Document Title	Specification of Global Time Synchronization over Ethernet
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	676
Document Classification	Standard

Document Status	Final
Part of AUTOSAR Release	4.2.2

Document Change History

Release	Changed by	Change Description
4.2.2	AUTOSAR Release Management	• <Bus>TSyn_SetTransmissionMode changed to return "void" • Call of StbM_BusSetGlobalTime() added - sequence diagrams corrected • 'const' added to input arguments passed by pointer
4.2.1	AUTOSAR Release Management	• Initial Release

Disclaimer

Advice for users

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

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

Table of Contents

1 Introduction and functional overview

The EthTSyn module handles the distribution of time information over Ethernet.

The Ethernet mechanism is based on existing PTP (precision time protocol) mechanisms that are described in standards like IEEE1588 (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems) and IEEE802.1AS (Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks).

IEEE802.1AS can be seen as a profile (or subset) for using IEEE1588.

However, neither IEEE1588 nor IEEE802.1AS has been developed considering automotive requirements. Currently an AVnu subgroup is working on an automotive profile of IEEE802.1AS, which is still ongoing.

Therefore, the time synchronization over Ethernet uses the current mechanisms as defined in IEEE802.1AS.

Automotive Ethernet networks deviate from commercial Ethernet networks in terms of the following items:

- Role and functions of ECUs is known and defined a priori
- The network is static, i.e. components like ECUs, switches and characteristics like cable length' don't change during 'operation' or even after switching off and switching on the vehicle. Components of course may be unavailable (due to failure situations or by purpose) but mostly only change when the vehicle is at a service facility.

Therefore, dynamic mechanisms like determining the Global Time Master (denoted as grandmaster in IEEE802.1AS) by the best master clock algorithm (BMCA) during operation are not required.
It is also possible to omit the cyclic measurement of link delays on Ethernet links due to the static nature of the automotive network and restrict mechanisms that belonging to dynamic network topology.

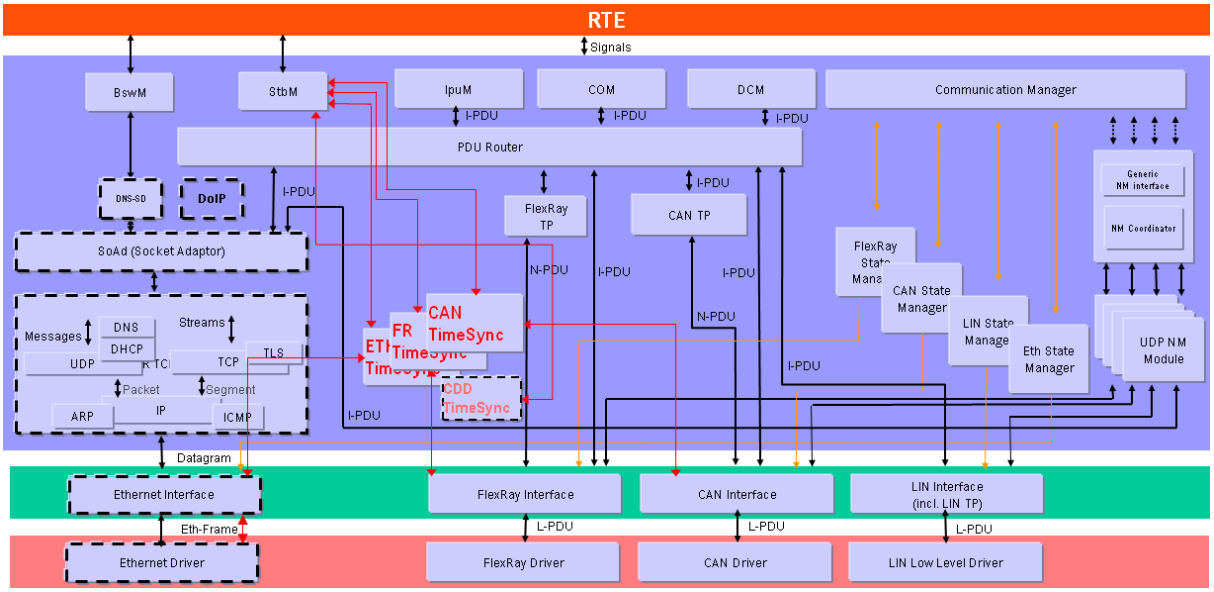


Figure 1: Clustering of TimeSync modules

2 Acronyms, Abbreviations and Definitions

This section lists module local Abbreviations and Definitions. For a complete set of Synchronized Time Base related Terms refer to the corresponding chapter in [3].

Abbreviation / Acronym:	Description
(G)TD	(Global) Time Domain
(G)TM	(Global)Time Master
<Bus>TSyn	A bus specific Time Synchronization Provider module
AVB	Audio Video Bridging
BMCA	Best Master Clock Algorithm
CRC	Cyclic Redundancy Checksum
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ETH	Ethernet
EthTSyn	Time Synchronization Provider module for Ethernet
Follow_Up message	Time adjustment message (Follow-Up)
GM(C)	Grand Master (Clock)
OFNS message	Offset adjustment message
OFS message	Offset synchronization message
Pdelay_Req message	Latency adjustment message (Request)
Pdelay_Resp message	Latency adjustment message (Response)
Pdelay_Resp_Follow_Up message	Latency adjustment message (Follow-Up)
PTP	Precision Time Protocol
StbM	Synchronized Time-Base Manager
Sync message	Time synchronization message
TG	Time Gateway
TS	Time Slave
TSD	Time Sub-domain

3 Related documentation

3.1 Input documents

[1] AUTOSAR Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] AUTOSAR General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

[3] AUTOSAR Synchronized Time Base Manager
AUTOSAR_SWS_SynchronizedTimeBaseManager.pdf

3.2 Related standards

[4] IEEE Standard 802.1AS™- 30 of March 2011
http://standards.ieee.org/getieee802/download/802.1AS-2011.pdf

3.3 Related specification

AUTOSAR provides a General Specification on Basic Software (SWS BSW General [2]) which is also valid for EthTSyn.

Thus, the General Specification on Basic Software (SWS BSW General) shall be considered additionally and as required specification for EthTSyn.

4 Constraints and assumptions

4.1 Limitations

1. Limited to Time Domain “0” (i.e. Global Time Domain)
   → EthTSynGlobalTimeDomainId (**ECUC_EthTSyn_00005 : (Obsolete)**) obsolete
2. No support of User Data within time synchronization messages
3. No support of BMCA protocol, like specified in [4].

Time Master and Time Slaves shall work with a Time Base reference clock accuracy as defined in“[4], ANNEX B.1.2 Time measurement granularity”.

The reception of a `Pdelay_Req` message is not taken as a pre-condition to start with the transmission `Sync` messages.

4.2 Applicability to car domains

Systems requiring a common Time Base to ECUs independent to which bus system the ECU is connected.

5 Dependencies to other modules

The Global Time Synchronization over Ethernet (EthTSyn) has interfaces towards the Synchronized Time Base Manager (StbM), the Ethernet Interface (EthIf), the Basic Software Mode Manager (BswM) and the Development Error Tracer (DET).

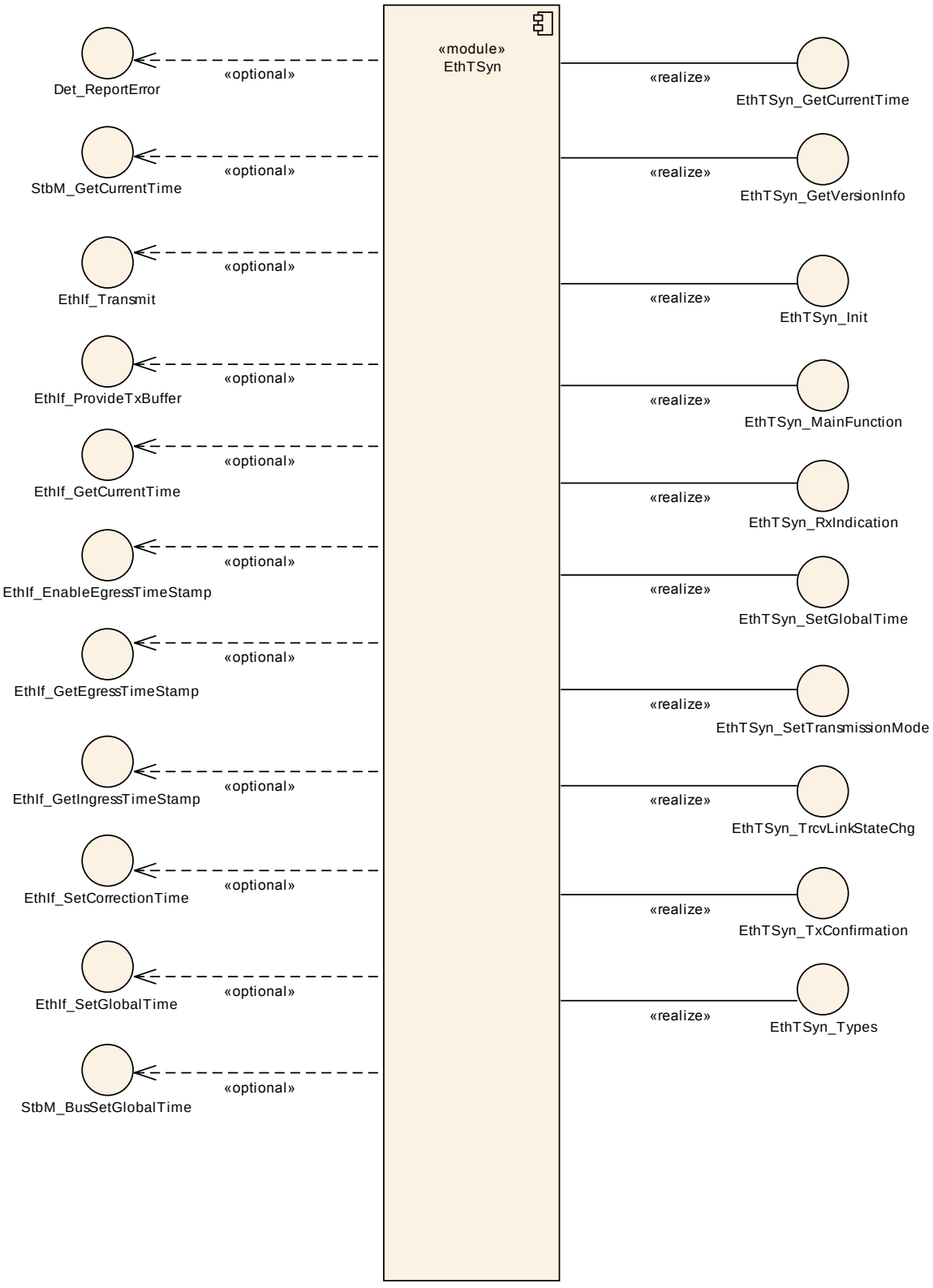


Figure 2: Module dependencies of the EthTSyn module

- StbM – Get and set the current time value
- EthIf – Receiving and transmitting messages
- BswM – Coordination of network access

- DET – Reporting of development errors

5.1 File structure

5.1.1 Code file structure

For details, refer to the section 5.1.6 "Code file structure" of the SWS BSW General [2].

