Document Title	Specification of Time Synchronization over FlexRay
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	675

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R21-11

Document Change History Change History

Date	Release	Changed by	Description
2021-11-25	R21-11	AUTOSAR Release Management	• Hysteresis added for sequence counter validation • Small enhancement to improve precision of Global Time • Bugfix for Time Validation
2020-11-30	R20-11	AUTOSAR Release Management	• Time Validation updated for Time Gateways • Post build variant value corrected for FrTSynGlobalTimeMasterHandleId and FrTSynGlobalTimeSlaveHandleId
2019-11-28	R19-11	AUTOSAR Release Management	• Time Validation (draft) • Clarification regarding messages with stuck sequence counter • Clarification regarding cyclic operation entry after timebase startup • Clarification regarding transmission and reception of User Bytes • Changed Document Status from Final to published

2018-10-31	4.4.0	AUTOSAR Release Management	• Modifications to enhance the precision of Global Time Synchronization • Additional minor corrections / clarifications / editorial changes; For details please refer to the Change Documentation
2016-11-30	4.3.0	AUTOSAR Release Management	• Offset message formats changed • Immediate Time Synchronization message transmission • Various enhancements and corrections
2015-07-31	4.2.2	AUTOSAR Release Management	• Error code FRTSYN_E_INVALID_PDU_SDU_ID replaced by FRTSYN_E_INVALID_PDUID • FlexRay communication state handling simplified (FrIf_GetPOCStatus replaced by FrIf_GetState)
2014-10-31	4.2.1	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

The FrTSyn module handles the distribution of time information over FlexRay buses.

The FlexRay mechanism is much simpler than the mechanism for CAN since it is based on the fact, that FlexRay nodes are synchronized to each other, otherwise no messages can be transmitted on FlexRay.

Both, Time Master and Time Slaves have the same view on the FlexRay global time. It is therefore just necessary to define the same point in (FlexRay) time and to transmit the time information, which will be valid at that point in (FlexRay) time.

Although this same point in (FlexRay) time could be in theory any FlexRay macrotick within a FlexRay cycle, the start of a FlexRay cycle simplifies this mechanism. In addition, the mechanism does not just use any cycle start but uses the cycle start of the subsequent cycle with cycle counter value 0, i.e. the Time Master transmits time information located in the future.

On FlexRay only one Time Synchronization message is needed.

The Time Master uses its current FlexRay time, i.e. macrotick counter and cycle counter, and the current time, which shall be distributed and calculates the resulting time at the start of the next cycle 0. Once this resulting time has been calculated, it is neither very time critical, when exactly the FlexRay frame is transmitted, nor when it is received and processed.

Every Time Slave receiving the transmitted time information will use it in combination with the current FlexRay macrotick counter and cycle counter to determine the actual master time and set its slave time.

Figure 1.1 illustrates the Time Synchronization mechanism on FlexRay.

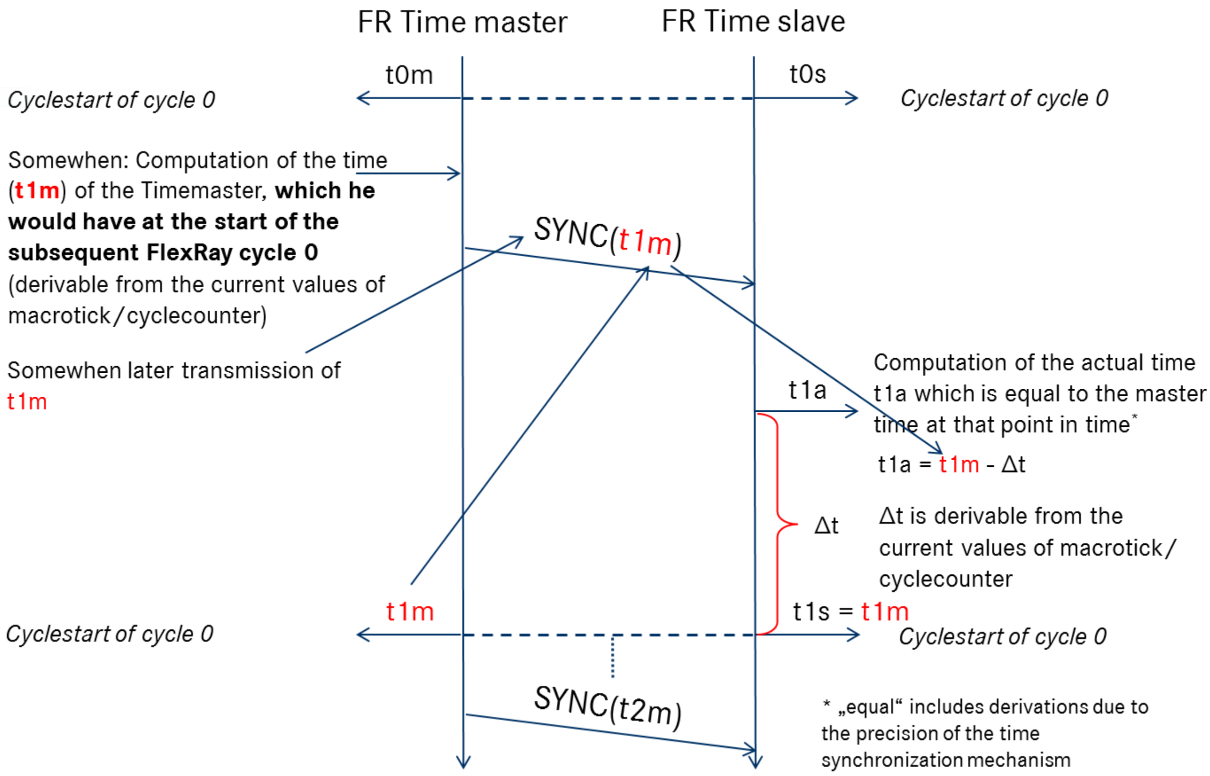

FR Time master
FR Time slave
t0m
t0s
Cyclestart of cycle 0
Cyclestart of cycle 0
Somewhen: Computation of the time (t1m) of the Timemaster, which he would have at the start of the subsequent FlexRay cycle 0 (derivable from the current values of macrotick/cyclecounter)
SYNC(t1m)
Somewhen later transmission of t1m
Computation of the actual time t1a which is equal to the master time at that point in time*
t1a
t1a = t1m - Δt
Δt
Δt is derivable from the current values of macrotick/ cyclecounter
t1m
t1s = t1m
Cyclestart of cycle 0
Cyclestart of cycle 0
SYNC(t2m)
* „equal“ includes derivations due to the precision of the time synchronization mechanism

Figure 1.1: FlexRay Time Synchronization Mechanism

2 Acronyms, Abbreviations, and Definitions

This section lists module local abbreviations and definitions. For additional Time Synchronization related abbreviations and definitions refer to chapter 3 in the RS Time Synchronization [1]. For general terms and abbreviations refer to the AUTOSAR Glossary [2].

Abbreviation	Description
GTM	Global Time Master
BswM	BSW Mode Manager module
<Bus>TSyn	Bus specific Time Synchronization module
CRC	Cyclic Redundancy Checksum
Debounce Time	Minimum gap between two Tx messages with the same PDU
Det	Default Error Tracer module
FCNT	FlexRay Cycle Counter
FR	FlexRay
FrIf	FlexRay interface module
FrTSyn	Time Synchronization over FlexRay module
FUP message	Follow-Up message
OFNS message	Offset adjustment message
OFS message	Offset Synchronization message
SC	Sequence Counter in Time Synchronization messages
SGW	"Synchronized to Gateway" state of Time Synchronization
StbM	Synchronized Time-Base Manager

Abbreviation	Description
SYNC message	Time Synchronization message
Timesync	Time Synchronization

3 Related documentation

3.1 Input documents & related standards and norms

[1] Requirements on Time Synchronization
AUTOSAR_RS_TimeSync

[2] Glossary
AUTOSAR_TR_Glossary

[3] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral

[4] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral

[5] Specification of Synchronized Time-Base Manager
AUTOSAR_SWS_SynchronizedTimeBaseManager

[6] Specification of CRC Routines
AUTOSAR_SWS_CRCLibrary

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [3, SWS BSW General], which is also valid for FrTSyn.

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

4 Constraints and assumptions

4.1 Limitations

- Time Masters, Time Gateways and Time Slaves shall work with a Time Base reference clock with a worst-case accuracy of 2μs.
- The Time Base in the OFS messages is limited to 32 bit, wherefore the maximum supported time value is 4294967295 seconds (2^{32}-1).

- "CRC secured" in the context of this document refers to CRC integrity protection mechanism and does not imply that CRC is used as a cybersecurity solution.

4.2 Applicability to car domains

Automotive systems requiring a common Time Base for ECUs regardless of which bus system the ECUs are connected to.

5 Dependencies to other modules

The Time Synchronization over FlexRay (FrTSyn) has interfaces towards the Synchronized Time-Base Manager (StbM), the FlexRay Interface (FrIf), the BSW Mode Manager (BswM) and the Default Error Tracer (Det).

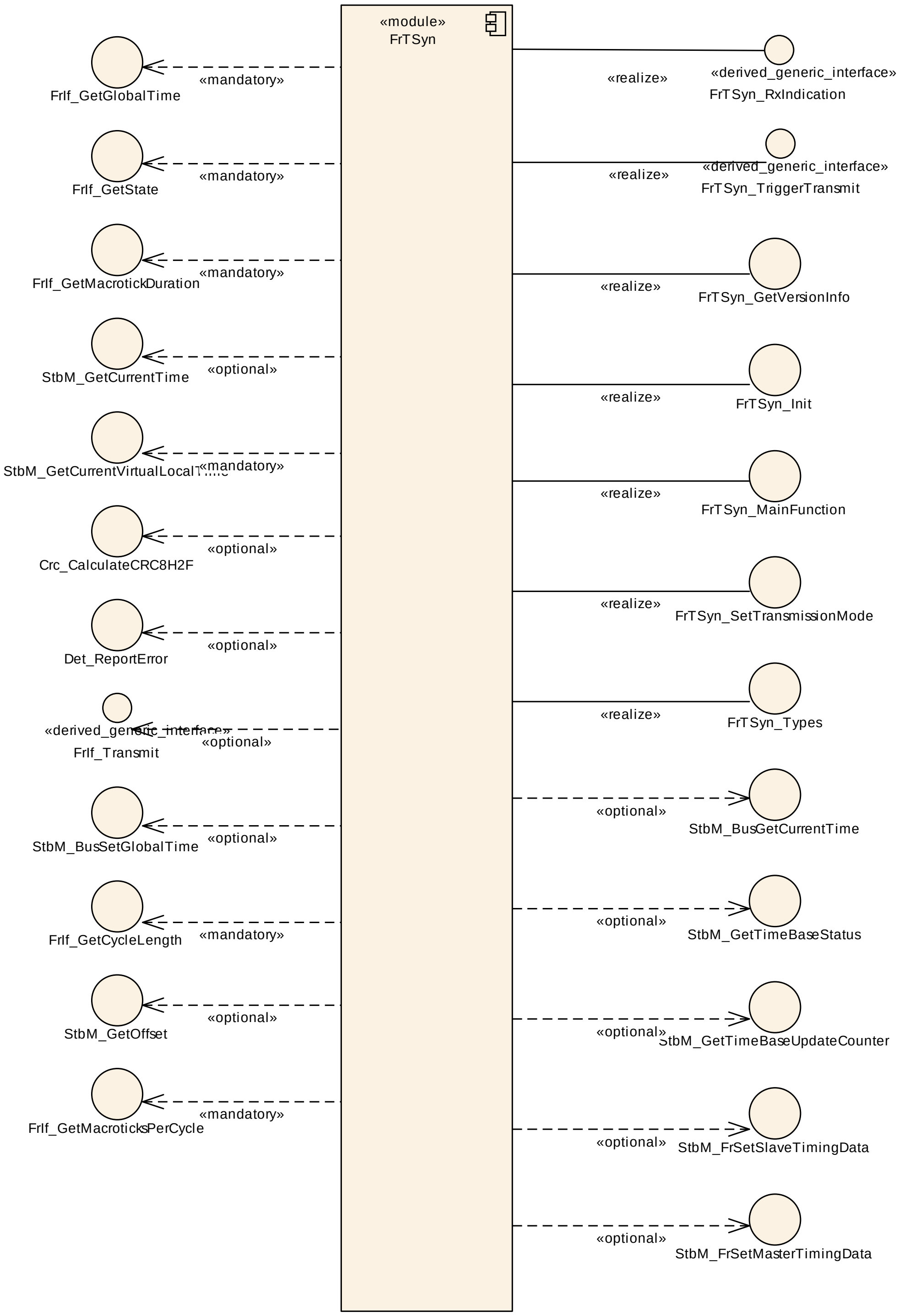


Figure 5.1: Module dependencies of the FrTSyn module

- StbM - Get and set the current time value
- FrIf - Receiving and transmitting messages
- BswM - Coordination of network access (via FrTSyn_SetTransmissionMode)
- 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 [3].

5.1.2 Header file structure

For details, refer to the section 5.1.7 "Header file structure" of the SWS BSW General [3].

6 Requirements Tracing

The following tables reference the requirements specified in [1, RS TimeSync] and [4, SRS BSWGeneral] and links to the fulfillment of these. Please note that if column "Satisfied by" is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_TS_00002]	The Implementation of Time Synchronization shall maintain its own Time Base independently of the acting role.	[SWS_FrTSyn_00999]
[RS_TS_00003]	The TS shall initialize the Local Time Base with a configurable startup value	[SWS_FrTSyn_00003]
[RS_TS_00004]	The Implementation of Time Synchronization shall initialize the Global Time Base with a configurable startup value.	[SWS_FrTSyn_00003]
[RS_TS_00005]	The Implementation of Time Synchronization shall allow customers to have access to the Synchronized Time Base	[SWS_FrTSyn_00999]
[RS_TS_00006]	The Implementation of Time Synchronization shall provide time information to TSP modules	[SWS_FrTSyn_00999]
[RS_TS_00007]	The Implementation of Time Synchronization shall synchronize the Time Base of a Time Slave, on reception of a Time Master value	[SWS_FrTSyn_00999]
[RS_TS_00008]	The Implementation of Time Synchronization shall continuously maintain its Time Bases based on a Time Base reference clock	[SWS_FrTSyn_00999]

Requirement	Description	Satisfied by
[RS_TS_00009]	The Implementation of Time Synchronization shall maintain the synchronization status of a Time Base	[SWS_FrTSyn_00999]
[RS_TS_00010]	The Implementation of Time Synchronization shall allow customer on master side to set the Global Time	[SWS_FrTSyn_00999]
[RS_TS_00011]	The Implementation of Time Synchronization shall allow customers on master side to trigger time transmission by the TSP module	[SWS_FrTSyn_00999]
[RS_TS_00012]	The Implementation of Time Synchronization shall allow customers and TSP modules to read the offset value of an Offset Time Base	[SWS_FrTSyn_00999]
[RS_TS_00013]	The Implementation of Time Synchronization shall allow the customers and TSP modules to set the offset value of an Offset Master Time Base	[SWS_FrTSyn_00999]
[RS_TS_00014]	The Implementation of Time Synchronization shall allow customers to read User Data propagated via the TSP modules.	[SWS_FrTSyn_00999]
[RS_TS_00015]	The Implementation of Time Synchronization shall allow customers to set User Data propagated via the TSP modules.	[SWS_FrTSyn_00999]
[RS_TS_00016]	The Implementation of Time Synchronization shall notify customers about status events	[SWS_FrTSyn_00999]
[RS_TS_00017]	The Implementation of Time Synchronization shall notify customers about elapsed pre-defined time span.	[SWS_FrTSyn_00999]
[RS_TS_00018]	The Implementation of Time Synchronization shall support rate correction	[SWS_FrTSyn_00999]
[RS_TS_00019]	The Implementation of Time Synchronization shall support damping offset correction	[SWS_FrTSyn_00999]
[RS_TS_00021]	The Implementation of Time Synchronization shall provide interfaces to query the synchronization status	[SWS_FrTSyn_00999]

