** | 0x08 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in)*** | TxPduId | ID of the PDU that has been transmitted. |
| | result | `E_OK`: The PDU was transmitted.<br>`E_NOT_OK`:Transmission of the PDU failed. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |

▽

△

| | |
|---|---|
| ***Description*** | The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.<br>**Tags:** atp.Status=draft |
| ***Available via*** | IEEE1722Tp.h |

⌋

### 8.3.1.5 IEEE1722Tp_RxIndication

**[CP_SWS_IEEE1722Tp_91030] Definition of callback function IEEE1722Tp_RxIndication**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | IEEE1722Tp_RxIndication (draft) | |
| ***Syntax*** | `void IEEE1722Tp_RxIndication (`<br>`  PduIdType RxPduId,`<br>`  const PduInfoType* PduInfoPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x09 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in)*** | RxPduId | ID of the PDU which is used for reception. |
| | PduInfoPtr | Length of and pointer to the PDU data and pointer to MetaData. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | The lower layer communication interface module indicates the reception of a PDU.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

#### 8.3.1.6 IEEE1722Tp_ReleaseRxBuffer

### [CP_SWS_IEEE1722Tp_91026] Definition of API function IEEE1722Tp_ReleaseRxBuffer

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | IEEE1722Tp_ReleaseRxBuffer (draft) | |
| ***Syntax*** | `void IEEE1722Tp_ReleaseRxBuffer ( PduIdType RxPduId )` | |
| ***Service ID [hex]*** | 0x05 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different PduIds. Non reentrant for the same PduId | |
| ***Parameters (in)*** | RxPduId | Identifier of the received PDU. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Indication from the upper layer to release the lower layer reception buffer.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

#### 8.3.1.7 IEEE1722Tp_ActivateStream

### [CP_SWS_IEEE1722Tp_91027] Definition of API function IEEE1722Tp_ActivateStream

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | IEEE1722Tp_ActivateStream (draft) | |
| ***Syntax*** | `Std_ReturnType IEEE1722Tp_ActivateStream ( IEEE1722Tp_StreamIndexType StreamIndex )` | |
| ***Service ID [hex]*** | 0x06 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different stream index values. Non reentrant for the same stream index value. | |
| ***Parameters (in)*** | StreamIndex | Identifier of the stream to be activated |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | `E_OK`: Request to activate a stream has been accepted.<br>`E_NOT_OK`:Request to activate a stream has been rejected. |
| ***Description*** | Request to activate a stream of the given stream index.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

#### 8.3.1.8 IEEE1722Tp_DeactivateStream

**[CP_SWS_IEEE1722Tp_91028] Definition of API function IEEE1722Tp_DeactivateStream**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | IEEE1722Tp_DeactivateStream (draft) | |
| ***Syntax*** | `Std_ReturnType IEEE1722Tp_DeactivateStream ( IEEE1722Tp_StreamIndexType StreamIndex )` | |
| ***Service ID [hex]*** | 0x07 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different stream index values. Non reentrant for the same stream index value. | |
| ***Parameters (in)*** | StreamIndex | Identifier of the stream to be activated |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | `E_OK`: Request to de-activate a stream has been accepted.<br>`E_NOT_OK`: Request to de-activate a stream has been rejected. |
| ***Description*** | Request to de-activate a stream of the given stream index.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | IEEE1722Tp.h | |

⌋

### 8.3.2 API Parameter Checking

The IEEE1722Tp module reports the development error `IEEE1722TP_E_PARAM_POINTER` when a `NULL_PTR` is not accepted as an argument to a service or callback function. The exact behavior is specified in [SWS_BSW_00050] and [SWS_BSW_00212].

**[CP_SWS_IEEE1722Tp_00226]**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00386

⌈If development error detection is enabled by `IEEE1722TpDevErrorDetect`, the IEEE1722Tp module shall check the `PduIdType` parameters of its service functions against the configured parameter of `IEEE1722TpStreamTxPduId`, `IEEE1722TpStreamRxPduId`, `IEEE1722TpLowerLayerTxPduId` and `IEEE1722TpLowerLayerRxPduId`, and report the development error `IEEE1722TP_E_INVALID_PDU_SDU_ID` when an unknown ID is provided by the call:

- `TxPduId` of callback function `IEEE1722Tp_TxConfirmation`
- `RxPduId` of callback function `IEEE1722Tp_RxIndication`
- `TxPduId` of function `IEEE1722Tp_Transmit`
- `RxPduId` of function `IEEE1722Tp_ReleaseRxBuffer`

⌋

**[CP_SWS_IEEE1722Tp_00227]**
*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00386

⌈If development error detection is enabled by `IEEE1722TpDevErrorDetect`, the IEEE1722Tp module shall check the `IEEE1722Tp_StreamIndexType` parameters of its service functions against the configured parameters `IEEE1722TpStreamIndex`, and shall report the development error `IEEE1722TP_E_INVALID_STREAM_ID` when an unknown stream index value is provided by the call.⌋

## 8.4 Scheduled functions

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

### 8.4.1 IEEE1722Tp_MainFunctionTx

**[CP_SWS_IEEE1722Tp_91031] Definition of scheduled function IEEE1722Tp_MainFunctionTx**

*Status:* DRAFT

⌈

| | |
|---|---|
| ***Service Name*** | IEEE1722Tp_MainFunctionTx (draft) |
| ***Syntax*** | `void IEEE1722Tp_MainFunctionTx ( void )` |
| ***Service ID [hex]*** | 0x0a |
| ***Description*** | The function issues transmission requests in polling mode<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_IEEE1722Tp.h |

⌋

### 8.4.2 IEEE1722Tp_MainFunctionRx

**[CP_SWS_IEEE1722Tp_91032] Definition of scheduled function IEEE1722Tp_MainFunctionRx**

*Status:* DRAFT

⌈

| | |
|---|---|
| ***Service Name*** | IEEE1722Tp_MainFunctionRx (draft) |
| ***Syntax*** | `void IEEE1722Tp_MainFunctionRx ( void )` |
| ***Service ID [hex]*** | 0x0b |
| ***Description*** | The function issues reception indications in polling mode<br>**Tags:** atp.Status=draft |
| ***Available via*** | SchM_IEEE1722Tp.h |

⌋

## 8.5 Expected interfaces

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

### 8.5.1 Mandatory interfaces

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

**[CP_SWS_IEEE1722Tp_91035] Definition of mandatory interfaces required by module IEEE1722Tp** ⌈

| *API Function* | *Header File* | *Description* |
|---|---|---|
| Det_ReportRuntimeError | Det.h | Service to report runtime errors. If a callout has been configured then this callout shall be called. |
| LSduR_IEEE1722TpReleaseRxBuffer (draft) | LSduR_<module>.h | Indication from the upper layer to release the lower layer reception buffer. |
| LSduR_IEEE1722TpRxIndication (draft) | LSduR_<module>.h | Indication of a received PDU from a lower layer communication interface module. |
| LSduR_IEEE1722TpTransmit (draft) | LSduR_<module>.h | Requests transmission of a PDU. |
| LSduR_IEEE1722TpTxConfirmation (draft) | LSduR_<module>.h | The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU. |
| StbM_GetCurrentTime | StbM.h | Returns a time tuple (Local time, Global time and Timebase status) and user data details Note: This API shall be called with locked interrupts / within an Exclusive Area to prevent interruption (i.e., the risk that the time stamp is outdated on return of the function call). |

▽

△

| API Function | Header File | Description |
|---|---|---|
| StbM_GetTimeBaseStatus | StbM.h | Returns detailed status information for a Synchronized (or Pure Local) Time Base. |

⌋

### 8.5.2 Optional interfaces

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

**[CP_SWS_IEEE1722Tp_91034] Definition of optional interfaces requested by module IEEE1722Tp** ⌈

| API Function | Header File | Description |
|---|---|---|
| EthIf_GetCurrentTimeTuple (draft) | EthIf.h | Reads the current time of the timestamp clock and the current time of the PHC in an atomic operation.<br>**Tags:** atp.Status=draft |

⌋

## 8.6 Service Interfaces

The IEEE1722Tp module does not define service interfaces.

# 9 Sequence diagrams

This chapter shows explanational sequence diagrams for IEEE1722 stream handling

## 9.1 IEEE1722 stream transmission

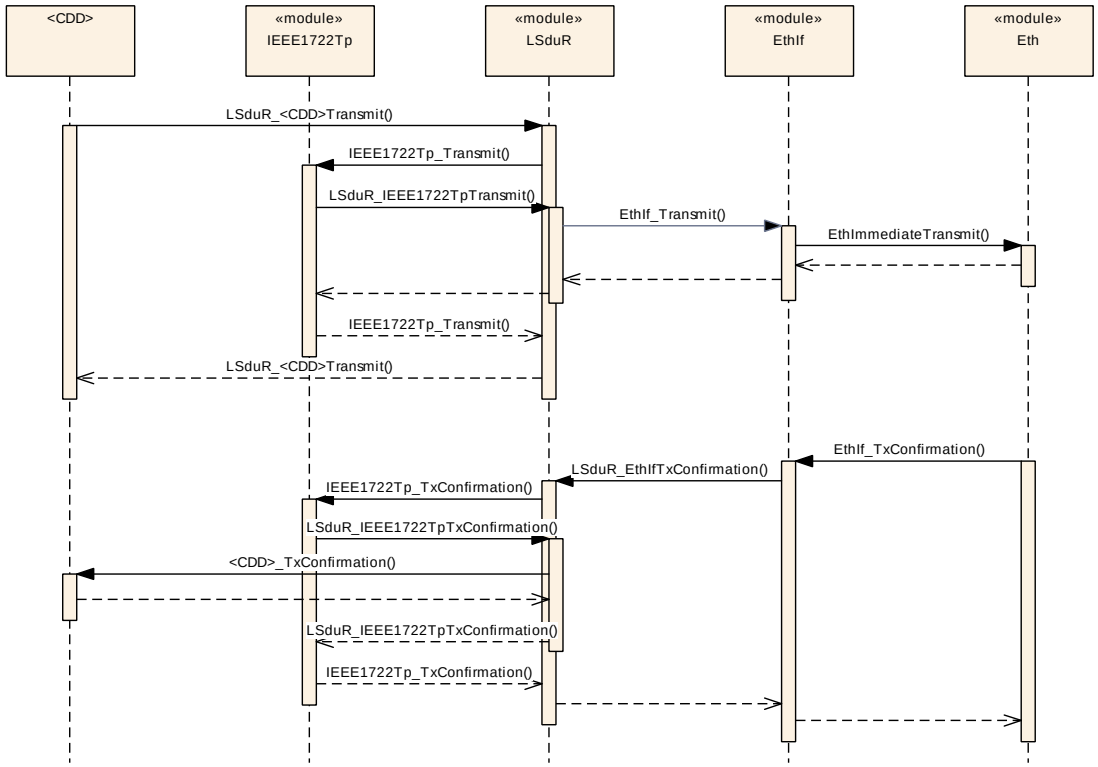


**Figure 9.1: IEEE1722 stream transmission**

## 9.2 IEEE1722 stream reception

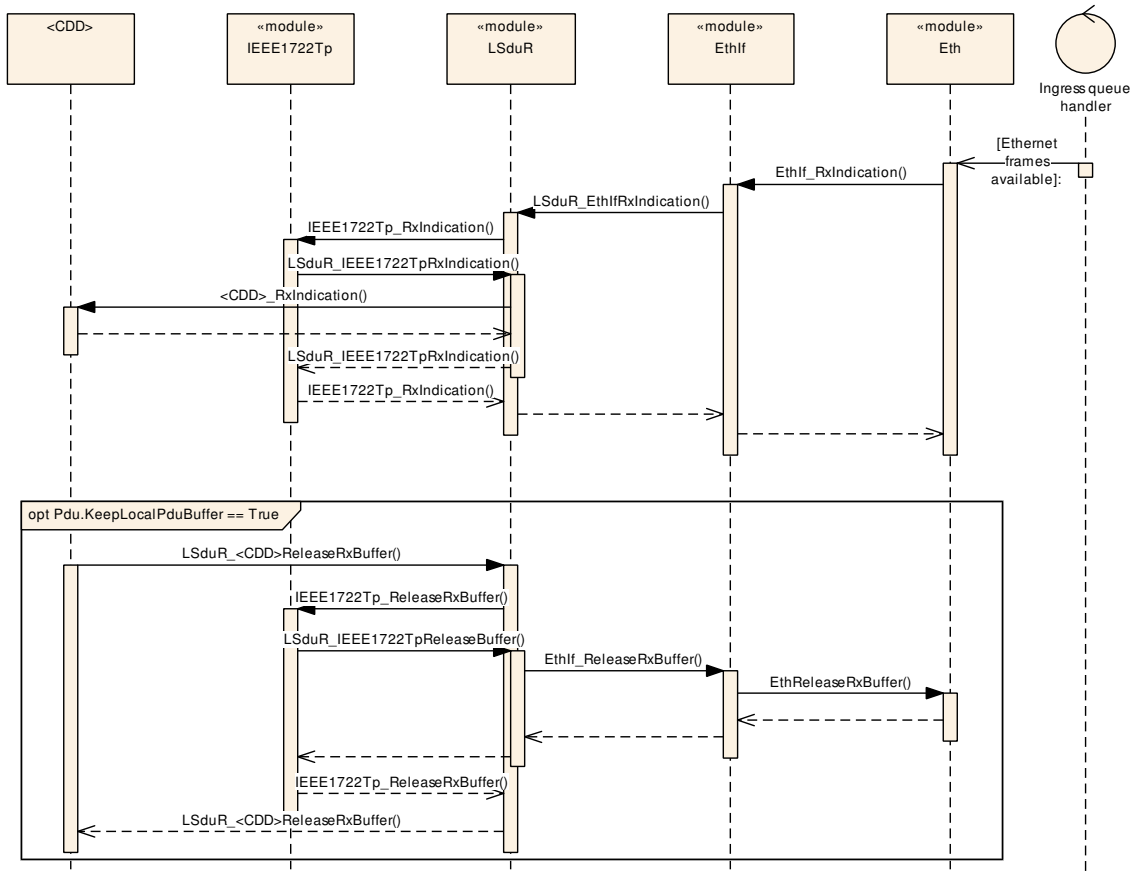


**Figure 9.2: IEEE1722 stream reception**

# 10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. or general information about the definition of containers and parameters, refer to [5, SWS BSW General] 10.1 *"Introduction to configuration specification"*.

Section 10.1 specifies the structure (containers) and the parameters of the `IEEE1722Tp` module.

