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

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.4.0

Document Change History

Date	Release	Changed by	Change Description
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

Table of Contents 3

1 Introduction and functional overview 5

2 Acronyms, Abbreviations, and Definitions 7

3 Related documentation 8

3.1 Input documents 8
3.2 Related specification 8

4 Constraints and assumptions 9

4.1 Limitations 9
4.2 Applicability to car domains 9

5 Dependencies to other modules 10

5.1 File structure 11
5.1.1 Code file structure 11
5.1.2 Header file structure 11

6 Requirements traceability 12

7 Functional specification 14

7.1 Overview 14
7.2 Module Handling 14
7.2.1 Initialization 14
7.2.2 FlexRay Interface 14
7.3 Message Format 15
7.3.1 SYNC message 16
7.3.2 OFS message 16
7.4 Acting as Time Master 17
7.4.1 SYNC message processing 18
7.4.2 OFS message processing 19
7.4.3 Transmission mode 19
7.4.4 Debounce Time 20
7.4.5 Immediate Time Synchronization 20
7.4.6 Calculation and Assembling of Time Synchronization Messages 21
7.5 Acting as Time Slave 24
7.5.1 SYNC message processing 24
7.5.2 OFS message processing 24
7.5.3 Validation and Disassembling of Time Synchronization Messages 25
7.6 Global Time Measurement Support 28
7.7 Error Classification 28
7.7.1 Development Errors 28
7.7.2 Runtime Errors 28
7.7.3 Transient Faults 28
7.7.4 Production Errors 29
7.7.5 Extended Production Errors 29

8 API specification 30
8.1 API 30
8.1.1 Imported types 30
8.1.2 Type definitions 30
8.1.3 Function definitions 31
8.1.4 Call-back notifications 32
8.1.5 Scheduled functions 34
8.1.6 Expected Interfaces 35
9 Sequence diagrams 37
9.1 FlexRay Time Synchronization (Time Master) 37
9.2 FlexRay Time Synchronization (Time Slave) 38
10 Configuration specification 39
10.1 How to read this chapter 39
10.2 Containers and configuration parameters 40
10.2.1 Variants 40
10.2.2 FrTSyn 40
10.2.3 FrTSynGeneral 41
10.2.4 FrTSynGlobalTimeDomain 42
10.2.5 FrTSynGlobalTimeSyncDataIDList 45
10.2.6 FrTSynGlobalTimeSyncDataIDListElement 47
10.2.7 FrTSynGlobalTimeOfsDataIDList 47
10.2.8 FrTSynGlobalTimeOfsDataIDListElement 49
10.2.9 FrTSynGlobalTimeMaster 50
10.2.10 FrTSynGlobalTimeMasterPdu 54
10.2.11 FrTSynGlobalTimeSlave 54
10.2.12 FrTSynGlobalTimeSlavePdu 59
10.3 Published Information 60

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.

The following Figure shows the Time Synchronization mechanism on FlexRay.

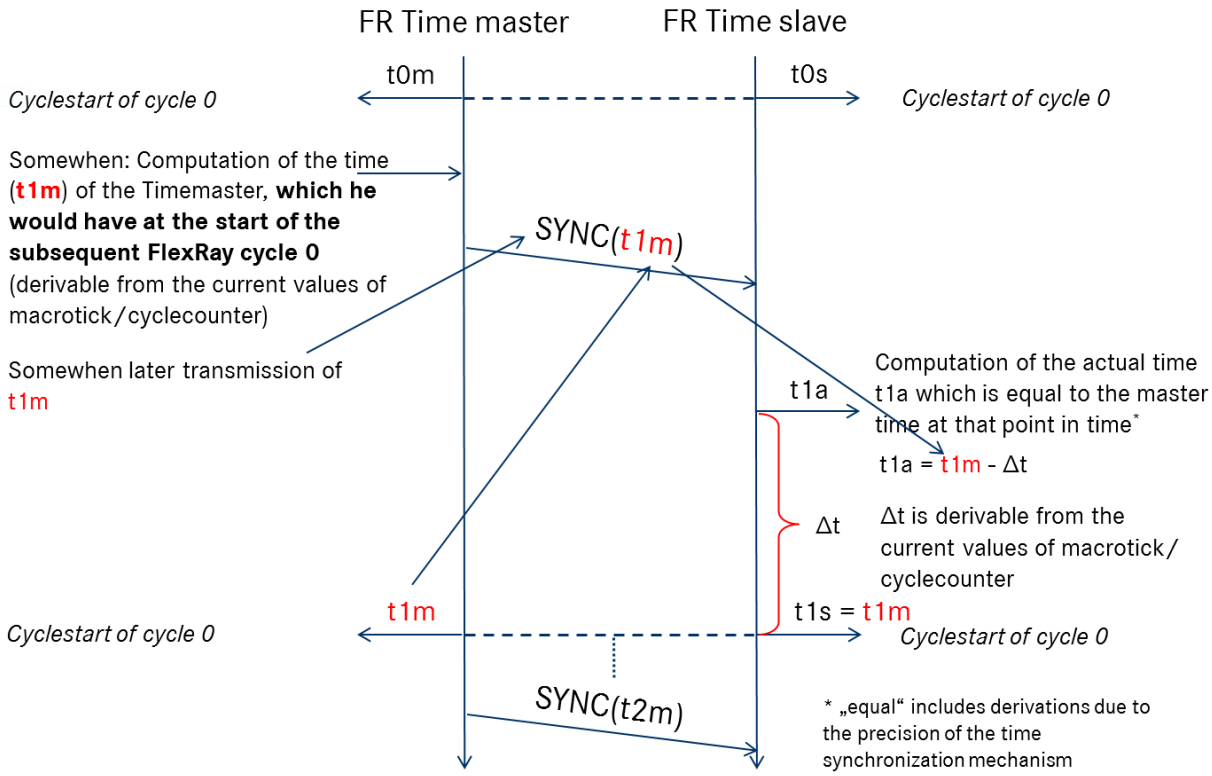


Figure 1: FlexRay Time Synchronization mechanism

2 Acronyms, Abbreviations, and Definitions

This section lists module local Abbreviations and Definitions. For a complete set of Synchronized Time Base related Abbreviations and Definitions refer to the corresponding chapter in [4].