Requirement	Description	Satisfied by
[RS_TS_00024]	The Implementation of Time Synchronization shall support storage of the Time Base value at shutdown if configured as Time Master	[SWS_FrTSyn_00999]
[RS_TS_00025]	The Implementation of Time Synchronization shall provide fault detection mechanisms	[SWS_FrTSyn_00999]
[RS_TS_00026]	The Implementation of Time Synchronization shall provide to the customers a specific API per type of Time Base Resource	[SWS_FrTSyn_00999]
[RS_TS_00027]	The TS shall provide a bus independent customer interface	[SWS_FrTSyn_00999]
[RS_TS_00029]	The configuration of the Time Synchronization implementation shall allow the implementation to behave as a (vehicle wide) Time Master	[SWS_FrTSyn_00999]
[RS_TS_00030]	The configuration of the Time Synchronization implementation shall allow the implementation to behave as a Time Slave	[SWS_FrTSyn_00999]
[RS_TS_00031]	The configuration of the Time Synchronization implementation shall allow the implementation to behave as a Time Gateway	[SWS_FrTSyn_00999]
[RS_TS_00032]	The Implementation of Time Synchronization shall trigger registered customers	[SWS_FrTSyn_00999]
[RS_TS_00033]	The Implementation of Time Synchronization shall use a time format with a resolution of 1 ns	[SWS_FrTSyn_00999]
[RS_TS_00034]	The Implementation of Time Synchronization shall provide measurement data to the application	[SWS_FrTSyn_00092] [SWS_FrTSyn_00096] [SWS_FrTSyn_00097] [SWS_FrTSyn_00098] [SWS_FrTSyn_00099] [SWS_FrTSyn_00100] [SWS_FrTSyn_00101]
[RS_TS_00035]	The Implementation of Time Synchronization shall provide a system service interface to applications	[SWS_FrTSyn_00999]
[RS_TS_00036]	The Implementation of Time Synchronization shall provide a bus independent customer interface	[SWS_FrTSyn_00999]
[RS_TS_00037]	The configuration of the Time Synchronization implementation shall allow the interaction with different types of customers	[SWS_FrTSyn_00999]

Requirement	Description	Satisfied by
[RS_TS_00038]	The Implementation of Time Synchronization shall copy Time Base information upon user request	[SWS_FrTSyn_00999]
[RS_TS_20031]	The Timesync over CAN module shall trigger Time Base Synchronization transmission	[SWS_FrTSyn_00999]
[RS_TS_20032]	The Timesync over CAN module shall provide the Time Base after reception of a valid Timesync/TS messages	[SWS_FrTSyn_00999]
[RS_TS_20033]	The Timesync over CAN module shall support means to protect the Time synchronization protocol	[SWS_FrTSyn_00999]
[RS_TS_20034]	The Timesync over CAN module shall detect and handle timeout and integrity errors in the Time Synchronization protocol	[SWS_FrTSyn_00999]
[RS_TS_20035]	The Timesync over CAN module shall support a protocol for precise time measurement and synchronization over CAN	[SWS_FrTSyn_00999]
[RS_TS_20036]	The Timesync over CAN module shall use the time measurement and synchronization protocol to transmit and receive an offset value	[SWS_FrTSyn_00999]
[RS_TS_20037]	The Timesync over CAN module shall support user specific data within the time measurement and synchronization protocol	[SWS_FrTSyn_00999]
[RS_TS_20038]	The Timesync over CAN module configuration shall allow the Implementation of Time Synchronization for CAN to support different roles for a Time Base	[SWS_FrTSyn_00999]
[RS_TS_20039]	The Timesync over FlexRay module shall trigger Time Base Synchronization transmission	[SWS_FrTSyn_00019] [SWS_FrTSyn_00023] [SWS_FrTSyn_00026] [SWS_FrTSyn_00027] [SWS_FrTSyn_00084] [SWS_FrTSyn_00085] [SWS_FrTSyn_00086] [SWS_FrTSyn_00087] [SWS_FrTSyn_00088] [SWS_FrTSyn_00089] [SWS_FrTSyn_00090] [SWS_FrTSyn_00091] [SWS_FrTSyn_00093]

Requirement	Description	Satisfied by
[RS_TS_20040]	The Timesync over FlexRay module shall provide a Time Base after reception of a valid protocol information	[SWS_FrTSyn_00041] [SWS_FrTSyn_00045] [SWS_FrTSyn_00078] [SWS_FrTSyn_00094]
[RS_TS_20041]	The Timesync over FlexRay module shall support means to protect the Time Synchronization protocol	[SWS_FrTSyn_00006] [SWS_FrTSyn_00014] [SWS_FrTSyn_00015] [SWS_FrTSyn_00021] [SWS_FrTSyn_00025] [SWS_FrTSyn_00030] [SWS_FrTSyn_00031] [SWS_FrTSyn_00035] [SWS_FrTSyn_00036] [SWS_FrTSyn_00078] [SWS_FrTSyn_00079] [SWS_FrTSyn_00080]
[RS_TS_20042]	The Timesync over FlexRay module shall detect and handle timeout and integrity errors in the Time Synchronization protocol	[SWS_FrTSyn_00015] [SWS_FrTSyn_00038] [SWS_FrTSyn_00041] [SWS_FrTSyn_00042] [SWS_FrTSyn_00045] [SWS_FrTSyn_00048] [SWS_FrTSyn_00049] [SWS_FrTSyn_00050] [SWS_FrTSyn_00054] [SWS_FrTSyn_00055] [SWS_FrTSyn_00057] [SWS_FrTSyn_00058] [SWS_FrTSyn_00080] [SWS_FrTSyn_00081] [SWS_FrTSyn_00082] [SWS_FrTSyn_00094] [SWS_FrTSyn_00102]

Requirement	Description	Satisfied by
[RS_TS_20043]	The Timesync over FlexRay module shall support a protocol for precise time measurement and synchronization over Flex Ray	[SWS_FrTSyn_00007] [SWS_FrTSyn_00009] [SWS_FrTSyn_00010] [SWS_FrTSyn_00014] [SWS_FrTSyn_00015] [SWS_FrTSyn_00018] [SWS_FrTSyn_00019] [SWS_FrTSyn_00020] [SWS_FrTSyn_00021] [SWS_FrTSyn_00026] [SWS_FrTSyn_00027] [SWS_FrTSyn_00028] [SWS_FrTSyn_00030] [SWS_FrTSyn_00031] [SWS_FrTSyn_00035] [SWS_FrTSyn_00036] [SWS_FrTSyn_00037] [SWS_FrTSyn_00038] [SWS_FrTSyn_00039] [SWS_FrTSyn_00040] [SWS_FrTSyn_00041] [SWS_FrTSyn_00046] [SWS_FrTSyn_00048] [SWS_FrTSyn_00049] [SWS_FrTSyn_00050] [SWS_FrTSyn_00054] [SWS_FrTSyn_00055] [SWS_FrTSyn_00056] [SWS_FrTSyn_00057] [SWS_FrTSyn_00060] [SWS_FrTSyn_00061] [SWS_FrTSyn_00062] [SWS_FrTSyn_00063] [SWS_FrTSyn_00064] [SWS_FrTSyn_00065] [SWS_FrTSyn_00066] [SWS_FrTSyn_00069] [SWS_FrTSyn_00071] [SWS_FrTSyn_00072] [SWS_FrTSyn_00074] [SWS_FrTSyn_00075] [SWS_FrTSyn_00081]

Requirement	Description	Satisfied by
[RS_TS_20044]	The Timesync over FlexRay module shall use the time measurement and synchronization protocol to transmit and receive an offset value	[SWS_FrTSyn_00007] [SWS_FrTSyn_00009] [SWS_FrTSyn_00010] [SWS_FrTSyn_00020] [SWS_FrTSyn_00022] [SWS_FrTSyn_00023] [SWS_FrTSyn_00025] [SWS_FrTSyn_00026] [SWS_FrTSyn_00027] [SWS_FrTSyn_00029] [SWS_FrTSyn_00030] [SWS_FrTSyn_00031] [SWS_FrTSyn_00035] [SWS_FrTSyn_00036] [SWS_FrTSyn_00037] [SWS_FrTSyn_00042] [SWS_FrTSyn_00043] [SWS_FrTSyn_00044] [SWS_FrTSyn_00045] [SWS_FrTSyn_00047] [SWS_FrTSyn_00048] [SWS_FrTSyn_00049] [SWS_FrTSyn_00050] [SWS_FrTSyn_00054] [SWS_FrTSyn_00055] [SWS_FrTSyn_00056] [SWS_FrTSyn_00057] [SWS_FrTSyn_00079] [SWS_FrTSyn_00080] [SWS_FrTSyn_00082]
[RS_TS_20045]	The Timesync over FlexRay module shall support user specific data within the time measurement and synchronization protocol	[SWS_FrTSyn_00010] [SWS_FrTSyn_00011] [SWS_FrTSyn_00012] [SWS_FrTSyn_00013]
[RS_TS_20046]	The configuration for Time synchronization over FlexRay shall allow the FlexRay Time Synchronization module to support different roles for a Time Base	[SWS_FrTSyn_00077]
[RS_TS_20047]	The Timesync over Ethernet module shall trigger Time Base Synchronization transmission	[SWS_FrTSyn_00999]
[RS_TS_20048]	The Timesync over Ethernet module shall support IEEE 802.1AS as well as AUTOSAR extensions	[SWS_FrTSyn_00999]
[RS_TS_20051]	The Timesync over Ethernet module shall detect and handle errors in synchronization protocol / communication	[SWS_FrTSyn_00999]

Requirement	Description	Satisfied by
[RS_TS_20052]	The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Master	[SWS_FrTSyn_00999]
[RS_TS_20053]	The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Slave	[SWS_FrTSyn_00999]
[RS_TS_20054]	The Implementation of the Time Synchronization shall evaluate and propagate Time Gateway relevant information	[SWS_FrTSyn_00999]
[RS_TS_20058]	The Timesync over Ethernet module shall provide the precision of Synchronized Time Bases	[SWS_FrTSyn_00999]
[RS_TS_20059]	The Timesync over Ethernet module shall access all communication ports belonging to Time Synchronization	[SWS_FrTSyn_00999]
[RS_TS_20060]	The Timesync over Ethernet module shall provide a Time Base after reception of a valid protocol information	[SWS_FrTSyn_00999]
[RS_TS_20061]	The Timesync over Ethernet module shall support means to protect the Time Synchronization protocol	[SWS_FrTSyn_00999]
[RS_TS_20062]	The Timesync over Ethernet module shall support user specific data within the time measurement and synchronization protocol	[SWS_FrTSyn_00999]
[RS_TS_20063]	The Timesync over Ethernet module shall use the Time Synchronization protocol for Synchronized Time Bases to transmit and receive Offset Time Bases	[SWS_FrTSyn_00999]
[RS_TS_20066]	The Timesync over Ethernet module shall support a static (pre)configuration of IEEE 802.1AS Pdelay	[SWS_FrTSyn_00999]
[RS_TS_20068]	The Timesync over CAN module shall support classic CAN and CAN FD	[SWS_FrTSyn_00999]
[RS_TS_20069]	The TimeSync over Ethernet module shall provide read / write access to bus protocol specific parameters	[SWS_FrTSyn_00999]
[RS_TS_20070]	The Timesync over CAN module shall support hardware and software timestamping	[SWS_FrTSyn_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_FrTSyn_00058] [SWS_FrTSyn_00067] [SWS_FrTSyn_00070] [SWS_FrTSyn_00095]
[SRS_BSW_00337]	Classification of development errors	[SWS_FrTSyn_00067] [SWS_FrTSyn_00070] [SWS_FrTSyn_00095]
[SRS_BSW_00385]	List possible error notifications	[SWS_FrTSyn_00059]

7 Functional specification

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

7.1 Overview

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

Time Synchronization principles and common wording is described in [5] and [1].

7.2 Module Handling

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

7.2.1 Initialization

The Time Synchronization over FlexRay is initialized via FrTSyn_Init. Except for FrTSyn_GetVersionInfo and FrTSyn_Init, the API functions of the Time Synchronization over FlexRay may only be called when the module has been properly initialized.

[SWS_FrTSyn_00003] ⌈A call to FrTSyn_Init initializes all internal variables and sets the Time Synchronization over FlexRay to the initialized state. ⌋*(RS_TS_00003, RS_TS_00004)*

[SWS_FrTSyn_00006] ⌈The Sequence Counter (SC) shall be initialized with 0. ⌋*(RS_TS_20041)*

7.2.2 FlexRay Interface

[SWS_FrTSyn_00078] ⌈The FrTSyn module shall call FrIf_GetGlobalTime only if FrIf_GetState returns FRIF_STATE_ONLINE. This is to ensure that FrIf_GetGlobalTime returns valid time information, i.e. that the FlexRay communication controller is synchronous to the FlexRay global time.⌋*(RS_TS_20040, RS_TS_20041)*

7.2.3 Error Handling

[SWS_FrTSyn_00058] ⌈On errors and exceptions, the FrTSyn module shall not modify its current module state but shall simply report the error event.⌋*(RS_TS_20042, SRS_BSW_00323)*

7.3 Message Format

SYNC and OFS messages may share the same FR PDU by using a multiplexed signal group. The multiplexer is located in Byte 0, named Type.

For different Time Domains the same FR PDU may be used if Time Synchronization messages are sent by the same Time Master or Time Gateway.

For different Time Domains different FR PDUs shall be used if Time Synchronization messages are sent by different Time Masters or Time Gateways.

The usage of CRC is optional. To ensure a great variability between several time observing units, the configuration decides of how to handle CRC secured time synchronization messages if the receiver does not support the CRC calculation. Hence it might be possible, that a receiver is just using the given Time Base value, without evaluating the CRC.

[SWS_FrTSyn_00007] ⌈The byte order for time values inside Time Synchronization messages is "Big Endian".⌋*(RS_TS_20043, RS_TS_20044)*

[SWS_FrTSyn_00009] ⌈The PayloadLength is 16.⌋*(RS_TS_20043, RS_TS_20044)*

[SWS_FrTSyn_00010] ⌈Time Synchronization messages contain User Data according to the given message format.⌋*(RS_TS_20043, RS_TS_20044, RS_TS_20045)*

[SWS_FrTSyn_00011] ⌈User Data shall be read consistently from the incoming Time Synchronization messages.⌋*(RS_TS_20045)*

[SWS_FrTSyn_00012] ⌈User Data shall be written consistently to outgoing Time Synchronization messages.