## 10.1 Containers and configuration parameters

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

### 10.1.1 IEEE1722Tp

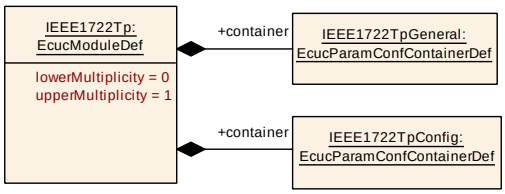


**Figure 10.1: IEEE1722Tp**

**[ECUC_IEEE1722Tp_00001] Definition of EcucModuleDef IEEE1722Tp**

*Status:* DRAFT

⌈

| | |
|---|---|
| **Module Name** | IEEE1722Tp |
| **Description** | Configuration of the IEEE1722Tp module. |
| **Post-Build Selectable Support** | true |
| **Post-Build Loadable Support** | true |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpConfig | 1 | This container contains the configuration parameters and sub containers of the AUTOSAR IEEE1722Tp module.<br>**Tags:** atp.Status=draft |
| IEEE1722TpGeneral | 1 | Specifies the general configuration parameters of the IEEE1722 Tp.<br>**Tags:** atp.Status=draft |

⌋

### 10.1.2 IEEE1722TpGeneral

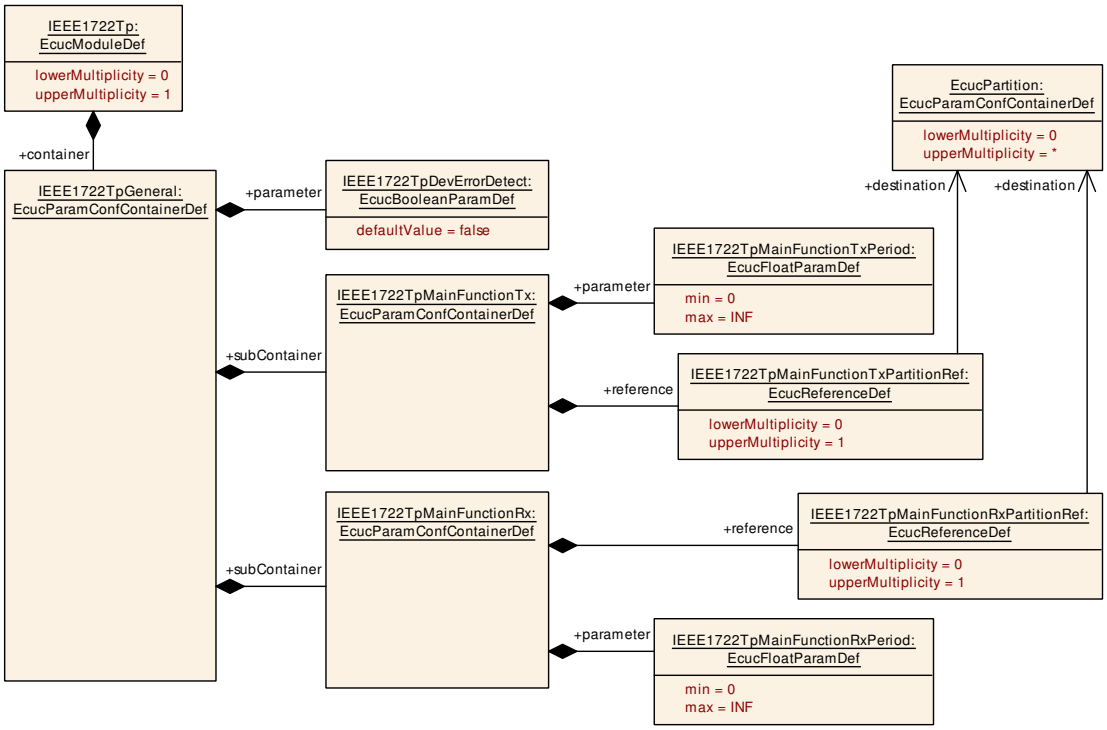


**Figure 10.2: IEEE1722TpGeneral**

## [ECUC_IEEE1722Tp_00002] Definition of EcucParamConfContainerDef IEEE1722TpGeneral

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpGeneral |
| **Parent Container** | IEEE1722Tp |
| **Upstream mapping defined** | – |
| **Description** | Specifies the general configuration parameters of the IEEE1722Tp.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpDevErrorDetect | 1 | [ECUC_IEEE1722Tp_00003] |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpMainFunctionRx | 1 | Defines the IEEE1722TpMainFunctionRx.<br>**Tags:** atp.Status=draft |
| IEEE1722TpMainFunctionTx | 1 | Defines the IEEE1722TpMainFunctionTx.<br>**Tags:** atp.Status=draft |

⌋

### [ECUC_IEEE1722Tp_00003] Definition of EcucBooleanParamDef IEEE1722Tp DevErrorDetect

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpDevErrorDetect |
| **Parent Container** | IEEE1722TpGeneral |
| **Upstream mapping defined** | – |
| **Description** | Switches the development error detection and notification on or off.<br>• true: detection and notification is enabled.<br>• false: detection and notification is disabled.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucBooleanParamDef |
| **Default value** | false |
| **Pre-Compile only** | true |

⌋

### [ECUC_IEEE1722Tp_00104] Definition of EcucParamConfContainerDef IEEE1722 TpMainFunctionRx

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpMainFunctionRx |
| **Parent Container** | IEEE1722TpGeneral |
| **Upstream mapping defined** | – |
| **Description** | Defines the IEEE1722TpMainFunctionRx.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpMainFunctionRxPeriod | 1 | [ECUC_IEEE1722Tp_00102] |
| IEEE1722TpMainFunctionRxPartitionRef | 0..1 | [ECUC_IEEE1722Tp_00106] |

| **No Included Containers** |
|---|

⌋

### [ECUC_IEEE1722Tp_00102] Definition of EcucFloatParamDef IEEE1722TpMainFunctionRxPeriod

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpMainFunctionRxPeriod |
| **Parent Container** | IEEE1722TpMainFunctionRx |
| **Upstream mapping defined** | – |
| **Description** | Specifies the period of the Rx main function IEEE1722Tp_MainFunctionRx in seconds.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucFloatParamDef |
| **Range** | ]0 .. INF[ |
| **Default value** | – |
| **Pre-Compile only** | true |

⌋

### [ECUC_IEEE1722Tp_00106] Definition of EcucReferenceDef IEEE1722TpMainFunctionRxPartitionRef

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpMainFunctionRxPartitionRef |
| **Parent Container** | IEEE1722TpMainFunctionRx |
| **Upstream mapping defined** | – |
| **Description** | Reference to EcucPartition, where the IEEE1722Tp_MainFunctionRx is assigned to.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Reference to EcucPartition |
| **Pre-Compile only** | true |

⌋

### [ECUC_IEEE1722Tp_00103] Definition of EcucParamConfContainerDef IEEE1722TpMainFunctionTx

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpMainFunctionTx |
| **Parent Container** | IEEE1722TpGeneral |
| **Upstream mapping defined** | – |
| **Description** | Defines the IEEE1722TpMainFunctionTx.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpMainFunctionTxPeriod | 1 | [ECUC_IEEE1722Tp_00097] |
| IEEE1722TpMainFunctionTxPartitionRef | 0..1 | [ECUC_IEEE1722Tp_00105] |

| No Included Containers |
|---|

⌋

### [ECUC_IEEE1722Tp_00097] Definition of EcucFloatParamDef IEEE1722TpMainFunctionTxPeriod

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpMainFunctionTxPeriod | |
| **Parent Container** | IEEE1722TpMainFunctionTx | |
| **Upstream mapping defined** | – | |
| **Description** | Specifies the period of the Tx main function IEEE1722Tp_MainFunctionTx in seconds.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucFloatParamDef | |
| **Range** | ]0 .. INF[ | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_IEEE1722Tp_00105] Definition of EcucReferenceDef IEEE1722TpMainFunctionTxPartitionRef

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpMainFunctionTxPartitionRef |
| **Parent Container** | IEEE1722TpMainFunctionTx |
| **Upstream mapping defined** | – |
| **Description** | Reference to EcucPartition, where the IEEE1722Tp_MainFunctionTx is assigned to.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Reference to EcucPartition |
| **Pre-Compile only** | true |

⌋

### 10.1.3 IEEE1722TpConfig

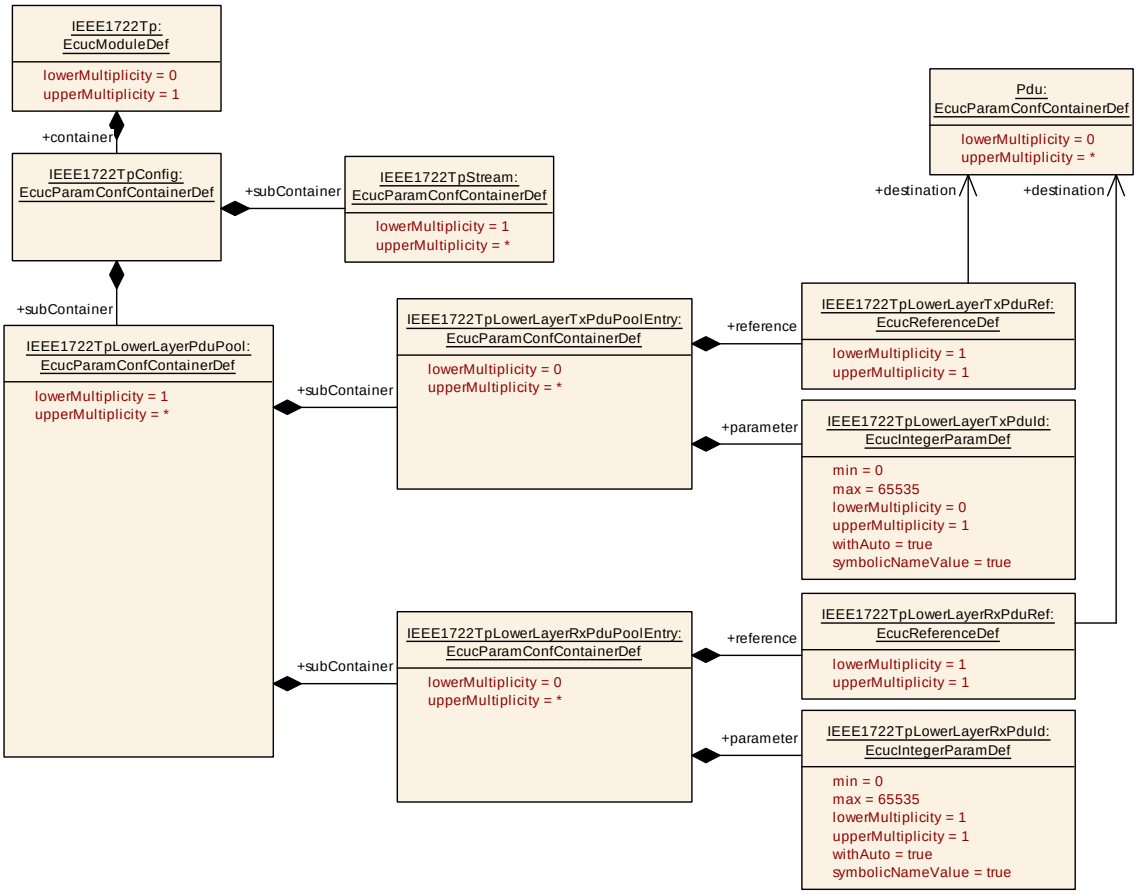


**Figure 10.3: IEEE1722TpConfig**

## [ECUC_IEEE1722Tp_00005] Definition of EcucParamConfContainerDef IEEE1722TpConfig

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpConfig |
| **Parent Container** | IEEE1722Tp |
| **Upstream mapping defined** | – |
| **Description** | This container contains the configuration parameters and sub containers of the AUTOSAR IEEE1722Tp module.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| |
|---|
| **No Included Parameters** |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpLowerLayerPduPool | 1..* | Each container defines one lower layer Pdu pool (either for Tx or Rx). Each Stream then may select over which Pdu pool their transport shall be done.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStream | 1..* | Definition of an IEEE1722Tp stream.<br>**Tags:** atp.Status=draft |

⌋

### 10.1.3.1 IEEE1722Tp Lower Layer Pdu configuration

### [ECUC_IEEE1722Tp_00031] Definition of EcucParamConfContainerDef IEEE1722TpLowerLayerPduPool

*Status:* DRAFT

⌈

| **Container Name** | IEEE1722TpLowerLayerPduPool |
|---|---|
| **Parent Container** | IEEE1722TpConfig |
| **Upstream mapping defined** | – |
| **Description** | Each container defines one lower layer Pdu pool (either for Tx or Rx). Each Stream then may select over which Pdu pool their transport shall be done.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| No Included Parameters |
|---|

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpLowerLayerRxPduPoolEntry | 0..* | This container defines one entry in the IEEE1722TpLowerLayerPduPool for Rx direction to be used for the transport of Rx Pdus from the lower layer.<br>This container is only required if at least one stream is consumed by the IEEE1722Tp module.<br>Supported MetaData entry:<br>• TIMETUPLE_TYPE_PTR<br>**Tags:** atp.Status=draft |

▽

△

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpLowerLayerTxPduPoolEntry | 0..* | This container defines one entry in the IEEE1722TpLowerLayerPduPool for Tx direction to be used for the transport of Tx Pdus to the lower layer.<br>This container is only required if at least one stream is produced by the IEEE1722Tp module.<br>Supported MetaData entries:<br>• LISTELEM_PTR<br>• ETHERNET_MAC_64<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_IEEE1722Tp_00025] Definition of EcucParamConfContainerDef IEEE1722TpLowerLayerTxPduPoolEntry

*Status:* DRAFT

⌈

| **Container Name** | IEEE1722TpLowerLayerTxPduPoolEntry |
|---|---|
| **Parent Container** | IEEE1722TpLowerLayerPduPool |
| **Upstream mapping defined** | – |
| **Description** | This container defines one entry in the IEEE1722TpLowerLayerPduPool for Tx direction to be used for the transport of Tx Pdus to the lower layer.<br>This container is only required if at least one stream is produced by the IEEE1722Tp module.<br>Supported MetaData entries:<br>• LISTELEM_PTR<br>• ETHERNET_MAC_64<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpLowerLayerTxPduId | 0..1 | [ECUC_IEEE1722Tp_00027] |
| IEEE1722TpLowerLayerTxPduRef | 1 | [ECUC_IEEE1722Tp_00026] |

| No Included Containers |
|---|

⌋

### [ECUC_IEEE1722Tp_00027] Definition of EcucIntegerParamDef IEEE1722TpLowerLayerTxPduId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpLowerLayerTxPduId | |
| **Parent Container** | IEEE1722TpLowerLayerTxPduPoolEntry | |
| **Upstream mapping defined** | – | |
| **Description** | Definition of the Handle Pdu Id used by the lower layer for Tx Pdu confirmation.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **With Auto** | true | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_IEEE1722Tp_00026] Definition of EcucReferenceDef IEEE1722TpLowerLayerTxPduRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpLowerLayerTxPduRef | |
| **Parent Container** | IEEE1722TpLowerLayerTxPduPoolEntry | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to the EcuC Pdu used for the transport of Tx Pdu to the lower layer.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to Pdu | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00028] Definition of EcucParamConfContainerDef IEEE1722TpLowerLayerRxPduPoolEntry

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpLowerLayerRxPduPoolEntry |
| **Parent Container** | IEEE1722TpLowerLayerPduPool |
| **Upstream mapping defined** | – |

▽

△

| | |
|---|---|
| **Description** | This container defines one entry in the IEEE1722TpLowerLayerPduPool for Rx direction to be used for the transport of Rx Pdus from the lower layer.<br>This container is only required if at least one stream is consumed by the IEEE1722Tp module.<br>Supported MetaData entry:<br>• TIMETUPLE_TYPE_PTR<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpLowerLayerRxPduId | 1 | [ECUC_IEEE1722Tp_00030] |
| IEEE1722TpLowerLayerRxPduRef | 1 | [ECUC_IEEE1722Tp_00029] |

| **No Included Containers** |
|---|

⌋

## [ECUC_IEEE1722Tp_00030] Definition of EcucIntegerParamDef IEEE1722Tp LowerLayerRxPduId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpLowerLayerRxPduId | |
| **Parent Container** | IEEE1722TpLowerLayerRxPduPoolEntry | |
| **Upstream mapping defined** | – | |
| **Description** | Definition of the Handle Pdu Id used by the lower layer for Rx Pdu indication.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **With Auto** | true | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_IEEE1722Tp_00029] Definition of EcucReferenceDef IEEE1722TpLowerLayerRxPduRef

*Status:* DRAFT

⌈

| Parameter Name | IEEE1722TpLowerLayerRxPduRef | |
|---|---|---|
| Parent Container | IEEE1722TpLowerLayerRxPduPoolEntry | |
| Upstream mapping defined | – | |
| Description | Reference to the EcuC Pdu used for the transport of Rx Pdu from the lower layer.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | Reference to Pdu | |
| Post-Build Value | Loadable | Selectable |