Abbreviation / Acronym:	Description
(G)TD	(Global) Time Domain
(G)TM	(Global) Time Master
<Bus>TSyn	A bus specific Time Synchronization module
CRC	Cyclic Redundancy Checksum
Debounce Time	Minimum gap between two Tx messages with the same PDU
DEM	Diagnostic Event Manager
DET	Default Error Tracer
FR	FlexRay
FUP message	Follow-Up message
OFNS message	Offset adjustment message
OFS message	Offset Synchronization message
StbM	Synchronized Time-Base Manager
SYNC message	Time Synchronization message
TG	Time Gateway
Timesync	Time Synchronization
TS	Time Slave
TSD	Time Sub-domain

3 Related documentation

3.1 Input documents

[1] Requirements on Synchronized Time-Base Manager
AUTOSAR_SRS_SynchronizedTimeBaseManager.pdf

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[3] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

[4] Specification of Synchronized Time-Base Manager
AUTOSAR_SWS_SynchronizedTimeBaseManager.pdf

[5] Specification of CRC Routines
AUTOSAR_SWS_CRCLibrary.pdf

[6] Specification of FlexRay Interface
AUTOSAR_SWS_FlexRayInterface.pdf

[7] Specification of Default Error Tracer
AUTOSAR_SWS_DefaultErrorTracer.pdf

[8] Specification of Basic Software Mode Manager
AUTOSAR_SWS_BSWModeManager.pdf

3.2 Related specification

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

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

4.2 Applicability to car domains

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

5 Dependencies to other modules

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

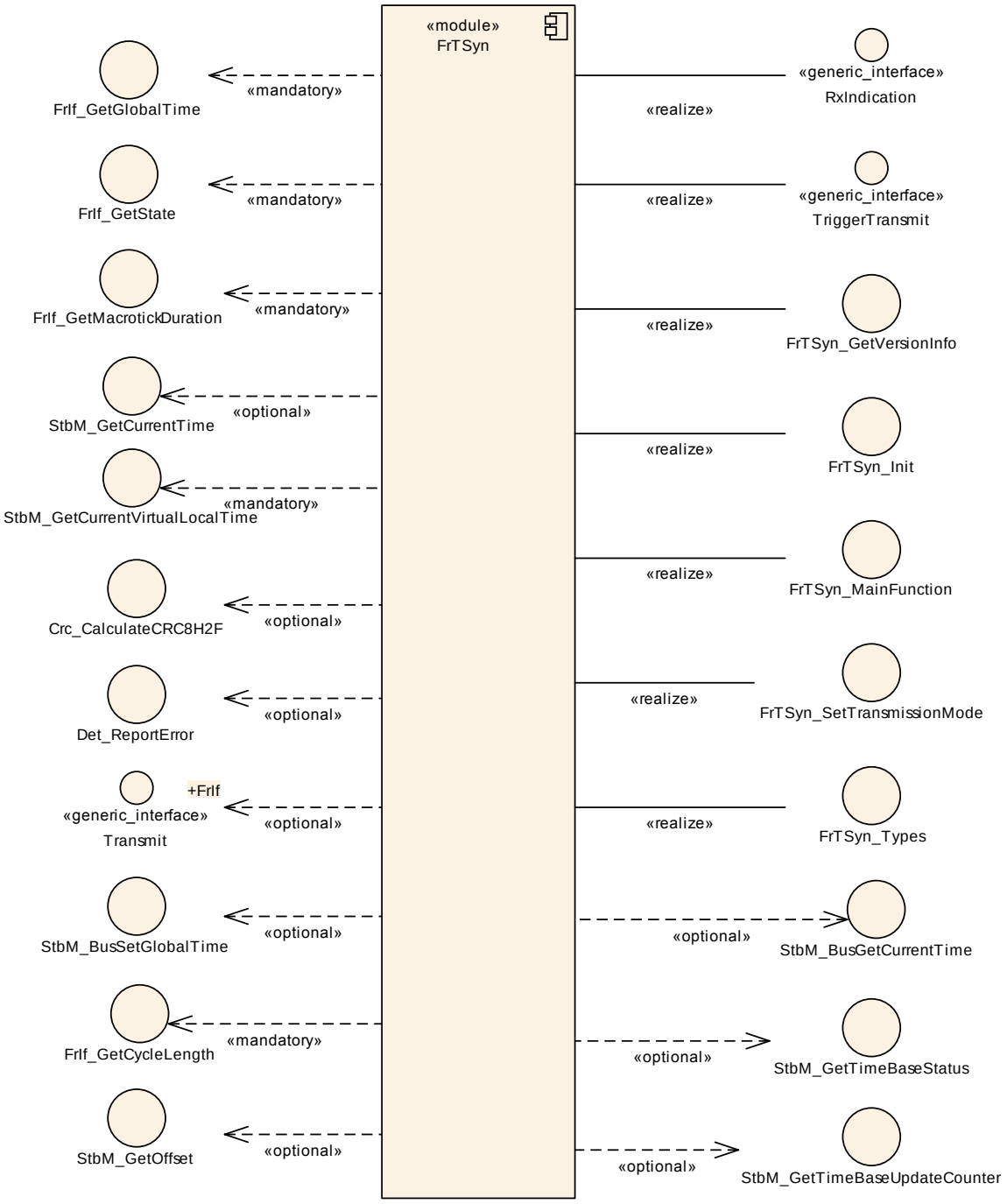


Figure 2: Module dependencies of the FrTSyn module

- 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 traceability

Requirement	Description	Satisfied by
RS_TS_00003	The Implementation of Time Synchronization shall initialize the Local Time Base with zero at startup	SWS_FrTSyn_00003, SWS_FrTSyn_00005
RS_TS_00004	The Implementation of Time Synchronization shall initialize the Global Time Base with a configurable startup value.	SWS_FrTSyn_00003, SWS_FrTSyn_00005
RS_TS_00034	The Implementation of Time Synchronization shall provide measurement data to the application	SWS_FrTSyn_00092
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
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
RS_TS_20043	The Timesync over FlexRay module shall support a protocol for precise time measurement and synchronization over FlexRay	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
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
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 [4].

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_00005][
When `FrTSyn_Init()` is called in initialized state, the Time Synchronization over FlexRay shall re-initialize its internal variables.
](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)

7.3 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.
](RS_TS_20045)

[SWS_FrTSyn_00013][
User Data shall be mapped to the `StbM_UserDataType`, whereas the byte number given in the message and by the `StbM_UserDataType` shall match (User Byte 0

mapped to `StbM_UserDataType.userByte0` etc.). Afterwards `StbM_UserDataType.userDataLength` shall be set accordingly.
](RS_TS_20045)

7.3.1 SYNC message

[SWS_FrTSyn_00014][
SYNC not CRC secured message format:
Byte 0: *Type* = 0x10
Byte 1: User Byte 2, default: 0
Byte 2: *D* = Time Domain 0 to 15 (Bit 7 to Bit 4)
SC = Sequence Counter (Bit 3 to Bit 0)
Byte 3: *FCNT* = FlexRay Cycle Counter from 0 to 63 (Bit 7 to Bit 2)
SGW (Bit 1)
SyncToGTM = 0
SyncToSubDomain = 1
reserved (Bit 0), default: 0
Byte 4: User Byte 0, default: 0
Byte 5: User Byte 1, default: 0
Byte 6-11: *SyncTimeSec* = 48 Bit time value in seconds
Byte 12-15: *SyncTimeNSec* = 32 Bit time value in nanoseconds
](RS_TS_20041, RS_TS_20043)