5.1.2 Header file structure

Besides the files defined in section 5.1.7 “Header file structure” of the SWS BSW General [2], the Global Time Synchronization over Ethernet needs to include the files defined below.

[SWS_EthTSyn_00001]
[The implementation header files shall include *ComStack_Types.h*.
](SRS_BSW_00301, SRS_BSW_00456)

The following picture shows the include hierarchy of the Global Time Synchronization over Ethernet.

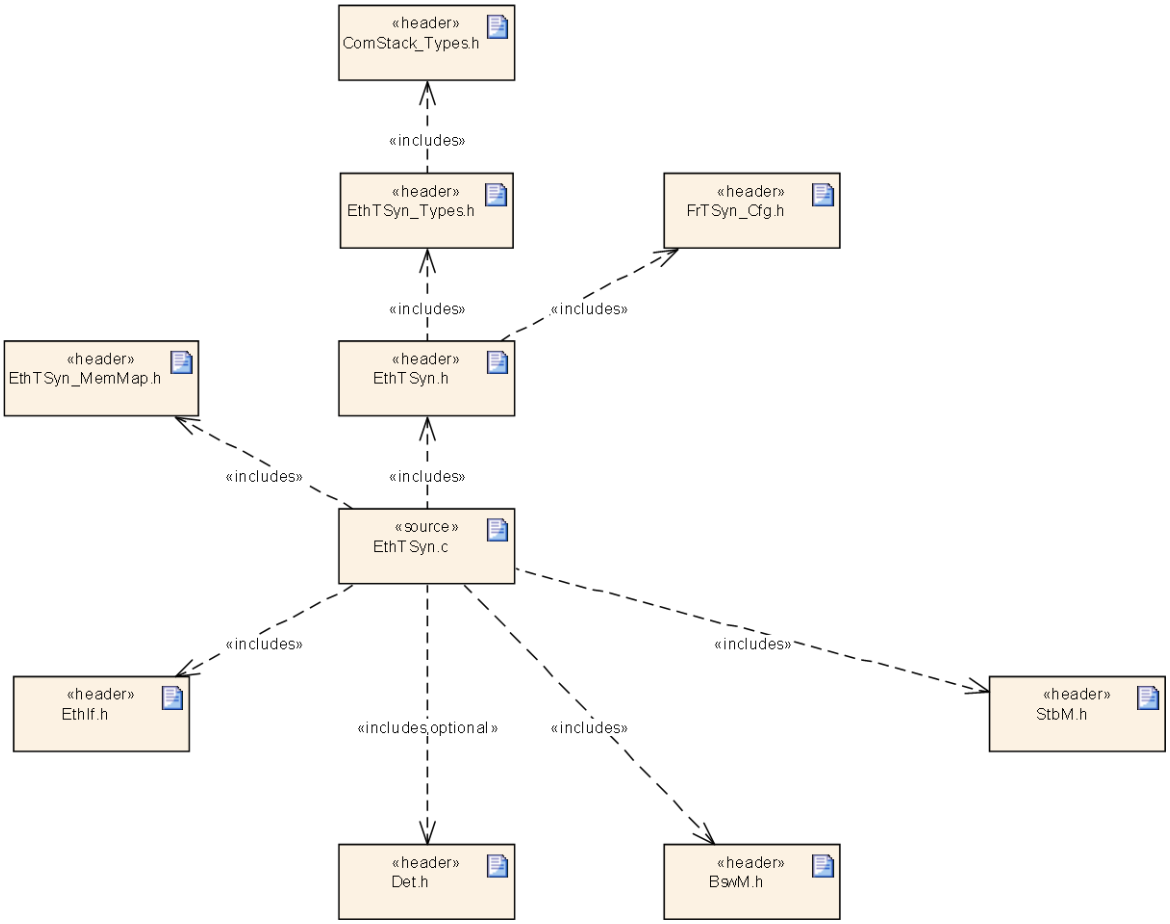


Figure 3: File structure of EthTSyn

6 Requirements traceability

Requirement	Description	Satisfied by
-	-	SWS_EthTSyn_00048
SRS_BSW_00301	All AUTOSAR Basic Software Modules shall only import the necessary information	SWS_EthTSyn_00001
SRS_BSW_00323	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	SWS_EthTSyn_00029
SRS_BSW_00337	Classification of development errors	SWS_EthTSyn_00007, SWS_EthTSyn_00041, SWS_EthTSyn_00046
SRS_BSW_00385	List possible error notifications	SWS_EthTSyn_00030
SRS_BSW_00456	- A Header file shall be defined in order to harmonize BSW Modules	SWS_EthTSyn_00001
SRS_StbM_20017	The StbM shall initialize the local time base if configured as Time Slave	SWS_EthTSyn_00006, SWS_EthTSyn_00008
SRS_StbM_20018	The StbM shall initialize the local time base with 0 at startup if configured as Time Slave	SWS_EthTSyn_00006, SWS_EthTSyn_00008
SRS_StbM_20019	The StbM shall initialize the global time base with a configurable startup value if configured as Time Master	SWS_EthTSyn_00006, SWS_EthTSyn_00008
SRS_StbM_20047	The Ethernet TimeSync module shall trigger time base synchronization transmission	SWS_EthTSyn_00016, SWS_EthTSyn_00050
SRS_StbM_20048	The Ethernet TimeSync module shall support IEEE 802.1AS	SWS_EthTSyn_00002, SWS_EthTSyn_00003, SWS_EthTSyn_00004, SWS_EthTSyn_00005, SWS_EthTSyn_00009, SWS_EthTSyn_00010, SWS_EthTSyn_00011, SWS_EthTSyn_00012, SWS_EthTSyn_00013, SWS_EthTSyn_00014, SWS_EthTSyn_00015, SWS_EthTSyn_00016, SWS_EthTSyn_00017, SWS_EthTSyn_00018, SWS_EthTSyn_00019, SWS_EthTSyn_00020, SWS_EthTSyn_00021, SWS_EthTSyn_00022, SWS_EthTSyn_00023, SWS_EthTSyn_00024, SWS_EthTSyn_00025, SWS_EthTSyn_00026, SWS_EthTSyn_00028, SWS_EthTSyn_00031, SWS_EthTSyn_00032, SWS_EthTSyn_00033, SWS_EthTSyn_00034, SWS_EthTSyn_00035,

		SWS_EthTSyn_00036, SWS_EthTSyn_00037, SWS_EthTSyn_00038, SWS_EthTSyn_00039, SWS_EthTSyn_00040, SWS_EthTSyn_00042, SWS_EthTSyn_00043, SWS_EthTSyn_00044, SWS_EthTSyn_00045, SWS_EthTSyn_00047, SWS_EthTSyn_00049, SWS_EthTSyn_00050, SWS_EthTSyn_00052
SRS_StbM_20049	The Ethernet TimeSync module shall provide access to the local time stamp unit in the Ethernet HW	SWS_EthTSyn_00009, SWS_EthTSyn_00011, SWS_EthTSyn_00012, SWS_EthTSyn_00013, SWS_EthTSyn_00015, SWS_EthTSyn_00016, SWS_EthTSyn_00017, SWS_EthTSyn_00018, SWS_EthTSyn_00024, SWS_EthTSyn_00026, SWS_EthTSyn_00049
SRS_StbM_20050	The Ethernet TimeSync module shall support rate correction	SWS_EthTSyn_00009, SWS_EthTSyn_00017, SWS_EthTSyn_00024
SRS_StbM_20051	The Ethernet TimeSync module shall detect and handle errors in synchronization protocol / communication	SWS_EthTSyn_00004, SWS_EthTSyn_00019, SWS_EthTSyn_00020, SWS_EthTSyn_00021, SWS_EthTSyn_00022, SWS_EthTSyn_00025, SWS_EthTSyn_00029
SRS_StbM_20052	The Ethernet TimeSync configuration shall allow the EthTSyn to support different roles for a time base	SWS_EthTSyn_00051

7 Functional specification

This chapter defines the behavior of the Global Time Synchronization over Ethernet. The API of the module is defined in chapter 8, while the configuration is defined in chapter 10.

7.1 Overview

The Global Time Synchronization over Ethernet is responsible to ensure the collection and distribution of synchronized time information across the Ethernet network. It interacts with the StbM and provides all Ethernet specific functions to the StbM.

[SWS_EthTSyn_00002]
⌈IEEE802.1AS [4] specifies default configuration values like the MAC destination address or Ethernet frame type. The EthTSyn shall use these default configuration values if not otherwise specified within this document.
⌋(SRS_StbM_20048)

[SWS_EthTSyn_00003]
⌈The EthTSyn module uses `Pdelay_Req`, `Pdelay_Resp`, `Pdelay_Resp_Follow_Up` for latency calculation.
⌋(SRS_StbM_20048)

[SWS_EthTSyn_00004]
⌈A `Pdelay_Res` timeout or incomplete `Pdelay` protocol shall stop the latency calculation algorithm. In such cases, the device shall use the latest successful calculated latency value.
⌋(SRS_StbM_20048, SRS_StbM_20051)

[SWS_EthTSyn_00005]
⌈All messages belonging to the IEEE Rapid Spanning Tree Protocol (`PortAnnounceReceive`, `PortAnnounceInformation`, `PortRoleSelection`, `PortAnnounceTransmit`) shall be ignored on the receiver side. Therefore, Time Master and Time Slave shall start their protocol state machines without announce-message recognition.
⌋(SRS_StbM_20048)

Note: Collaboration with devices that transmit (and receive) announce-messages is still possible, but not recommended by this standard, due to the fact, that those devices start their state machines in a different way.

7.2 Module Handling

This section contains description of auxiliary functionality of the Global Time Synchronization over Ethernet.

7.2.1 Initialization

The Global Time Synchronization over Ethernet is initialized via `EthTSyn_Init()`. Except for `EthTSyn_GetVersionInfo()` and `EthTSyn_Init()`, the API functions of the EthTSyn module may only be called when the module has been properly initialized.