If the number of User Data Fields in a Time Synchronization message is greater than the number of User Data Bytes provided by the StbM, the remaining User Data Fields shall be set to 0 (default value).⌋*(RS_TS_20045)*

[SWS_FrTSyn_00013] ⌈User Data shall be mapped to the StbM_UserDataType, where the byte number given in the message and by the StbM_UserDataType shall match (User Byte 0 mapped to StbM_UserDataType.userByte0 etc.). StbM_UserDataType.userDataLength shall be set to the Time Synchronization message type specific number of User Bytes.⌋*(RS_TS_20045)*

7.3.1 SYNC message

[SWS_FrTSyn_00014] ⌈SYNC not CRC secured message format:

Byte	Bit Position	Field Name	Field Value Range	Description
0		Type	0x10	Message Type
1		User Byte 2	default: 0	
2	7..4	D	0..15	Time Domain Id
	3..0	SC	0..15	Sequence Counter
3	7..2	FCNT	0..63	FlexRay Cycle Counter
	1	SGW	0 = SyncToGTM 1 = SyncToSubDomain	
	0	reserved	0	
4		User Byte 0	default: 0	
5		User Byte 1	default: 0	
6..11		SyncTimeSec		48 Bit time value in seconds
12..15		SyncTimeNSec		32 Bit time value in nanoseconds

Table 7.1: SYNC not CRC secured message format

⌋*(RS_TS_20041, RS_TS_20043)*

[SWS_FrTSyn_00015] ⌈SYNC CRC secured message format:

Byte	Bit Position	Field Name	Field Value Range	Description
0		Type	0x20	Message Type
1		CRC	0..255	Checksum
2	7..4	D	0..15	Time Domain Id
	3..0	SC	0..15	Sequence Counter
3	7..2	FCNT	0..63	FlexRay Cycle Counter
	1	SGW	0 = SyncToGTM 1 = SyncToSubDomain	
	0	reserved	0	
4		User Byte 0	default: 0	
5		User Byte 1	default: 0	
6..11		SyncTimeSec		48 Bit time value in seconds
12..15		SyncTimeNSec		32 Bit time value in nanoseconds

Table 7.2: SYNC CRC secured message format

⌋*(RS_TS_20041, RS_TS_20042, RS_TS_20043)*

7.3.2 OFS message

Offset messages can be multiplexed with SYNC messages (using the same PDU, etc.).

[SWS_FrTSyn_00079] ⌈OFS not CRC secured message format:

Byte	Bit Position	Field Name	Field Value Range	Description
0		Type	0x34	Message Type
1		User Byte 2	default: 0	
2	7..4	D	16..31	Time Domain Id
	3..0	SC	0..15	Sequence Counter
3	7..2	reserved	0	
	1	SGW	0 = SyncToGTM 1 = SyncToSubDomain	
	0	reserved	0	
4		User Byte 0	default: 0	
5		User Byte 1	default: 0	
6		reserved	0	
7		reserved	0	
8..11		OfsTimeSec		32 Bit offset time value in seconds
12..15		OfsTimeNSec		32 Bit offset time value in nanoseconds

Table 7.3: OFS not CRC secured message format

⌋*(RS_TS_20041, RS_TS_20044)*

[SWS_FrTSyn_00080] ⌈OFS CRC secured message format:

Byte	Bit Position	Field Name	Field Value Range	Description
0		Type	0x44	Message Type
1		CRC	0..255	Checksum
2	7..4	D	16..31	Time Domain Id
	3..0	SC	0..15	Sequence Counter
3	7..2	reserved	0	
	1	SGW	0 = SyncToGTM 1 = SyncToSubDomain	
	0	reserved	0	
4		User Byte 0	default: 0	
5		User Byte 1	default: 0	
6		reserved	0	
7		reserved	0	
8..11		OfsTimeSec		32 Bit offset time value in seconds
12..15		OfsTimeNSec		32 Bit offset time value in nanoseconds

Table 7.4: OFS CRC secured message format

⌋*(RS_TS_20041, RS_TS_20042, RS_TS_20044)*

7.4 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 Global Time Base, the Time Base from which all further Time Bases are derived from, then it is the Global Time Master. A Time Gateway typically consists of one Time Master port which is connected to one or more Time Slaves. 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.

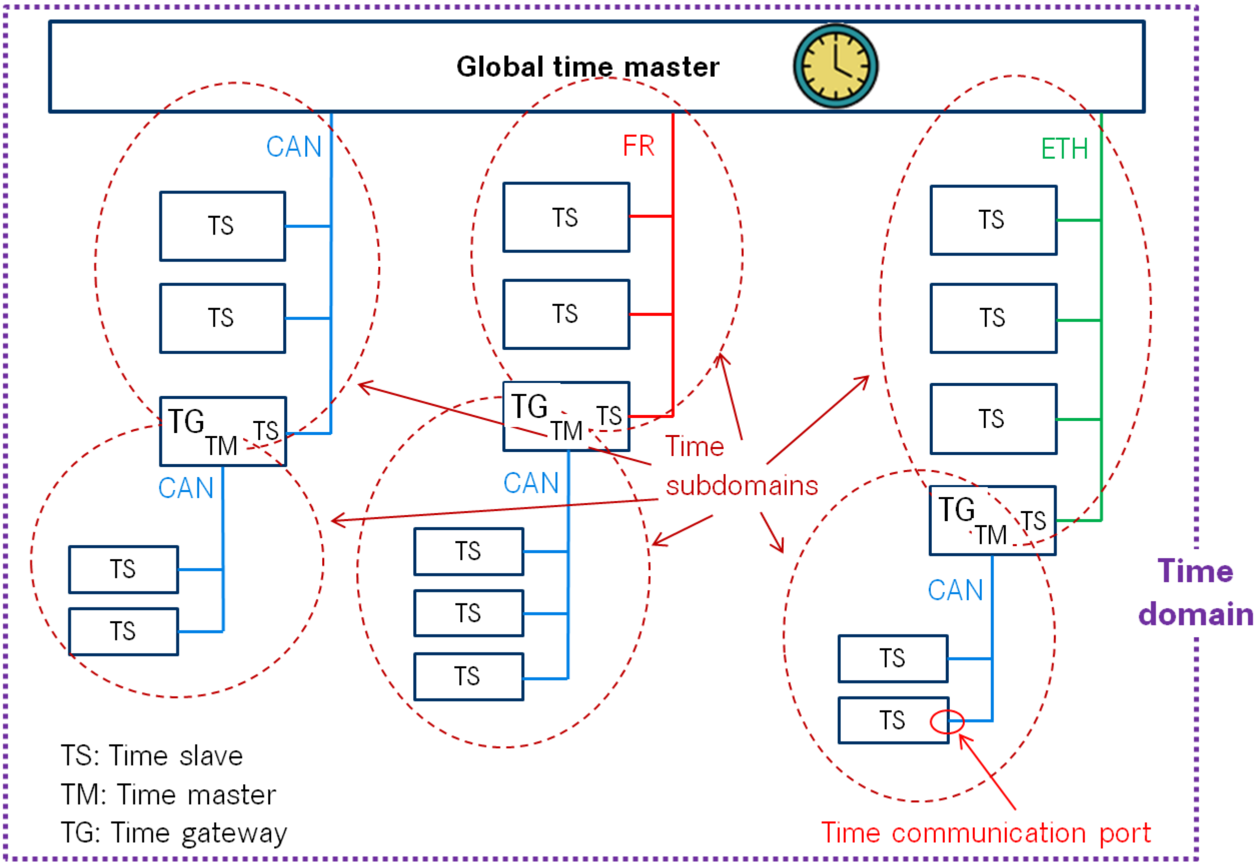


Figure 7.1: Terminology Example

7.4.1 SYNC message processing

[SWS_FrTSyn_00018] ⌈A Time Synchronization message sequence consists of a SYNC message per Time Domain.⌋*(RS_TS_20043)*

Note: Refer to figure 9.1 for the sequence diagram of a Time Master.

[SWS_FrTSyn_00019] ⌈For each configured Time Master (refer to FrTSynGlobalTimeMaster) the FrTSyn module shall periodically transmit SYNC messages with the cycle FrTSynGlobalTimeTxPeriod including the time value, which will be valid at the start of the next FlexRay cycle 0 and User Data, if the GLOBAL_TIME_BASE bit within the timeBaseStatus is set and FrTSynGlobalTimeTxPeriod is unequal to 0 and if the associated cyclicMsgResumeCounter is not running.

The cyclic transmission shall be started in the earliest possible FrTSyn_MainFunction call once the requirements above are fulfilled. ⌋*(RS_TS_20039, RS_TS_20043)*

Note: "earliest possible" means:

- In the next FrTSyn_MainFunction, because GLOBAL_TIME_BASE is set outside the FrTSyn_MainFunction.
- In the current FrTSyn_MainFunction, when switching from immediate to cyclic transmission (because this decision is made inside the FrTSyn_MainFunction). For details on immediate transmission refer to chapter 7.4.5.

[SWS_FrTSyn_00021] ⌈Depending on FrTSynGlobalTimeTxCrcSecured the SYNC message shall be of type:

FrTSynGlobalTimeTxCrcSecured **Value**	**SYNC Message Type**
CRC_NOT_SUPPORTED	0x10 SYNC not CRC secured message
CRC_SUPPORTED	0x20 SYNC CRC secured message

Table 7.5

⌋*(RS_TS_20041, RS_TS_20043)*

7.4.2 OFS message processing

[SWS_FrTSyn_00022] ⌈An offset message sequence consists of an OFS message per Time Domain.⌋*(RS_TS_20044)*

[SWS_FrTSyn_00023] ⌈For each configured Time Master (FrTSynGlobalTimeMaster) the FrTSyn module shall periodically transmit OFS messages with the cycle FrTSynGlobalTimeTxPeriod) including the Offset Time value and User Data, if the GLOBAL_TIME_BASE bit within the timeBaseStatus is set and FrTSynGlobalTimeTxPeriod is unequal to 0 and if the associated cyclicMsgResumeCounter is not running.

The cyclic transmission shall be started in the earliest possible FrTSyn_MainFunction call once the requirements above are fulfilled.⌋*(RS_TS_20039, RS_TS_20044)*

Note: "earliest possible" means:

- In the next FrTSyn_MainFunction, because GLOBAL_TIME_BASE is set outside the FrTSyn_MainFunction.

- In the current FrTSyn_MainFunction, when switching from immediate to cyclic transmission (because this decision is made inside the FrTSyn_MainFunction). For details on immediate transmission refer to chapter 7.4.5).

[SWS_FrTSyn_00025] ⌈Depending on FrTSynGlobalTimeTxCrcSecured the OFS message shall be of type:

FrTSynGlobalTimeTxCrcSecured **Value**	**OFS Message Type**
CRC_NOT_SUPPORTED	0x34 OFS not CRC secured message
CRC_SUPPORTED	0x44 OFS CRC secured message

Table 7.6

⌋*(RS_TS_20041, RS_TS_20044)*

7.4.3 Transmission mode

[SWS_FrTSyn_00026] ⌈If FrTSyn_SetTransmissionMode(Controller, Mode) is called and parameter Mode equals FRTSYN_TX_OFF, all transmit requests from FrTSyn shall be omitted on this FlexRay channel. ⌋*(RS_TS_20039, RS_TS_20043, RS_TS_20044)*

[SWS_FrTSyn_00027] ⌈If FrTSyn_SetTransmissionMode(Controller, Mode) is called and parameter Mode equals FRTSYN_TX_ON, all transmit requests from FrTSyn on this FlexRay channel shall be able to be transmitted. ⌋*(RS_TS_20039, RS_TS_20043, RS_TS_20044)*

7.4.4 Debounce Time

[SWS_FrTSyn_00084] ⌈If FrTSynGlobalTimeDebounceTime) is greater than 0 for a Time Base, FrTSyn shall always do debouncing for the corresponding Timesync PDUs as described below, otherwise FrTSyn shall not do any debouncing. ⌋*(RS_TS_20039)*

[SWS_FrTSyn_00085] ⌈FrTSynGlobalTimeDebounceTime represents the debounce value of a debounceCounter of a Time Base. FrTSyn shall reload the debounceCounter after a Timesync PDU for the corresponding Time Base (SYNC and OFS) has been sent. FrTSyn shall decrement the debounceCounter value on each invocation of FrTSyn_MainFunction, if no Timesync PDU is transmitted. ⌋*(RS_TS_20039)*

[SWS_FrTSyn_00086] ⌈A new Timesync PDU shall only be sent if the corresponding debounceCounter has a value equal or less than zero. ⌋*(RS_TS_20039)*

7.4.5 Immediate Time Synchronization

In addition to the cyclic Timesync message transmission, an immediate message transmission might be required.

Depending on configuration, the FrTSyn module checks on each FrTSyn_MainFunction call the necessity for a Timesync message transmission for each Time Base, where a Master Port belongs to.

[SWS_FrTSyn_00087] ⌈If FrTSynImmediateTimeSync is set to TRUE for a Time Base, FrTSyn shall check on each FrTSyn_MainFunction call by calling StbM_GetTimeBaseUpdateCounter, if the timeBaseUpdateCounter of the corresponding Time Base has changed.⌋*(RS_TS_20039)*

[SWS_FrTSyn_00088] ⌈If FrTSynImmediateTimeSync is set to TRUE and the timeBaseUpdateCounter of a Time Base has changed and the GLOBAL_TIME_BASE bit of the timeBaseStatus is set, FrTSyn shall trigger an immediate transmission of Time Synchronization messages for the corresponding Time Base.⌋*(RS_TS_20039)*

Note: timeBaseStatus can be obtained by StbM_GetTimeBaseStatus, StbM_BusGetCurrentTime or StbM_GetCurrentTime.

Note: The debounceCounter as described in 7.4.4 shall always be considered.

[SWS_FrTSyn_00089] ⌈If FrTSynImmediateTimeSync is set to TRUE, cyclicMsgResumeCounter and FrTSynCyclicMsgResumeTime shall be considered. ⌋*(RS_TS_20039)*

[SWS_FrTSyn_00090] ⌈FrTSynCyclicMsgResumeTime represents the timeout value of a cyclicMsgResumeCounter that shall be started when either a SYNC or OFS message has been sent immediately, asynchronous to the cyclic Timesync message transmission. cyclicMsgResumeCounter shall be decremented on each invocation of FrTSyn_MainFunction, if no Timesync PDU is transmitted asynchronously.⌋*(RS_TS_20039)*

[SWS_FrTSyn_00091] ⌈If the cyclicMsgResumeCounter has reached a value equal or less than zero, FrTSyn shall resume cyclic Timesync message transmission by sending either a SYNC or OFS message.⌋*(RS_TS_20039)*

[SWS_FrTSyn_00093] ⌈If the cyclicMsgResumeCounter is started, FrTSyn shall stop cyclic Timesync message transmission.⌋*(RS_TS_20039)*

7.4.6 Calculation and Assembling of Time Synchronization Messages

This chapter describes the workflow, how the items of a Time Synchronization message will be calculated (1^{st} step) and how the message will be assembled (2^{nd} step).

7.4.6.1 Global Time Calculation

[SWS_FrTSyn_00028] ⌈The transmitter of a Synchronized Time Base (Time Master) shall perform the following steps to distribute the Synchronized Time Base:

1. Retrieve current Synchronized Time Base's Time Tuple as [T_{SYNC};$T0_{VLT}$] via `StbM_BusGetCurrentTime`
2. Protect the following two steps against interruptions:
   (a) Get `currentCycle` and `currentMacroticks` via `FrIf_GetGlobalTime`
   (b) Retrieve current Virtual Local Time value as $T1_{VLT}$ via `StbM_GetCurrentVirtualLocalTime`
3. Calculate the (future) time value of the Time Base at the start of the next FlexRay cycle by T0 = T_{SYNC} + ($T1_{VLT}$ - $T0_{VLT}$) + (64 - `currentCycle`) * `CycleLength` - (`CycleLength` / `MacroticksPerCycle`) * `currentMacroticks`
4. Calculate `SyncTimeSec` (second portion of T0) and `SyncTimeNSec` (nanosecond portion of T0)

⌋*(RS_TS_20043)*

Note: Refer to figure 9.1 for the Time Master sequence of actions.