[SWS_FrTSyn_00015][
SYNC CRC secured message format:
Byte 0: Type = 0x20
Byte 1: *CRC*
Byte 2: *D* = Time Domain 0 to 15 (Bit 7 to Bit 4)
SC = Sequence Counter (Bit 3 to Bit 0)
Byte 3: *FCNT* = FlexRay Cycle Counter from 0 to 63 (Bit 7 to Bit 2)
SGW (Bit 1)
SyncToGTM = 0
SyncToSubDomain = 1
reserved (Bit 0), default: 0
Byte 4: User Byte 0, default: 0
Byte 5: User Byte 1, default: 0
Byte 6-11: *SyncTimeSec* = 48 Bit time value in seconds
Byte 12-15: *SyncTimeNSec* = 32 Bit time value in nanoseconds
](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 0: *Type* = 0x34

Byte 1: User Byte 2, default: 0
Byte 2: *D* = Time Domain 16 to 31 (Bit 7 to Bit 4)
SC = Sequence Counter (Bit 3 to Bit 0)
Byte 3: reserved (Bit 7 to Bit 2), default: 0
SGW (Bit 1)
SyncToGTM = 0
SyncToSubDomain = 1
reserved (Bit 0), default: 0
Byte 4: User Byte 0, default: 0
Byte 5: User Byte 1, default: 0
Byte 6: reserved, default: 0
Byte 7: reserved, default: 0
Byte 8-11: *OfsTimeSec* = 32 Bit offset time value in seconds
Byte 12-15: *OfsTimeNSec* = 32 Bit offset time value in nanoseconds
](RS_TS_20041, RS_TS_20044)

[SWS_FrTSyn_00080][
OFS CRC secured message format:
Byte 0: Type = 0x44
Byte 1: *CRC*
Byte 2: *D* = Time Domain 16 to 31 (Bit 7 to Bit 4)
SC = Sequence Counter (Bit 3 to Bit 0)
Byte 3: reserved (Bit 7 to Bit 2), default: 0
SGW (Bit 1)
SyncToGTM = 0
SyncToSubDomain = 1
reserved (Bit 0), default: 0
Byte 4: User Byte 0, default: 0
Byte 5: User Byte 1, default: 0
Byte 6: reserved, default: 0
Byte 7: reserved, default: 0
Byte 8-11: *OfsTimeSec* = 32 Bit offset time value in seconds
Byte 12-15: *OfsTimeNSec* = 32 Bit offset time value in nanoseconds
](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.

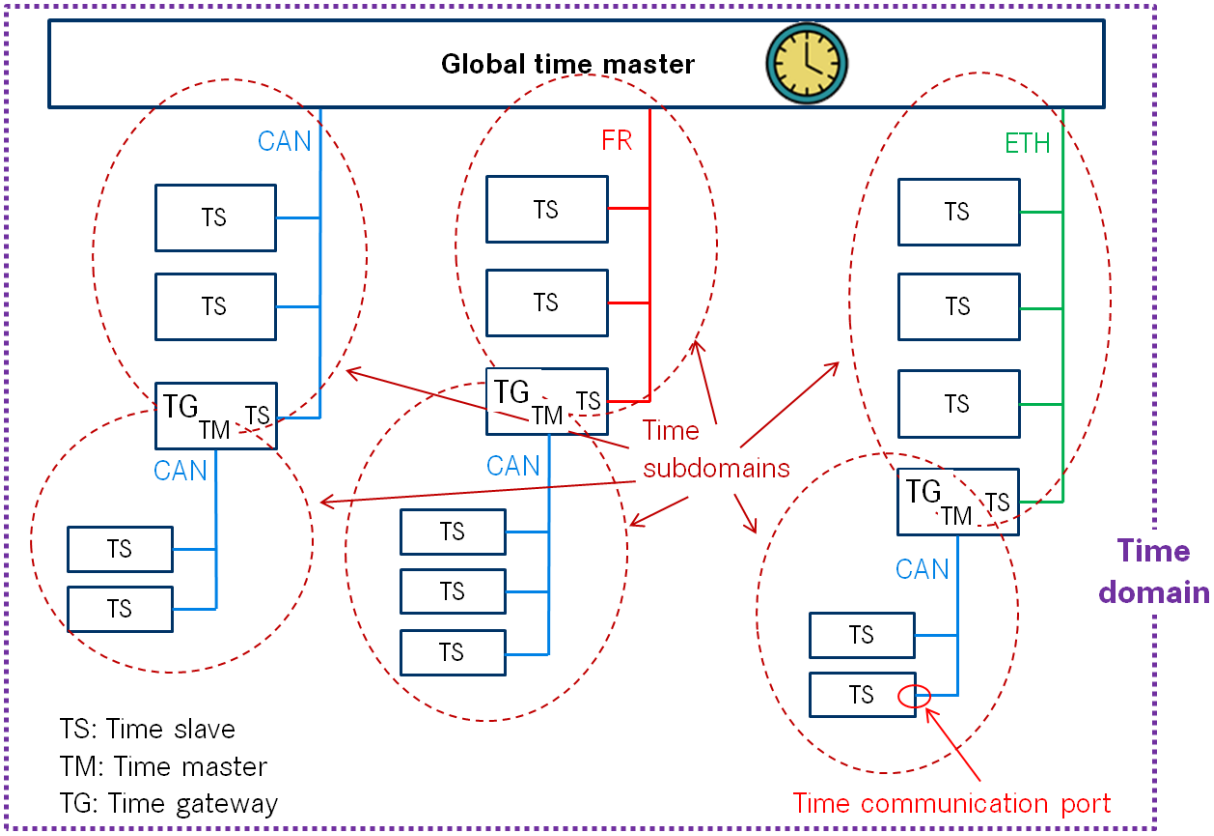


Figure 3: Terminology Example

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)

[SWS_FrTSyn_00019][
For each configured Time Master (`FrTSynGlobalTimeMaster`) the FrTSyn module shall periodically transmit SYNC messages with the cycle `FrTSynGlobalTimeTxPeriod` (**ECUC_FrTSyn_00014 :**) including the time value, which will be valid at the start of the next FlexRay cycle 0 (see Figure 4) 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 (see 7.4.5).
](RS_TS_20039, RS_TS_20043)

[SWS_FrTSyn_00021][
Depending on `FrTSynGlobalTimeTxCrcSecured` (**ECUC_FrTSyn_00013 :**) the SYNC message shall be of type:

FrTSynGlobalTimeTxCrcSecured	SYNC
CRC_NOT_SUPPORTED	0x10 SYNC not CRC secured message
CRC_SUPPORTED	0x20 SYNC CRC secured message

](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 (**ECUC_FrTSyn_00014 :**) 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 (see 7.4.5).
](RS_TS_20039, RS_TS_20044)

[SWS_FrTSyn_00025][
Depending on FrTSynGlobalTimeTxCrcSecured (**ECUC_FrTSyn_00013 :**) the OFS message shall be of type:

FrTSynGlobalTimeTxCrcSecured	OFS
CRC_NOT_SUPPORTED	0x34 OFS not CRC secured message
CRC_SUPPORTED	0x44 OFS CRC secured message