[SWS_EthTSyn_00006]
⌈A call to `EthTSyn_Init()` initializes all internal variables and sets the EthTSyn module to the initialized state.
⌋(SRS_StbM_20017, SRS_StbM_20018, SRS_StbM_20019)

[SWS_EthTSyn_00007]
⌈When DET reporting is enabled (see `EthTSynDevErrorDetect`), the EthTSyn module shall call `Det_ReportError()` with the error code `ETHTSYN_E_NOT_INITIALIZED` when any API other than `EthTSyn_GetVersionInfo()` or `EthTSyn_Init()` is called in uninitialized state.
⌋(SRS_BSW_00337)

[SWS_EthTSyn_00008]
⌈When `EthTSyn_Init()` is called in initialized state, the EthTSyn module shall re-initialize its internal variables.
⌋(SRS_StbM_20017, SRS_StbM_20018, SRS_StbM_20019)

[SWS_EthTSyn_00009]
⌈When `EthTSyn_Init()` is called in initialized state, the EthTSyn module shall set the rate correction to 0.
⌋(SRS_StbM_20048, SRS_StbM_20049, SRS_StbM_20050)

[SWS_EthTSyn_00010]
⌈When `EthTSyn_Init()` is called in initialized state, the EthTSyn module shall set the latency for ingress and egress to 0.
⌋(SRS_StbM_20048)

7.2.2 Pdelay Protocol for Latency Calculation

[SWS_EthTSyn_00011]
⌈Time Master and Time Slave shall periodically transmit `Pdelay_Req` messages for latency calculation with the cycle `EthTSynGlobalTimeTxPdelayReqPeriod` (**ECUC_EthTSyn_00019 :**) as defined in [4] chapter 11.1.2 “Propagation delay measurement”. For that, the following sequence shall be applied:

1) Get a free transmission buffer via `EthIf_ProvideTxBuffer()`
2) Activate the time stamping via `EthIf_EnableEgressTimeStamp()` if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE
3) Trigger transmit request via `EthIf_Transmit()`

⌋(SRS_StbM_20048, SRS_StbM_20049)

[SWS_EthTSyn_00012]
⌈Time Master and Time Slave shall transmit `Pdelay_Resp` messages for latency calculation as defined in [4] chapter 11.1.2 “Propagation delay measurement”. For that, the following sequence shall be applied:

1) Get a free transmission buffer via `EthIf_ProvideTxBuffer()`
2) Activate the time stamping via `EthIf_EnableEgressTimeStamp()` if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE
3) Trigger transmit request via `EthIf_Transmit()`

⌋(SRS_StbM_20048, SRS_StbM_20049)

[SWS_EthTSyn_00013]
⌈On invocation of `EthTSyn_TxConfirmation()` the egress time stamp shall be retrieved for `Pdelay_Req` and `Pdelay_Resp` from the EthIf according to Figure 6, if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** is set to TRUE.

Otherwise the `responseOriginTimestamp` of `Pdelay_Resp_Follow_up` shall be retrieved from the StbM via `StbM_GetCurrentTime()` on egress of `Pdelay_Resp` and the local time stamping uses for T1 (`Pdelay_Req`) `StbM_GetCurrentTimeRaw()` and for (T3-T2) (`Pdelay_Resp`) `StbM_GetCurrentTimeDiff()`.
⌋(SRS_StbM_20048, SRS_StbM_20049)

[SWS_EthTSyn_00014]
⌈The transmission timestamp of messages defined in **[SWS_EthTSyn_00003]** shall be used in the transmission of the `Pdelay_Resp_Follow_Up` message as defined in [4] chapter 11.1.2 “Propagation delay measurement” considering `EthTSynGlobalTimeTxFollowUpOffset` as time offset between `Pdelay_Resp` and `Pdelay_Resp_Follow_Up`. For that, the following sequence shall be applied:

1) Get a free transmission buffer via `EthIf_ProvideTxBuffer()`
2) Trigger transmit request with the transmission timestamp of **[SWS_EthTSyn_00003]** via `EthIf_Transmit()`

⌋(SRS_StbM_20048)

[SWS_EthTSyn_00049]
⌈On invocation of `EthTSyn_RxIndication()` the time stamp shall be retrieved for `Pdelay_Req` and `Pdelay_Resp` from the EthIf according to Figure 7, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE.
Otherwise the `requestReceiptTimestamp` of `Pdelay_Resp` shall be retrieved from the StbM via `StbM_GetCurrentTime()` on ingress of `Pdelay_Req` and the local time stamping uses for T2 (`Pdelay_Req`) `StbM_GetCurrentTimeRaw()` and for (T4-T1) (`Pdelay_Resp`) `StbM_GetCurrentTimeDiff()`.
⌋(SRS_StbM_20048, SRS_StbM_20049)

7.3 Acting as Time Master

A Time Master is an entity which is the master for a certain Time Base and which propagates this Time Base to a set of time slaves within a certain segment of a communication network, being a source for this time base.
If a Time Master is also the owner of the Time Base then he is the Global Time master. A time gateway typically consists of one Time Slave and one or more time masters. When mapping time entities to real ECUs it has to be noted, that an ECU could be Time Master (or even Global Time Master) for one Time Base and Time Slave for another time base.

[SWS_EthTSyn_00015]
⌈On invocation of `EthTSyn_SetGlobalTime()` the EthTSyn module shall forward this time to the Ethernet hardware via `EthIf_SetGlobalTime()`.
⌋(SRS_StbM_20048, SRS_StbM_20049)

7.3.1 Message processing

[SWS_EthTSyn_00050]
⌈The EthTSyn module shall support the transmission of `Sync` and `Follow_Up` messages according [4] as well as the transmission and reception of `Pdelay_Req`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up` (**[SWS_EthTSyn_00003]**, **[SWS_EthTSyn_00004]**).
⌋(SRS_StbM_20047, SRS_StbM_20048)

[SWS_EthTSyn_00016]
⌈If configured as Time Master (`EthTSynGlobalTimeMaster`) the EthTSyn module shall periodically transmit `Sync` messages only on Time Domain 0 with the cycle `EthTSynGlobalTimeTxPeriod` (**ECUC_EthTSyn_00010 :**) as defined in [4] chapter 11.1.3 “Transport of time-synchronization information”. For that, the following sequence shall be applied:

1) Get a free transmission buffer via `EthIf_ProvideTxBuffer()`
2) Activate the time stamping via `EthIf_EnableEgressTimeStamp()` if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE
3) Trigger transmit request via `EthIf_Transmit()`

⌋(SRS_StbM_20047, SRS_StbM_20048, SRS_StbM_20049)

[SWS_EthTSyn_00017]
⌈On invocation of `EthTSyn_TxConfirmation()` the egress time stamp shall be retrieved for `Sync` from the EthIf according to Figure 6, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE.

Otherwise the `preciseOriginTimestamp` of `Follow_up` shall be retrieved from the StbM via `StbM_GetCurrentTime()` on egress of `Sync` and the local time stamping

uses on egress of the same message for `correctionField` calculation ($T_{s,i} – T_{r,i}$) `StbM_GetCurrentTimeDiff()`.
](SRS_StbM_20048, SRS_StbM_20049, SRS_StbM_20050)

[SWS_EthTSyn_00018]
[The transmission timestamp of **[SWS_EthTSyn_00017]** shall be used in the transmission of the `Follow_Up` message as defined in [4] chapter 11.1.3 “Transport of time-synchronization information” considering `EthTSynGlobalTimeTxFollowUpOffset` as time offset between `Sync` and `Follow_Up`. For that, the following sequence shall be applied:

1) Get a free transmission buffer via `EthIf_ProvideTxBuffer()`
2) Trigger transmit request with the transmission timestamp of **[SWS_EthTSyn_00017]** via `EthIf_Transmit()`

](SRS_StbM_20048, SRS_StbM_20049)

7.3.2 Link State and Transmission Mode

[SWS_EthTSyn_00019]
[A transceiver link state change (notification call of `EthTSyn_TrcvLinkStateChg()`) from `ETHTRCV_LINK_STATE_ACTIVE` to `ETHTRCV_LINK_STATE_DOWN` resets the state machines for transmission and reception of time synchronization messages.
](SRS_StbM_20048, SRS_StbM_20051)

[SWS_EthTSyn_00020]
[A transceiver link state change (notification call of `EthTSyn_TrcvLinkStateChg()`) from `ETHTRCV_LINK_STATE_DOWN` to `ETHTRCV_LINK_STATE_ACTIVE` (re-)starts the transmission and reception of time synchronization messages.
](SRS_StbM_20048, SRS_StbM_20051)

[SWS_EthTSyn_00021]
[If `EthTSyn_SetTransmissionMode()` is called and the parameter `Mode` equals `ETHTSYN_TX_OFF`, all transmit request from EthTSyn shall be omitted on this Ethernet controller.
](SRS_StbM_20048, SRS_StbM_20051)

[SWS_EthTSyn_00022]
[If `EthTSyn_SetTransmissionMode()` is called and the parameter `Mode` equals `ETHTSYN_TX_ON`, all transmit request from EthTSyn on this Ethernet controller shall be able to be transmitted.
](SRS_StbM_20048, SRS_StbM_20051)

7.4 Acting as Time Slave

A Time Slave is an entity, which is the recipient for a certain Time Base within a certain segment of a communication network, being a consumer for this time base.

7.4.1 Message processing

[SWS_EthTSyn_00023]
[The EthTSyn module shall support the reception of `Sync` and `Follow_Up` messages according [4] as well as the transmission and reception of `Pdelay_Req`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up` (**[SWS_EthTSyn_00003]**, **[SWS_EthTSyn_00004]**).
](SRS_StbM_20048)

[SWS_EthTSyn_00024]
[On invocation of `EthTSyn_RxIndication()` the time stamp shall be retrieved for `Sync` from the EthIf according to Figure 7, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE.

Otherwise for local time stamping on ingress of the `Sync` message for `correctionField` calculation of $T_{r,i}$ `StbM_GetCurrentTimeRaw()` shall be used.
](SRS_StbM_20048, SRS_StbM_20049, SRS_StbM_20050)

[SWS_EthTSyn_00025]
[For each configured Time Slave (`EthTSynGlobalTimeSlave`) the EthTSyn module shall observe the *reception timeout* `EthTSynGlobalTimeFollowUpTimeout` (**ECUC_EthTSyn_00007 :**) between the `Sync` and its `Follow_Up` message. If the *reception timeout* occurs the sequence shall be reset (i.e. waiting for a new Sync message).
](SRS_StbM_20048, SRS_StbM_20051)

Note: The general timeout monitoring for the Time Base update is located in the StbM and not in the provider modules.