Note: It is inevitable to retrieve `currentCycle` and `currentMacroticks` of the FlexRay time and $T1_{VLT}$ of the Virtual Local Time in an atomic way, otherwise any delay between them will worsen the precision by the amount of the delay.

Note: If the calculation is done on an integer basis the rounding error of the term (`CycleLength` / `MacroticksPerCycle`) needs to be minimized. This can be done in multiple ways, e.g., by calculating

- ((`CycleLength` * `currentMacroticks`) / `MacroticksPerCycle`) on 64 bit architectures or
- ((((`CycleLength` * 256) / `MacroticksPerCycle`) * `currentMacroticks`) / 256) on 32 bit architectures (multiplication by 256 is acceptable for any possible FlexRay parameter configuration)

`CycleLength` and `MacroticksPerCycle` are retrieved via `FrIf_GetCycleLength` and `FrIf_GetMacroticksPerCycle`.

[SWS_FrTSyn_00029] ⌈The transmitter of an Offset Time Base (Time Master) shall perform the following steps to distribute the Offset Time Base:

1. Retrieve current Offset Time via `StbM_GetOffset`
2. Write second portion of the Offset Time to `OfsTimeSec`
3. Write nanosecond portion of the Offset Time to `OfsTimeNSec`

⌋(*RS_TS_20044*)

7.4.6.2 SGW Calculation

[SWS_FrTSyn_00020] ⌈The SGW value (Time Gateway synchronization status) shall be retrieved from the Time Base synchronization status. If the SYNC_TO_GATEWAY bit within timeBaseStatus is not set the SGW value shall be SyncToGTM. Otherwise the SGW value shall be set to SyncToSubDomain.⌋(*RS_TS_20043, RS_TS_20044*)

7.4.6.3 Sequence Counter Calculation

[SWS_FrTSyn_00030] ⌈A Sequence Counter (SC) of 4 bit is representing numbers from 0 to 15 per Time Domain. The Sequence Counter shall be independent between SYNC and OFS messages and shall be incremented by 1 on every transmission request of a SYNC or OFS message. It shall wrap around at 15 to 0 again. ⌋(*RS_TS_20041, RS_TS_20043, RS_TS_20044*)

7.4.6.4 CRC Calculation

[SWS_FrTSyn_00031] ⌈The function Crc_CalculateCRC8H2F as defined in [6] shall be used to calculate the CRC, if configured. ⌋(*RS_TS_20041, RS_TS_20043, RS_TS_20044*)

[SWS_FrTSyn_00035] ⌈The DataID shall be calculated as DataID = DataIDList[SC], where DataIDList is given by configuration for each message type (refer to FrTSynGlobalTimeSyncDataIDList and FrTSynGlobalTimeOfsDataIDList).⌋(*RS_TS_20041, RS_TS_20043, RS_TS_20044*)

Note: A specific DataID out of a predefined DataIDList ensures the identification of data elements of Time Synchronization messages.

[SWS_FrTSyn_00036] ⌈The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID, where Byte 2 is applied first, followed by the other bytes in ascending order, and DataID last. ⌋(*RS_TS_20041, RS_TS_20043, RS_TS_20044*)

7.4.6.5 Message Assembling

[SWS_FrTSyn_00037] ⌈For each transmission of a Time Synchronization message the FrTSyn module shall assemble the message as follows:

- Calculate SC

- Copy currentCycle (refer to [SWS_FrTSyn_00028]) to FCNT (for SYNC message)
- Calculate SGW
- Copy all data to the appropriate position within the related message
- Calculate CRC (configuration dependent)

⌋*(RS_TS_20043, RS_TS_20044)*

7.5 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.

7.5.1 SYNC message processing

[SWS_FrTSyn_00038] ⌈The FrTSyn shall only accept a SYNC message with Type equal to 0x20 and a correct CRC value if FrTSynRxCrcValidated is configured to CRC_VALIDATED. ⌋*(RS_TS_20042, RS_TS_20043)*

[SWS_FrTSyn_00039] ⌈The FrTSyn shall only accept a SYNC message with Type equal to 0x10 if FrTSynRxCrcValidated is configured to CRC_NOT_VALIDATED.⌋ *(RS_TS_20043)*

[SWS_FrTSyn_00040] ⌈The FrTSyn shall only accept a SYNC message with Type equal to 0x10 or 0x20 if FrTSynRxCrcValidated is configured to CRC_IGNORED.⌋ *(RS_TS_20043)*

[SWS_FrTSyn_00081] ⌈The FrTSyn shall only accept a SYNC message with Type equal to 0x10 or a SYNC message with Type equal to 0x20 and a correct CRC value if FrTSynRxCrcValidated is configured to CRC_OPTIONAL. ⌋*(RS_TS_20042, RS_TS_20043)*

[SWS_FrTSyn_00041] ⌈For valid SYNC messages a new Time Tuple, consisting of the Global Time value and the associated value of the Virtual Local Time, shall be calculated and forwarded to the StbM module via StbM_BusSetGlobalTime. ⌋*(RS_TS_20040, RS_TS_20042, RS_TS_20043)*

7.5.2 OFS message processing

[SWS_FrTSyn_00042] ⌈The FrTSyn shall only accept an OFS message with Type equal to 0x44 and a correct CRC value if FrTSynRxCrcValidated is configured to CRC_VALIDATED. ⌋*(RS_TS_20042, RS_TS_20044)*

[SWS_FrTSyn_00043] ⌈The FrTSyn shall only accept an OFS message with Type equal to 0x34 if FrTSynRxCrcValidated is configured to CRC_NOT_VALIDATED.⌋ *(RS_TS_20044)*

[SWS_FrTSyn_00044] ⌈The FrTSyn shall only accept an OFS message with Type equal to 0x34 or 0x44 if FrTSynRxCrcValidated is configured to CRC_IGNORED. ⌋*(RS_TS_20044)*

[SWS_FrTSyn_00082] ⌈The FrTSyn shall only accept an OFS message with Type equal to 0x34 or an OFS message with Type equal to 0x44 and a correct CRC value if FrTSynRxCrcValidated is configured to CRC_OPTIONAL. ⌋*(RS_TS_20042, RS_TS_20044)*

[SWS_FrTSyn_00045] ⌈For valid OFS messages a new Time Tuple, consisting of the Offset Time value and the associated value of the Virtual Local Time, shall be calculated (according [SWS_FrTSyn_00047]) and forwarded to the StbM module via StbM_BusSetGlobalTime.⌋*(RS_TS_20040, RS_TS_20042, RS_TS_20044)*

7.5.3 Validation and Disassembling of Time Synchronization Messages

This chapter describes the workflow how the items of a Time Synchronization message will be validated (1st step) and how the message will be disassembled (2nd step).

7.5.3.1 Global Time Calculation

[SWS_FrTSyn_00046] ⌈The receiver of a Synchronized Time Base shall perform the following steps to assemble the Synchronized Time Base:

1. On SYNC message RX indication (or in the subsequent MainFunction call) store received time value T0 (SyncTimeSec, SyncTimeNSec)
2. Protect the following two steps against interruptions:
   (a) Get currentCycle and currentMacroticks via FrIf_GetGlobalTime
   (b) Retrieve current Virtual Local Time value as $T1_{VLT}$ via StbM_GetCurrentVirtualLocalTime
3. Calculate Time Tuple [T1; $T1_{VLT}$] to update the Time Slave's local instance of the Time Base:
   (a) T1 = T0 + (CycleLength * currentCycle) + ((CycleLength / MacrotickPerCycle) * currentMacroticks)

(b) If currentCycle is greater or equal than the retrieved FCNT value from the transmitter (Time Master), then the calculated value T1 shall be subtracted by 64 times the FlexRay cycle duration: T1 = T1 - (CycleLength * 64)

⌋*(RS_TS_20043)*

Note: Refer to figure 9.2 for the Time Slave sequence of actions.

Note: It is inevitable to retrieve currentCycle and currentMacroticks of the FlexRay time and $T1_{VLT}$ of the Virtual Local Time atomic, otherwise any delay between them will worsen the precision by the amount of the delay.

Note: In order to minimize rounding errors for the term (CycleLength / MacrotickPerCycle) in case of integer calculation refer to note below [SWS_FrTSyn_00028].

[SWS_FrTSyn_00047] ⌈The receiver of an Offset Time Base shall perform the following steps to assemble the Offset Time:

1. Get second portion of the Offset Time out of OfsTimeSec
2. Get nanosecond portion of the Offset Time out of OfsTimeNSec
3. Retrieve current Virtual Local Time value via StbM_GetCurrentVirtualLocalTime

⌋*(RS_TS_20044)*

7.5.3.2 SGW Calculation

[SWS_FrTSyn_00094] ⌈If the SGW value (SYNC and OFS) is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to TRUE. Otherwise, it shall be set to FALSE.⌋*(RS_TS_20040, RS_TS_20042)*

7.5.3.3 Sequence Counter Validation

[SWS_FrTSyn_00048] ⌈The Sequence Counter Jump Width between two consecutive SYNC or two consecutive OFS messages of the same Time Domain shall be greater than 0 and smaller than or equal to FrTSynGlobalTimeSequenceCounterJumpWidth. Otherwise a Time Slave shall discard the respective SYNC / OFS message.

If the FrTSynGlobalTimeSequenceCounterJumpWidth value is set to 0, the Time Slave shall not do Sequence Counter Jump Width checks.⌋*(RS_TS_20042, RS_TS_20043, RS_TS_20044)*

[SWS_FrTSyn_00049] ⌈Upon reception of a SYNC (or OFS) message a Time Slave shall check the Sequence Counter of the received message per Time Domain against the configured value of FrTSynGlobalTimeSequenceCounterJumpWidth (according to [SWS_FrTSyn_00048]), unless it is the first message

- at Startup or
- after a Time Base update timeout has been detected (TIMEOUT bit set in Time Base synchronization status timeBaseStatus).

⌋*(RS_TS_20042, RS_TS_20043, RS_TS_20044)*

Note: There are scenarios where it makes sense to skip the check of the Sequence Counter Jump Width, e.g. at startup (Time Slaves start asynchronously to the Time Master) or after a message timeout to allow for Sequence Counter (re-)synchronization. In case of a timeout the error has been detected already by the timeout monitoring, there is no benefit in generating a subsequent error by the jump width check.

Note: According to [SWS_FrTSyn_00048] the Sequence Counter validation will still discard messages with a Sequence Counter Jump Width being zero (i.e., stuck Sequence Counter) during Time Base update timeout.

[SWS_FrTSyn_00102] ⌈While a Time Base Timeout is present (TIMEOUT bit is set in Time Base synchronization status timeBaseStatus), FrTSyn shall discard SYNC (or OFS) messages until it has successfully validated (refer to [SWS_FrTSyn_00048]) 'n' consecutive SYNC (or OFS) messages ('n' is given by the parameter FrTSynGlobalTimeSequenceCounterHysteresis).⌋*(RS_TS_20042)*

Note: [SWS_FrTSyn_00102] improves robustness against a scenario with a buggy master implementation or injection of invalid master messages (sequence counter increments greater than FrTSynGlobalTimeSequenceCounterJumpWidth. In such a scenario any valid SYNC (or OFS) message would cause the Time Slave to leave the Timeout state (refer to [SWS_FrTSyn_00049]) although the sequence counter is not incremented correctly. An additional hysteresis avoids this.

7.5.3.4 CRC Validation

[SWS_FrTSyn_00050] ⌈The function Crc_CalculateCRC8H2F as defined in [6] shall be used to validate the CRC, if configured. ⌋*(RS_TS_20042, RS_TS_20043, RS_TS_20044)*

[SWS_FrTSyn_00054] ⌈The DataID shall be calculated as DataID = DataIDList [SC], where DataIDList is given by configuration for each message Type. ⌋*(RS_TS_20042, RS_TS_20043, RS_TS_20044)*

Note: A specific DataID out of a predefined DataIDList ensures the identification of data elements of Time Synchronization messages.

[SWS_FrTSyn_00055] ⌈The CRC shall be calculated over Time Synchronization message Byte 2 to Byte 15 and DataID, where Byte 2 is applied first, followed by the other bytes in ascending order, and DataID last. ⌋*(RS_TS_20042, RS_TS_20043, RS_TS_20044)*

7.5.3.5 Message Disassembling

[SWS_FrTSyn_00056] ⌈For each received Time Synchronization message the FrTSyn shall validate the message as follows (all conditions must match):

1. Type matches depending on the FrTSynRxCrcValidated parameter
2. SC value is within the accepted range (refer to [SWS_FrTSyn_00048] and [SWS_FrTSyn_00049])
3. D matches to the defined Time Domain range for each Type
4. D matches to one of the configured Time Domains
5. SyncTimeNSec (SYNC message) or OfsTimeNSec (OFS message) matches the defined range of StbM_TimeStampType.nanoseconds.
6. CRC (including DataID) matches depending on the FrTSynRxCrcValidated parameter.

⌋*(RS_TS_20043, RS_TS_20044)*

[SWS_FrTSyn_00057] ⌈For each received Time Synchronization message the FrTSyn shall disassemble the message after successful validation [SWS_FrTSyn_00056].⌋*(RS_TS_20042, RS_TS_20043, RS_TS_20044)*

7.6 Time Recording

7.6.1 Global Time Measurement Support

[SWS_FrTSyn_00092] ⌈On an invocation of StbM_BusSetGlobalTime the member pathDelay of the measureDataPtr structure shall be set to 0.⌋*(RS_TS_00034)*

7.6.2 Time Validation

[SWS_FrTSyn_00096] ⌈The FrTSyn shall support Time Validation, if FrTSynTimeValidationSupport set to TRUE.⌋*(RS_TS_00034)*

[SWS_FrTSyn_00097] ⌈
If

- FrTSynTimeValidationSupport is enabled and
- FrTSynEnableTimeValidation for the Time Domain is enabled

FrTSyn shall do time recording for Time Validation for that Time Domain.
⌋*(RS_TS_00034)*

[SWS_FrTSyn_00098] ⌈
If

- time recording for Time Validation is enabled for a Time Domain (refer to [SWS_FrTSyn_00096] and [SWS_FrTSyn_00097]) and
- FrTSyn is configured as Time Slave for that Time Domain,

FrTSyn shall call StbM_FrSetSlaveTimingData upon successful reception of a SYNC message.

StbM_FrSetSlaveTimingData shall be called after StbM_BusSetGlobalTime.⌋ *(RS_TS_00034)*

Note: StbM_BusSetGlobalTime shall be called first, because it updates the Sync-local Time Tuple (refer to [5]), which is required by StbM_FrSetSlaveTimingData. Refer to Figure 9.2 for the overall sequence of API calls for a Time Slave.

[SWS_FrTSyn_00099] ⌈Upon invocation of StbM_FrSetSlaveTimingData FrTSyn shall pass following values

- the Sequence Counter as received in the SYNC message,
- the segment id of the physical channel on which the SYNC message has been received (refer to parameter FrTSynGlobalTimeNetworkSegmentId)
- currentCycle and currentMacroticks and FCNT as read upon reception of the SYNC message (refer to step 2 in [SWS_FrTSyn_00046]),
- CycleLength and MacrotickDuration
- the Sync ingress timestamp $T1_{VLT}$ as retrieved in step 1 in [SWS_FrTSyn_00046])
- T0 as received in the SYNC message (refer to step 1 in [SWS_FrTSyn_00046]),

by the parameter measureDataPtr.