](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` (**ECUC_FrTSyn_00033 :**) 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` (**ECUC_FrTSyn_00033 :**) 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` (**ECUC_FrTSyn_00031 :**) 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` (**ECUC_FrTSyn_00031 :**) 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` (**ECUC_FrTSyn_00031 :**) is set to `TRUE`, `cyclicMsgResumeCounter` and `FrTSynCyclicMsgResumeTime` (**ECUC_FrTSyn_00032 :**) shall be considered.
](RS_TS_20039)

[SWS_FrTSyn_00090][
`FrTSynCyclicMsgResumeTime` (**ECUC_FrTSyn_00032 :**) 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 (1st step) and how the message will be assembled (2nd 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 (refer to Figure 4):

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
   $T0 = T_{SYNC} + (T1_{VLT} - T0_{VLT}) + (64 - \texttt{currentCycle}) * \texttt{CycleLength} - (\texttt{currentMacroticks} * \texttt{MacrotickDuration})$

4. Calculate *SyncTimeSec* (second portion of T0) and *SyncTimeNSec* (nanosecond portion of T0)

](RS_TS_20043)

Note: `CycleLength` and `MacrotickDuration` are given statically by configuration. In order to minimize rounding errors due to the granularity of `MacrotickDuration` (i.e., ns) the calculation uses `CycleLength` instead of the term ("MacroticksPerCycle" * `MacrotickDuration`).

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.

[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 [5] 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` (**ECUC_FrTSyn_00023 : ECUC_FrTSyn_00024 :**) is given by configuration for each message *Type*.
](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:

1. Calculate *SC*
2. Copy `currentCycle` (**[SWS_FrTSyn_00028]**) to FCNT (for SYNC message)
3. Calculate *SGW*
4. Copy all data to the appropriate position within the related message
5. 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()` (see Figure 5:).
](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 (refer to Figure 5):

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`) + (`MacrotickDuration` * `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 FR cycle duration:
   T1 = T1 – (`CycleLength` * 64)

](RS_TS_20043)

Note: `CycleLength` and `MacrotickDuration` are given statically by configuration. In order to minimize rounding errors due to the granularity of `MacrotickDuration` (i.e., ns) the calculation uses `CycleLength` instead of the term ("MacroticksPerCycle" * `MacrotickDuration`).

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.

[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.

The `FrTSynGlobalTimeSequenceCounterJumpWidth` value 0 is not allowed.
](RS_TS_20042, RS_TS_20043, RS_TS_20044)

[SWS_FrTSyn_00049][
At Startup or if a Time Base update timeout has been detected (`TIMEOUT` bit set in Time Base synchronization status `timeBaseStatus`), a Time Slave shall not check the Sequence Counter of the 1st received SYNC (or OFS) message per Time Domain against the defined Sequence Counter Jump Width.
](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.

7.5.3.4 CRC Validation

[SWS_FrTSyn_00050][
The function `Crc_CalculateCRC8H2F()` as defined in [5] 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 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)

7.7 Error Classification

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

[SWS_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)

7.7.1 Development Errors

The detection of development errors is configurable (see section 10.2, `FrTSynDevErrorDetect`).

[SWS_FrTSyn_00059][
FrTSyn shall use following development errors:

Type or error	*Related error code*	*Value [hex]*
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`
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

No Runtime Errors defined.

7.7.3 Transient Faults

No Transient Faults defined.

7.7.4 Production Errors

No Production Errors defined.

7.7.5 Extended Production Errors

No Extended Production Errors defined.

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	ComStackTypes.h	PduIdType
	ComStackTypes.h	PduInfoType
FrIf	FrIf.h	FrIf_StateType
StbM	Rte_StbM_Type.h	StbM_SynchronizedTimeBaseType
	Rte_StbM_Type.h	StbM_TimeBaseStatusType
	Rte_StbM_Type.h	StbM_TimeStampType
	Rte_StbM_Type.h	StbM_UserDataType
	StbM.h	StbM_MeasurementType
	StbM.h	StbM_VirtualLocalTimeType
Std_Types	StandardTypes.h	Std_ReturnType
	StandardTypes.h	Std_VersionInfoType

⌋ (RS_TS_20043)

8.1.2 Type definitions

8.1.2.1 FrTSyn_ConfigType

[SWS_FrTSyn_00061] ⌈

Name:	`FrTSyn_ConfigType`		
Type:	`Structure`		
Element:	`void`	`implementation specific`	--
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`

Type:	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 FR Interface 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 PDU ID (`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 PDU ID (`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
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Main function for cyclic call / resp. 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` (refer to **ECUC_FrTSyn_00016 :**).
](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_GetGlobalTime(). Important Note: FrIf_GetGlobalTime may be called within an exclusive area.
FrIf_GetMacrotickDuration	FrIf.h	Retrieves the Duration of a Macrotick in ns
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., the Received Time Tuple) to the StbM.
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)