[SWS_EthTSyn_00026]
[On reception of a valid Follow_Up message, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to TRUE, a time slave shall check the time difference between its local time and Global Time provided by that message. If the threshold `EthTSynTimeHardwareCorrectionThreshold` (**ECUC_EthTSyn_00014 :**) is exceeded, the time slave shall update the time in the Ethernet hardware via `EthIf_SetCorrectionTime()`](SRS_StbM_20048, SRS_StbM_20049)

[SWS_EthTSyn_00052]
[On reception of a valid Follow_Up message, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to FALSE, a time slave shall calculate the Global Time (from the time information provided by that message) and forward it to the StbM via `StbM_BusSetGlobalTime()`.] (SRS_StbM_20048)

7.5 Message Format

[SWS_EthTSyn_00028]
[The message format, etc. shall be derived from [4] chapter 10. “Media-independent layer specification” and chapter 11. “Media-dependent layer specification for full-

duplex, point-to-point links". The default values shall be used if not specified different in this document.
](SRS_StbM_20048)

7.6 Error Classification

This chapter lists and classifies all errors that can be detected by this software module. Each error is classified to relevance (development / production) and the related error code (unique label for the error). For development errors this table also specifies the unique values, which correspond to the error codes.

[SWS_EthTSyn_00029]
⌈On errors and exceptions, the EthTSyn module shall not modify its current module state but shall simply report the error event.
⌋(SRS_StbM_20051, SRS_BSW_00323)

[SWS_EthTSyn_00030]
⌈EthTSyn shall use following errors:

Type or error	*Relevance*	*Related error code*	*Valu [hex]*
API service used in un-initialized state	Development	`ETHTSYN_E_NOT_INITIALIZED`	`0x20`

⌋(SRS_BSW_00385)

7.7 Error Detection

The detection of development errors is configurable (see section 10.1.1, `EthTSynDevErrorDetect`).

7.8 Error Notification

The module ID 164 of EthTSyn, which is used as a parameter in the `Det_ReportError()` call, is exported via the macro definition `ETHTSYN_MODULE_ID` in *`EthTSyn.h`*.

8 API specification

8.1 API

8.1.1 Imported types

In this section all types included from the following files are listed:

[SWS_EthTSyn_00031]
[

Module	*Imported Type*
ComStack_Types	BufReq_ReturnType
Eth_GeneralTypes	EthTrcv_LinkStateType
	Eth_BufIdxType
	Eth_DataType
	Eth_FrameType
	Eth_RateRatioType
	Eth_TimeIntDiffType
	Eth_TimeStampQualType
	Eth_TimeStampType
StbM	StbM_SynchronizedTimeBaseType
	StbM_TimeStampType
	StbM_UserDataType
Std_Types	Std_ReturnType
	Std_VersionInfoType

](SRS_StbM_20048)

8.1.2 Type definitions

8.1.2.1 EthTSyn_ConfigType

[SWS_EthTSyn_00032]
[

Name:	`EthTSyn_ConfigType`		
Type:	`Structure`		
Element:	`void`	`implementation specific`	--
Description:	This is the base type for the configuration of the Global Time Synchronization over Ethernet. A pointer to an instance of this structure will be used in the initialization of the Global Time Synchronization over Ethernet. The content of this structure is defined in chapter 10 Configuration specification.		

](SRS_StbM_20048)

8.1.2.2 EthTSyn_TransmissionModeType

[SWS_EthTSyn_00033]

[

Name:	EthTSyn_TransmissionModeType		
Type:	--		
Range:	ETHTSYN_TX_OFF	0	Transmission Disabled
	ETHTSYN_TX_ON	1	Transmission Enabled
Description:	Handles the enabling and disabling of the transmission mode		

](SRS_StbM_20048)

8.1.2.3 EthTSyn_SyncStateType

[SWS_EthTSyn_00034]

[

Name:	EthTSyn_SyncStateType		
Type:	--		
Range:	ETHTSYN_SYNC	0	Ethernet time synchronous
	ETHTSYN_UNSYNC	1	Ethernet not time synchronous
	ETHTSYN_UNCERTAIN	2	Ethernet Sync state uncertain
	ETHTSYN_NEVERSYNC	3	No Sync message received between EthTSyn_Init() and current requested state.
Description:	Depending on the HW, quality information regarding the evaluated Sync state might be supported. If not supported, the value shall be always ETHTSYN_SYNC. For ETHTSYN_UNSYNC and ETHTSYN_UNCERTAIN values, the upper layer shall discard the time synchronous information. Within this enumeration, ETHTSYN_NEVERSYNC is having a higher priority than ETHTSYN_UNSYNC.		

](SRS_StbM_20048)

8.1.3 Function definitions

8.1.3.1 EthTSyn_Init

[SWS_EthTSyn_00035]
[

Service name:	EthTSyn_Init	

Syntax:	`void EthTSyn_Init(`
	`  const EthTSyn_ConfigType* configPtr`
	`)`	

Service ID[hex]:	0x01	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	configPtr	Pointer to selected configuration structure
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	This function initializes the Time Synchronization over Ethernet.	

](SRS_StbM_20048)

See section 7.2.1 for details.

8.1.3.2 EthTSyn_GetVersionInfo

[SWS_EthTSyn_00036]
[

Service name:	EthTSyn_GetVersionInfo	

Syntax:	`void EthTSyn_GetVersionInfo(`
	`  Std_VersionInfoType* versioninfo`
	`)`	

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

](SRS_StbM_20048)

8.1.3.3 EthTSyn_GetCurrentTime

[SWS_EthTSyn_00037]

[

Service name:	EthTSyn_GetCurrentTime	
Syntax:	`Std_ReturnType EthTSyn_GetCurrentTime( StbM_SynchronizedTimeBaseType timeBaseId, StbM_TimeStampType* timeStampPtr, EthTSyn_SyncStateType* syncState )`	
Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	timeBaseId	time base reference index
Parameters (inout):	None	
Parameters (out):	timeStampPtr	current time stamp
	syncState	to inform whether Ethernet is time synchronous or not
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Returns a time value according its definition out of the HW registers.	

](SRS_StbM_20048)

8.1.3.4 EthTSyn_SetGlobalTime

[SWS_EthTSyn_00038]

[

Service name:	EthTSyn_SetGlobalTime	
Syntax:	`Std_ReturnType EthTSyn_SetGlobalTime( StbM_SynchronizedTimeBaseType timeBaseId, const StbM_TimeStampType* timeStampPtr )`	
Service ID[hex]:	0x04	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	timeBaseId	time base reference index
	timeStampPtr	new time stamp
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Allows the Time Master to adjust the global ETH Reference clock in HW. This method is used to set a Global Time Base on ETH in general or to synchronize the Global ETH Time Base with another time base, e.g. Ethernet.	

](SRS_StbM_20048)

8.1.3.5 EthTSyn_SetTransmissionMode

[SWS_EthTSyn_00039]
⌈

Service name:	EthTSyn_SetTransmissionMode	
Syntax:	`void EthTSyn_SetTransmissionMode( uint8 CtrlIdx, EthTSyn_TransmissionModeType Mode )`	
Service ID[hex]:	0x05	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	CtrlIdx	Index of the Ethernet controller
	Mode	ETHTSYN_TX_OFF ETHTSYN_TX_ON
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	This API is used to turn on and off the TX capabilities of the EthTSyn.	

⌋(SRS_StbM_20048)

8.1.4 Call-back notifications

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

8.1.4.1 EthTSyn_RxIndication

[SWS_EthTSyn_00040]

[

Service name:	EthTSyn_RxIndication	

Syntax:	`void EthTSyn_RxIndication(`
	`  uint8 CtrlIdx,`
	`  Eth_FrameType FrameType,`
	`  boolean IsBroadcast,`
	`  const uint8* PhysAddrPtr,`
	`  uint8* DataPtr,`
	`  uint16 LenByte`
	`)`	

Service ID[hex]:	0x06	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	CtrlIdx	Index of the Ethernet controller
	FrameType	frame type of received Ethernet frame
	IsBroadcast	parameter to indicate a broadcast frame
	PhysAddrPtr	pointer to Physical source address (MAC address in network byte order) of received Ethernet frame
	DataPtr	Pointer to payload of the received Ethernet frame (i.e. Ethernet header is not provided).
	LenByte	Length of received data.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	By this API service the EthTSyn gets an indication and the data of a received frame.	

](SRS_StbM_20048)

[SWS_EthTSyn_00041]

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

- EthTSyn was not initialized (`ETHTSYN_E_NOT_INITIALIZED`)

](SRS_BSW_00337)

8.1.4.2 EthTSyn_TxConfirmation

[SWS_EthTSyn_00042]

[

Service name:	EthTSyn_TxConfirmation	

Syntax:	`void EthTSyn_TxConfirmation(`
	`  uint8 CtrlIdx,`
	`  uint8 BufIdx`
	`)`	

Service ID[hex]:	0x07	
Sync/Async:	Synchronous	
Reentrancy:	Dont care	
Parameters (in):	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface
	BufIdx	Index of the buffer resource
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Confirms the transmission of an Ethernet frame	

](SRS_StbM_20048)

8.1.4.3 EthTSyn_TrcvLinkStateChg

[SWS_EthTSyn_00043]

[

Service name:	EthTSyn_TrcvLinkStateChg	

Syntax:	`Std_ReturnType EthTSyn_TrcvLinkStateChg(`
	`  uint8 CtrlIdx,`
	`  EthTrcv_LinkStateType TrcvLinkState`
	`)`	

Service ID[hex]:	0x08	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	CtrlIdx	Index of the Ethernet controller
	TrcvLinkState	ETHTRCV_LINK_STATE_DOWN ETHTRCV_LINK_STATE_ACTIVE
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: successful E_NOT_OK: failed
Description:	Allows resetting state machine in case of unexpected Link loss to avoid inconsistent Sync and Follow_Up sequences	

](SRS_StbM_20048)

8.1.5 Scheduled functions

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

8.1.5.1 EthTSyn_MainFunction

[SWS_EthTSyn_00044]
⌈

Service name:	EthTSyn_MainFunction

Syntax:	`void EthTSyn_MainFunction(`
	`  void`
	`)`

Service ID[hex]:	0x09
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Main function for cyclic call / resp. Sync, Follow_Up and Pdelay_Req transmissions

⌋(SRS_StbM_20048)

[SWS_EthTSyn_00045]
⌈The frequency of invocations of `EthTSyn_MainFunction()` is determined by the configuration parameter `EthTSynMainFunctionPeriod`.
⌋(SRS_StbM_20048)

[SWS_EthTSyn_00046]
⌈The scheduled function `EthTSyn_MainFunction()` shall inform the DET, if development error detection is enabled (`ETHTSYN_DEV_ERROR_DETECT` is set to TRUE) and if function call has failed because of the following reasons:

- The EthTSyn module was not initialized (`ETHTSYN_E_NOT_INITIALIZED`).