Struct members

- measureDataPtr→referenceLocalTimestamp and
- measureDataPtr→referenceGlobalTimestamp

shall be passed as 0.⌋*(RS_TS_00034)*

Note: MacrotickDuration is calculated as CycleLength / MacroticksPerCycle

Note: The FrTSyn passes 0 to avoid undefined values. The structure members referenceLocalTimestamp and referenceGlobalTimestamp will be set by the StbM StbM_FrSetSlaveTimingData internally (refer to [SWS_StbM_00471] in [5]).

[SWS_FrTSyn_00100] ⌈
If

- time recording for Time Validation is enabled for a Time Domain (refer to [SWS_FrTSyn_00096] and [SWS_FrTSyn_00097]) and
- FrTSyn is configured as Time Master for that Time Domain,

FrTSyn shall call StbM_FrSetMasterTimingData upon successful transmission of a SYNC message.⌋*(RS_TS_00034)*

Note: Refer to Figure 9.1 for the overall sequence of API calls for a Time Master.

[SWS_FrTSyn_00101] ⌈Upon invocation of StbM_FrSetMasterTimingData FrTSyn shall pass the following data

- the Sequence Counter as sent in the SYNC message
- the segment id of the physical channel on which the SYNC message has been sent (refer to parameter FrTSynGlobalTimeNetworkSegmentId)
- currentCycle and currentMacroticks read upon construction of the Sync message (refer to step 2 in [SWS_FrTSyn_00028]),
- cycleLength and macrotickDuration
- the reference timestamp $T1_{VLT}$ (refer to step 2 In [SWS_FrTSyn_00028]),
- T0 as sent in the SYNC message (refer to step 3 In [SWS_FrTSyn_00028]),

by the parameter measureDataPtr.⌋*(RS_TS_00034)*

7.7 Error Classification

Section 7.2 "Error Handling" of the document "General Specification of Basic Software Modules" [3] 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

[SWS_FrTSyn_00059] ⌈

Type of error	*Related error code*	*Error value*
API service called with wrong PDU or SDU.	FRTSYN_E_INVALID_PDUID	0x01
API service used in un-initialized state	FRTSYN_E_UNINIT	0x20
A pointer is invalid	FRTSYN_E_NULL_POINTER	0x21

▽

△

Type of error	*Related error code*	*Error value*
FrTSyn initialization failed	FRTSYN_E_INIT_FAILED	0x22
API called with invalid parameter	FRTSYN_E_PARAM	0x23
Invalid Controller index	FRTSYN_E_INV_CTRL_IDX	0x24

⌋(*SRS_BSW_00385*)

7.7.2 Runtime Errors

There are no runtime errors.

7.7.3 Transient Faults

There are no transient faults.

7.7.4 Production Errors

There are no production errors.

7.7.5 Extended Production Errors

There are no extended production errors.

8 API specification

8.1 API

8.1.1 Imported types

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

[SWS_FrTSyn_00060] ⌈

Module	*Header File*	*Imported Type*
ComStack_Types	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType

▽

△

Module	***Header File***	***Imported Type***
	ComStack_Types.h	PduLengthType
FrIf	FrIf.h	FrIf_StateType
StbM	Rte_StbM_Type.h	StbM_FrTimeMasterMeasurementType
	Rte_StbM_Type.h	StbM_FrTimeSlaveMeasurementType
	Rte_StbM_Type.h	StbM_SynchronizedTimeBaseType
	Rte_StbM_Type.h	StbM_TimeBaseStatusType
	Rte_StbM_Type.h	StbM_TimeStampShortType
	Rte_StbM_Type.h	StbM_TimeStampType
	Rte_StbM_Type.h	StbM_UserDataType
	StbM.h	StbM_MeasurementType
	StbM.h	StbM_VirtualLocalTimeType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋*(RS_TS_20043)*

8.1.2 Type definitions

8.1.2.1 FrTSyn_ConfigType

[SWS_FrTSyn_00061] ⌈

Name	FrTSyn_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	–
Description	This is the base type for the configuration of the Time Synchronization over FlexRay. A pointer to an instance of this structure will be used in the initialization of the Time Synchronization over FlexRay. The content of this structure is defined in chapter 10 Configuration specification.	
Available via	FrTSyn.h	

⌋*(RS_TS_20043)*

8.1.2.2 FrTSyn_TransmissionModeType

[SWS_FrTSyn_00062] ⌈

Name	FrTSyn_TransmissionModeType		
Kind	Enumeration		
Range	FRTSYN_TX_OFF	–	Transmission Disabled
	FRTSYN_TX_ON	–	Transmission Enabled
Description	Handles the enabling and disabling of the transmission mode		
Available via	FrTSyn.h		

⌋(*RS_TS_20043*)

8.1.3 Function definitions

8.1.3.1 FrTSyn_Init

[SWS_FrTSyn_00063] ⌈

Service Name	FrTSyn_Init	

Syntax	`void FrTSyn_Init (`
	`  const FrTSyn_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 FlexRay.	
Available via	FrTSyn.h	

⌋(*RS_TS_20043*)

See section 7.2.1 for details.

8.1.3.2 FrTSyn_GetVersionInfo

[SWS_FrTSyn_00064] ⌈

Service Name	FrTSyn_GetVersionInfo

Syntax	`void FrTSyn_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.	
Available via	FrTSyn.h	

⌋*(RS_TS_20043)*

8.1.3.3 FrTSyn_SetTransmissionMode

[SWS_FrTSyn_00065] ⌈

Service Name	FrTSyn_SetTransmissionMode	

Syntax	`void FrTSyn_SetTransmissionMode (`
	`  uint8 CtrlIdx,`
	`  FrTSyn_TransmissionModeType Mode`
	`)`	

Service ID [hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Index of the FlexRay channel
	Mode	FRTSYN_TX_OFF FRTSYN_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 FrTSyn.	
Available via	FrTSyn.h	

⌋*(RS_TS_20043)*

[SWS_FrTSyn_00095] ⌈The function `FrTSyn_SetTransmissionMode` shall inform the `Det`, if development error detection is enabled (`FrTSynDevErrorDetect` is set to `TRUE`) and if function call has failed because of the following reasons:

- Invalid CtrlIdx (FRTSYN_E_INV_CTRL_IDX)
- Invalid Mode (FRTSYN_E_PARAM)

⌋*(SRS_BSW_00323, SRS_BSW_00337)*

8.1.4 Call-back notifications

This is a list of functions provided for other modules.

8.1.4.1 FrTSyn_RxIndication

[SWS_FrTSyn_00066] ⌈

Service Name	FrTSyn_RxIndication	

Syntax	`void FrTSyn_RxIndication (`
	`  PduIdType RxPduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`	

Service ID [hex]	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indication of a received PDU from a lower layer communication interface module.	
Available via	FrTSyn.h	

⌋*(RS_TS_20043)*

Note: The callback function FrTSyn_RxIndication called by the FrIf module and implemented by the FrTSyn module. It is called in case of a receive indication event of the FR Driver.

[SWS_FrTSyn_00067] ⌈The callback function FrTSyn_RxIndication shall inform the Det, if development error detection is enabled FrTSynDevErrorDetect is set to TRUE) and if function call has failed because of the following reasons:

- Invalid RxPduId (FRTSYN_E_INVALID_PDUID)
- PduInfoPtr or SduDataPtr equals NULL_PTR (FRTSYN_E_NULL_POINTER)

⌋*(SRS_BSW_00323, SRS_BSW_00337)*

Caveats of FrTSyn_RxIndication

- The FrTSyn module is initialized correctly.

8.1.4.2 FrTSyn_TriggerTransmit

[SWS_FrTSyn_00069] ⌈

Service Name	FrTSyn_TriggerTransmit	

Syntax	`Std_ReturnType FrTSyn_TriggerTransmit (`
	`  PduIdType TxPduId,`
	`  PduInfoType* PduInfoPtr`
	`)`	

Service ID [hex]	0x41	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	ID of the SDU that is requested to be transmitted.
Parameters (inout)	PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
Description	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr.	
Available via	FrTSyn.h	

⌋*(RS_TS_20043)*

Note: The function `FrTSyn_TriggerTransmit` might be called by the `FrTSyn` module's environment in an interrupt context.

[SWS_FrTSyn_00070] ⌈The callback function `FrTSyn_TriggerTransmit` shall inform the `Det`, if development error detection is enabled `FrTSynDevErrorDetect` is set to `TRUE`) and if function call has failed because of the following reasons:

- Invalid `TxPduId` (FRTSYN_E_INVALID_PDUID)
- `PduInfoPtr` or `SduDataPtr` equals `NULL_PTR` (FRTSYN_E_NULL_POINTER)

⌋*(SRS_BSW_00323, SRS_BSW_00337)*

8.1.5 Scheduled functions

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

8.1.5.1 FrTSyn_MainFunction

[SWS_FrTSyn_00071] ⌈

Service Name	FrTSyn_MainFunction

Syntax	`void FrTSyn_MainFunction (`
	`  void`
	`)`

Service ID [hex]	0x04
Description	Main function for cyclic call / resp. Timesync message transmission
Available via	FrTSyn_SchM.h

⌋*(RS_TS_20043)*

[SWS_FrTSyn_00072] ⌈The frequency of invocations of `FrTSyn_MainFunction` is determined by the configuration parameter `FrTSynMainFunctionPeriod`. ⌋*(RS_TS_20043)*

8.1.6 Expected Interfaces

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

8.1.6.1 Mandatory Interfaces

This section defines all interfaces that are required to fulfill a mandatory functionality of the module.

[SWS_FrTSyn_00074] ⌈

API Function	***Header File***	***Description***
FrIf_GetCycleLength	FrIf.h	This API returns the configured time of the configuration parameter "GdCycle" in nanoseconds for the FlexRay controller with index FrIf_CtrlIdx.
FrIf_GetGlobalTime	FrIf.h	Wraps the FlexRay Driver API function Fr_Get GlobalTime(). Important Note: FrIf_GetGlobalTime may be called within an exclusive area.
FrIf_GetMacrotickDuration	FrIf.h	Retrieves the Duration of a Macrotick in ns
FrIf_GetMacroticksPerCycle	FrIf.h	Retrieves the amount of Macroticks per Cycle
FrIf_GetState	FrIf.h	Get current FrIf state.
StbM_GetCurrentVirtualLocalTime	StbM.h	Returns the Virtual Local Time of the referenced Time Base.

⌋*(RS_TS_20043)*

8.1.6.2 Optional Interfaces

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

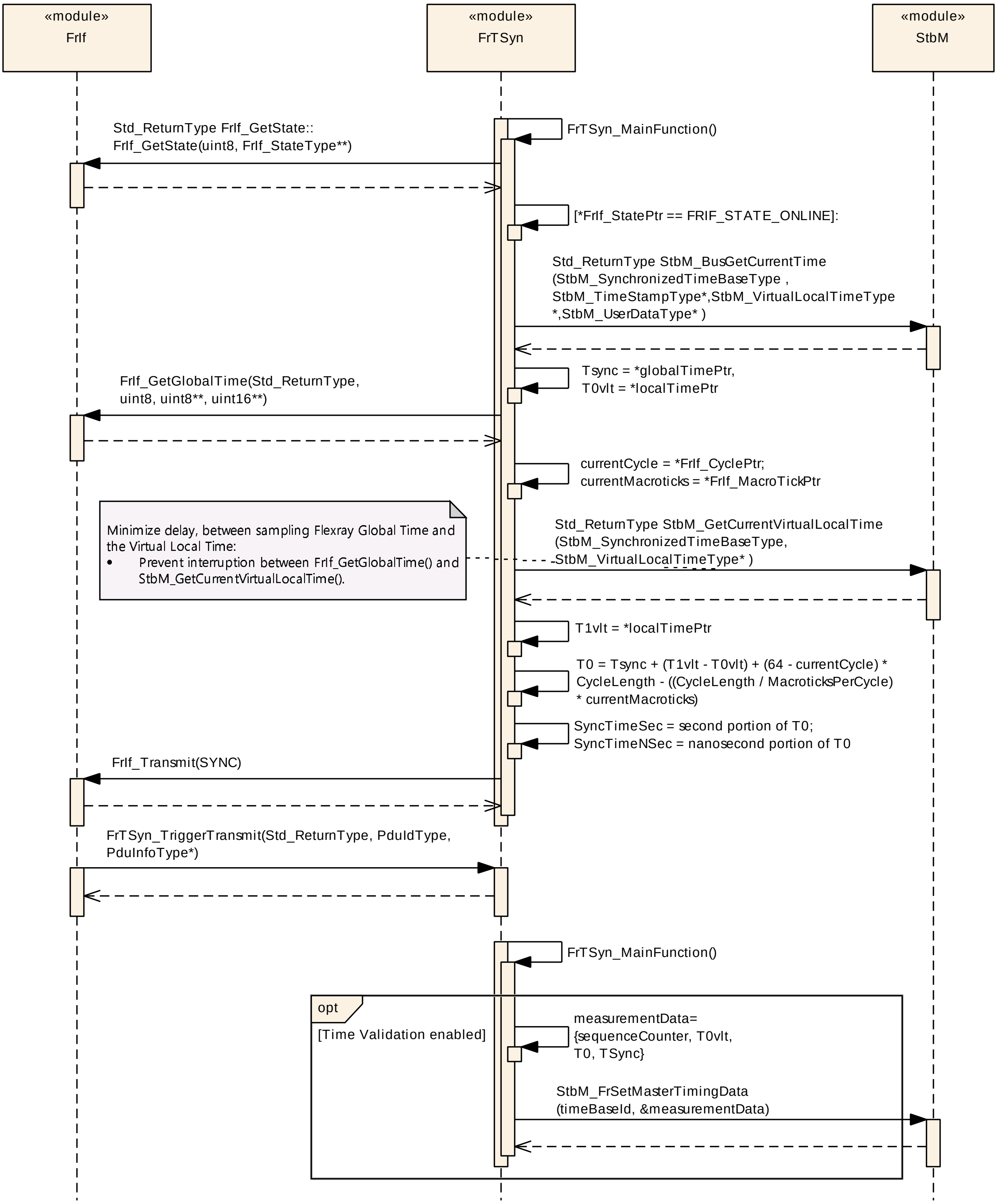
[SWS_FrTSyn_00075] ⌈

API Function	***Header File***	***Description***
Crc_CalculateCRC8H2F	Crc.h	This service makes a CRC8 calculation with the Polynomial 0x2F on Crc_Length
Det_ReportError	Det.h	Service to report development errors.
FrIf_Transmit	FrIf.h	Requests transmission of a PDU.
StbM_BusGetCurrentTime	StbM.h	Returns the current Time Tuple, status and User Data of the Time Base.
StbM_BusSetGlobalTime	StbM.h	Allows the Time Base Provider Modules to forward a new Global Time tuple (i.e., Rx Time Tuple) to the StbM.
StbM_FrSetMasterTimingData (draft)	StbM_FrTSyn.h	Provides Flexray Timesyn module specific data for a Time Master to the StbM. **Tags:** atp.Status=draft
StbM_FrSetSlaveTimingData (draft)	StbM_FrTSyn.h	Allows the FrTSyn Module to forward Flexray specific details to the StbM. **Tags:** atp.Status=draft
StbM_GetCurrentTime	StbM.h	Returns a time value (Local Time Base derived from Global Time Base) in standard format. 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).
StbM_GetOffset	StbM.h	Allows the Timesync Modules to get the current Offset Time and User Data.
StbM_GetTimeBaseStatus	StbM.h	Returns detailed status information for a Synchronized (or Pure Local) Time Base and, if called for an Offset Time Base, for the Offset Time Base and the underlying Synchronized Time Base.
StbM_GetTimeBaseUpdateCounter	StbM.h	Allows the Timesync Modules to detect, whether a Time Base should be transmitted immediately in the subsequent <Bus>TSyn_MainFunction() cycle.