⌋

### 10.1.4 IEEE1722Tp stream configuration

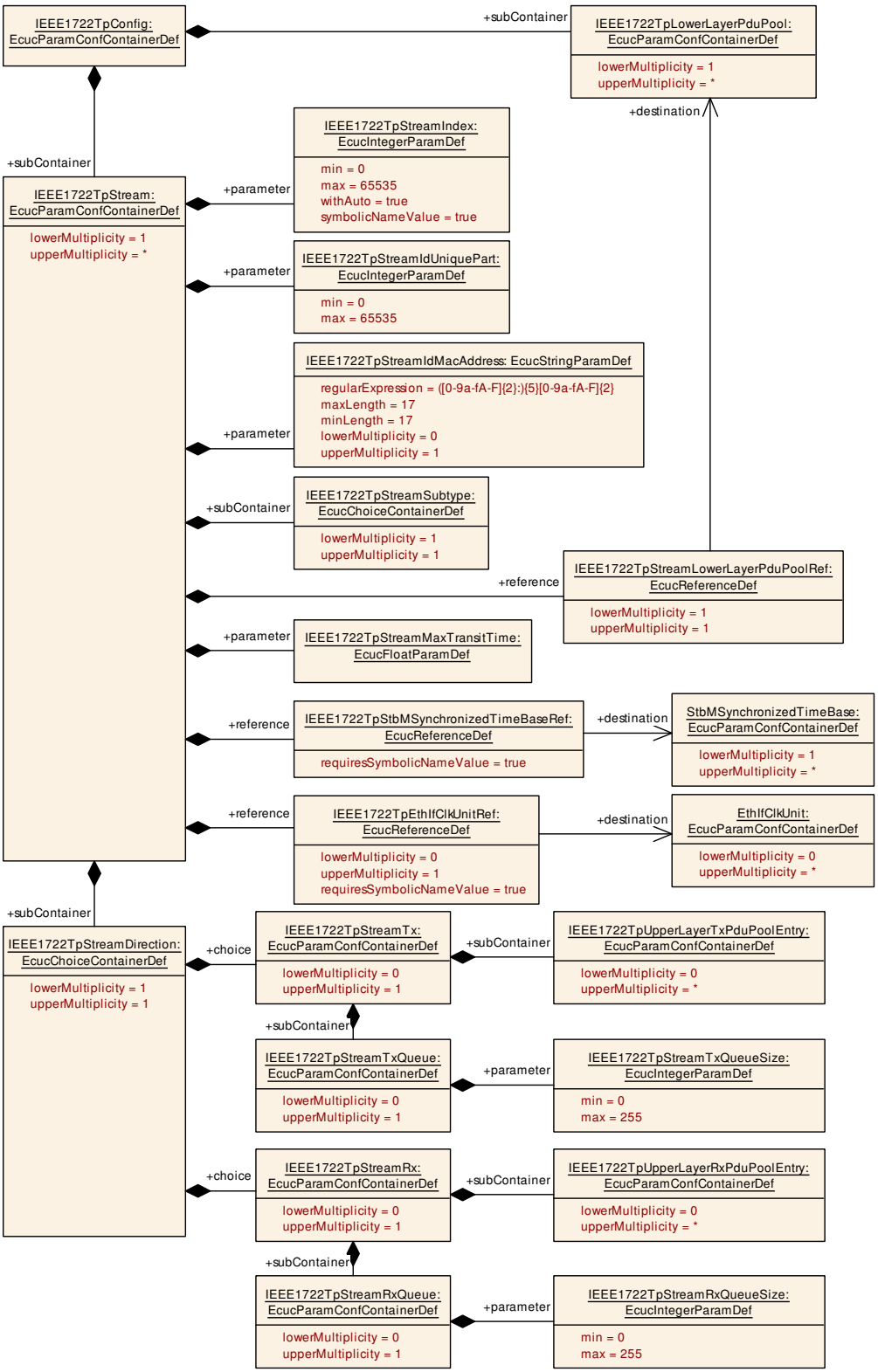


**Figure 10.4: IEEE1722TpStream**

### [ECUC_IEEE1722Tp_00006] Definition of EcucParamConfContainerDef IEEE1722 TpStream

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | IEEE1722TpStream | |
| **Parent Container** | IEEE1722TpConfig | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of an IEEE1722Tp stream.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1..* | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamIdMacAddress | 0..1 | [ECUC_IEEE1722Tp_00008] |
| IEEE1722TpStreamIdUniquePart | 1 | [ECUC_IEEE1722Tp_00007] |
| IEEE1722TpStreamIndex | 1 | [ECUC_IEEE1722Tp_00091] |
| IEEE1722TpStreamMaxTransitTime | 1 | [ECUC_IEEE1722Tp_00033] |
| IEEE1722TpStreamVersion | 1 | [ECUC_IEEE1722Tp_00039] |
| IEEE1722TpEthIfClkUnitRef | 0..1 | [ECUC_IEEE1722Tp_00101] |
| IEEE1722TpStbMSynchronizedTimeBaseRef | 1 | [ECUC_IEEE1722Tp_00100] |
| IEEE1722TpStreamLowerLayerPduPoolRef | 1 | [ECUC_IEEE1722Tp_00032] |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamDirection | 1 | Choice of the IEEE1722 stream direction, either Tx or Rx is configurable.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamSubtype | 1 | Choice of the IEEE1722 stream subtype.<br>**Tags:** atp.Status=draft |

⌋

### [ECUC_IEEE1722Tp_00008] Definition of EcucStringParamDef IEEE1722Tp StreamIdMacAddress

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamIdMacAddress |
| **Parent Container** | IEEE1722TpStream |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of the MAC address part of the Stream Id.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |

▽

△

| | | |
|---|---|---|
| **Type** | EcucStringParamDef | |
| **Default value** | – | |
| **Length** | 17-17 | |
| **Regular Expression** | ([0-9a-fA-F]{2}:){5}[0-9a-fA-F]{2} | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00007] Definition of EcucIntegerParamDef IEEE1722Tp StreamIdUniquePart

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIdUniquePart | |
| **Parent Container** | IEEE1722TpStream | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the unique ID part of the Stream Id.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00091] Definition of EcucIntegerParamDef IEEE1722Tp StreamIndex

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIndex | |
| **Parent Container** | IEEE1722TpStream | |
| **Upstream mapping defined** | – | |
| **Description** | Definition of the Handle Index to identify this stream for API access in the communication stack.<br>This value is NOT related to the stream id combined out of IEEE1722TpStreamId UniquePart and IEEE1722TpStreamIdMacAddress.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **With Auto** | true | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_IEEE1722Tp_00033] Definition of EcucFloatParamDef IEEE1722Tp StreamMaxTransitTime

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamMaxTransitTime | |
| **Parent Container** | IEEE1722TpStream | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the max transit time for the stream.<br>Value in seconds.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucFloatParamDef | |
| **Range** | [-INF .. INF] | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00039] Definition of EcucIntegerParamDef IEEE1722Tp StreamVersion

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamVersion | |
| **Parent Container** | IEEE1722TpStream | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the stream version.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 7 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00101] Definition of EcucReferenceDef IEEE1722TpEthIfClkUnitRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpEthIfClkUnitRef | |
| **Parent Container** | IEEE1722TpStream | |
| **Upstream mapping defined** | – | |

▽

△

| | |
|---|---|
| **Description** | Reference to the EthIfClkUnit from which the current synchronized time could be retrieved (e.g. determine avtp timestamp for transmission).<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to EthIfClkUnit |
| **Pre-Compile only** | true |

⌋

### [ECUC_IEEE1722Tp_00100] Definition of EcucReferenceDef IEEE1722TpStbMSynchronizedTimeBaseRef

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStbMSynchronizedTimeBaseRef |
| **Parent Container** | IEEE1722TpStream |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Reference to the StbMSynchronizedTimeBase from which the current synchronized time could be retrieved (e.g. determine avtp timestamp for transmission).<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | Symbolic name reference to StbMSynchronizedTimeBase |
| **Pre-Compile only** | true |

⌋

### [ECUC_IEEE1722Tp_00032] Definition of EcucReferenceDef IEEE1722TpStreamLowerLayerPduPoolRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamLowerLayerPduPoolRef | |
| **Parent Container** | IEEE1722TpStream | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to one Pdu collection for the transport towards/from the lower layer.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to IEEE1722TpLowerLayerPduPool | |
| **Post-Build Value** | Loadable | Selectable |

⌋

#### 10.1.4.1 IEEE1722Tp Upper Layer Pdu configuration

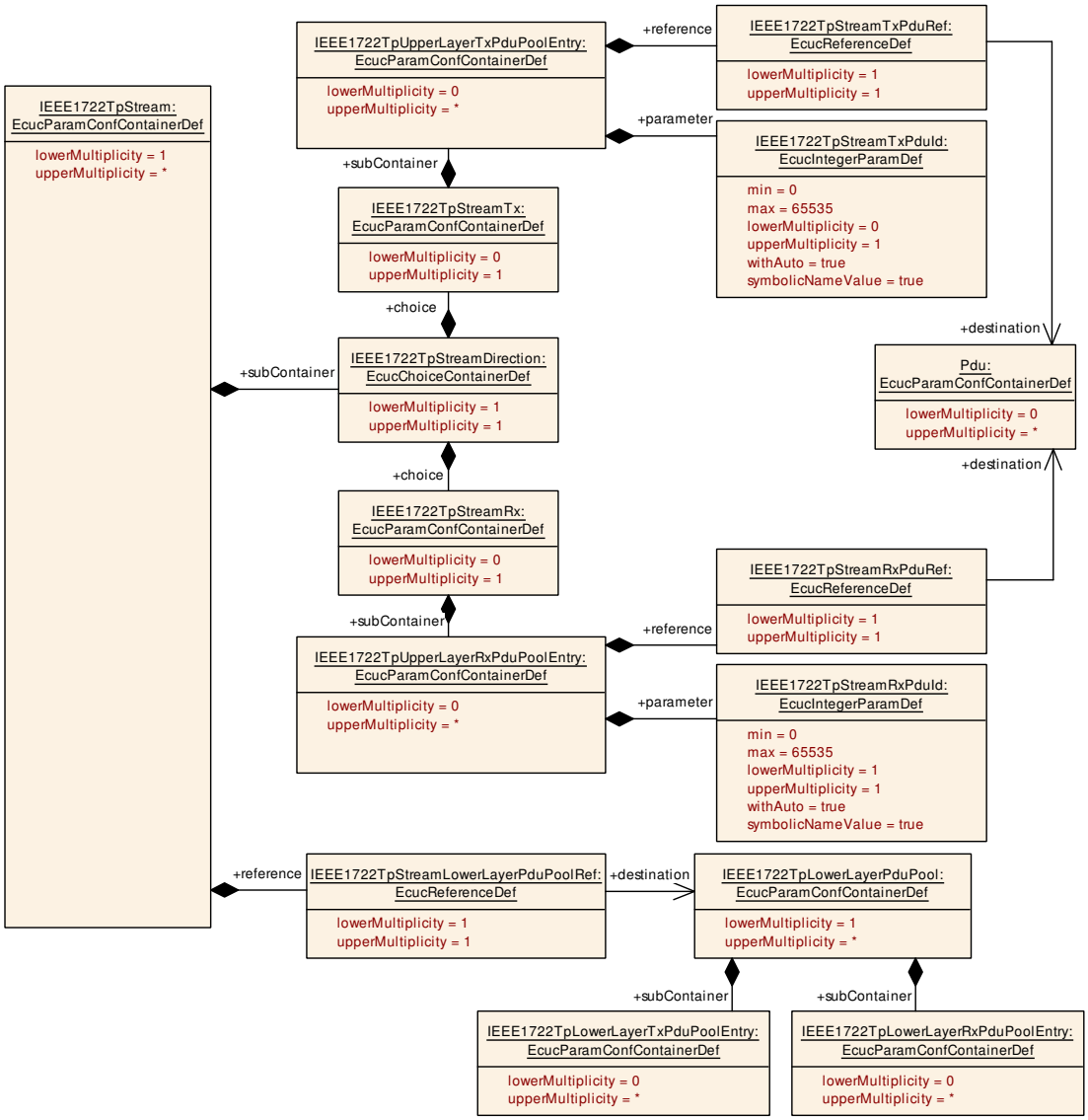


**Figure 10.5: IEEE1722TpStreamPdus**

### [ECUC_IEEE1722Tp_00107] Definition of EcucChoiceContainerDef IEEE1722Tp StreamDirection

*Status:* DRAFT

⌈

| | |
|---|---|
| **Choice Container Name** | IEEE1722TpStreamDirection |
| **Parent Container** | IEEE1722TpStream |
| **Upstream mapping defined** | – |

▽

△

| | |
|---|---|
| **Description** | Choice of the IEEE1722 stream direction, either Tx or Rx is configurable.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |

| **No Included Parameters** |
|---|

| **Container Choices** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamRx | 0..1 | This container defines exclusive parameters for Tx direction.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamTx | 0..1 | This container defines exclusive parameters for Tx direction.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_IEEE1722Tp_00108] Definition of EcucParamConfContainerDef IEEE1722TpStreamTx

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamTx |
| **Parent Container** | IEEE1722TpStreamDirection |
| **Upstream mapping defined** | – |
| **Description** | This container defines exclusive parameters for Tx direction.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **No Included Parameters** |
|---|

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamTxQueue | 0..1 | Definition of an IEEE1722Tp stream Tx queue.<br>**Tags:** atp.Status=draft |
| IEEE1722TpUpperLayerTxPduPoolEntry | 0..* | This container defines one entry in the IEEE1722TpUpperLayerPduPool for Tx direction to be used for the transport of Tx Pdus from the upper layer.<br>This container is only required if the stream is produced by the IEEE1722Tp module and is not an ACF stream.<br>Supported MetaData entries:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• IEEE1722TP_TX_IEC61883_IIDC_PTR<br>• IEEE1722TP_TX_IEC61883_PTR<br>• IEEE1722TP_TX_IEC61883_CIP_NO_SPH_PTR<br>• IEEE1722TP_TX_IEC61883_CIP_WITH_SPH_PTR<br>• IEEE1722TP_TX_AAF_PCM_PTR<br>• IEEE1722TP_TX_AAF_AES3_PTR<br>• IEEE1722TP_TX_RVF_PTR<br>• IEEE1722TP_TX_CRF_PTR<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_IEEE1722Tp_00023] Definition of EcucParamConfContainerDef IEEE1722TpUpperLayerTxPduPoolEntry

*Status:* DRAFT

⌈

| **Container Name** | IEEE1722TpUpperLayerTxPduPoolEntry |
|---|---|
| **Parent Container** | IEEE1722TpStreamTx |
| **Upstream mapping defined** | – |
| **Description** | This container defines one entry in the IEEE1722TpUpperLayerPduPool for Tx direction to be used for the transport of Tx Pdus from the upper layer.<br>This container is only required if the stream is produced by the IEEE1722Tp module and is not an ACF stream.<br>Supported MetaData entries:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• IEEE1722TP_TX_IEC61883_IIDC_PTR<br>• IEEE1722TP_TX_IEC61883_PTR<br>• IEEE1722TP_TX_IEC61883_CIP_NO_SPH_PTR<br>• IEEE1722TP_TX_IEC61883_CIP_WITH_SPH_PTR<br>• IEEE1722TP_TX_AAF_PCM_PTR<br>• IEEE1722TP_TX_AAF_AES3_PTR<br>• IEEE1722TP_TX_RVF_PTR<br>• IEEE1722TP_TX_CRF_PTR<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamTxPduId | 0..1 | [ECUC_IEEE1722Tp_00020] |
| IEEE1722TpStreamTxPduRef | 1 | [ECUC_IEEE1722Tp_00019] |

| No Included Containers |
|---|

⌋

### [ECUC_IEEE1722Tp_00020] Definition of EcucIntegerParamDef IEEE1722TpStreamTxPduId

*Status:* DRAFT

⌈

| **Parameter Name** | IEEE1722TpStreamTxPduId | |
|---|---|---|
| **Parent Container** | IEEE1722TpUpperLayerTxPduPoolEntry | |
| **Upstream mapping defined** | – | |
| **Description** | Definition of the Handle Pdu Id used by the upper layer for Tx Pdu confirmation.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **With Auto** | true | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_IEEE1722Tp_00019] Definition of EcucReferenceDef IEEE1722TpStreamTxPduRef

*Status:* DRAFT

⌈

| **Parameter Name** | IEEE1722TpStreamTxPduRef | |
|---|---|---|
| **Parent Container** | IEEE1722TpUpperLayerTxPduPoolEntry | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to the EcuC Pdu used for the transport of Tx stream Pdu from the upper layer.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to Pdu | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00109] Definition of EcucParamConfContainerDef IEEE1722TpStreamRx

*Status:* DRAFT

⌈

| Container Name | IEEE1722TpStreamRx |
|---|---|
| Parent Container | IEEE1722TpStreamDirection |
| Upstream mapping defined | – |
| Description | This container defines exclusive parameters for Tx direction.<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Pre-Compile only | true |
| Configuration Parameters | |

| No Included Parameters |
|---|

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamRxQueue | 0..1 | Definition of an IEEE1722Tp stream Rx queue.<br>**Tags:** atp.Status=draft |
| IEEE1722TpUpperLayerRxPduPoolEntry | 0..* | This container defines one entry in the IEEE1722TpUpperLayerRxPduPool for Rx direction to be used for the transport of Rx Pdus to the upper layer.<br>This container is only required if the stream is consumed by the IEEE1722Tp module and is not an ACF stream.<br>Supported MetaData entries:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• IEEE1722TP_RX_IEC61883_IIDC_PTR<br>• IEEE1722TP_RX_IEC61883_PTR<br>• IEEE1722TP_RX_IEC61883_CIP_NO_SPH_PTR<br>• IEEE1722TP_RX_IEC61883_CIP_WITH_SPH_PTR<br>• IEEE1722TP_RX_AAF_PCM_PTR<br>• IEEE1722TP_RX_AAF_AES3_PTR<br>• IEEE1722TP_RX_RVF_PTR<br>• IEEE1722TP_RX_CRF_PTR<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_IEEE1722Tp_00024] Definition of EcucParamConfContainerDef IEEE1722TpUpperLayerRxPduPoolEntry