⌋(SRS_BSW_00337)

8.1.6 Expected Interfaces

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

8.1.6.1 Mandatory Interfaces

There are no mandatory interfaces defined.

8.1.6.2 Optional Interfaces

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

[SWS_EthTSyn_00047]

[

API function	*Description*
Det_ReportError	Service to report development errors.
EthIf_EnableEgressTimeStamp	Activates egress time stamping on a dedicated message object. Some HW does store once the egress time stamp marker and some HW needs it always before transmission. There will be no â€œdisableâ€• functionality, due to the fact, that the message type is always "time stamped" by network design.
EthIf_GetCurrentTime	Returns a time value out of the HW registers according to the capability of the HW. Is the HW resolution is lower than the Eth_TimeStampType resolution resp. range, than an the remaining bits will be filled with 0.
EthIf_GetEgressTimeStamp	Reads back the egress time stamp on a dedicated message object. It must be called within the TxConfirmation() function.
EthIf_GetIngressTimeStamp	Reads back the ingress time stamp on a dedicated message object. It must be called within the RxIndication() function.
EthIf_ProvideTxBuffer	Provides access to a transmit buffer of the specified Ethernet controller.
EthIf_SetCorrectionTime	Allows the Time Slave to adjust the local ETH Reference clock in HW.
EthIf_SetGlobalTime	Allows the Time Master to adjust the global ETH Reference clock in HW. We can use this method to set a global time base on ETH in general or to synchronize the global ETH time base with another time base, e.g. FlexRay.
EthIf_Transmit	Triggers transmission of a previously filled transmit buffer
StbM_BusSetGlobalTime	Allows the Timebase Provider Modules to forward a new Global Time to the StbM, which has been received from different busses.
StbM_GetCurrentTime	Returns a time value (Local Time Base derived from Global Time Base) in standard format.

](SRS_StbM_20048)