⌋*(RS_TS_20043)*

9 Sequence diagrams

9.1 FlexRay Time Synchronization (Time Master)


«module»
FrIf
«module»
FrTSyn
«module»
StbM
FrTSyn_MainFunction()
Std_ReturnType FrIf_GetState::
FrIf_GetState(uint8, FrIf_StateType**)
[*FrIf_StatePtr == FRIF_STATE_ONLINE]:
Std_ReturnType StbM_BusGetCurrentTime
(StbM_SynchronizedTimeBaseType ,
StbM_TimeStampType*,StbM_VirtualLocalTimeType
,StbM_UserDataType)
Tsync = *globalTimePtr,
T0vlt = *localTimePtr
FrIf_GetGlobalTime(Std_ReturnType,
uint8, uint8**, uint16**)
currentCycle = *FrIf_CyclePtr;
currentMacroticks = *FrIf_MacroTickPtr
Std_ReturnType StbM_GetCurrentVirtualLocalTime
(StbM_SynchronizedTimeBaseType,
StbM_VirtualLocalTimeType*)
T1vlt = *localTimePtr
T0 = Tsync + (T1vlt - T0vlt) + (64 - currentCycle) *
CycleLength - ((CycleLength / MacroticksPerCycle)
* currentMacroticks)
SyncTimeSec = second portion of T0;
SyncTimeNSec = nanosecond portion of T0
FrIf_Transmit(SYNC)
FrTSyn_TriggerTransmit(Std_ReturnType, PduIdType,
PduInfoType*)
FrTSyn_MainFunction()
opt
[Time Validation enabled]
measurementData=
{sequenceCounter, T0vlt,
T0, TSync}
StbM_FrSetMasterTimingData
(timeBaseId, &measurementData)

Minimize delay, between sampling Flexray Global Time and the Virtual Local Time:

- Prevent interruption between FrIf_GetGlobalTime() and StbM_GetCurrentVirtualLocalTime().

Figure 9.1: FlexRay Time Synchronization (Time Master)

9.2 FlexRay Time Synchronization (Time Slave)

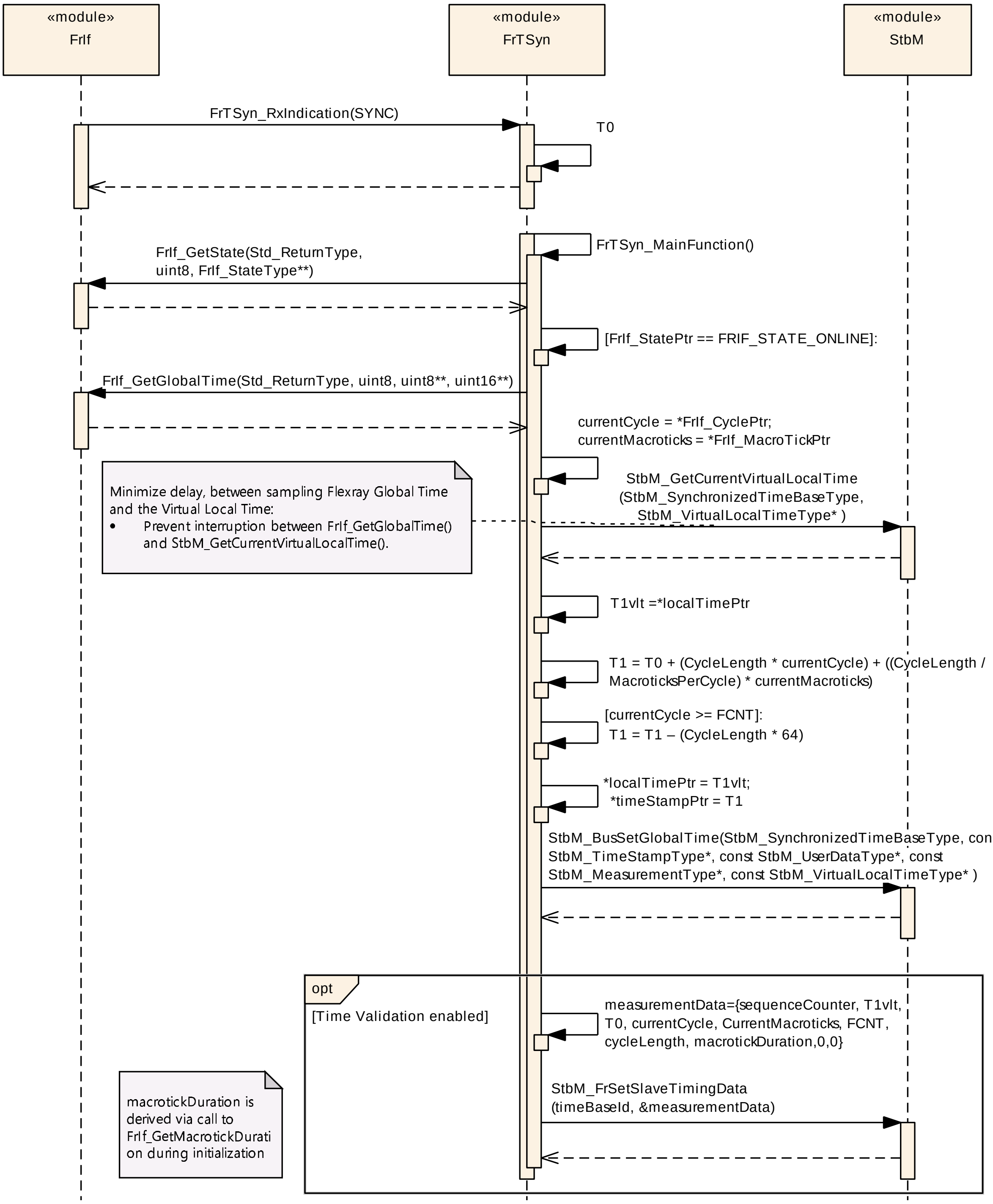

«module»
FrIf
«module»
FrTSyn
«module»
StbM
FrTSyn_RxIndication(SYNC)
T0
FrTSyn_MainFunction()
FrIf_GetState(Std_ReturnType,
uint8, FrIf_StateType**)
[FrIf_StatePtr == FRIF_STATE_ONLINE]:
FrIf_GetGlobalTime(Std_ReturnType, uint8, uint8**, uint16**)
currentCycle = *FrIf_CyclePtr;
currentMacroticks = *FrIf_MacroTickPtr
StbM_GetCurrentVirtualLocalTime
(StbM_SynchronizedTimeBaseType,
StbM_VirtualLocalTimeType*)
Minimize delay, between sampling Flexray Global Time and the Virtual Local Time:
• Prevent interruption between FrIf_GetGlobalTime() and StbM_GetCurrentVirtualLocalTime().
T1vlt =*localTimePtr
T1 = T0 + (CycleLength * currentCycle) + ((CycleLength / MacroticksPerCycle) * currentMacroticks)
[currentCycle >= FCNT]:
T1 = T1 – (CycleLength * 64)
*localTimePtr = T1vlt;
*timeStampPtr = T1
StbM_BusSetGlobalTime(StbM_SynchronizedTimeBaseType, const StbM_TimeStampType*, const StbM_UserDataType*, const StbM_MeasurementType*, const StbM_VirtualLocalTimeType*)
opt
[Time Validation enabled]
measurementData={sequenceCounter, T1vlt, T0, currentCycle, CurrentMacroticks, FCNT, cycleLength, macrotickDuration,0,0}
StbM_FrSetSlaveTimingData
(timeBaseId, &measurementData)
macrotickDuration is derived via call to FrIf_GetMacrotickDurati on during initialization

Figure 9.2: FlexRay Time Synchronization (Time Slave)

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 10.2 specifies the structure (containers) and the parameters of the Time Synchronization over FlexRay.

Section 10.3 specifies published information of the Time Synchronization over FlexRay.

10.1 How to read this chapter

For details, refer to the chapter 10.1 "Introduction to configuration specification" in SWS BSW General [3].

10.2 Containers and configuration parameters

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

10.2.1 Variants

[SWS_FrTSyn_00077] ⌈The Time Synchronization over FlexRay shall support the configuration for Time Master, Time Slave and Time Gateway.⌋*(RS_TS_20046)*

The module supports different post-build variants (previously known as post-build selectable configuration sets), but not post-build loadable configuration.

10.2.2 FrTSyn

Module SWS Item	ECUC_FrTSyn_00001	
Module Name	FrTSyn	
Module Description	This represents the specific configuration variant for the TSyn on Flexray.	
Post-Build Variant Support	true	
Supported Config Variants	VARIANT-PRE-COMPILE	
Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**

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

10.2.3 FrTSynGeneral

SWS Item	[ECUC_FrTSyn_00003]
Container Name	FrTSynGeneral
Parent Container	FrTSyn
Description	This container holds the general parameters of the Flexray-specific Synchronized Time-base Manager
Configuration Parameters	

Name	FrTSynDevErrorDetect [ECUC_FrTSyn_00002]		
Parent Container	FrTSynGeneral		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynMainFunctionPeriod [ECUC_FrTSyn_00016]	
Parent Container	FrTSynGeneral	
Description	Schedule period of the main function FrTSyn_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		

Name	FrTSynTimeValidationSupport [ECUC_FrTSyn_00040]		
Parent Container	FrTSynGeneral		
Description	Switches support for Time Validation on or off. • true: Time Validation is enabled. • false:Time Validation is disabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynVersionInfoApi [ECUC_FrTSyn_00019]		
Parent Container	FrTSynGeneral		
Description	Activate/Deactivate the version information API (FrTSyn_GetVersionInfo). True: version information API activated False: version information API deactivated.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

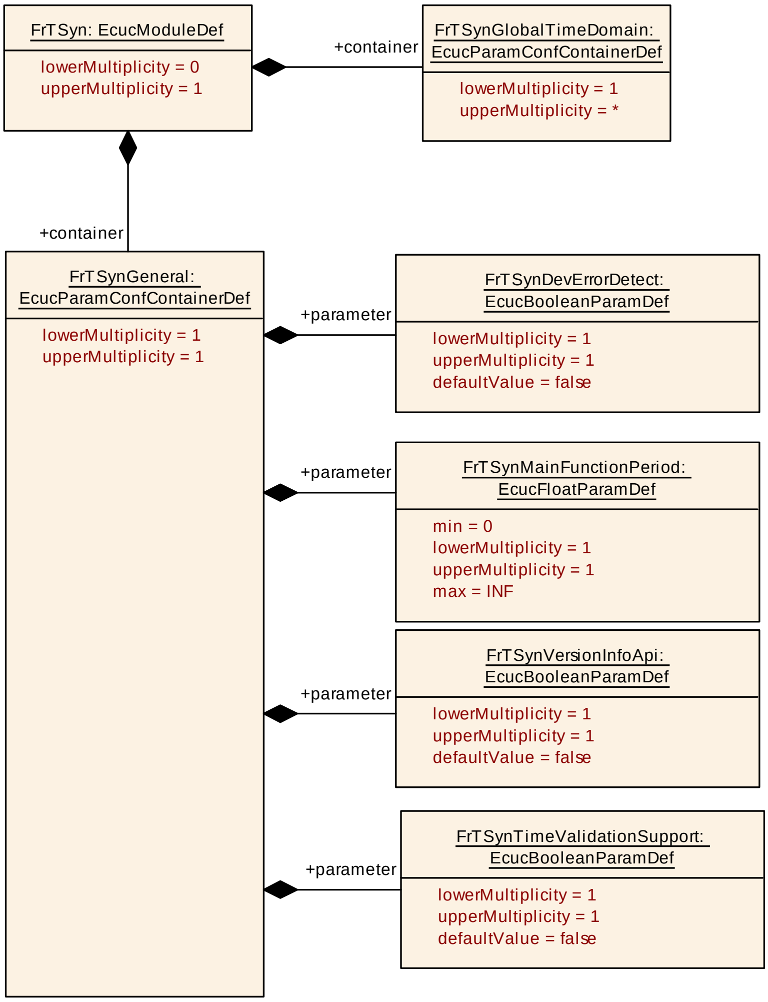


Figure 10.1: FrTSynGeneral

10.2.4 FrTSynGlobalTimeDomain

SWS Item	[ECUC_FrTSyn_00004]
Container Name	FrTSynGlobalTimeDomain
Parent Container	FrTSyn
Description	This represents the existence of a global time domain on Flexray. The FrTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the FrTSyn exists it is assumed that at least one global time domain exists.
Configuration Parameters	

Name	FrTSynEnableTimeValidation [ECUC_FrTSyn_00041]		
Parent Container	FrTSynGlobalTimeDomain		
Description	Enables/disables time recording for Time Validation for a specific Time Domain.		
Multiplicity	0..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 dependency: Only valid if FrTSynTimeValidationSupport is TRUE. Value set according to parameter StbMEnableTimeValidation of the referenced Time Base in the StbM.		

Name	FrTSynGlobalTimeDomainId [ECUC_FrTSyn_00005]		
Parent Container	FrTSynGlobalTimeDomain		
Description	The global time domain ID.		
Multiplicity	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		

Name	FrTSynGlobalTimeNetworkSegmentId [ECUC_FrTSyn_00042]		
Parent Container	FrTSynGlobalTimeDomain		
Description	This represents the numerical identifier of the network on system level scope where this Global Time has been communicated on.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
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		

Name	FrTSynGlobalTimeSecureTmacLength [ECUC_FrTSyn_00034]		
Parent Container	FrTSynGlobalTimeDomain		
Description	Represents the number of bytes for the used Truncated Message Authentication Code (TMAC). If 0, no message authentication will be used. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 16		
Default Value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynUseExtendedMsgFormat [ECUC_FrTSyn_00035]		
Parent Container	FrTSynGlobalTimeDomain		
Description	• true: use at least 32 byte for Timesync messages (depending on configuration) • false: use always 16 byte for Timesync messages **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynSynchronizedTimeBaseRef [ECUC_FrTSyn_00018]		
Parent Container	FrTSynGlobalTimeDomain		
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**
FrTSynGlobalTimeMaster	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.
FrTSynGlobalTimeOfsDataIDList	0..1	The DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.
FrTSynGlobalTimeSlave	0..1	This represents the time slave for the enclosing global time domain.
FrTSynGlobalTimeSyncDataIDList	0..1	The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.