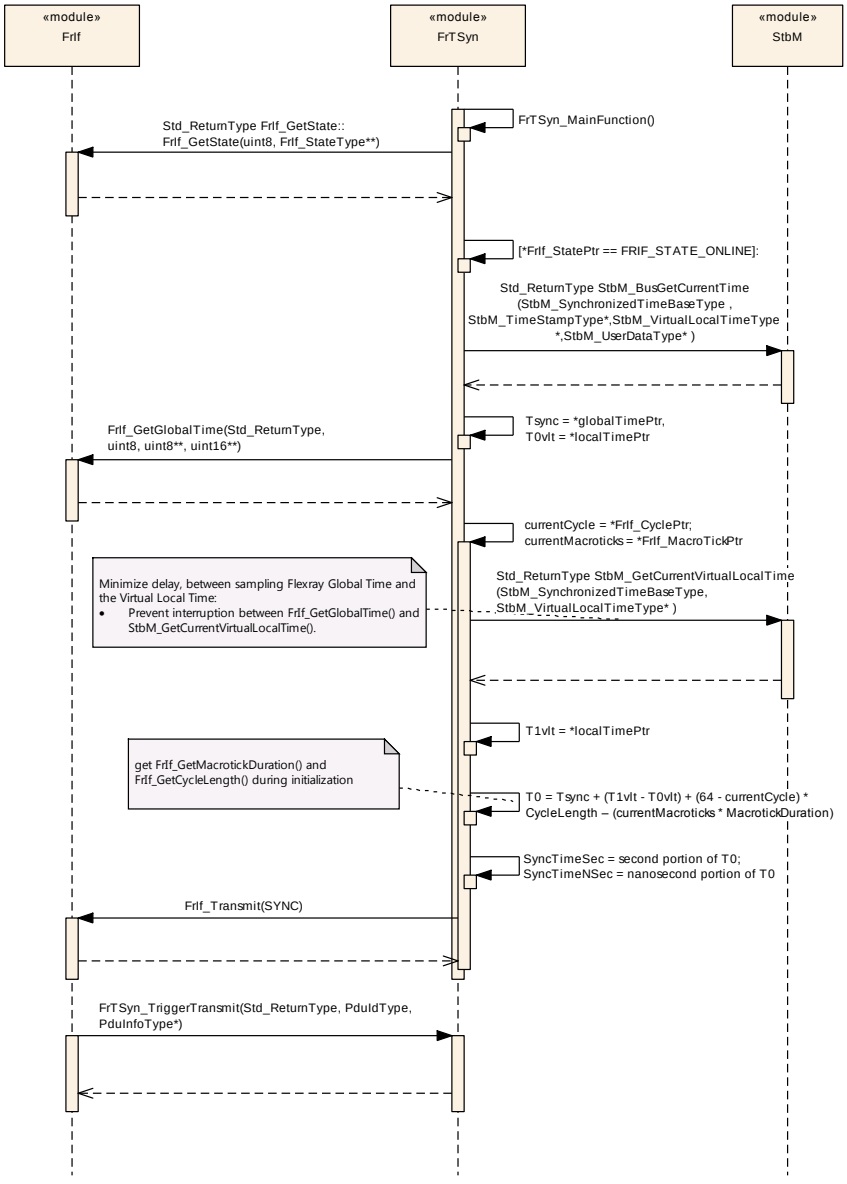


Figure 4: FlexRay Time Synchronization (Time Master)

9.2 FlexRay Time Synchronization (Time Slave)

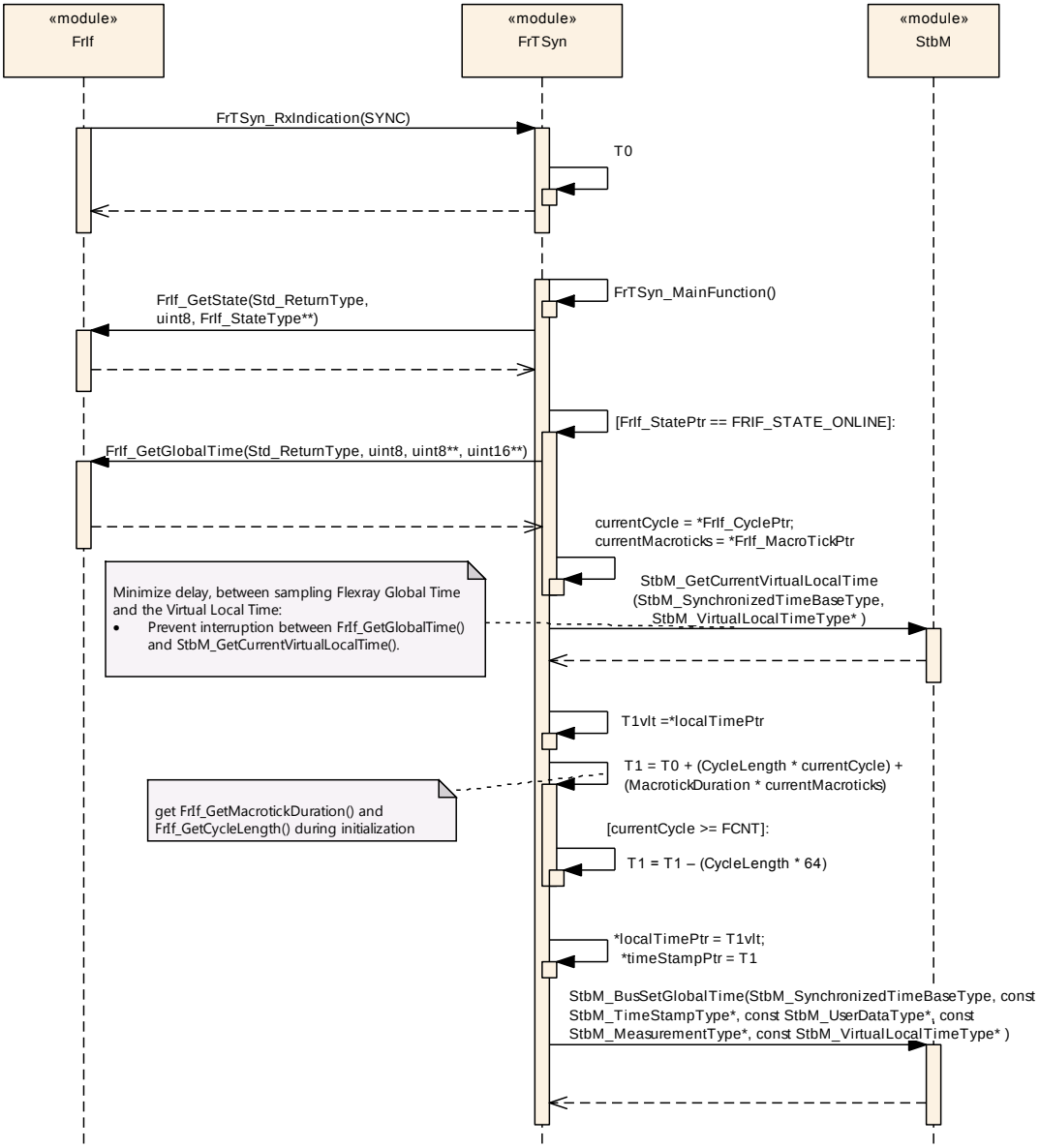


Figure 5: FlexRay Time Synchronization (Time Slave)

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_BSWGeneral*.

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

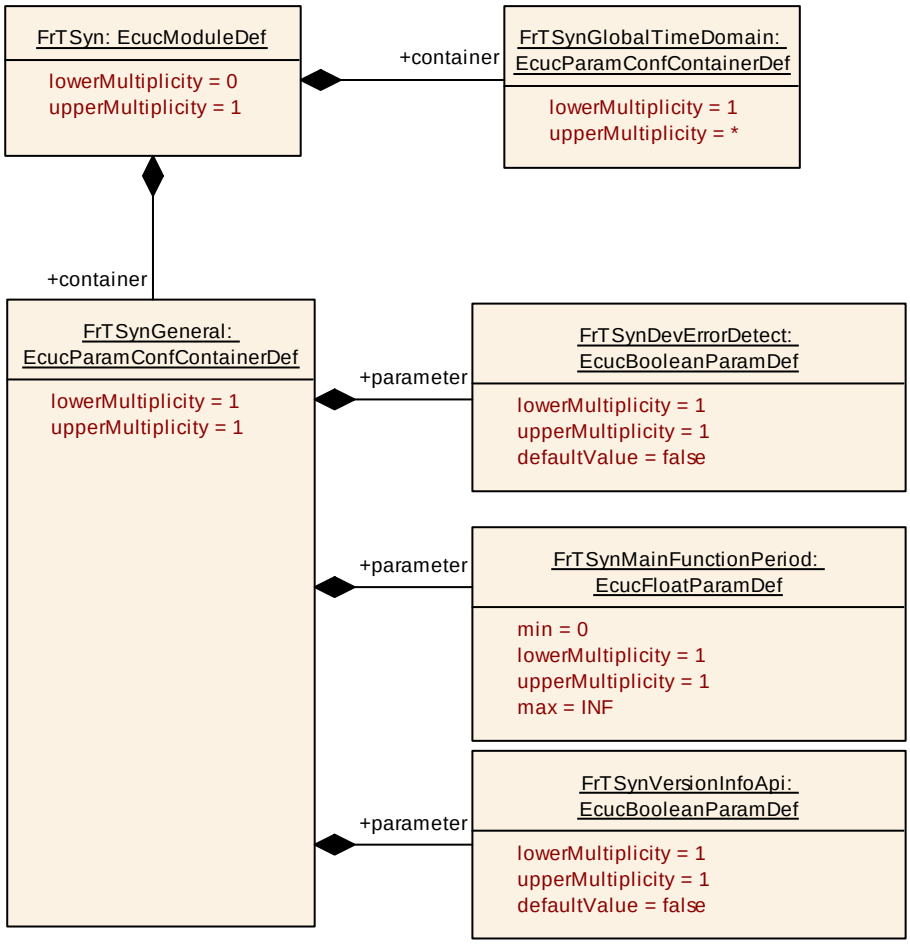
[SWS_FrTSyn_00077][
The Time Synchronization over FlexRay shall support the configuration for Time Master, Time Slave and Time Gateway.
](RS_TS_20046)