9 Sequence diagrams

9.1 EthTSyn_GetCurrentTime

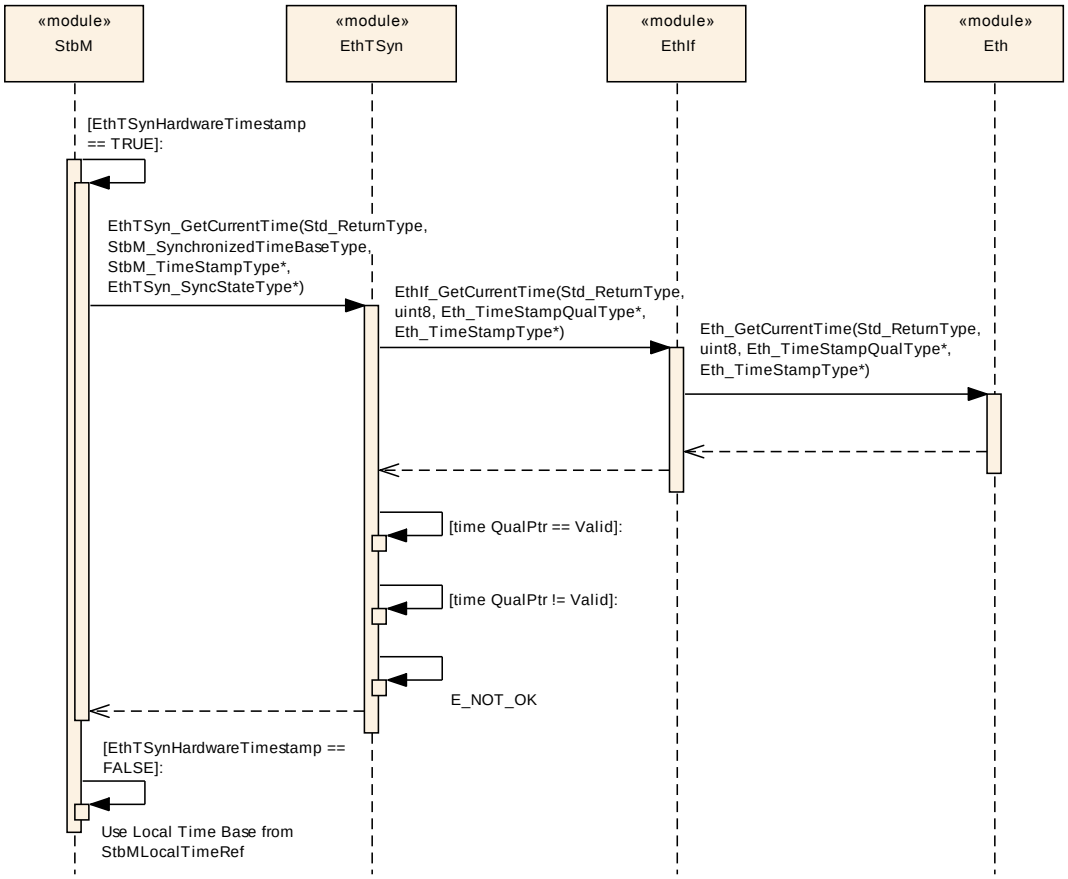


Figure 4: EthTSyn_GetCurrentTime

9.2 EthIf_EnableEgressTimeStamp

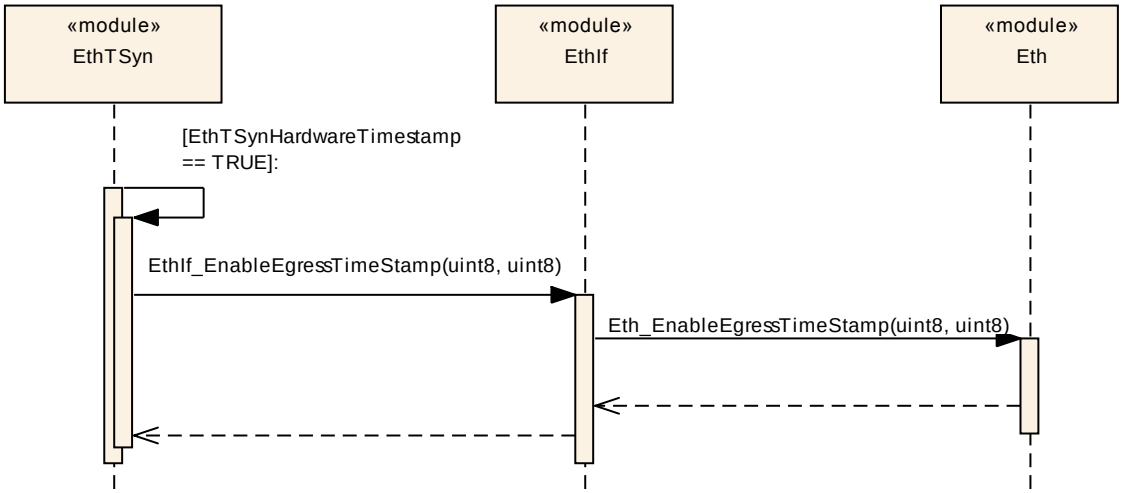


Figure 5: EthIf_EnableEgressTimeStamp

9.3 EthTSyn Egress Time Stamping

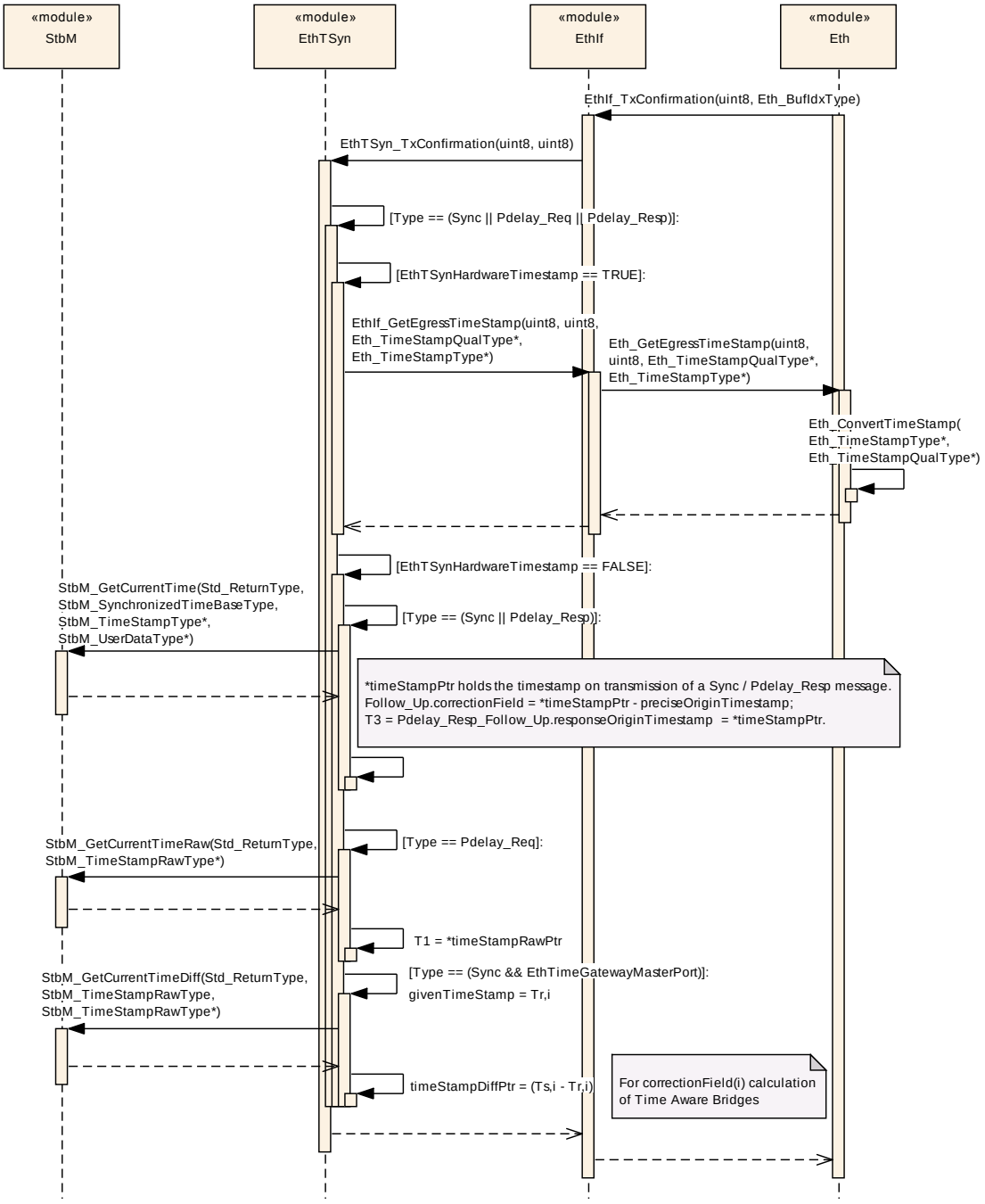


Figure 6: EthTSyn Egress Time Stamping

9.4 EthTSyn Ingress Time Stamping

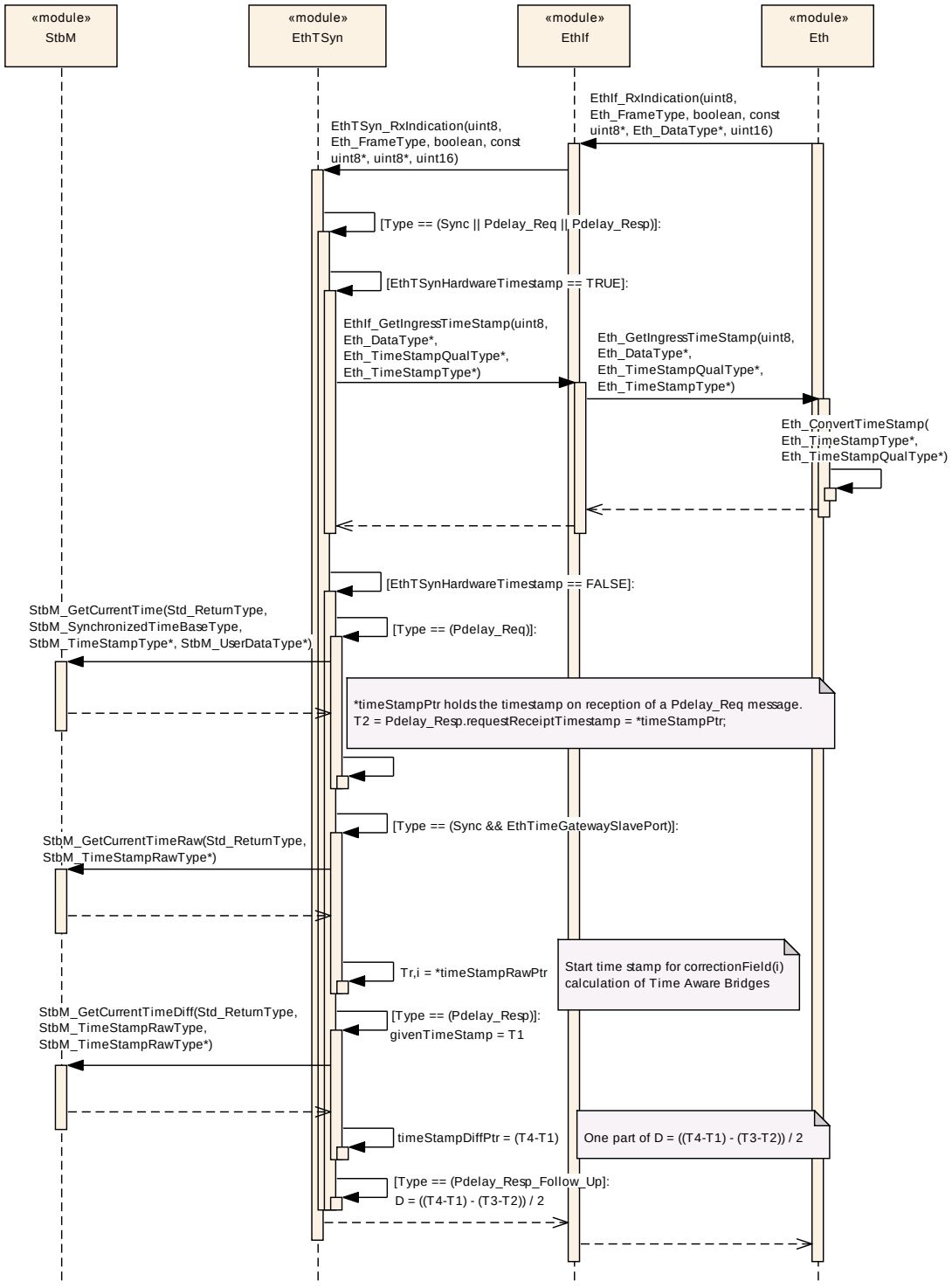


Figure 7: EthTSyn Ingress Time Stamping

9.5 EthTSyn_SetGlobalTime

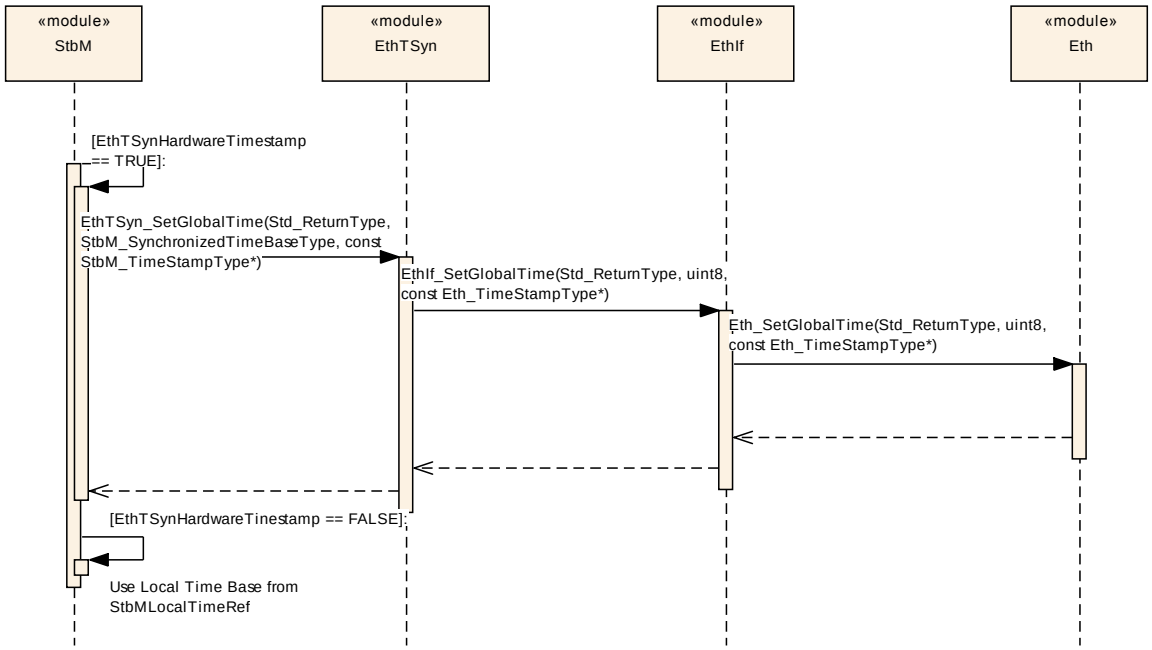


Figure 8: EthTSyn_SetGlobalTime

10 Configuration specification

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

Section 10.1.1 specifies the structure (containers) and the parameters of the Global Time Synchronization over Ethernet.

Section 10.3 specifies published information of the Global Time Synchronization over Ethernet.

10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in *SWS_BSWGeneral*.

10.1.1 Variants

[SWS_EthTSyn_00048]
[The Global Time Synchronization over Ethernet shall support the configuration variant VARIANT-PRE-COMPILE.
](SRS_BSW_00396

[SWS_EthTSyn_00051]
[The Time Synchronization over CAN shall support the configuration for Time Master, Time Slave and Time Gateway.
](SRS_StbM_20052)

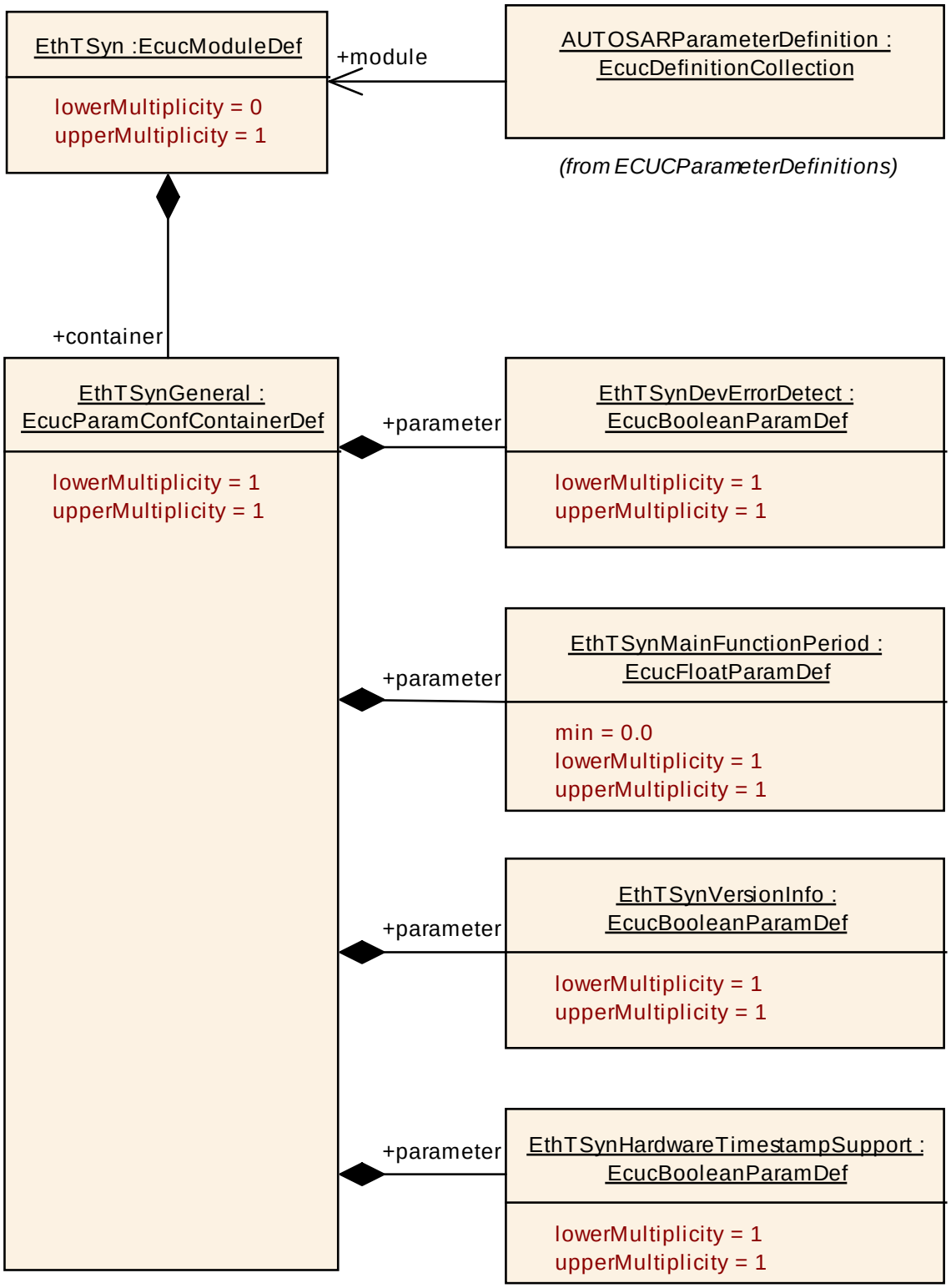
10.2 Containers and configuration parameters

The following sections summarize all configuration parameters of the Global Time Synchronization over Ethernet. The detailed meaning of the parameters is described in chapters 7 and 8.