*Status:* DRAFT

⌈

| Container Name | IEEE1722TpUpperLayerRxPduPoolEntry |
|---|---|
| Parent Container | IEEE1722TpStreamRx |
| Upstream mapping defined | – |

▽

△

| | |
|---|---|
| **Description** | This container defines one entry in the IEEE1722TpUpperLayerRxPduPool for Rx direction to be used for the transport of Rx Pdus to the upper layer.<br>This container is only required if the stream is consumed by the IEEE1722Tp module and is not an ACF stream.<br>Supported MetaData entries:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• IEEE1722TP_RX_IEC61883_IIDC_PTR<br>• IEEE1722TP_RX_IEC61883_PTR<br>• IEEE1722TP_RX_IEC61883_CIP_NO_SPH_PTR<br>• IEEE1722TP_RX_IEC61883_CIP_WITH_SPH_PTR<br>• IEEE1722TP_RX_AAF_PCM_PTR<br>• IEEE1722TP_RX_AAF_AES3_PTR<br>• IEEE1722TP_RX_RVF_PTR<br>• IEEE1722TP_RX_CRF_PTR<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamRxPduId | 1 | [ECUC_IEEE1722Tp_00022] |
| IEEE1722TpStreamRxPduRef | 1 | [ECUC_IEEE1722Tp_00021] |

| **No Included Containers** |
|---|

⌋

## [ECUC_IEEE1722Tp_00022] Definition of EcucIntegerParamDef IEEE1722Tp StreamRxPduId

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamRxPduId |
| **Parent Container** | IEEE1722TpUpperLayerRxPduPoolEntry |
| **Upstream mapping defined** | – |
| **Description** | Definition of the Handle Pdu Id used by the upper layer for Rx Pdu indication.<br>This handle Id is only required if the stream is consumed by the IEEE1722Tp module.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) |
| **Range** | 0 .. 65535 |
| **Default value** | – |
| **With Auto** | true |
| **Pre-Compile only** | true |

⌋

### [ECUC_IEEE1722Tp_00021] Definition of EcucReferenceDef IEEE1722TpStreamRxPduRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRxPduRef | |
| **Parent Container** | IEEE1722TpUpperLayerRxPduPoolEntry | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to the EcuC Pdu used for the transport of Rx stream Pdu to the upper layer.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to Pdu | |
| **Post-Build Value** | Loadable | Selectable |

⌋

## 10.1.4.2 IEEE1722TpStreamTxQueue

### [ECUC_IEEE1722Tp_00017] Definition of EcucParamConfContainerDef IEEE1722TpStreamTxQueue

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamTxQueue |
| **Parent Container** | IEEE1722TpStreamTx |
| **Upstream mapping defined** | – |
| **Description** | Definition of an IEEE1722Tp stream Tx queue.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamTxQueueSize | 1 | [ECUC_IEEE1722Tp_00018] |

| **No Included Containers** |
|---|

⌋

### [ECUC_IEEE1722Tp_00018] Definition of EcucIntegerParamDef IEEE1722Tp StreamTxQueueSize

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamTxQueueSize | |
| **Parent Container** | IEEE1722TpStreamTxQueue | |
| **Upstream mapping defined** | – | |
| **Description** | Definition of the queue size for the stream Tx queue. The queue is configured in number of to be queued elements.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

## 10.1.4.3 IEEE1722TpStreamRxQueue

### [ECUC_IEEE1722Tp_00035] Definition of EcucParamConfContainerDef IEEE1722 TpStreamRxQueue

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamRxQueue |
| **Parent Container** | IEEE1722TpStreamRx |
| **Upstream mapping defined** | – |
| **Description** | Definition of an IEEE1722Tp stream Rx queue.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamRxQueueSize | 1 | [ECUC_IEEE1722Tp_00037] |

| **No Included Containers** |
|---|

⌋

### [ECUC_IEEE1722Tp_00037] Definition of EcucIntegerParamDef IEEE1722Tp StreamRxQueueSize

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRxQueueSize | |
| **Parent Container** | IEEE1722TpStreamRxQueue | |
| **Upstream mapping defined** | – | |
| **Description** | Definition of the queue size for the stream Rx queue. The queue is configured in number of to be queued elements.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

#### 10.1.4.4 IEEE1722TpStreamSubtype

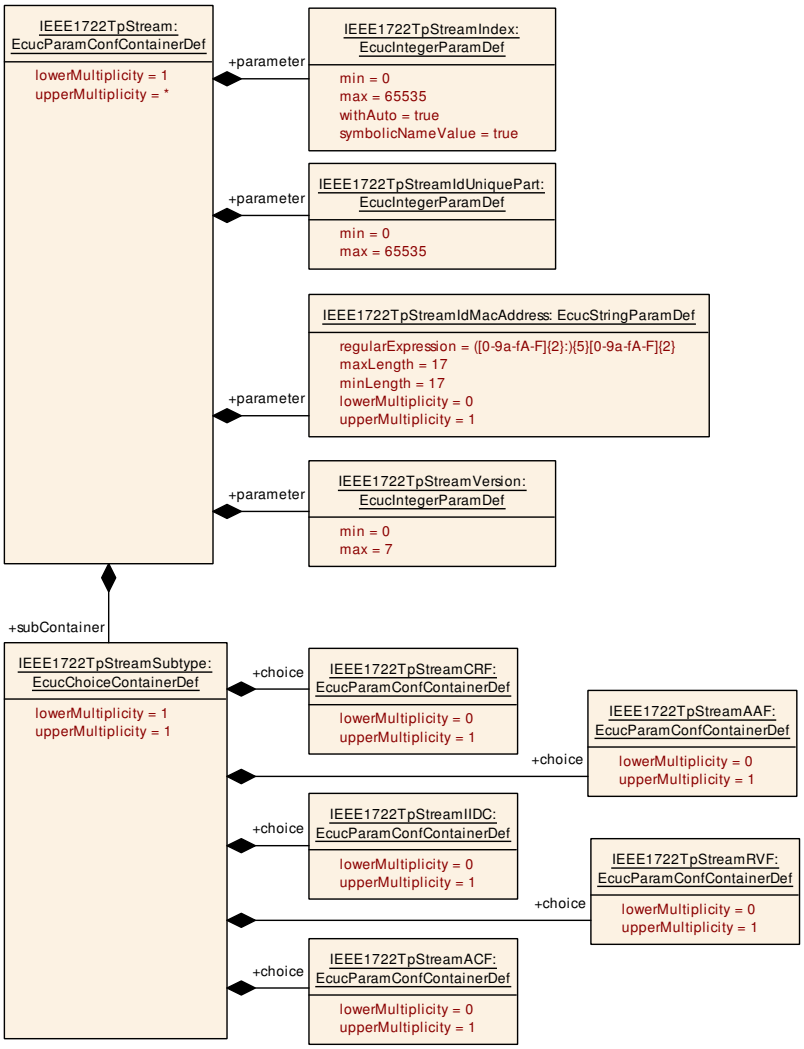


**Figure 10.6: IEEE1722TpStreamSubtype**

### [ECUC_IEEE1722Tp_00009] Definition of EcucChoiceContainerDef IEEE1722TpStreamSubtype

*Status:* DRAFT

⌈

| | |
|---|---|
| **Choice Container Name** | IEEE1722TpStreamSubtype |
| **Parent Container** | IEEE1722TpStream |
| **Upstream mapping defined** | – |
| **Description** | Choice of the IEEE1722 stream subtype.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |

| **No Included Parameters** |
|---|

| Container Choices | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAAF | 0..1 | Definition of an IEEE1722 AVTP Audio Format (AAF) stream.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamACF | 0..1 | Definition of an IEEE1722 AVTP Control Format (ACF) stream.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamCRF | 0..1 | Definition of an IEEE1722 Clock Reference Format (CRF) stream.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamIIDC | 0..1 | Definition of an IEEE1722 61883_IIDC (IEC 61883/IIDC over AVTP) stream.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamRVF | 0..1 | Definition of an IEEE1722 Raw Video Format (RVF) stream.<br>**Tags:** atp.Status=draft |

⌋

### 10.1.4.5 IEEE1722TpStreamCRF

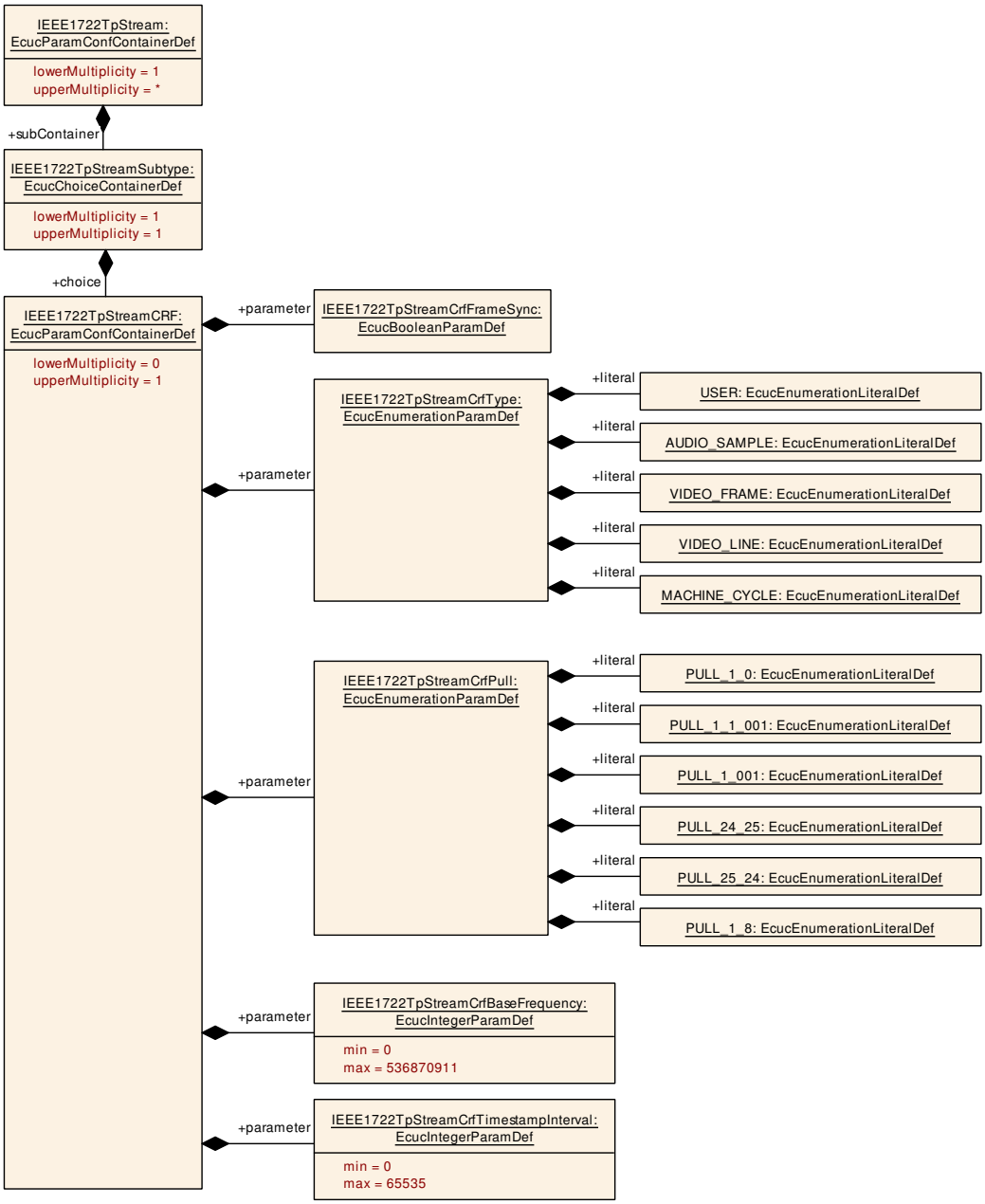


**Figure 10.7: IEEE1722TpStreamCRF**

### [ECUC_IEEE1722Tp_00010] Definition of EcucParamConfContainerDef IEEE1722 TpStreamCRF

*Status:* DRAFT

⌈

| Container Name | IEEE1722TpStreamCRF |
|---|---|
| Parent Container | IEEE1722TpStreamSubtype |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] |
| Description | Definition of an IEEE1722 Clock Reference Format (CRF) stream.<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Pre-Compile only | true |
| Configuration Parameters | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamCrfBaseFrequency | 1 | [ECUC_IEEE1722Tp_00042] |
| IEEE1722TpStreamCrfFrameSync | 1 | [ECUC_IEEE1722Tp_00038] |
| IEEE1722TpStreamCrfPull | 1 | [ECUC_IEEE1722Tp_00041] |
| IEEE1722TpStreamCrfTimestampInterval | 1 | [ECUC_IEEE1722Tp_00058] |
| IEEE1722TpStreamCrfType | 1 | [ECUC_IEEE1722Tp_00040] |

| No Included Containers |
|---|

⌋

### [ECUC_IEEE1722Tp_00042] Definition of EcucIntegerParamDef IEEE1722TpStreamCrfBaseFrequency

*Status:* DRAFT

⌈

| Parameter Name | IEEE1722TpStreamCrfBaseFrequency | |
|---|---|---|
| Parent Container | IEEE1722TpStreamCRF | |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] | |
| Description | Definition of the CRF stream base frequency. This is defined in Hz.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 536870911 | |
| Default value | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00038] Definition of EcucBooleanParamDef IEEE1722Tp StreamCrfFrameSync

*Status:* DRAFT

⌈

| Parameter Name | IEEE1722TpStreamCrfFrameSync | |
|---|---|---|
| Parent Container | IEEE1722TpStreamCRF | |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] | |
| Description | Defines the CRF stream frame sync (fs).<br>true = 1<br>false = 0<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucBooleanParamDef | |
| Default value | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00041] Definition of EcucEnumerationParamDef IEEE1722 TpStreamCrfPull

*Status:* DRAFT

⌈

| Parameter Name | IEEE1722TpStreamCrfPull | |
|---|---|---|
| Parent Container | IEEE1722TpStreamCRF | |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] | |
| Description | Definition of the CRF stream pull value.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucEnumerationParamDef | |
| Range | PULL_1_0 | value=0x00, Multiply base_frequency field by 1.0<br>**Tags:** atp.Status=draft |
| | PULL_1_001 | value=0x02, Multiply base_frequency field by 1.001<br>**Tags:** atp.Status=draft |
| | PULL_1_1_001 | value=0x01, Multiply base_frequency field by 1/1.001<br>**Tags:** atp.Status=draft |
| | PULL_1_8 | value=0x05, Multiply base_frequency field by 1/8<br>**Tags:** atp.Status=draft |
| | PULL_24_25 | value=0x03, Multiply base_frequency field by 24/25<br>**Tags:** atp.Status=draft |
| | PULL_25_24 | value=0x04, Multiply base_frequency field by 25/24<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00058] Definition of EcucIntegerParamDef IEEE1722Tp StreamCrfTimestampInterval

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamCrfTimestampInterval | |
| **Parent Container** | IEEE1722TpStreamCRF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the CRF stream timestamp interval. This is defined as multiple of base frequency.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00040] Definition of EcucEnumerationParamDef IEEE1722 TpStreamCrfType

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamCrfType | |
| **Parent Container** | IEEE1722TpStreamCRF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the CRF stream type.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | AUDIO_SAMPLE | value=0x01, CRF_AUDIO_SAMPLE, Audio sample timestamp<br>**Tags:** atp.Status=draft |
| | MACHINE_CYCLE | value=0x04, CRF_MACHINE_CYCLE, Machine cycle timestamp<br>**Tags:** atp.Status=draft |
| | USER | value=0x00, CRF_USER, User specified<br>**Tags:** atp.Status=draft |
| | VIDEO_FRAME | value=0x02, CRF_VIDEO_FRAME, Video frame sync timestamp<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| | VIDEO_LINE | value=0x03, CRF_VIDEO_LINE, Video line sync timestamp<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

#### 10.1.4.6 IEEE1722TpStreamAAF

The configuration of AAF depends whether the `IEEE1722TpStreamAafFormat` defines a PCM or an AES3 format.