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

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

SWS Item	**ECUC_FrTSyn_00002 :**		
Name	FrTSynDevErrorDetect		
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		

SWS Item	**ECUC_FrTSyn_00016 :**		
Name	FrTSynMainFunctionPeriod		
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		

SWS Item	**ECUC_FrTSyn_00019 :**		
Name	FrTSynVersionInfoApi		
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

10.2.4 FrTSynGlobalTimeDomain

SWS Item	**ECUC_FrTSyn_00004 :**
Container Name	FrTSynGlobalTimeDomain
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	

SWS Item	**ECUC_FrTSyn_00005 :**	
Name	FrTSynGlobalTimeDomainId	
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		

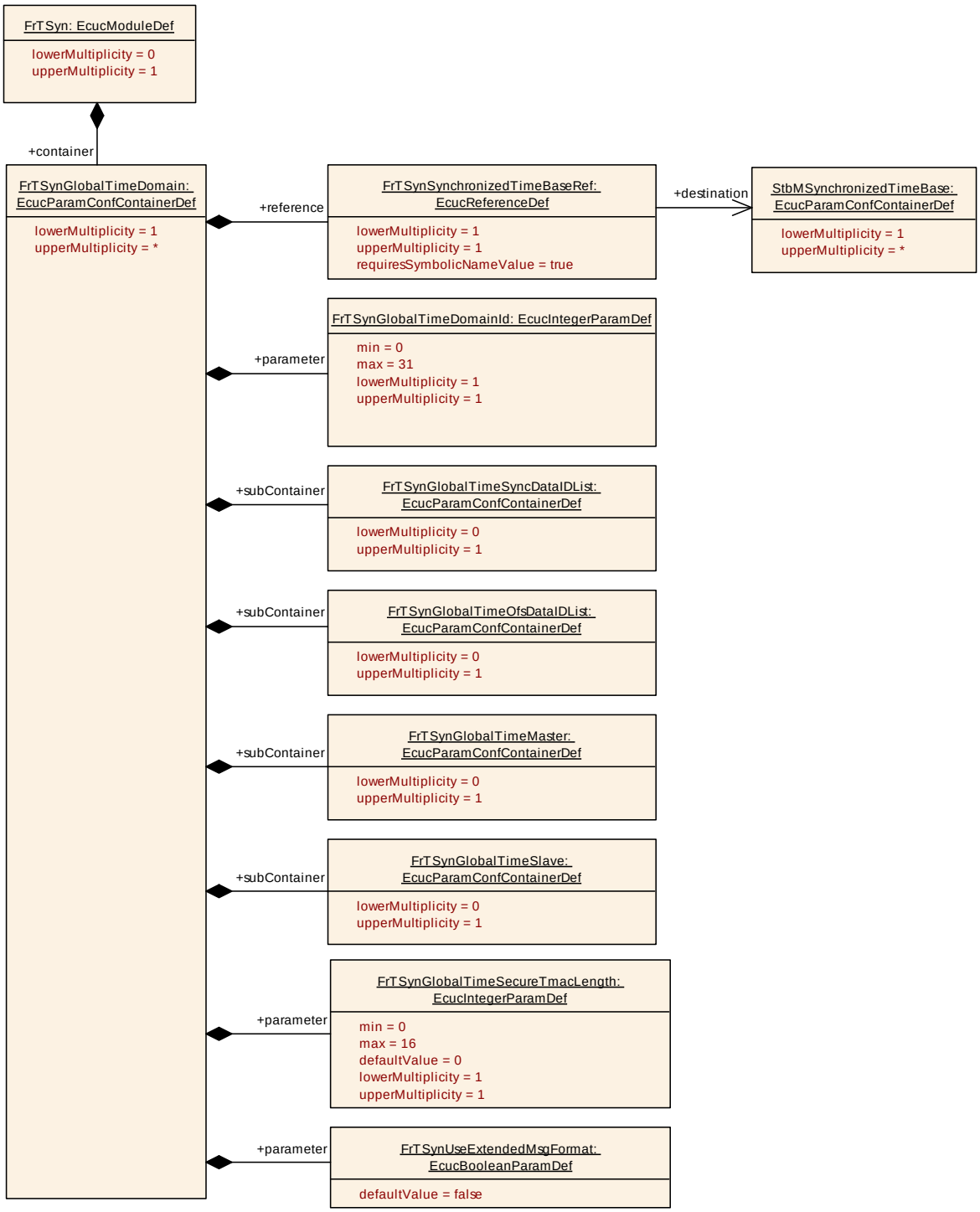
SWS Item	**ECUC_FrTSyn_00034 :**		
Name	FrTSynGlobalTimeSecureTmacLength		
Parent Container	FrTSynGlobalTimeDomain		
Description	Represents the number of bytes for the used Truncated Message Authentication Code (TMAC). If 0, no message authentication will be used. **Tags:** atp.Status=draft		
Multiplicity	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		

SWS Item	**ECUC_FrTSyn_00035 :**		
Name	FrTSynUseExtendedMsgFormat		
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		

SWS Item	**ECUC_FrTSyn_00018 :**		
Name	FrTSynSynchronizedTimeBaseRef		
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.