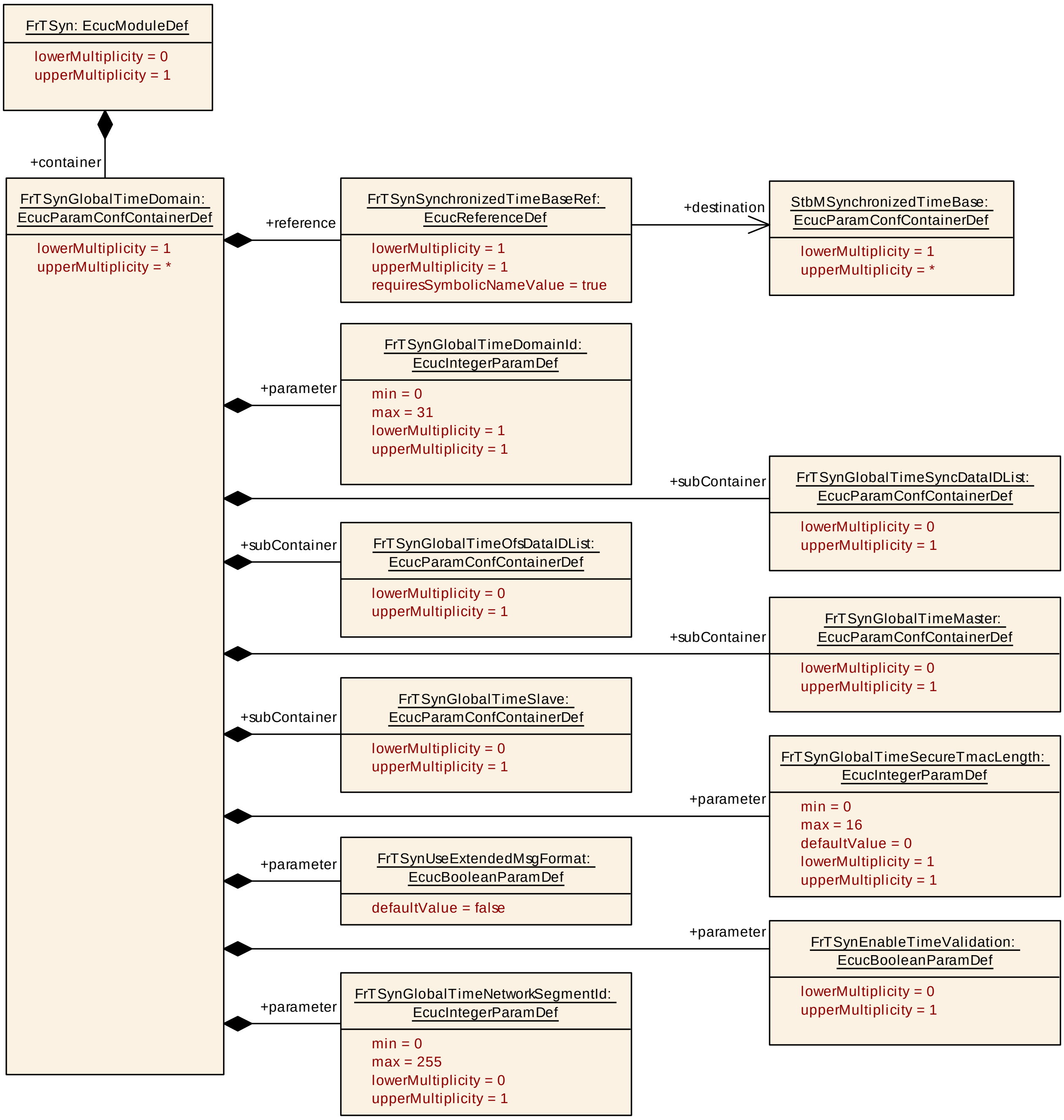


Figure 10.2: FrTSynGlobalTimeDomain

10.2.5 FrTSynGlobalTimeSyncDataIDList

SWS Item	[ECUC_FrTSyn_00023]
Container Name	FrTSynGlobalTimeSyncDataIDList
Parent Container	FrTSynGlobalTimeDomain
Description	The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.

Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
FrTSynGlobalTimeSyncDataIDListElement	16	Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.

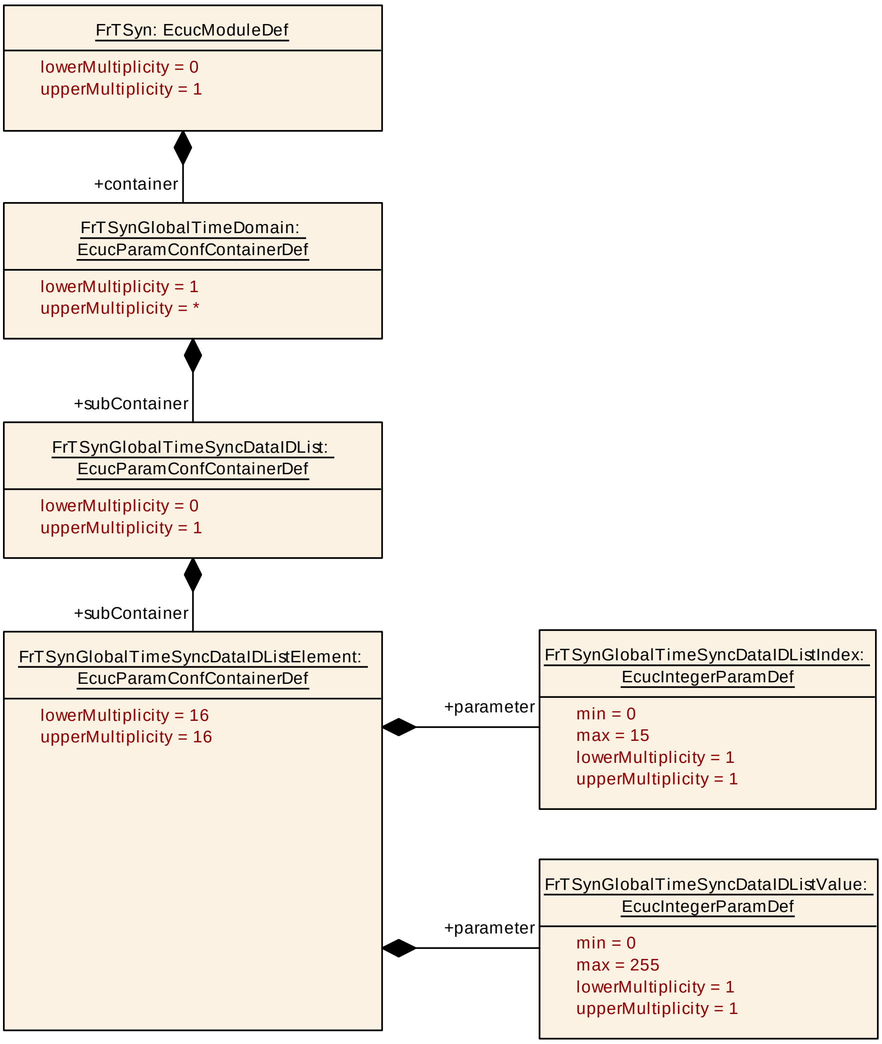


Figure 10.3: FrTSynGlobalTimeSyncDataIDList

10.2.6 FrTSynGlobalTimeSyncDataIDListElement

SWS Item	[ECUC_FrTSyn_00025]
Container Name	FrTSynGlobalTimeSyncDataIDListElement
Parent Container	FrTSynGlobalTimeSyncDataIDList
Description	Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.
Configuration Parameters	

Name	FrTSynGlobalTimeSyncDataIDListIndex [ECUC_FrTSyn_00026]		
Parent Container	FrTSynGlobalTimeSyncDataIDListElement		
Description	Index of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 15		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeSyncDataIDListValue [ECUC_FrTSyn_00027]		
Parent Container	FrTSynGlobalTimeSyncDataIDListElement		
Description	Value of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.7 FrTSynGlobalTimeOfsDataIDList

SWS Item	[ECUC_FrTSyn_00024]

Container Name	FrTSynGlobalTimeOfsDataIDList		
Parent Container	FrTSynGlobalTimeDomain		
Description	The DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
FrTSynGlobalTimeOfsDataIDListElement	16	Element of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.

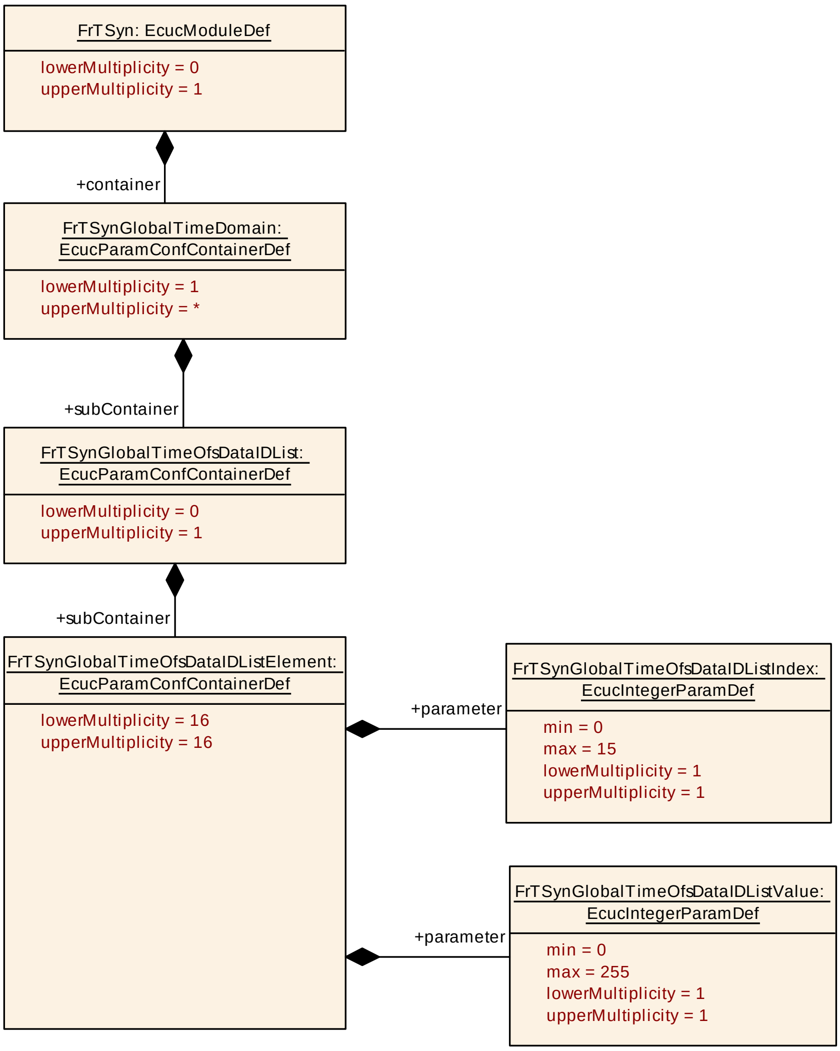


Figure 10.4: FrTSynGlobalTimeOfsDataIDList

10.2.8 FrTSynGlobalTimeOfsDataIDListElement

SWS Item	[ECUC_FrTSyn_00028]
Container Name	FrTSynGlobalTimeOfsDataIDListElement
Parent Container	FrTSynGlobalTimeOfsDataIDList
Description	Element of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.
Configuration Parameters	

Name	FrTSynGlobalTimeOfsDataIDListIndex [ECUC_FrTSyn_00029]		
Parent Container	FrTSynGlobalTimeOfsDataIDListElement		
Description	Index of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 15		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeOfsDataIDListValue [ECUC_FrTSyn_00030]		
Parent Container	FrTSynGlobalTimeOfsDataIDListElement		
Description	Value of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.9 FrTSynGlobalTimeMaster

SWS Item	[ECUC_FrTSyn_00006]		
Container Name	FrTSynGlobalTimeMaster		
Parent Container	FrTSynGlobalTimeDomain		
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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	FrTSynCyclicMsgResumeTime [ECUC_FrTSyn_00032]		
Parent Container	FrTSynGlobalTimeMaster		
Description	Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeDebounceTime [ECUC_FrTSyn_00033]		
Parent Container	FrTSynGlobalTimeMaster		
Description	This represents the configuration of a TX debounce time for SYNC and OFS messages compared to a message before with the same PDU. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF[		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeTxCrcSecured [ECUC_FrTSyn_00013]		
Parent Container	FrTSynGlobalTimeMaster		
Description	This represents the configuration of whether or not CRC is supported.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	CRC_NOT_SUPPORTED	This represents a configuration where CRC is not supported.	
	CRC_SUPPORTED	This represents a configuration where CRC is supported.	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeTxPeriod [ECUC_FrTSyn_00014]		
Parent Container	FrTSynGlobalTimeMaster		
Description	This represents the TX period. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynImmediateTimeSync [ECUC_FrTSyn_00031]		
Parent Container	FrTSynGlobalTimeMaster		
Description	Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within FrTSyn_MainFunction().		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynTxTmacCalculated [ECUC_FrTSyn_00036]		
Parent Container	FrTSynGlobalTimeMaster		
Description	This parameter controls whether or not TMAC calculation shall be supported. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TMAC_CALCULATED	The Timesync module shall calculate the TMAC.	
	TMAC_NOT_CALCULATED	The Timesync module shall not calculate any TMAC.	
Post-Build Variant Value	true		
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**
FrTSynGlobalTimeMasterPdu	1	This container carries all properties required to configure the PDU sent by the global time master for the given global time domain.

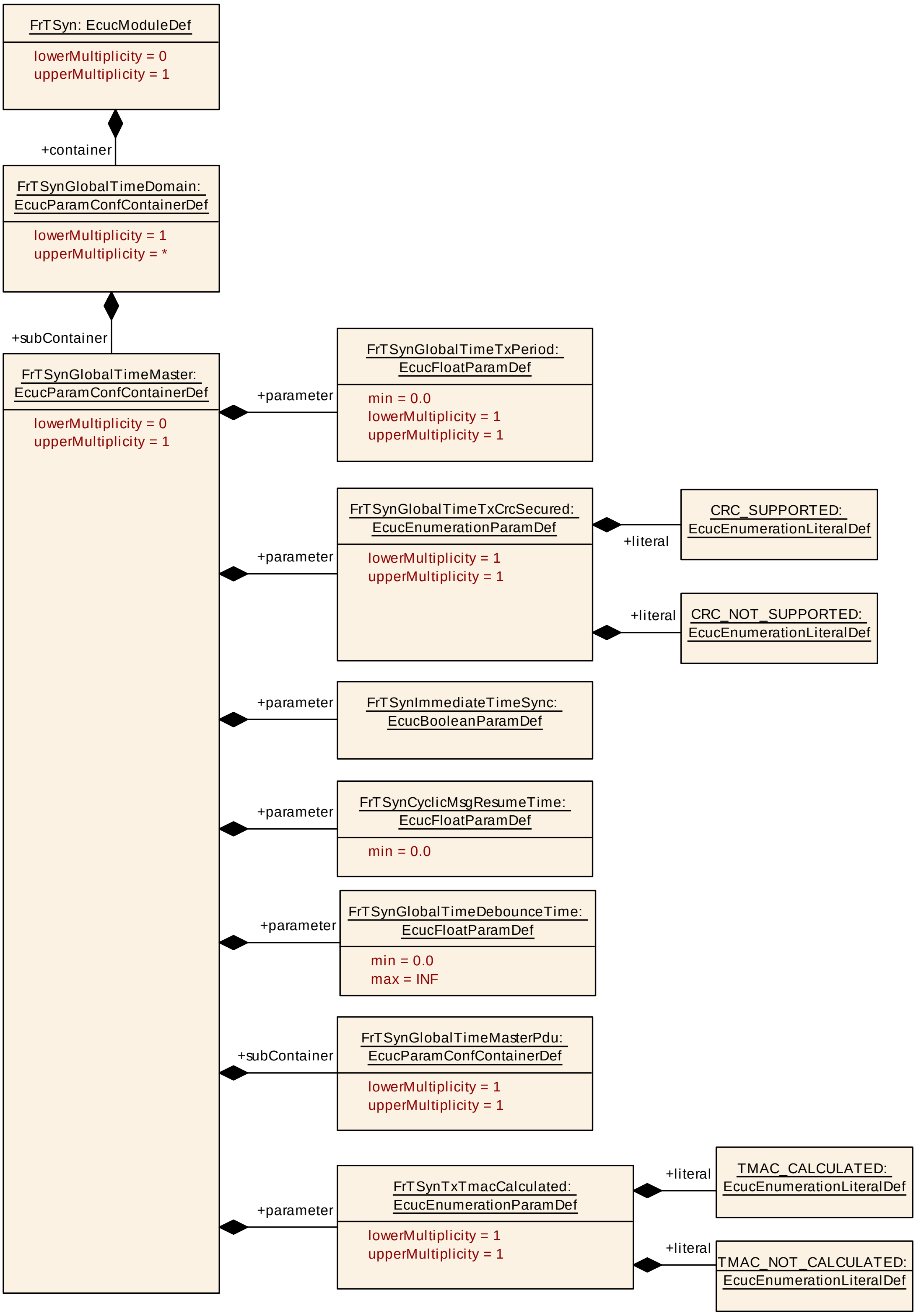

FrTSyn: EcucModuleDef
lowerMultiplicity = 0
upperMultiplicity = 1
+container
FrTSynGlobalTimeDomain: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
FrTSynGlobalTimeMaster: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
FrTSynGlobalTimeTxPeriod: EcucFloatParamDef
min = 0.0
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
FrTSynGlobalTimeTxCrcSecured: EcucEnumerationParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
+literal
CRC_SUPPORTED: EcucEnumerationLiteralDef
+literal
CRC_NOT_SUPPORTED: EcucEnumerationLiteralDef
+parameter
FrTSynImmediateTimeSync: EcucBooleanParamDef
+parameter
FrTSynCyclicMsgResumeTime: EcucFloatParamDef
min = 0.0
+parameter
FrTSynGlobalTimeDebounceTime: EcucFloatParamDef
min = 0.0
max = INF
+subContainer
FrTSynGlobalTimeMasterPdu: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
FrTSynTxTmacCalculated: EcucEnumerationParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
+literal
TMAC_CALCULATED: EcucEnumerationLiteralDef
+literal
TMAC_NOT_CALCULATED: EcucEnumerationLiteralDef