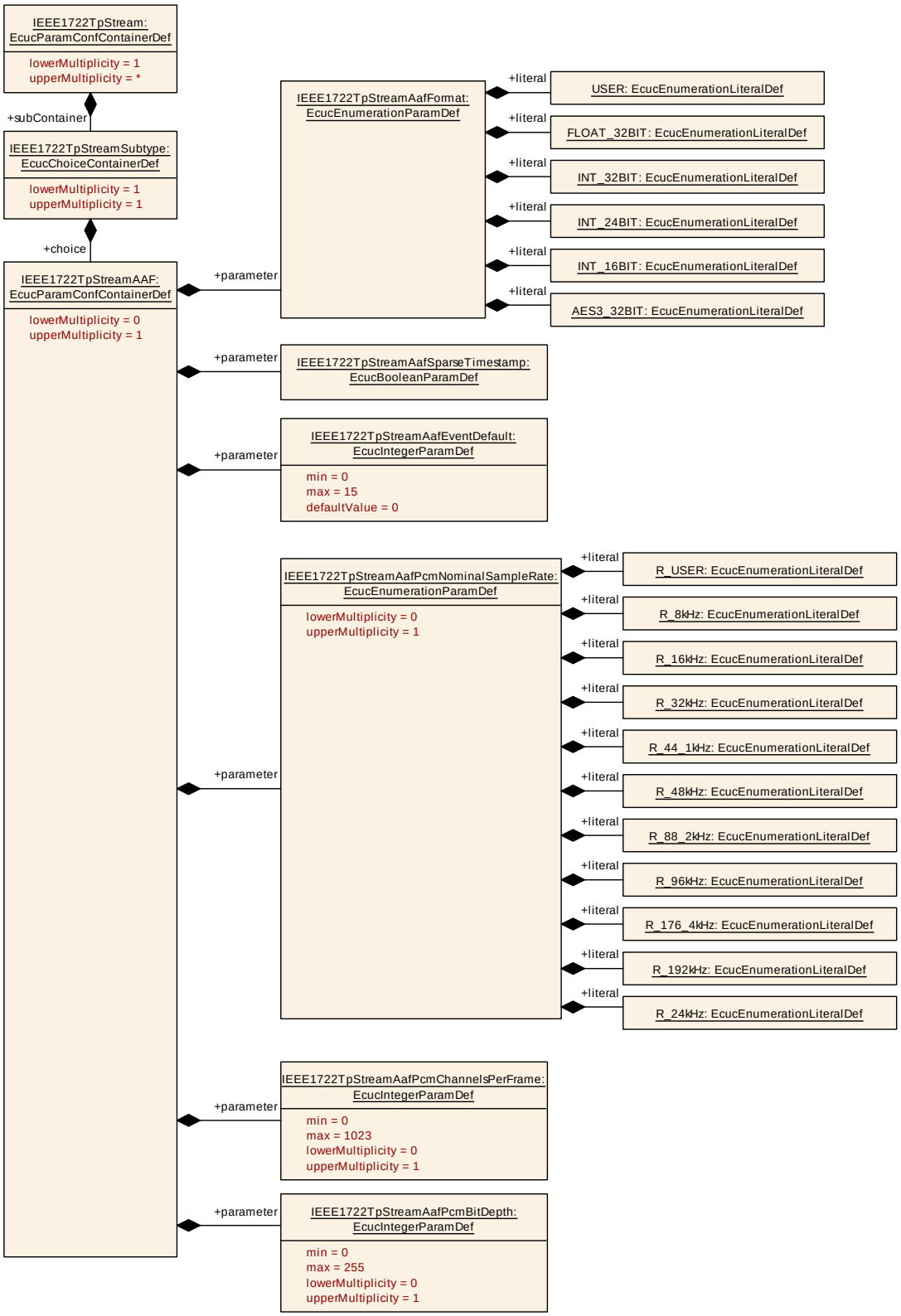


**Figure 10.8: IEEE1722TpStreamAAF with PCM**

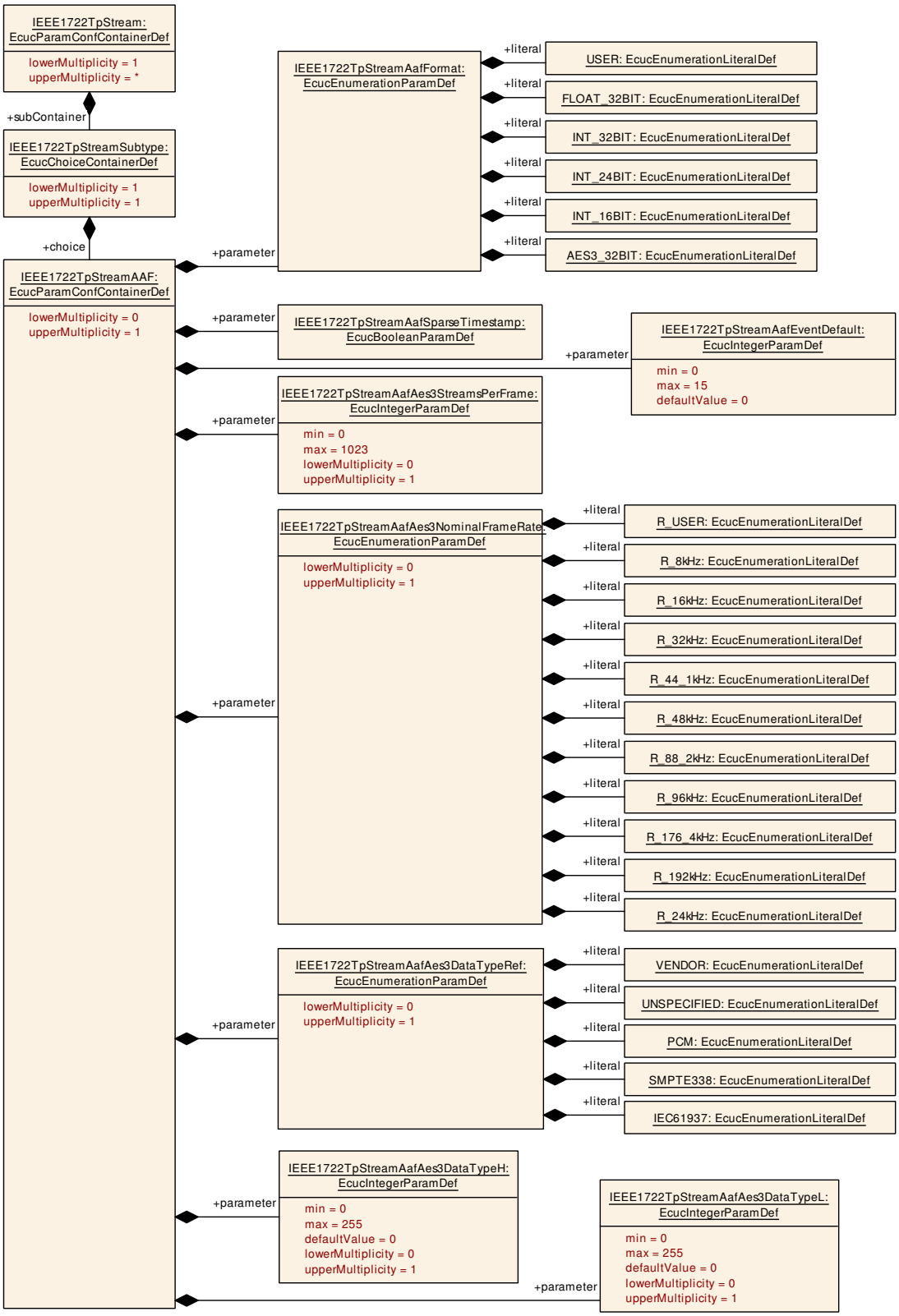


**Figure 10.9: IEEE1722TpStreamAAF with AES3**

### [ECUC_IEEE1722Tp_00011] Definition of EcucParamConfContainerDef IEEE1722TpStreamAAF

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamAAF |
| **Parent Container** | IEEE1722TpStreamSubtype |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of an IEEE1722 AVTP Audio Format (AAF) stream.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAafAes3DataTypeH | 0..1 | [ECUC_IEEE1722Tp_00049] |
| IEEE1722TpStreamAafAes3DataTypeL | 0..1 | [ECUC_IEEE1722Tp_00063] |
| IEEE1722TpStreamAafAes3DataTypeRef | 0..1 | [ECUC_IEEE1722Tp_00062] |
| IEEE1722TpStreamAafAes3NominalFrameRate | 0..1 | [ECUC_IEEE1722Tp_00060] |
| IEEE1722TpStreamAafAes3StreamsPerFrame | 0..1 | [ECUC_IEEE1722Tp_00061] |
| IEEE1722TpStreamAafEventDefault | 1 | [ECUC_IEEE1722Tp_00048] |
| IEEE1722TpStreamAafFormat | 1 | [ECUC_IEEE1722Tp_00043] |
| IEEE1722TpStreamAafPcmBitDepth | 0..1 | [ECUC_IEEE1722Tp_00047] |
| IEEE1722TpStreamAafPcmChannelsPerFrame | 0..1 | [ECUC_IEEE1722Tp_00045] |
| IEEE1722TpStreamAafPcmNominalSampleRate | 0..1 | [ECUC_IEEE1722Tp_00044] |
| IEEE1722TpStreamAafSparseTimestamp | 1 | [ECUC_IEEE1722Tp_00046] |

| **No Included Containers** |
|---|

⌋

### [ECUC_IEEE1722Tp_00049] Definition of EcucIntegerParamDef IEEE1722TpStreamAafAes3DataTypeH

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafAes3DataTypeH | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF AES3 aes3_data_type_h default value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | 0 | |

▽

△

| | | |
|---|---|---|
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00063] Definition of EcucIntegerParamDef IEEE1722Tp StreamAafAes3DataTypeL

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafAes3DataTypeL | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF AES3 aes3_data_type_l default value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | 0 | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00062] Definition of EcucEnumerationParamDef IEEE1722 TpStreamAafAes3DataTypeRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafAes3DataTypeRef | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF AES3 stream aes3_data_type reference (aes3_dt_ref).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | IEC61937 | value=0x3, Data type reference is IEC 61937-2<br>**Tags:** atp.Status=draft |
| | PCM | value=0x1, Data type is PCM<br>**Tags:** atp.Status=draft |
| | SMPTE338 | value=0x2, Data type reference is SMPTE ST 338<br>**Tags:** atp.Status=draft |
| | UNSPECIFIED | value=0x0, Data type not specified<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| | VENDOR | value=0x4, Data type reference is defined by vendor<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00060] Definition of EcucEnumerationParamDef IEEE1722TpStreamAafAes3NominalFrameRate

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafAes3NominalFrameRate | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF AES3 stream nominal frame rate (nfr).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | R_16kHz | value=0x2, 16 kHz<br>**Tags:** atp.Status=draft |
| | R_176_4kHz | value=0x8, 176.4 kHz<br>**Tags:** atp.Status=draft |
| | R_192kHz | value=0x9, 192 kHz<br>**Tags:** atp.Status=draft |
| | R_24kHz | value=0xA, 24 kHz<br>**Tags:** atp.Status=draft |
| | R_32kHz | value=0x3, 32 kHz<br>**Tags:** atp.Status=draft |
| | R_44_1kHz | value=0x4, 44.1 kHz<br>**Tags:** atp.Status=draft |
| | R_48kHz | value=0x5, 48 kHz<br>**Tags:** atp.Status=draft |
| | R_88_2kHz | value=0x6, 88.2 kHz<br>**Tags:** atp.Status=draft |
| | R_8kHz | value=0x1, 8 kHz<br>**Tags:** atp.Status=draft |
| | R_96kHz | value=0x7, 96 kHz<br>**Tags:** atp.Status=draft |
| | R_USER | value=0x0, User specified<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00061] Definition of EcucIntegerParamDef IEEE1722Tp StreamAafAes3StreamsPerFrame

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafAes3StreamsPerFrame | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF AES3 stream streams_per_frame.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 1023 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00048] Definition of EcucIntegerParamDef IEEE1722Tp StreamAafEventDefault

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafEventDefault | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF stream event (evt) default value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 15 | |
| **Default value** | 0 | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00043] Definition of EcucEnumerationParamDef IEEE1722 TpStreamAafFormat

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamAafFormat |
| **Parent Container** | IEEE1722TpStreamAAF |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of the AAF stream format.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |

▽

△

| | | |
|---|---|---|
| **Type** | EcucEnumerationParamDef | |
| **Range** | AES3_32BIT | value=0x05, AES3_32BIT, 32-bit AES3 format, AES3<br>**Tags:** atp.Status=draft |
| | FLOAT_32BIT | value=0x01, FLOAT_32BIT, 32bit floating, PCM<br>**Tags:** atp.Status=draft |
| | INT_16BIT | value=0x04, INT_16BIT, 16 bit integer, PCM<br>**Tags:** atp.Status=draft |
| | INT_24BIT | value=0x03, INT_24BIT, 24 bit integer, PCM<br>**Tags:** atp.Status=draft |
| | INT_32BIT | value=0x02, INT_32BIT, 32bit integer, PCM<br>**Tags:** atp.Status=draft |
| | USER | value=0x00, USER, user specific, PCM<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00047] Definition of EcucIntegerParamDef IEEE1722Tp StreamAafPcmBitDepth

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafPcmBitDepth | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF PCM stream bit_depth.<br>e.g. 16, 24, 32<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00045] Definition of EcucIntegerParamDef IEEE1722Tp StreamAafPcmChannelsPerFrame

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamAafPcmChannelsPerFrame |
| **Parent Container** | IEEE1722TpStreamAAF |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Definition of the AAF PCM stream channels_per_frame.<br>e.g. 1: mono, 2: stereo, 8: 7.1 multicannel<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 1023 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_IEEE1722Tp_00044] Definition of EcucEnumerationParamDef IEEE1722TpStreamAafPcmNominalSampleRate

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafPcmNominalSampleRate | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the AAF PCM stream nominal sample rate (nsr).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | R_16kHz | value=0x2, 16 kHz<br>**Tags:** atp.Status=draft |
| | R_176_4kHz | value=0x8, 176.4 kHz<br>**Tags:** atp.Status=draft |
| | R_192kHz | value=0x9, 192 kHz<br>**Tags:** atp.Status=draft |
| | R_24kHz | value=0xA, 24 kHz<br>**Tags:** atp.Status=draft |
| | R_32kHz | value=0x3, 32 kHz<br>**Tags:** atp.Status=draft |
| | R_44_1kHz | value=0x4, 44.1 kHz<br>**Tags:** atp.Status=draft |
| | R_48kHz | value=0x5, 48 kHz<br>**Tags:** atp.Status=draft |
| | R_88_2kHz | value=0x6, 88.2 kHz<br>**Tags:** atp.Status=draft |
| | R_8kHz | value=0x1, 8 kHz<br>**Tags:** atp.Status=draft |
| | R_96kHz | value=0x7, 96 kHz<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| | R_USER | value=0x0, User specified<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00046] Definition of EcucBooleanParamDef IEEE1722Tp StreamAafSparseTimestamp

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAafSparseTimestamp | |
| **Parent Container** | IEEE1722TpStreamAAF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Defines the CRF stream sparce timestamp (sp).<br>true = 1<br>false = 0<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.1.4.7 IEEE1722TpStreamIIDC

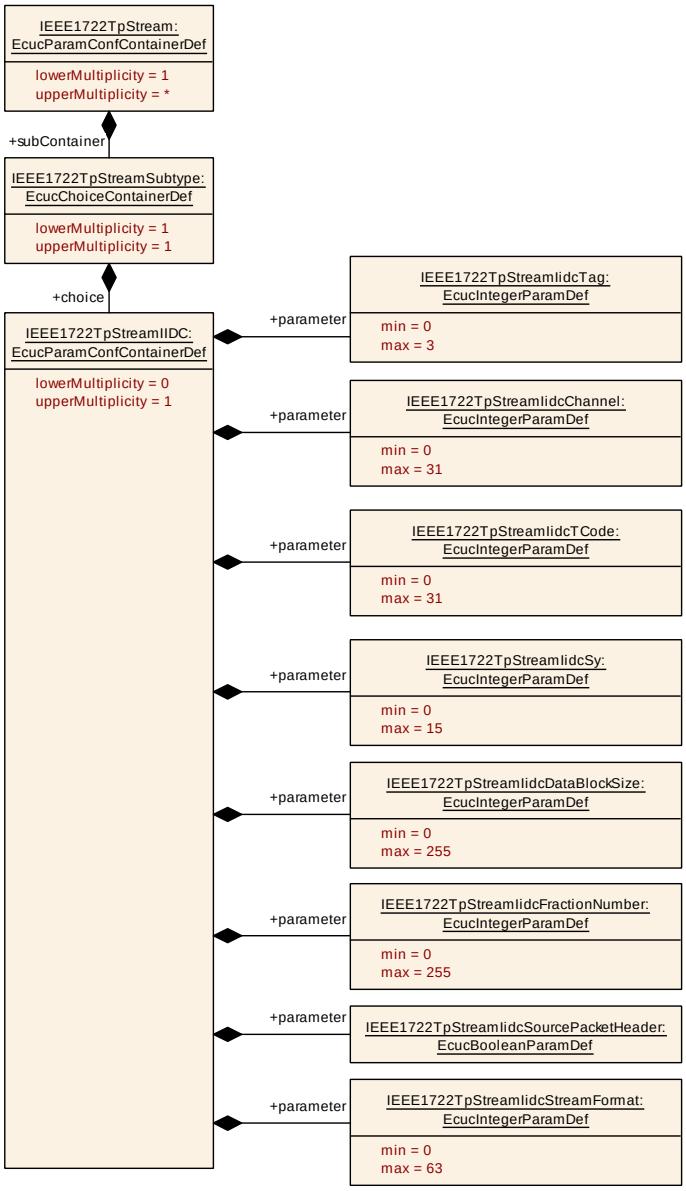


**Figure 10.10: IEEE1722TpStreamIIDC**

## [ECUC_IEEE1722Tp_00012] Definition of EcucParamConfContainerDef IEEE1722TpStreamIIDC

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamIIDC |
| **Parent Container** | IEEE1722TpStreamSubtype |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |

▽

△

| | |
|---|---|
| **Description** | Definition of an IEEE1722 61883_IIDC (IEC 61883/IIDC over AVTP) stream.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamIidcChannel | 1 | [ECUC_IEEE1722Tp_00051] |
| IEEE1722TpStreamIidcDataBlockSize | 1 | [ECUC_IEEE1722Tp_00053] |
| IEEE1722TpStreamIidcFractionNumber | 1 | [ECUC_IEEE1722Tp_00054] |
| IEEE1722TpStreamIidcSourcePacketHeader | 1 | [ECUC_IEEE1722Tp_00055] |
| IEEE1722TpStreamIidcStreamFormat | 1 | [ECUC_IEEE1722Tp_00056] |
| IEEE1722TpStreamIidcSy | 1 | [ECUC_IEEE1722Tp_00059] |
| IEEE1722TpStreamIidcTag | 1 | [ECUC_IEEE1722Tp_00050] |
| IEEE1722TpStreamIidcTCode | 1 | [ECUC_IEEE1722Tp_00052] |

| **No Included Containers** |
|---|

⌋

## [ECUC_IEEE1722Tp_00051] Definition of EcucIntegerParamDef IEEE1722Tp StreamIidcChannel

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIidcChannel | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IIDC and IEC61883 channel.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 31 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00053] Definition of EcucIntegerParamDef IEEE1722Tp StreamIdcDataBlockSize

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIdcDataBlockSize | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IEC61883 data block size (DBS).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00054] Definition of EcucIntegerParamDef IEEE1722Tp StreamIdcFractionNumber

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIdcFractionNumber | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IEC61883 fractionNumber (FN).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00055] Definition of EcucBooleanParamDef IEEE1722Tp StreamIdcSourcePacketHeader

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamIdcSourcePacketHeader |
| **Parent Container** | IEEE1722TpStreamIIDC |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Defines the IEC61883 source packet header (SPH).<br>true = 1<br>false = 0<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00056] Definition of EcucIntegerParamDef IEEE1722Tp StreamIidcStreamFormat

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIidcStreamFormat | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IEC61883 stream format (FMT).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 63 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00059] Definition of EcucIntegerParamDef IEEE1722Tp StreamIidcSy

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIidcSy | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IIDC and IEC61883 sy.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 15 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00050] Definition of EcucIntegerParamDef IEEE1722Tp StreamIidcTag

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIidcTag | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IIDC and IEC61883 tag.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 3 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00052] Definition of EcucIntegerParamDef IEEE1722Tp StreamIidcTCode

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamIidcTCode | |
| **Parent Container** | IEEE1722TpStreamIIDC | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the IIDC and IEC61883 tcode.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 31 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.1.4.8 IEEE1722TpStreamRVF

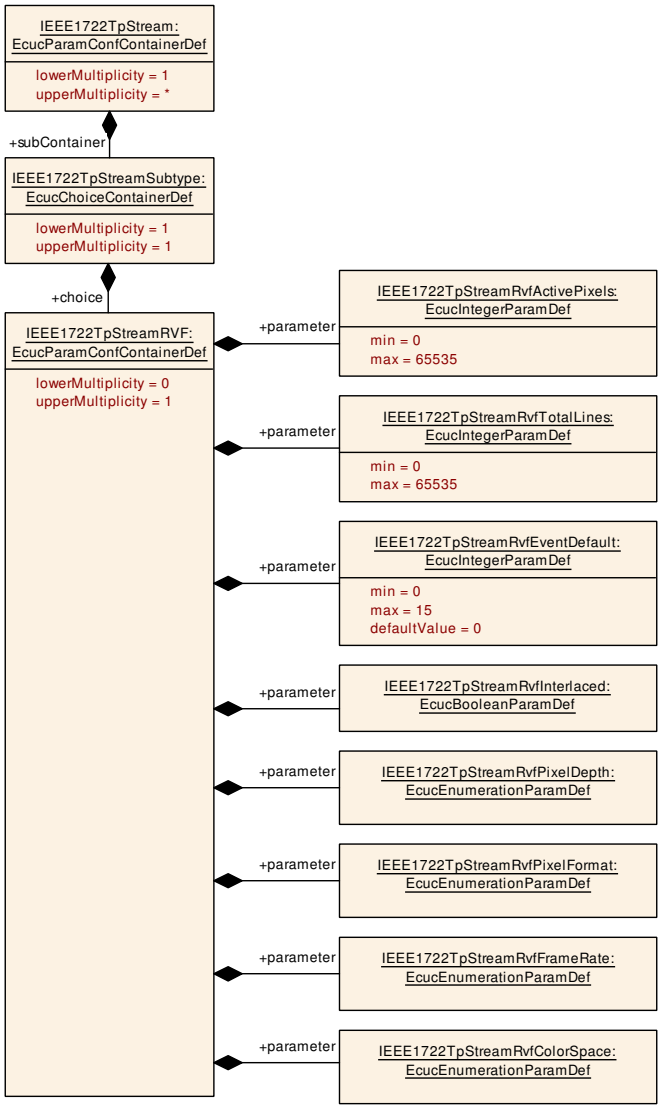


**Figure 10.11: IEEE1722TpStreamRVF**

## [ECUC_IEEE1722Tp_00013] Definition of EcucParamConfContainerDef IEEE1722TpStreamRVF

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamRVF |
| **Parent Container** | IEEE1722TpStreamSubtype |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of an IEEE1722 Raw Video Format (RVF) stream.<br>**Tags:** atp.Status=draft |

▽

△

| | |
|---|---|
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamRvfActivePixels | 1 | [ECUC_IEEE1722Tp_00064] |
| IEEE1722TpStreamRvfColorSpace | 1 | [ECUC_IEEE1722Tp_00071] |
| IEEE1722TpStreamRvfEventDefault | 1 | [ECUC_IEEE1722Tp_00066] |
| IEEE1722TpStreamRvfFrameRate | 1 | [ECUC_IEEE1722Tp_00070] |
| IEEE1722TpStreamRvfInterlaced | 1 | [ECUC_IEEE1722Tp_00067] |
| IEEE1722TpStreamRvfPixelDepth | 1 | [ECUC_IEEE1722Tp_00068] |
| IEEE1722TpStreamRvfPixelFormat | 1 | [ECUC_IEEE1722Tp_00069] |
| IEEE1722TpStreamRvfTotalLines | 1 | [ECUC_IEEE1722Tp_00065] |

| **No Included Containers** |
|---|

⌋

### [ECUC_IEEE1722Tp_00064] Definition of EcucIntegerParamDef IEEE1722TpStreamRvfActivePixels

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfActivePixels | |
| **Parent Container** | IEEE1722TpStreamRVF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the RVF stream active_pixels.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00071] Definition of EcucEnumerationParamDef IEEE1722TpStreamRvfColorSpace

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfColorSpace |
| **Parent Container** | IEEE1722TpStreamRVF |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Definition of the RVF stream colorspace.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | BT_Rec_601 | value=0x7, BT Rec.601<br>**Tags:** atp.Status=draft |
| | BT_Rec_709 | value=0x8, BT Rec.709<br>**Tags:** atp.Status=draft |
| | Grayscale | value=0x4, Grayscale<br>**Tags:** atp.Status=draft |
| | ITU_BT_2020 | value=0x9, ITU BT 2020<br>**Tags:** atp.Status=draft |
| | User | value=0xF, User defined<br>**Tags:** atp.Status=draft |
| | XYZ | value=0x5, XYZ<br>**Tags:** atp.Status=draft |
| | YCM | value=0x6, YCM<br>**Tags:** atp.Status=draft |
| | YCbCr | value=0x1, YCbCr<br>**Tags:** atp.Status=draft |
| | YCgCo | value=0x3, YCgCo<br>**Tags:** atp.Status=draft |
| | sRGB | value=0x2, sRGB<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00066] Definition of EcucIntegerParamDef IEEE1722Tp StreamRvfEventDefault

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfEventDefault | |
| **Parent Container** | IEEE1722TpStreamRVF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the RVF stream event (evt) default value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 15 | |
| **Default value** | 0 | |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_IEEE1722Tp_00070] Definition of EcucEnumerationParamDef IEEE1722 TpStreamRvfFrameRate

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfFrameRate | |
| **Parent Container** | IEEE1722TpStreamRVF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the RVF stream frame_rate.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | FR_1 | value=0x01, 1<br>**Tags:** atp.Status=draft |
| | FR_10 | value=0x10, 10<br>**Tags:** atp.Status=draft |
| | FR_100 | value=0x30, 100<br>**Tags:** atp.Status=draft |
| | FR_120 | value=0x31, 120<br>**Tags:** atp.Status=draft |
| | FR_15 | value=0x11, 15<br>**Tags:** atp.Status=draft |
| | FR_150 | value=0x32, 150<br>**Tags:** atp.Status=draft |
| | FR_2 | value=0x02, 2<br>**Tags:** atp.Status=draft |
| | FR_20 | value=0x12, 20<br>**Tags:** atp.Status=draft |
| | FR_200 | value=0x33, 200<br>**Tags:** atp.Status=draft |
| | FR_24 | value=0x13, 24<br>**Tags:** atp.Status=draft |
| | FR_240 | value=0x34, 240<br>**Tags:** atp.Status=draft |
| | FR_25 | value=0x14, 25<br>**Tags:** atp.Status=draft |
| | FR_30 | value=0x15, 30<br>**Tags:** atp.Status=draft |
| | FR_300 | value=0x35, 300<br>**Tags:** atp.Status=draft |
| | FR_48 | value=0x16, 48<br>**Tags:** atp.Status=draft |
| | FR_5 | value=0x03, 5<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| | FR_50 | value=0x17, 50<br>**Tags:** atp.Status=draft |
| | FR_60 | value=0x18, 60<br>**Tags:** atp.Status=draft |
| | FR_72 | value=0x19, 72<br>**Tags:** atp.Status=draft |
| | FR_85 | value=0x1A, 85<br>**Tags:** atp.Status=draft |
| | FR_User | value=0xFF, User defined<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00067] Definition of EcucBooleanParamDef IEEE1722Tp StreamRvfInterlaced

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfInterlaced | |
| **Parent Container** | IEEE1722TpStreamRVF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Defines the RVF stream interlaced (i).<br>true = 1<br>false = 0<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00068] Definition of EcucEnumerationParamDef IEEE1722 TpStreamRvfPixelDepth

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfPixelDepth |
| **Parent Container** | IEEE1722TpStreamRVF |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of the RVF stream pixel_depth.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucEnumerationParamDef |

▽

△

| | | |
|---|---|---|
| **Range** | PD_10 | value=0x02, 10<br>**Tags:** atp.Status=draft |
| | PD_12 | value=0x03, 12<br>**Tags:** atp.Status=draft |
| | PD_16 | value=0x04, 16<br>**Tags:** atp.Status=draft |
| | PD_8 | value=0x01, 8<br>**Tags:** atp.Status=draft |
| | User | value=0x0F, User defined<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00069] Definition of EcucEnumerationParamDef IEEE1722TpStreamRvfPixelFormat

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfPixelFormat | |
| **Parent Container** | IEEE1722TpStreamRVF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the RVF stream pixel_format.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | PF_4_1_1 | value=0x1, 4:1:1<br>**Tags:** atp.Status=draft |
| | PF_4_2_0 | value=0x2, 4:2:0<br>**Tags:** atp.Status=draft |
| | PF_4_2_2 | value=0x3, 4:2:2<br>**Tags:** atp.Status=draft |
| | PF_4_2_2_4 | value=0x6, 4:2:2:4<br>**Tags:** atp.Status=draft |
| | PF_4_4_4 | value=0x4, 4:4:4<br>**Tags:** atp.Status=draft |
| | PF_4_4_4_4 | value=0x7, 4:4:4:4<br>**Tags:** atp.Status=draft |
| | PF_Bayer_bggr | value=0xA, Bayer bggr<br>**Tags:** atp.Status=draft |
| | PF_Bayer_gbrg | value=0xB, Bayer gbrg<br>**Tags:** atp.Status=draft |
| | PF_Bayer_grbg | value=0x8, Bayer grbg<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| | PF_Bayer_rggb | value=0x9, Bayer rggb<br>**Tags:** atp.Status=draft |
| | PF_Monochrome | value=0x0, Monochrome<br>**Tags:** atp.Status=draft |
| | PF_User | value=0xF, User defined<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00065] Definition of EcucIntegerParamDef IEEE1722Tp StreamRvfTotalLines

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamRvfTotalLines | |
| **Parent Container** | IEEE1722TpStreamRVF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the RVF stream total_lines.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.1.4.9 IEEE1722TpStreamACF

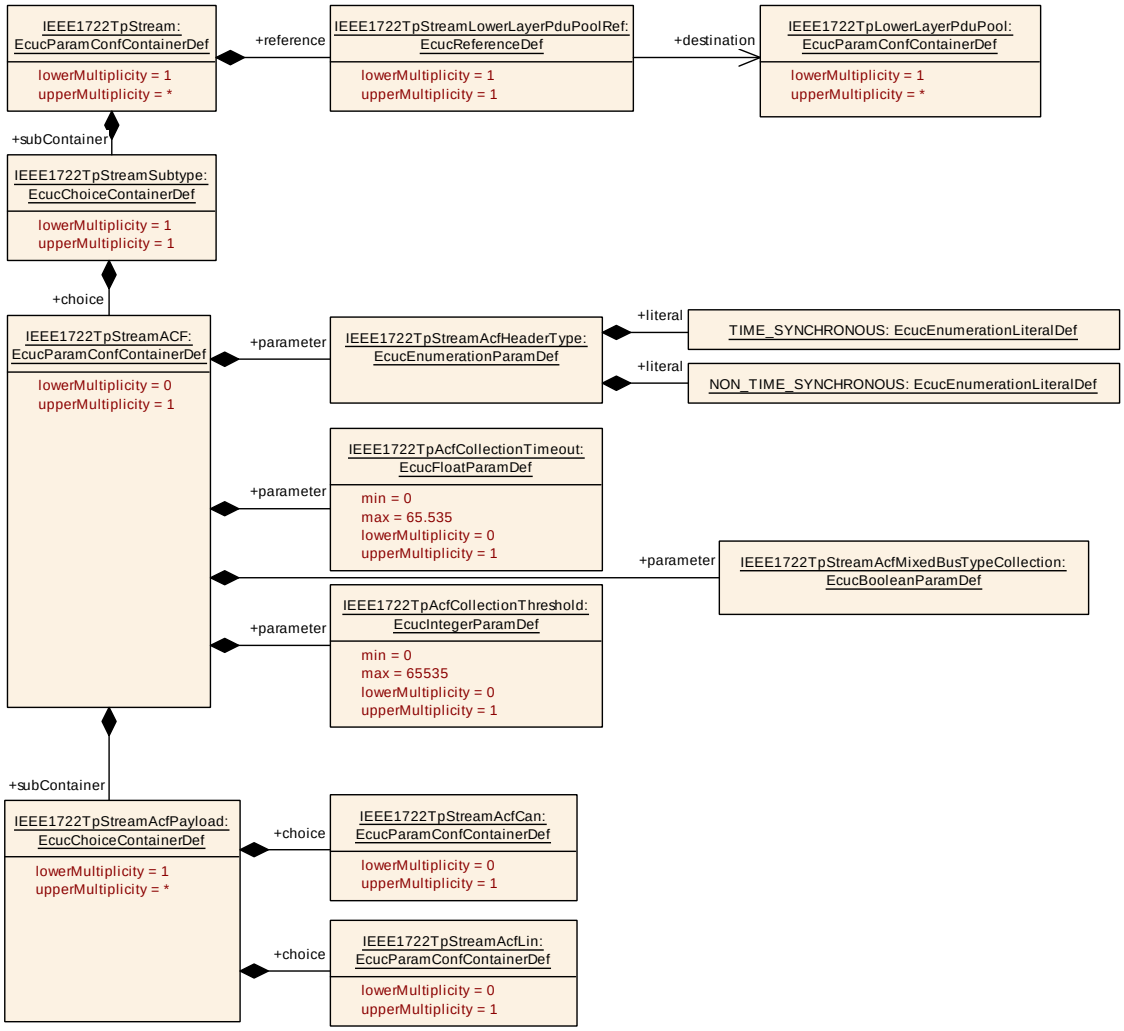


**Figure 10.12: IEEE1722TpStreamACF**

## [ECUC_IEEE1722Tp_00014] Definition of EcucParamConfContainerDef IEEE1722TpStreamACF

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamACF |
| **Parent Container** | IEEE1722TpStreamSubtype |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of an IEEE1722 AVTP Control Format (ACF) stream.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpAcfCollectionThreshold | 0..1 | [ECUC_IEEE1722Tp_00094] |
| IEEE1722TpAcfCollectionTimeout | 0..1 | [ECUC_IEEE1722Tp_00093] |
| IEEE1722TpStreamAcfHeaderType | 1 | [ECUC_IEEE1722Tp_00072] |
| IEEE1722TpStreamAcfMixedBusTypeCollection | 1 | [ECUC_IEEE1722Tp_00110] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfPayload | 1..* | Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_IEEE1722Tp_00094] Definition of EcucIntegerParamDef IEEE1722TpAcfCollectionThreshold