10.2.1 EthTSyn

SWS Item	**ECUC_EthTSyn_00001 :**
Module Name	*EthTSyn*
Module Description	Configuration of the Synchronized Time-base Manager (StbM) module with respect to global time handling on Ethernet.
Post-Build Variant Support	false

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
EthTSynGeneral	1	This container holds the general parameters of the Ethernet-specific Synchronized Time-base Manager
EthTSynGlobalTimeDomain	1..*	This represents the existence of a global time domain on Ethernet. The EthTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the EthTSyn exists it is assumed that at least one global time domain exists.



10.2.2 EthTSynGeneral

SWS Item	**ECUC_EthTSyn_00003 :**
Container Name	EthTSynGeneral

Description	This container holds the general parameters of the Ethernet-specific Synchronized Time-base Manager
Configuration Parameters	

SWS Item	**ECUC_EthTSyn_00002 :**		
Name	EthTSynDevErrorDetect		
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. • true: enabled (ON). • false: disabled (OFF).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00018 :**		
Name	EthTSynHardwareTimestampSupport		
Description	Activate/Deactivate the hardware time stamping functionality of the Ethernet hardware. True: Timestamp is retrieved from the Ethernet hardware False: Timestamp is retrieved from the StbM		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00012 :**		
Name	EthTSynMainFunctionPeriod		
Description	Schedule period of the main function EthTSyn_MainFunction. Unit: [s].		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00015 :**		
Name	EthTSynVersionInfo		
Description	Activate/Deactivate the version information API (EthTSyn_GetVersionInfo). True: version information API activated False: version information API deactivated.		
Multiplicity	1		
Type	EcucBooleanParamDef		

Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.3 EthTSynGlobalTimeDomain

SWS Item	**ECUC_EthTSyn_00004 :**
Container Name	EthTSynGlobalTimeDomain
Description	This represents the existence of a global time domain on Ethernet. The EthTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the EthTSyn exists it is assumed that at least one global time domain exists.
Configuration Parameters	

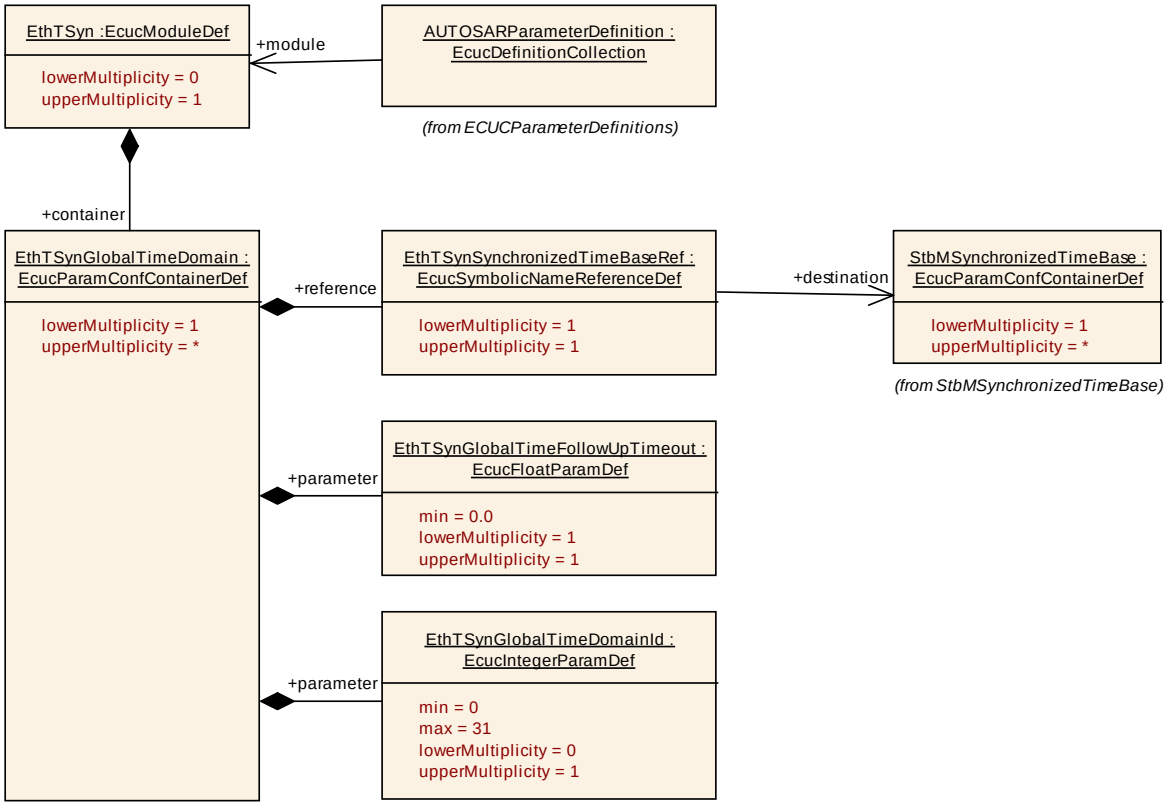
SWS Item	**ECUC_EthTSyn_00005 : (Obsolete)**		
Name	EthTSynGlobalTimeDomainId		
Description	This Parameter is deprecated and will be removed in future. Old description: The global time domain ID. **Tags:** atp.Status=obsolete atp.StatusRevisionBegin=4.2.2		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 31		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00007 :**		
Name	EthTSynGlobalTimeFollowUpTimeout		
Description	--		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00013 :**

Name	EthTSynSynchronizedTimeBaseRef		
Description	Mandatory reference to the required synchronized time-base.		
Multiplicity	1		
Type	Symbolic name reference to [StbMSynchronizedTimeBase]		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
EthTSynGlobalTimeMaster	0..1	Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.
EthTSynGlobalTimeSlave	0..1	Configuration of a time slave. Each global time domain is required to have at least one time slave. The configured ECU may or may not represent a time slave.



10.2.4 EthTSynGlobalTimeMaster

SWS Item	**ECUC_EthTSyn_00008 :**
Container Name	EthTSynGlobalTimeMaster
Description	Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.

Configuration Parameters

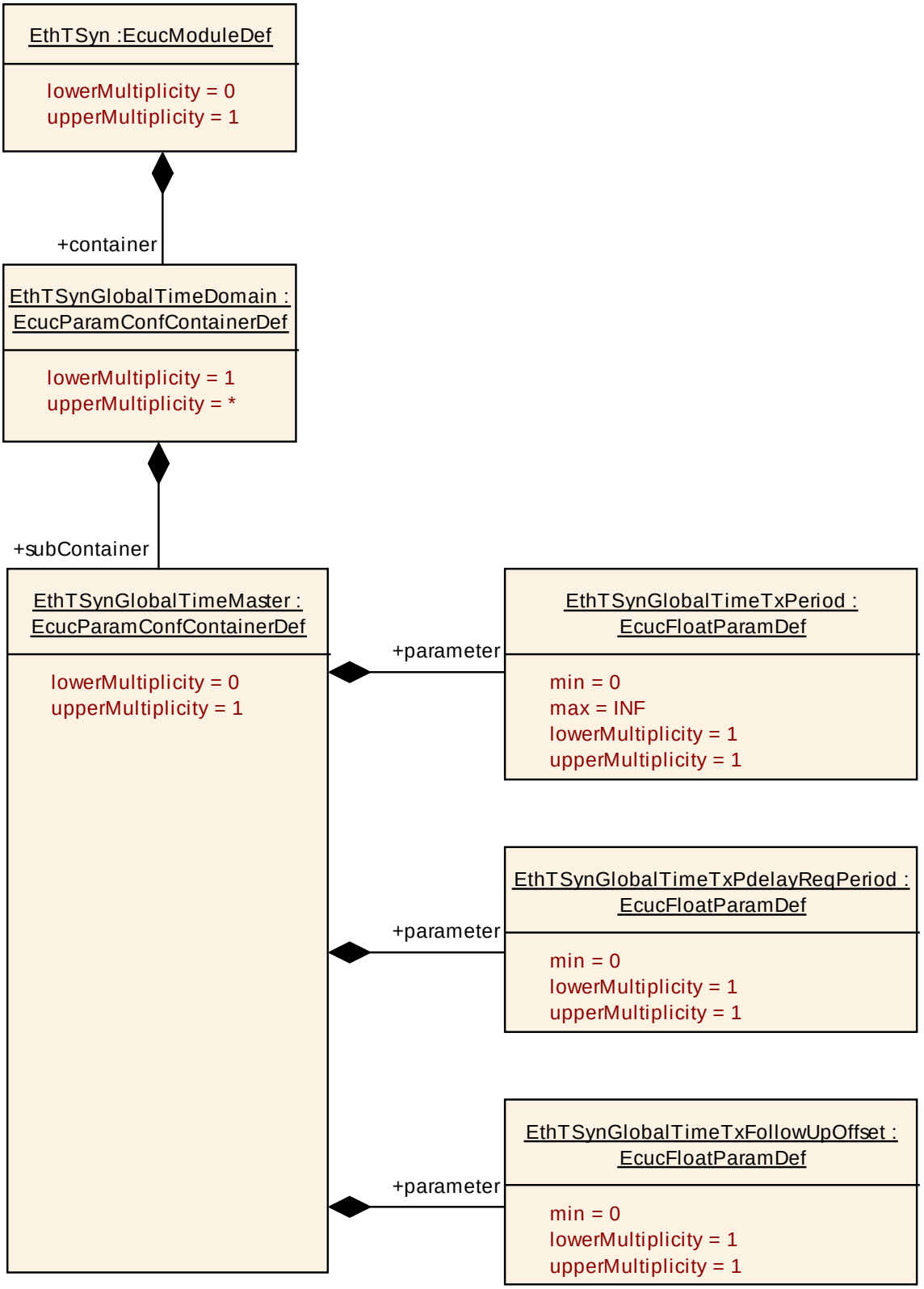
SWS Item	**ECUC_EthTSyn_00020 :**		
Name	EthTSynGlobalTimeTxFollowUpOffset		
Description	This represents configuration of the TX offset for Follow_Up / Pdelay_Resp_Follow_Up messages. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	*Pre-compile time*	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00019 :**		
Name	EthTSynGlobalTimeTxPdelayReqPeriod		
Description	This represents configuration of the TX period for Pdelay_Req messages. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	*Pre-compile time*	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00010 :**		
Name	EthTSynGlobalTimeTxPeriod		
Description	This represents configuration of the TX period. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	*Pre-compile time*	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_EthTSyn_00016 :**		
Name	EthTSynGlobalTimeEthIfRef		
Description	This represents the reference to the Ethernet interface taken to fetch the global time information.		
Multiplicity	1		
Type	Reference to [EthIfController]		
Post-Build Variant Value	false		
Value Configuration Class	*Pre-compile time*	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers



10.2.5 EthTSynGlobalTimeSlave

SWS Item	**ECUC_EthTSyn_00009 :**		
Container Name	EthTSynGlobalTimeSlave		
Description	Configuration of a time slave. Each global time domain is required to have at least one time slave. The configured ECU may or may not represent a time slave.		
Configuration Parameters			

SWS Item	**ECUC_EthTSyn_00022 :**		
Name	EthTSynGlobalTimeTxFollowUpOffset		
Description	This represents configuration of the TX offset for Follow_Up / Pdelay_Resp_Follow_Up messages. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

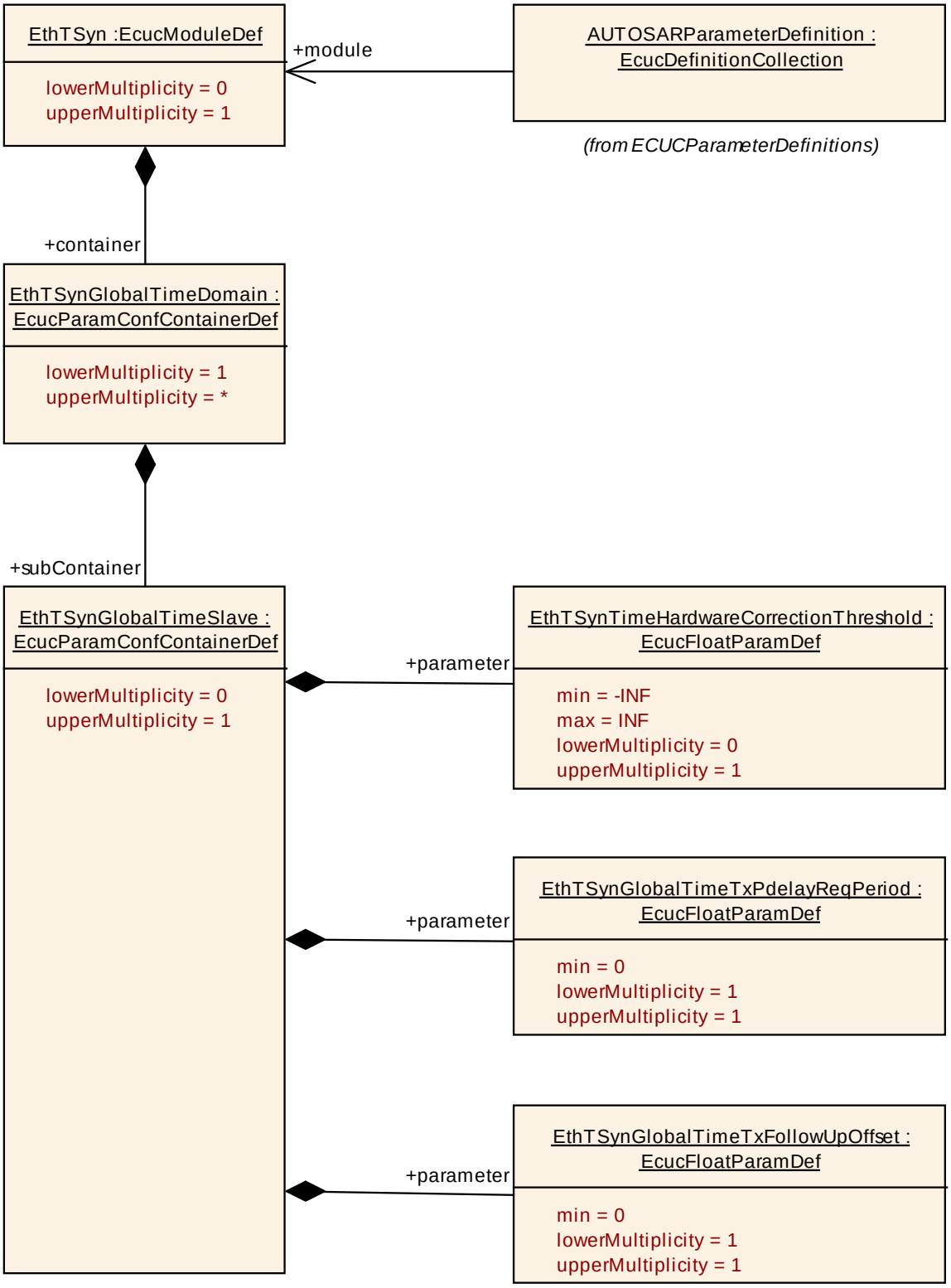
SWS Item	**ECUC_EthTSyn_00021 :**		
Name	EthTSynGlobalTimeTxPdelayReqPeriod		
Description	This represents configuration of the TX period for Pdelay_Req messages. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

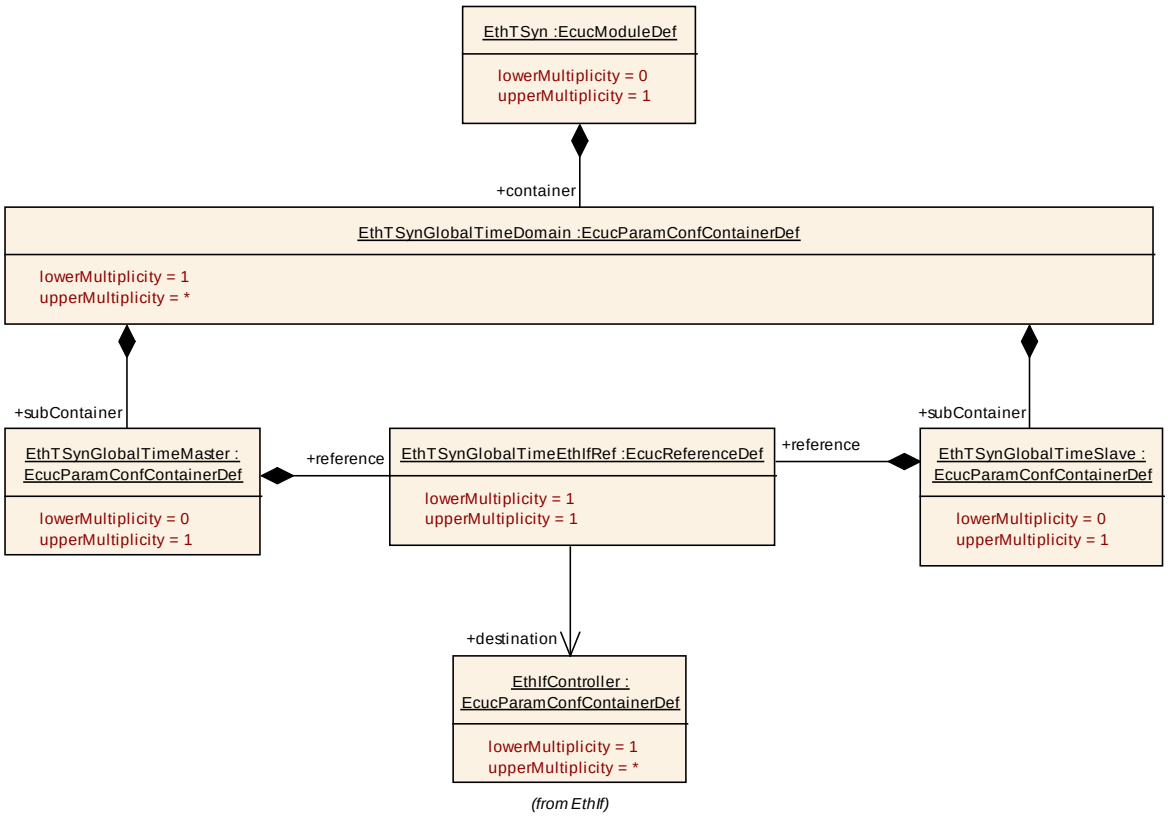
SWS Item	**ECUC_EthTSyn_00014 :**		
Name	EthTSynTimeHardwareCorrectionThreshold		
Description	This parameter can be taken to define the maximum deviation between the local time and the time obtained from SYNC message that triggers a correction of the time drift caused by the hardware on the local node.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	-INF .. INF		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	

Scope / Dependency	scope: local

SWS Item	**ECUC_EthTSyn_00017 :**		
Name	EthTSynGlobalTimeEthIfRef		
Description	This represents the reference to the Ethernet interface taken to fetch the global time information.		
Multiplicity	1		
Type	Reference to [EthIfController]		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers


EthTSyn :EcucModuleDef
lowerMultiplicity = 0
upperMultiplicity = 1
+module
AUTOSARParameterDefinition :
EcucDefinitionCollection
(from ECUCParameterDefinitions)
+container
EthTSynGlobalTimeDomain :
EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
EthTSynGlobalTimeSlave :
EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthTSynTimeHardwareCorrectionThreshold :
EcucFloatParamDef
min = -INF
max = INF
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
EthTSynGlobalTimeTxPdelayReqPeriod :
EcucFloatParamDef
min = 0
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
EthTSynGlobalTimeTxFollowUpOffset :
EcucFloatParamDef
min = 0
lowerMultiplicity = 1
upperMultiplicity = 1



10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in *SWS_BSWGeneral.*