Figure 10.5: FrTSynGlobalTimeMaster

10.2.10 FrTSynGlobalTimeMasterPdu

SWS Item	[ECUC_FrTSyn_00008]
Container Name	FrTSynGlobalTimeMasterPdu
Parent Container	FrTSynGlobalTimeMaster
Description	This container carries all properties required to configure the PDU sent by the global time master for the given global time domain.
Configuration Parameters	

Name	FrTSynGlobalTimeMasterHandleId [ECUC_FrTSyn_00007]		
Parent Container	FrTSynGlobalTimeMasterPdu		
Description	This represents the handle ID of the PDU that contains the global time information.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimePduRef [ECUC_FrTSyn_00020]		
Parent Container	FrTSynGlobalTimeMasterPdu		
Description	This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.11 FrTSynGlobalTimeSlave

SWS Item	[ECUC_FrTSyn_00010]
Container Name	FrTSynGlobalTimeSlave
Parent Container	FrTSynGlobalTimeDomain

Description	This represents the time slave for the enclosing global time domain.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Name	FrTSynGlobalTimeMinMsgGap [ECUC_FrTSyn_00038]		
Parent Container	FrTSynGlobalTimeSlave		
Description	This parameter represents the configuration of a minimum message gap time for received SYNC and OFS messages compared to a message before with the same PDU. If PDUs are received more often in between than this parameter allows, they shall be ignored. Unit: seconds **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default Value	0		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeSequenceCounterHysteresis [ECUC_FrTSyn_00043]		
Parent Container	FrTSynGlobalTimeSlave		
Description	FrTSynGlobalTimeSequenceCounterHysteresis specifies the number of consecutive valid SYNC (or OFS) messages that are required by the Time Slave while being in Timeout state until a Time Tuple is forwarded to the StbM.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 15		
Default Value	0		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeSequenceCounterJumpWidth [ECUC_FrTSyn_00022]		
Parent Container	FrTSynGlobalTimeSlave		
Description	The SequenceCounterJumpWidth specifies the maximum allowed gap of the Sequence Counter between two SYNC resp. two OFS messages.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 15		
Default Value	0		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimeTmacTimeout [ECUC_FrTSyn_00039]		
Parent Container	FrTSynGlobalTimeSlave		
Description	Rx timeout for the TMAC message. Unit:seconds **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default Value	0		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynRxCrcValidated [ECUC_FrTSyn_00017]	
Parent Container	FrTSynGlobalTimeSlave	
Description	This parameter controls whether or not CRC validation shall be supported.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	CRC_IGNORED	The Timesync module accepts Time Synchronization messages, which are CRC secured (without actually validating the CRC) and those, which are not CRC secured. That means, the Timesync module ignores the CRC.

	CRC_NOT_VALIDATED	The Timesync module accepts only Time Synchronization messages, which are not CRC secured. All other Time Synchronization messages are ignored.	
	CRC_OPTIONAL	The Timesync module accepts only Time Synchronization messages which are not CRC secured and Time Synchronization messages which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
	CRC_VALIDATED	The Timesync module accepts only Time Synchronization messages, which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynRxTmacValidated [ECUC_FrTSyn_00037]		
Parent Container	FrTSynGlobalTimeSlave		
Description	This parameter controls whether or not TMAC validation shall be supported. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TMAC_NOT_VALIDATED	The Timesync module shall not validate the TMAC.	
	TMAC_VALIDATED	The Timesync module shall validate the TMAC.	
Post-Build Variant Value	true		
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**
FrTSynGlobalTimeSlavePdu	1	This container carries all properties required to configure the PDU received by the time slave for the given global time domain.

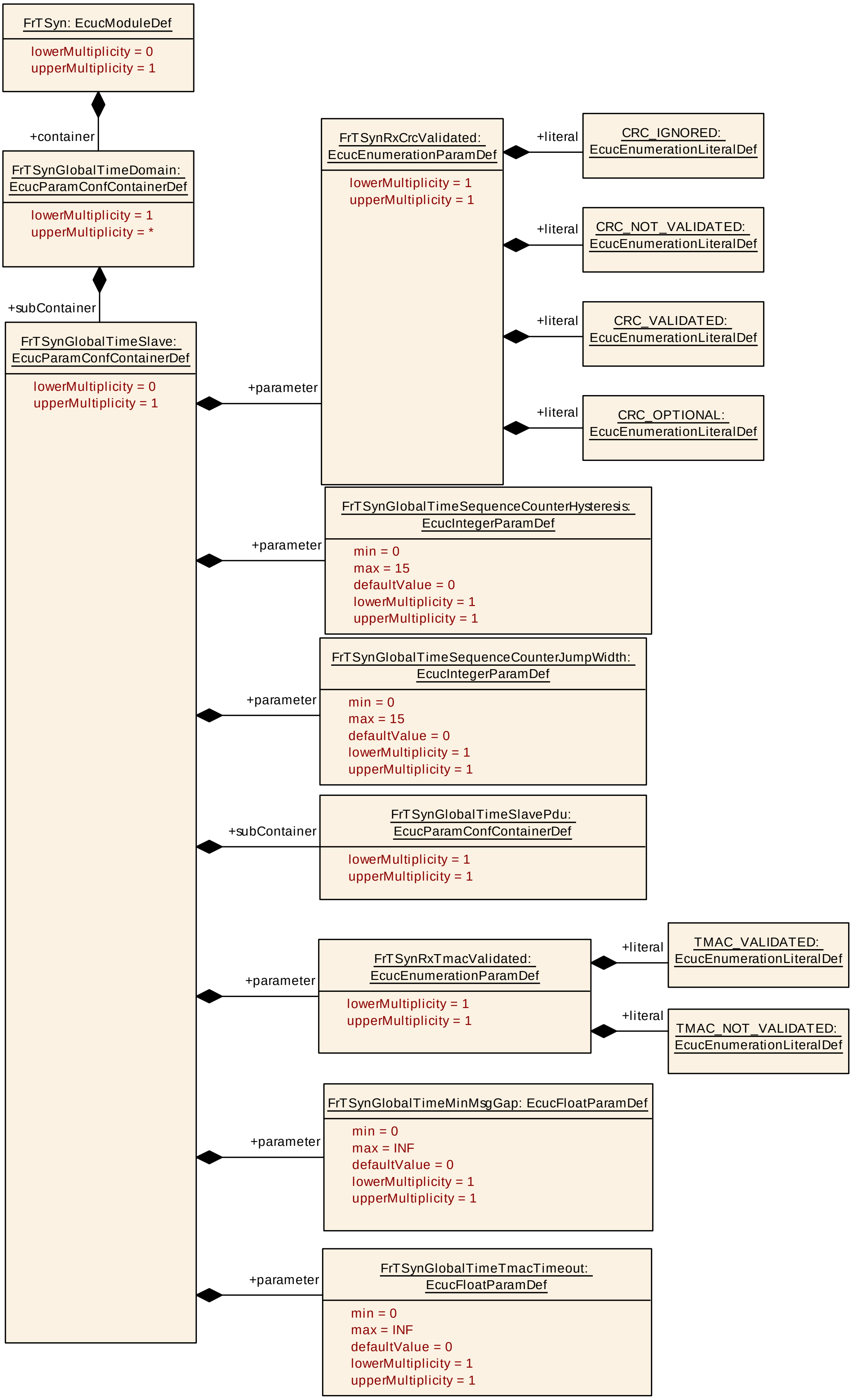


Figure 10.6: FrTSynGlobalTimeSlave

10.2.12 FrTSynGlobalTimeSlavePdu

SWS Item	[ECUC_FrTSyn_00012]
Container Name	FrTSynGlobalTimeSlavePdu
Parent Container	FrTSynGlobalTimeSlave
Description	This container carries all properties required to configure the PDU received by the time slave for the given global time domain.
Configuration Parameters	

Name	FrTSynGlobalTimeSlaveHandleId [ECUC_FrTSyn_00011]		
Parent Container	FrTSynGlobalTimeSlavePdu		
Description	This represents the handle ID of the PDU that contains the global time information.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	FrTSynGlobalTimePduRef [ECUC_FrTSyn_00021]		
Parent Container	FrTSynGlobalTimeSlavePdu		
Description	This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

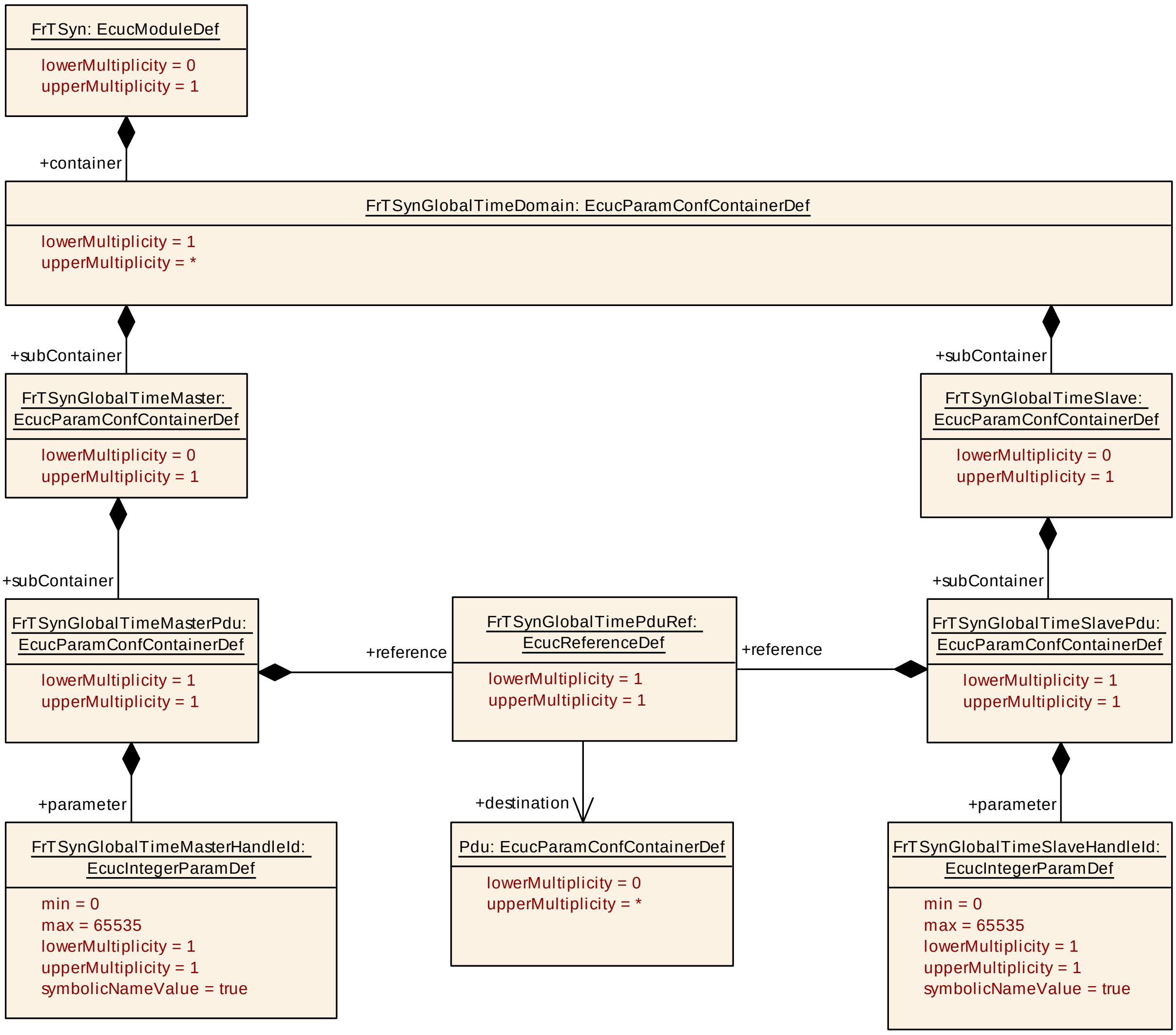


Figure 10.7: FrTSynGlobalTimePdu

10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in the SWS BSW General [3].

A Not applicable requirements

[SWS_FrTSyn_00999] ⌈These requirements on Time Synchronization from the RS Time Synchronization [1] are not applicable to FrTSyn, because they refer either to network types other than FlexRay or to the Time Base Manager module⌋*(RS_TS_00002, RS_TS_00005, RS_TS_00006, RS_TS_00007, RS_TS_00008,* *RS_TS_00009, RS_TS_00010, RS_TS_00011, RS_TS_00012, RS_TS_00013, RS_TS_00014, RS_TS_00015, RS_TS_00016, RS_TS_00017, RS_TS_00018, RS_TS_00019, RS_TS_00021, RS_TS_00024, RS_TS_00025, RS_TS_00026, RS_TS_00027, RS_TS_00029, RS_TS_00030, RS_TS_00031, RS_TS_00032, RS_TS_00033, RS_TS_00035, RS_TS_00036, RS_TS_00037, RS_TS_00038, RS_TS_20031, RS_TS_20032, RS_TS_20033, RS_TS_20034, RS_TS_20035, RS_TS_20036, RS_TS_20037, RS_TS_20038, RS_TS_20047, RS_TS_20048, RS_TS_20051, RS_TS_20052, RS_TS_20053, RS_TS_20054, RS_TS_20058, RS_TS_20059, RS_TS_20060, RS_TS_20061, RS_TS_20062, RS_TS_20063, RS_TS_20066, RS_TS_20068, RS_TS_20069, RS_TS_20070)*