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpAcfCollectionThreshold | |
| **Parent Container** | IEEE1722TpStreamACF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Defines the size threshold in bytes which, when exceeded, triggers the sending of the ACF message.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_IEEE1722Tp_00093] Definition of EcucFloatParamDef IEEE1722TpAcfCollectionTimeout

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpAcfCollectionTimeout |
| **Parent Container** | IEEE1722TpStreamACF |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Defines a timeout which, when exceeded, triggers the sending of the ACF message.<br>Defined in seconds.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | EcucFloatParamDef |

▽

△

| | | |
|---|---|---|
| **Range** | [0 .. 65.535] | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00072] Definition of EcucEnumerationParamDef IEEE1722TpStreamAcfHeaderType

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfHeaderType | |
| **Parent Container** | IEEE1722TpStreamACF | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the ACF stream header format.<br>Depending on this selection the AVTP stream data subtype will be defined.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | NON_TIME_SYNCHRONOUS | Defines the ACF stream to use the Non-Time-Synchronous Control Format header (NTSCF).<br>This defines the AVTP stream data subtype to be 0x82.<br>**Tags:** atp.Status=draft |
| | TIME_SYNCHRONOUS | Defines the ACF stream to use the Time-Synchronous Control Format header (TSCF).<br>This defines the AVTP stream data subtype to be 0x05.<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00110] Definition of EcucBooleanParamDef IEEE1722TpStreamAcfMixedBusTypeCollection

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfMixedBusTypeCollection |
| **Parent Container** | IEEE1722TpStreamACF |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |

▽

△

| | |
|---|---|
| **Description** | Defines if this ACF-stream is allowed to collect ACF-messages of different bus kinds (i.e. whether it is allowed to collect CAN and LIN ACF-messages in one ACF-stream message).<br>For the ACF-stream producer this configures the collection behavior.<br>For an ACF-stream consumer this configures that this ACF-stream was produced with this assumption.<br>true = 1: mixed collection is allowed<br>false = 0 : mixed collection is not allowed<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucBooleanParamDef |
| **Default value** | – |
| **Pre-Compile only** | true |

⌋

## [ECUC_IEEE1722Tp_00073] Definition of EcucChoiceContainerDef IEEE1722Tp StreamAcfPayload

*Status:* DRAFT

⌈

| | |
|---|---|
| **Choice Container Name** | IEEE1722TpStreamAcfPayload |
| **Parent Container** | IEEE1722TpStreamACF |
| **Upstream mapping defined** | – |
| **Description** | Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1..* |
| **Pre-Compile only** | true |

| **No Included Parameters** |
|---|

| **Container Choices** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfCan | 0..1 | Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload for CAN and CAN_BRIEF.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamAcfLin | 0..1 | Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload for LIN.<br>ACF message type = 0x03.<br>**Tags:** atp.Status=draft |

⌋

### 10.1.4.10 IEEE1722TpStreamAcfCan

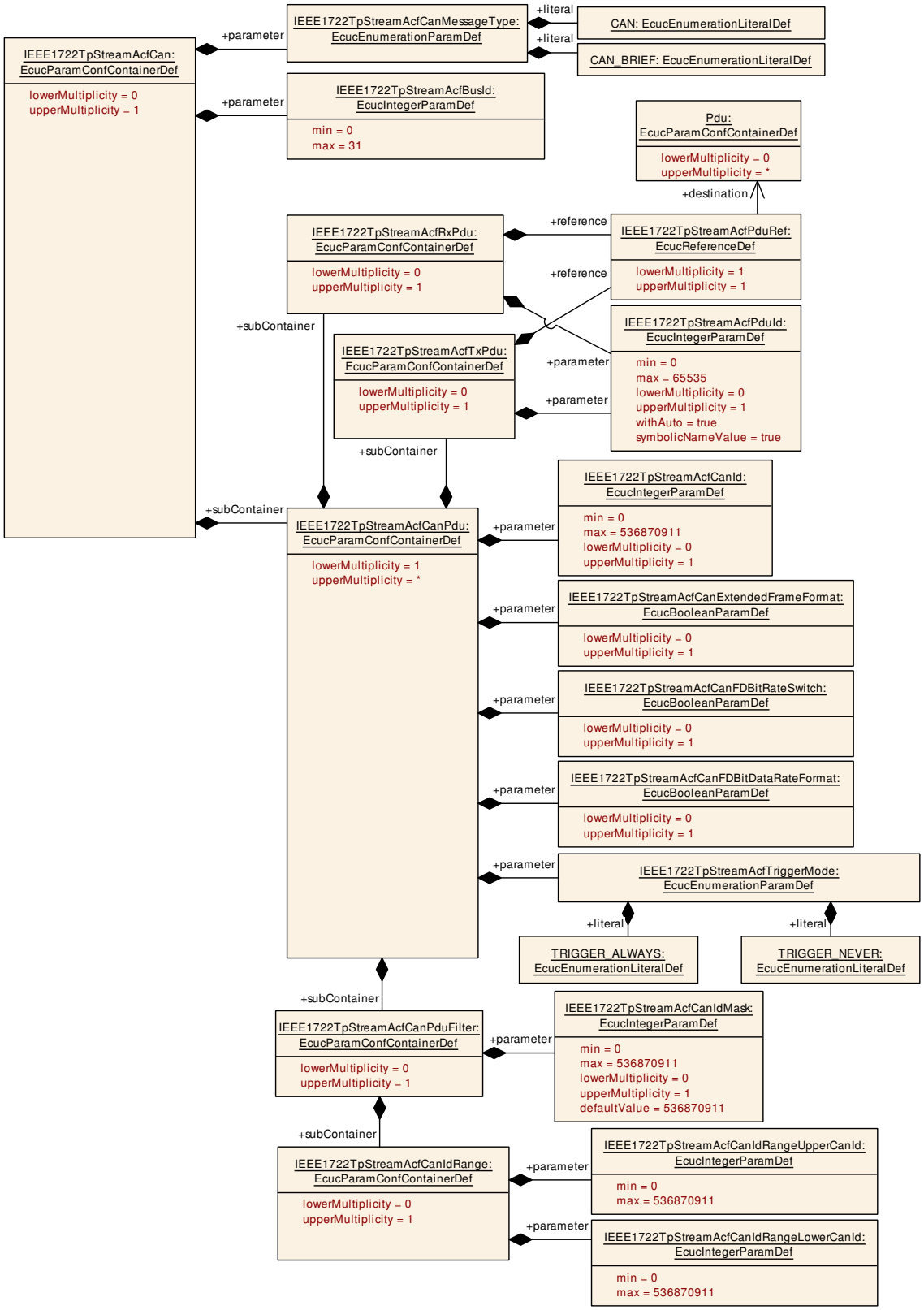


**Figure 10.13: IEEE1722TpStreamAcfCan**

### [ECUC_IEEE1722Tp_00074] Definition of EcucParamConfContainerDef IEEE1722TpStreamAcfCan

*Status:* DRAFT

⌈

| Container Name | IEEE1722TpStreamAcfCan |
|---|---|
| Parent Container | IEEE1722TpStreamAcfPayload |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] |
| Description | Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload for CAN and CAN_BRIEF.<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Pre-Compile only | true |
| Configuration Parameters | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfBusId | 1 | [ECUC_IEEE1722Tp_00078] |
| IEEE1722TpStreamAcfCanMessageType | 1 | [ECUC_IEEE1722Tp_00081] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfCanPdu | 1..* | Definition of a CAN Pdu transported on this ACF stream. Identification can be done by either explicit CAN Id or via meta-data.<br>**Tags:** atp.Status=draft |

⌋

### [ECUC_IEEE1722Tp_00078] Definition of EcucIntegerParamDef IEEE1722TpStreamAcfBusId

*Status:* DRAFT

⌈

| Parameter Name | IEEE1722TpStreamAcfBusId | |
|---|---|---|
| Parent Container | IEEE1722TpStreamAcfCan, IEEE1722TpStreamAcfLin | |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] | |
| Description | Definition of the ACF stream bus_id.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 31 | |
| Default value | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00081] Definition of EcucEnumerationParamDef IEEE1722TpStreamAcfCanMessageType

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfCanMessageType | |
| **Parent Container** | IEEE1722TpStreamAcfCan | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the ACF CAN stream message type.<br>Depending on this selection the ACF stream acf_msg_type will be defined.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | CAN | Defines the ACF CAN stream to use the ACF_CAN message type.<br>Controller Area Network (CAN)/CAN with Flexible Data-Rate (CAN FD) message.<br>This defines the ACF CAN stream acf_msg_type to be 0x01.<br>**Tags:** atp.Status=draft |
| | CAN_BRIEF | Defines the ACF CAN stream to use the ACF_CAN_BRIEF message type.<br>Abbreviated CAN/CAN FD message.<br>This defines the ACF CAN stream acf_msg_type to be 0x02.<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00082] Definition of EcucParamConfContainerDef IEEE1722TpStreamAcfCanPdu

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamAcfCanPdu |
| **Parent Container** | IEEE1722TpStreamAcfCan |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of a CAN Pdu transported on this ACF stream. Identification can be done by either explicit CAN Id or via meta-data.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfCanExtendedFrameFormat | 0..1 | [ECUC_IEEE1722Tp_00077] |
| IEEE1722TpStreamAcfCanFDBitDataRateFormat | 0..1 | [ECUC_IEEE1722Tp_00099] |
| IEEE1722TpStreamAcfCanFDBitRateSwitch | 0..1 | [ECUC_IEEE1722Tp_00098] |
| IEEE1722TpStreamAcfCanId | 0..1 | [ECUC_IEEE1722Tp_00079] |
| IEEE1722TpStreamAcfTriggerMode | 1 | [ECUC_IEEE1722Tp_00095] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfCanPdu Filter | 0..1 | Optional definition of a Can Id filter. This specifically applies to the case where the Can Id is taken from the meta-data and only certain PDUs shall actually pass.<br>If the definition is from Can network to stream, then this filter selects which PDUs shall be put to the stream.<br>If the definition is from stream to Can network, then this filter selects which PDUs shall be put on the Can network.<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamAcfRxPdu | 0..1 | This container defines the EcuC Rx Pdu used for the transport of the network specific payload (Can or Lin).<br>Dependency: Either IEEE1722TpStreamAcfTxPdu or IEEE1722 TpStreamAcfRxPdu shall be provided.<br>Supported MetaData entries for CAN:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• MESSAGE_TIMESTAMP_64<br>• MESSAGE_TIMESTAMP_VALID_8<br>• CAN_ID_32<br>• CAN_ID_PROPS_8<br>Supported MetaData entries for LIN:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• MESSAGE_TIMESTAMP_64<br>• MESSAGE_TIMESTAMP_VALID_8<br>• LIN_NAD_8<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamAcfTxPdu | 0..1 | This container defines the EcuC Tx Pdu used for the transport of the network specific payload (Can or Lin).<br>Dependency: Either IEEE1722TpStreamAcfTxPdu or IEEE1722 TpStreamAcfRxPdu shall be provided.<br>Supported MetaData entries for CAN:<br>• CAN_ID_32<br>• CAN_ID_PROPS_8<br>Supported MetaData entries for LIN:<br>• LIN_NAD_8<br>**Tags:** atp.Status=draft |

⌋

### [ECUC_IEEE1722Tp_00077] Definition of EcucBooleanParamDef IEEE1722Tp StreamAcfCanExtendedFrameFormat

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfCanExtendedFrameFormat | |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Defines the ACF CAN stream extended_frame_format (eff).<br>true = 1<br>false = 0<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00099] Definition of EcucBooleanParamDef IEEE1722Tp StreamAcfCanFDBitDataRateFormat

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfCanFDBitDataRateFormat | |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Represents the CAN-FD Data-rate (FD) format (fdf) switch. This is used, when a PDU is produced by an upper layer and transferred as encapsulated ACF_CAN via an Ethernet network.<br>**Tags:** atp.Status=draft<br>Dependency: If this parameter is configured, than also IEEE1722TpStreamAcfCan FDBitRateSwitch shall be configured. | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00098] Definition of EcucBooleanParamDef IEEE1722Tp StreamAcfCanFDBitRateSwitch

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfCanFDBitRateSwitch |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Represents the CAN-FD bit rate switch (brs). This is used, if a PDU is produced by an upper layer and transferred as encapsulated ACF_CAN via an Ethernet network.<br>**Tags:** atp.Status=draft<br>Dependency: If this parameter is configured, than also IEEE1722TpStreamAcfCan FDBitDataRateFormat shall be configured. | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00079] Definition of EcucIntegerParamDef IEEE1722Tp StreamAcfCanId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfCanId | |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the ACF stream CAN Id in case CAN Id is not taken from meta-data.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 536870911 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00095] Definition of EcucEnumerationParamDef IEEE1722 TpStreamAcfTriggerMode

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfTriggerMode |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu, IEEE1722TpStreamAcfLinPdu |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Defines whether putting this Can/Lin Pdu to the ACF message<br>• always or<br>• never<br>triggers immediate sending of the ACF message.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucEnumerationParamDef |

▽

△

| | | |
|---|---|---|
| **Range** | TRIGGER_ALWAYS | PDU triggers the transmission of the ACF-stream message.<br>**Tags:** atp.Status=draft |
| | TRIGGER_NEVER | PDU does not trigger the transmission of the ACF-stream message, other collection criteria might trigger the transmission.<br>**Tags:** atp.Status=draft |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00085] Definition of EcucParamConfContainerDef IEEE1722TpStreamAcfTxPdu

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamAcfTxPdu |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu, IEEE1722TpStreamAcfLinPdu |
| **Upstream mapping defined** | – |
| **Description** | This container defines the EcuC Tx Pdu used for the transport of the network specific payload (Can or Lin).<br>Dependency: Either IEEE1722TpStreamAcfTxPdu or IEEE1722TpStreamAcfRxPdu shall be provided.<br>Supported MetaData entries for CAN:<br>• CAN_ID_32<br>• CAN_ID_PROPS_8<br>Supported MetaData entries for LIN:<br>• LIN_NAD_8<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfPduId | 0..1 | [ECUC_IEEE1722Tp_00087] |
| IEEE1722TpStreamAcfPduRef | 1 | [ECUC_IEEE1722Tp_00086] |

| **No Included Containers** |
|---|

⌋

For parameter table [ECUC_IEEE1722Tp_00087] IEEE1722TpStreamAcfPduId, see definition below container IEEE1722TpStreamAcfRxPdu.

For parameter table [ECUC_IEEE1722Tp_00086] IEEE1722TpStreamAcfPduRef, see definition below container IEEE1722TpStreamAcfRxPdu.