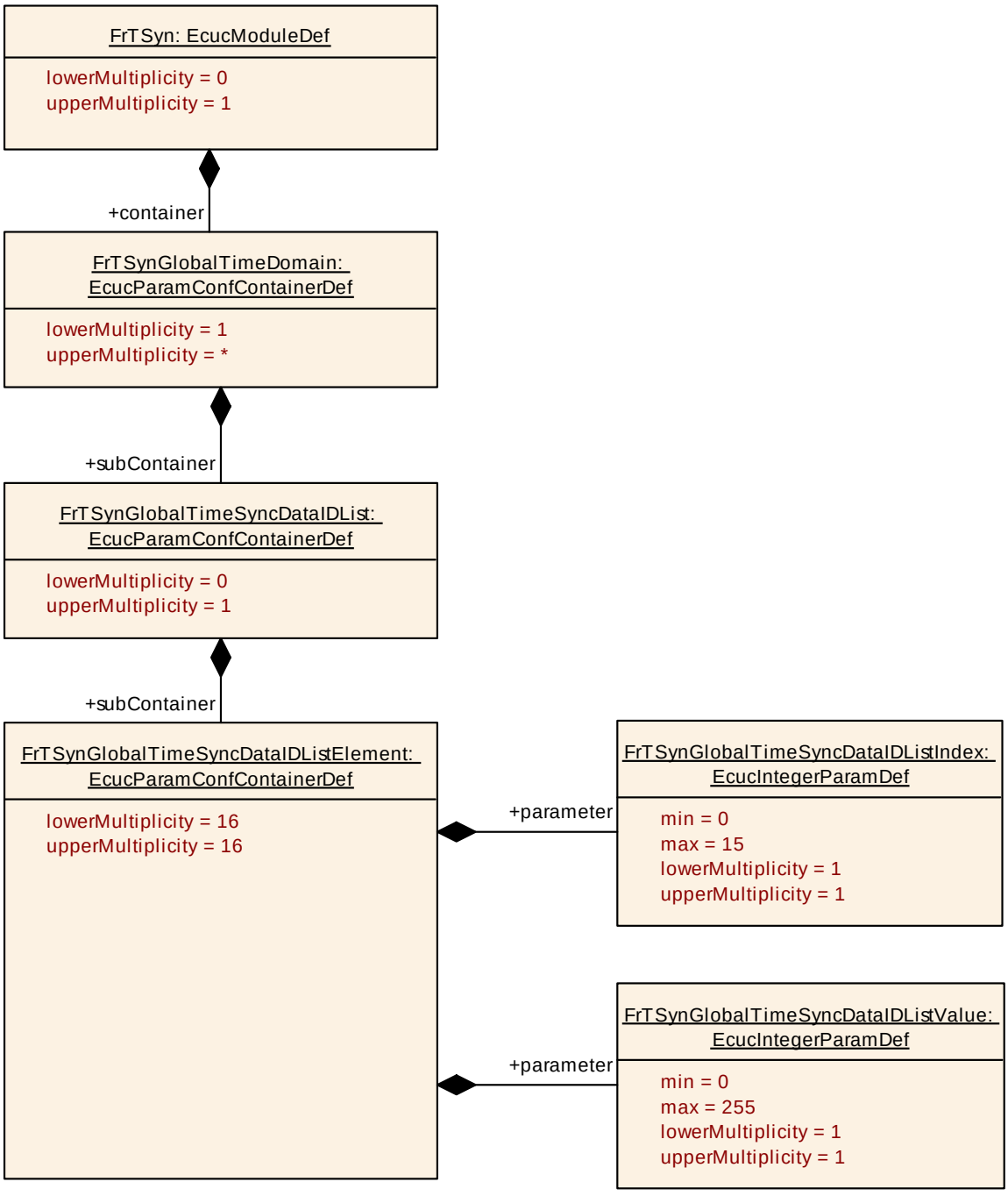


10.2.5 FrTSynGlobalTimeSyncDataIDList

SWS Item	**ECUC_FrTSyn_00023 :**
Container Name	FrTSynGlobalTimeSyncDataIDList
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.



10.2.6 FrTSynGlobalTimeSyncDataIDListElement

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

SWS Item	**ECUC_FrTSyn_00026 :**		
Name	FrTSynGlobalTimeSyncDataIDListIndex		
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		

SWS Item	**ECUC_FrTSyn_00027 :**		
Name	FrTSynGlobalTimeSyncDataIDListValue		
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		

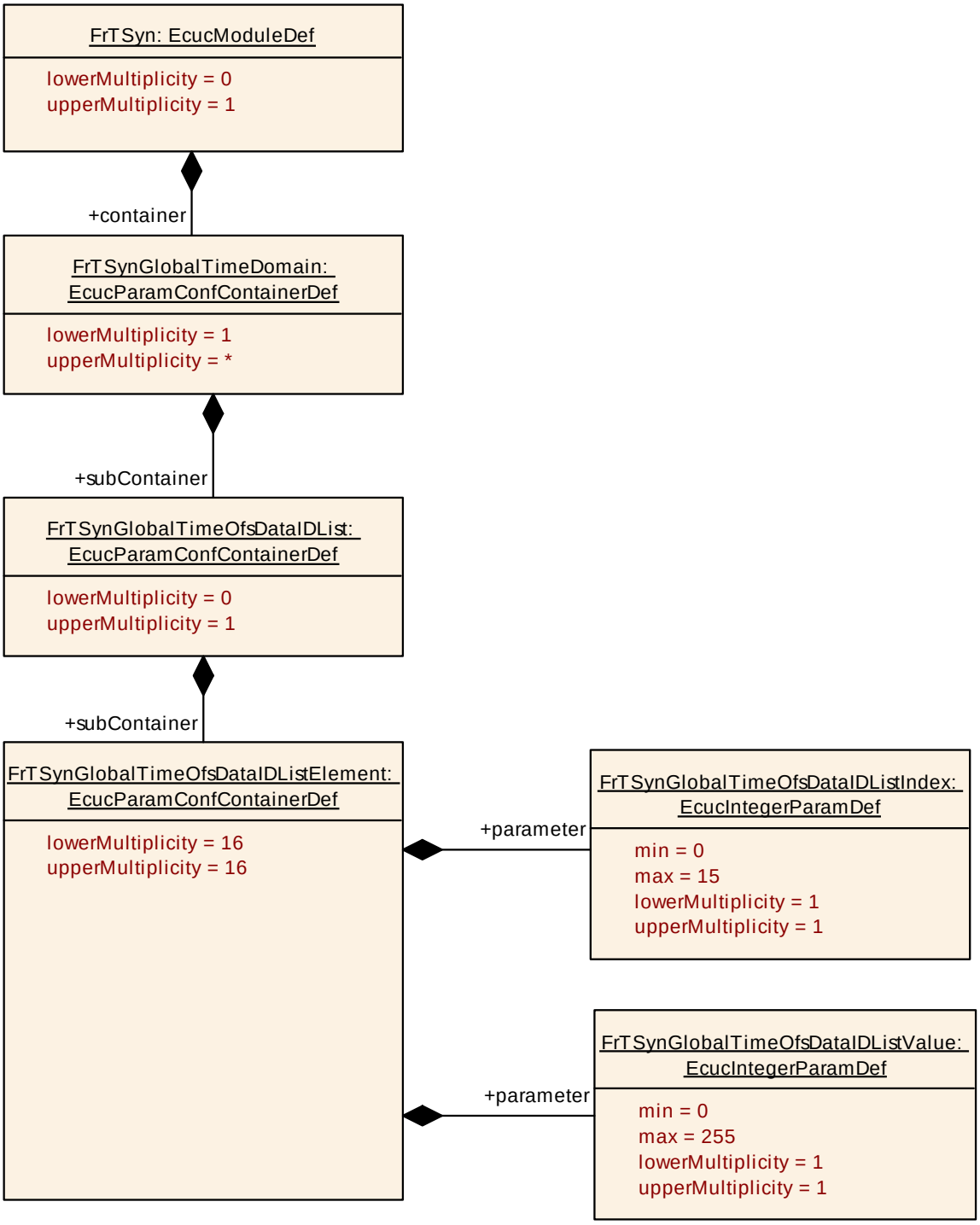
No Included Containers

10.2.7 FrTSynGlobalTimeOfsDataIDList

SWS Item	**ECUC_FrTSyn_00024 :**		
Container Name	FrTSynGlobalTimeOfsDataIDList		
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	***Pre-compile time***	X	All Variants

Class	***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.