### [ECUC_IEEE1722Tp_00111] Definition of EcucParamConfContainerDef IEEE1722TpStreamAcfRxPdu

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamAcfRxPdu |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu, IEEE1722TpStreamAcfLinPdu |
| **Upstream mapping defined** | – |
| **Description** | This container defines the EcuC Rx Pdu used for the transport of the network specific payload (Can or Lin).<br>Dependency: Either IEEE1722TpStreamAcfTxPdu or IEEE1722TpStreamAcfRxPdu shall be provided.<br>Supported MetaData entries for CAN:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• MESSAGE_TIMESTAMP_64<br>• MESSAGE_TIMESTAMP_VALID_8<br>• CAN_ID_32<br>• CAN_ID_PROPS_8<br>Supported MetaData entries for LIN:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• MESSAGE_TIMESTAMP_64<br>• MESSAGE_TIMESTAMP_VALID_8<br>• LIN_NAD_8<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfPduId | 0..1 | [ECUC_IEEE1722Tp_00087] |
| IEEE1722TpStreamAcfPduRef | 1 | [ECUC_IEEE1722Tp_00086] |

| **No Included Containers** |
|---|

⌋

### [ECUC_IEEE1722Tp_00087] Definition of EcucIntegerParamDef IEEE1722TpStreamAcfPduId

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfPduId |
| **Parent Container** | IEEE1722TpStreamAcfRxPdu, IEEE1722TpStreamAcfTxPdu |
| **Upstream mapping defined** | – |

▽

△

| | | |
|---|---|---|
| **Description** | Definition of the Handle Pdu Id.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **With Auto** | true | |
| **Pre-Compile only** | true | |

⌋

## [ECUC_IEEE1722Tp_00086] Definition of EcucReferenceDef IEEE1722TpStream AcfPduRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | IEEE1722TpStreamAcfPduRef | |
| **Parent Container** | IEEE1722TpStreamAcfRxPdu, IEEE1722TpStreamAcfTxPdu | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Reference to the EcuC Pdu.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to Pdu | |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_IEEE1722Tp_00084] Definition of EcucParamConfContainerDef IEEE1722 TpStreamAcfCanPduFilter

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | IEEE1722TpStreamAcfCanPduFilter | |
| **Parent Container** | IEEE1722TpStreamAcfCanPdu | |
| **Upstream mapping defined** | – | |
| **Description** | Optional definition of a Can Id filter. This specifically applies to the case where the Can Id is taken from the meta-data and only certain PDUs shall actually pass.<br>If the definition is from Can network to stream, then this filter selects which PDUs shall be put to the stream.<br>If the definition is from stream to Can network, then this filter selects which PDUs shall be put on the Can network.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfCanIdMask | 0..1 | [ECUC_IEEE1722Tp_00083] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfCanId Range | 0..1 | Identifier range definition. This parameter defines a CAN Identifier range in an alternative way to IEEE1722TpStreamAcf CanIdMask. |

⌋

### [ECUC_IEEE1722Tp_00083] Definition of EcucIntegerParamDef IEEE1722Tp StreamAcfCanIdMask

*Status:* DRAFT

⌈

| **Parameter Name** | IEEE1722TpStreamAcfCanIdMask | |
|---|---|---|
| **Parent Container** | IEEE1722TpStreamAcfCanPduFilter | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Identifier mask which denotes relevant bits in the CAN Identifier. This parameter defines a CAN Identifier range in an alternative way to IEEE1722TpStreamAcfCanId Range. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier.<br>Range: 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 536870911 | |
| **Default value** | 536870911 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00088] Definition of EcucParamConfContainerDef IEEE1722 TpStreamAcfCanIdRange ⌈

| **Container Name** | IEEE1722TpStreamAcfCanIdRange | |
|---|---|---|
| **Parent Container** | IEEE1722TpStreamAcfCanPduFilter | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Identifier range definition. This parameter defines a CAN Identifier range in an alternative way to IEEE1722TpStreamAcfCanIdMask. | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfCanIdRangeLowerCanId | 1 | [ECUC_IEEE1722Tp_00090] |
| IEEE1722TpStreamAcfCanIdRangeUpperCanId | 1 | [ECUC_IEEE1722Tp_00089] |

| No Included Containers |
|---|

⌋

### [ECUC_IEEE1722Tp_00090] Definition of EcucIntegerParamDef IEEE1722Tp StreamAcfCanIdRangeLowerCanId ⌈

| Parameter Name | IEEE1722TpStreamAcfCanIdRangeLowerCanId | |
|---|---|---|
| Parent Container | IEEE1722TpStreamAcfCanIdRange | |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] | |
| Description | Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId. | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 536870911 | |
| Default value | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_IEEE1722Tp_00089] Definition of EcucIntegerParamDef IEEE1722Tp StreamAcfCanIdRangeUpperCanId ⌈

| Parameter Name | IEEE1722TpStreamAcfCanIdRangeUpperCanId | |
|---|---|---|
| Parent Container | IEEE1722TpStreamAcfCanIdRange | |
| Upstream mapping defined | [7, CP_TPS_SystemTemplate] | |
| Description | Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId. | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 536870911 | |
| Default value | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### 10.1.4.11 IEEE1722TpStreamAcfLin

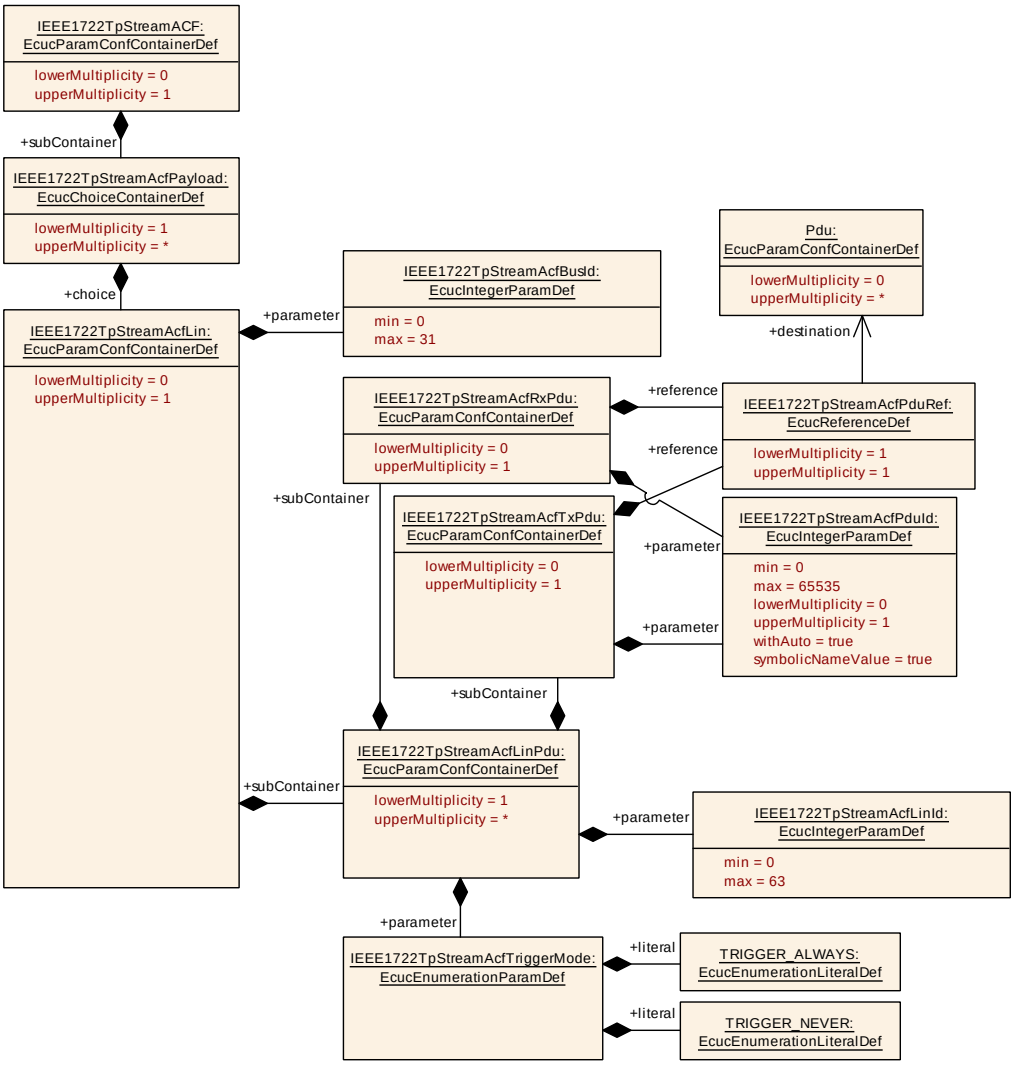


**Figure 10.14: IEEE1722TpStreamAcfLin**

## [ECUC_IEEE1722Tp_00076] Definition of EcucParamConfContainerDef IEEE1722TpStreamAcfLin

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamAcfLin |
| **Parent Container** | IEEE1722TpStreamAcfPayload |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload for LIN.<br>ACF message type = 0x03.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |

▽

△

| | |
|---|---|
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfBusId | 1 | [ECUC_IEEE1722Tp_00078] |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfLinPdu | 1..* | Definition of a LIN Pdu transported on this ACF stream. Identification can be done by either explicit LIN Id or via meta-data.<br>**Tags:** atp.Status=draft |

⌋

For parameter table [ECUC_IEEE1722Tp_00078] IEEE1722TpStreamAcfBusId, see definition below container IEEE1722TpStreamAcfCan.

### [ECUC_IEEE1722Tp_00096] Definition of EcucParamConfContainerDef IEEE1722TpStreamAcfLinPdu

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | IEEE1722TpStreamAcfLinPdu |
| **Parent Container** | IEEE1722TpStreamAcfLin |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] |
| **Description** | Definition of a LIN Pdu transported on this ACF stream. Identification can be done by either explicit LIN Id or via meta-data.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1..* |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| IEEE1722TpStreamAcfLinId | 1 | [ECUC_IEEE1722Tp_00080] |
| IEEE1722TpStreamAcfTriggerMode | 1 | [ECUC_IEEE1722Tp_00095] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| IEEE1722TpStreamAcfRxPdu | 0..1 | This container defines the EcuC Rx Pdu used for the transport of the network specific payload (Can or Lin).<br>Dependency: Either IEEE1722TpStreamAcfTxPdu or IEEE1722 TpStreamAcfRxPdu shall be provided.<br>Supported MetaData entries for CAN:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• MESSAGE_TIMESTAMP_64<br>• MESSAGE_TIMESTAMP_VALID_8<br>• CAN_ID_32<br>• CAN_ID_PROPS_8<br>Supported MetaData entries for LIN:<br>• IEEE1722TP_COMMON_STREAM_HEADER_PTR<br>• MESSAGE_TIMESTAMP_64<br>• MESSAGE_TIMESTAMP_VALID_8<br>• LIN_NAD_8<br>**Tags:** atp.Status=draft |
| IEEE1722TpStreamAcfTxPdu | 0..1 | This container defines the EcuC Tx Pdu used for the transport of the network specific payload (Can or Lin).<br>Dependency: Either IEEE1722TpStreamAcfTxPdu or IEEE1722 TpStreamAcfRxPdu shall be provided.<br>Supported MetaData entries for CAN:<br>• CAN_ID_32<br>• CAN_ID_PROPS_8<br>Supported MetaData entries for LIN:<br>• LIN_NAD_8<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_IEEE1722Tp_00080] Definition of EcucIntegerParamDef IEEE1722Tp StreamAcfLinId

*Status:* DRAFT

⌈

| **Parameter Name** | IEEE1722TpStreamAcfLinId | |
|---|---|---|
| **Parent Container** | IEEE1722TpStreamAcfLinPdu | |
| **Upstream mapping defined** | [7, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the ACF stream Lin Id.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 63 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

For parameter table [ECUC_IEEE1722Tp_00095] IEEE1722TpStreamAcfTrigger Mode, see definition below container IEEE1722TpStreamAcfCanPdu.

# A Not applicable requirements

**[CP_SWS_IEEE1722Tp_NA_00999]**

*Upstream requirements:* SRS_BSW_00493, SRS_BSW_00491, SRS_BSW_00004, SRS_BSW_00159, SRS_BSW_00167, SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00171, SRS_BSW_00323, SRS_BSW_00337, SRS_BSW_00339, SRS_BSW_00344, SRS_BSW_00345, SRS_BSW_00369, SRS_BSW_00375, SRS_BSW_00380, SRS_BSW_00383, SRS_BSW_00384, SRS_BSW_00388, SRS_BSW_00389, SRS_BSW_00390, SRS_BSW_00392, SRS_BSW_00393, SRS_BSW_00395, SRS_BSW_00396, SRS_BSW_00397, SRS_BSW_00398, SRS_BSW_00399, SRS_BSW_00400, SRS_BSW_00402, SRS_BSW_00403, SRS_BSW_00405, SRS_BSW_00409, SRS_BSW_00416, SRS_BSW_00417, SRS_BSW_00419, SRS_BSW_00422, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_BSW_00429, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00437, SRS_BSW_00438, SRS_BSW_00451, SRS_BSW_00452, SRS_BSW_00458, SRS_BSW_00461, SRS_BSW_00467, SRS_BSW_00469, SRS_BSW_00470, SRS_BSW_00471, SRS_BSW_00472, SRS_BSW_00478, SRS_BSW_00488, SRS_BSW_00489, SRS_BSW_00490

⌈These requirements are not applicable to this specification.⌋

# B Change history of AUTOSAR traceable items

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

## B.1 Traceable item history of this document according to AUTOSAR Release R24-11

### B.1.1 Added Constraints in R24-11

none

### B.1.2 Changed Constraints in R24-11

| Number | Heading |
|---|---|
| [CP_SWS_IEEE1722Tp_CONSTR_00006] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00007] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00008] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00009] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00010] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00011] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00012] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00013] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00014] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00015] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00016] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00017] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00018] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00019] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00020] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00021] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00022] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00023] | |
| [CP_SWS_IEEE1722Tp_CONSTR_00024] | |

**Table B.1: Changed Constraints in R24-11**

### B.1.3 Deleted Constraints in R24-11

none

### B.1.4 Added Specification Items in R24-11

none

### B.1.5 Changed Specification Items in R24-11

| Number | Heading |
|---|---|
| [CP_SWS_IEEE1722Tp_00033] | |
| [CP_SWS_IEEE1722Tp_91035] | Definition of mandatory interfaces required by module IEEE1722Tp |
| [CP_SWS_IEEE1722Tp_91036] | Definition of imported datatypes of module IEEE1722Tp |

**Table B.2: Changed Specification Items in R24-11**

### B.1.6 Deleted Specification Items in R24-11

none

## B.2 Traceable item history of this document according to AUTOSAR Release R25-11

### B.2.1 Added Constraints in R25-11

| Number | Heading |
|---|---|
| [CP_SWS_IEEE1722Tp_CONSTR_00025] | IEEE1722TpEthIfClkUnitRef optional |

**Table B.3: Added Constraints in R25-11**

### B.2.2 Changed Constraints in R25-11

none

### B.2.3 Deleted Constraints in R25-11

none

### B.2.4 Added Specification Items in R25-11

none

### B.2.5 Changed Specification Items in R25-11

none

### B.2.6 Deleted Specification Items in R25-11

none

## B.3 Traceable item history of this document according to AUTOSAR Release R26-11

### B.3.1 Added Constraints in R26-11

none

### B.3.2 Changed Constraints in R26-11

none

### B.3.3 Deleted Constraints in R26-11

none

### B.3.4 Added Specification Items in R26-11

none

### B.3.5 Changed Specification Items in R26-11

| Number | Heading |
|---|---|
| [CP_SWS_IEEE1722Tp_00003] | |
| [CP_SWS_IEEE1722Tp_00010] | |
| [CP_SWS_IEEE1722Tp_00013] | |
| [CP_SWS_IEEE1722Tp_00024] | |
| [CP_SWS_IEEE1722Tp_00036] | |
| [CP_SWS_IEEE1722Tp_00043] | |
| [CP_SWS_IEEE1722Tp_00054] | |

▽

△

| Number | Heading |
|---|---|
| [CP_SWS_IEEE1722Tp_00058] | |
| [CP_SWS_IEEE1722Tp_00073] | |
| [CP_SWS_IEEE1722Tp_00102] | |
| [CP_SWS_IEEE1722Tp_91036] | Definition of imported datatypes of module IEEE1722Tp |

**Table B.4: Changed Specification Items in R26-11**

### B.3.6 Deleted Specification Items in R26-11

| Number | Heading |
|---|---|
| [CP_SWS_IEEE1722Tp_91024] | Definition of API function IEEE1722Tp_GetVersionInfo |
| [ECUC_IEEE1722Tp_00004] | Definition of EcucBooleanParamDef IEEE1722TpVersion InfoApi |

**Table B.5: Deleted Specification Items in R26-11**