10.2.8 FrTSynGlobalTimeOfsDataIDListElement

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

SWS Item	**ECUC_FrTSyn_00029 :**		
Name	FrTSynGlobalTimeOfsDataIDListIndex		
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		

SWS Item	**ECUC_FrTSyn_00030 :**		
Name	FrTSynGlobalTimeOfsDataIDListValue		
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		
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			

SWS Item	**ECUC_FrTSyn_00032 :**
Name	FrTSynCyclicMsgResumeTime
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		

SWS Item	**ECUC_FrTSyn_00033 :**		
Name	FrTSynGlobalTimeDebounceTime		
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		

SWS Item	**ECUC_FrTSyn_00013 :**		
Name	FrTSynGlobalTimeTxCrcSecured		
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		

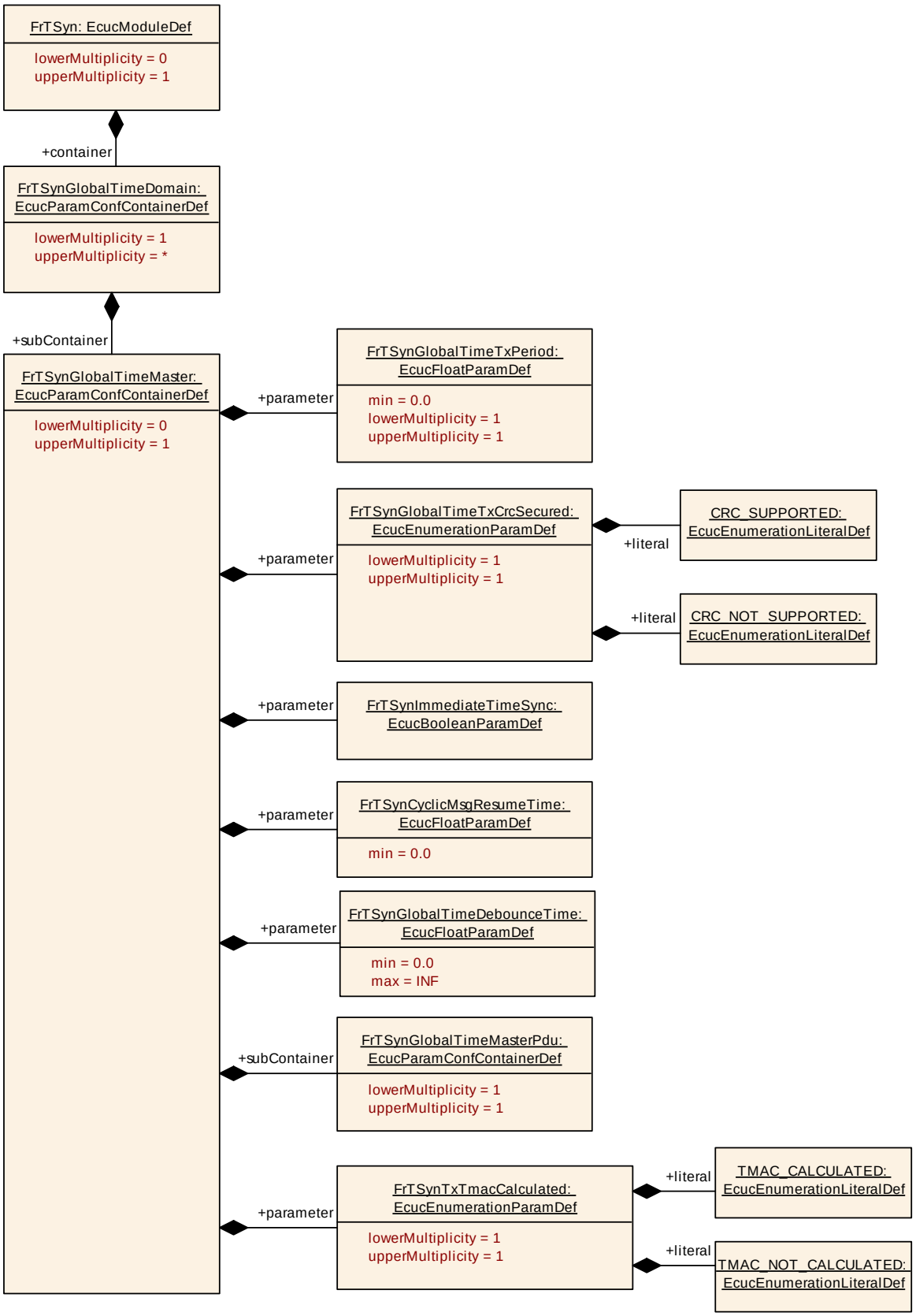
SWS Item	**ECUC_FrTSyn_00014 :**		
Name	FrTSynGlobalTimeTxPeriod		
Parent Container	FrTSynGlobalTimeMaster		
Description	This represents the TX period. Unit: seconds		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF]		
Default value	--		

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

SWS Item	**ECUC_FrTSyn_00031 :**		
Name	FrTSynImmediateTimeSync		
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		

SWS Item	**ECUC_FrTSyn_00036 :**		
Name	FrTSynTxTmacCalculated		
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

10.2.10 FrTSynGlobalTimeMasterPdu

SWS Item	**ECUC_FrTSyn_00008 :**		
Container Name	FrTSynGlobalTimeMasterPdu		
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			

SWS Item	**ECUC_FrTSyn_00007 :**		
Name	FrTSynGlobalTimeMasterHandleId		
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	true		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_FrTSyn_00020 :**		
Name	FrTSynGlobalTimePduRef		
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		
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			

SWS Item	ECUC_FrTSyn_00038 :		
Name	FrTSynGlobalTimeMinMsgGap		
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		

SWS Item	ECUC_FrTSyn_00022 :		
Name	FrTSynGlobalTimeSequenceCounterJumpWidth		
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	1 .. 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		

SWS Item	ECUC_FrTSyn_00039 :		
Name	FrTSynGlobalTimeTmacTimeout		
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		

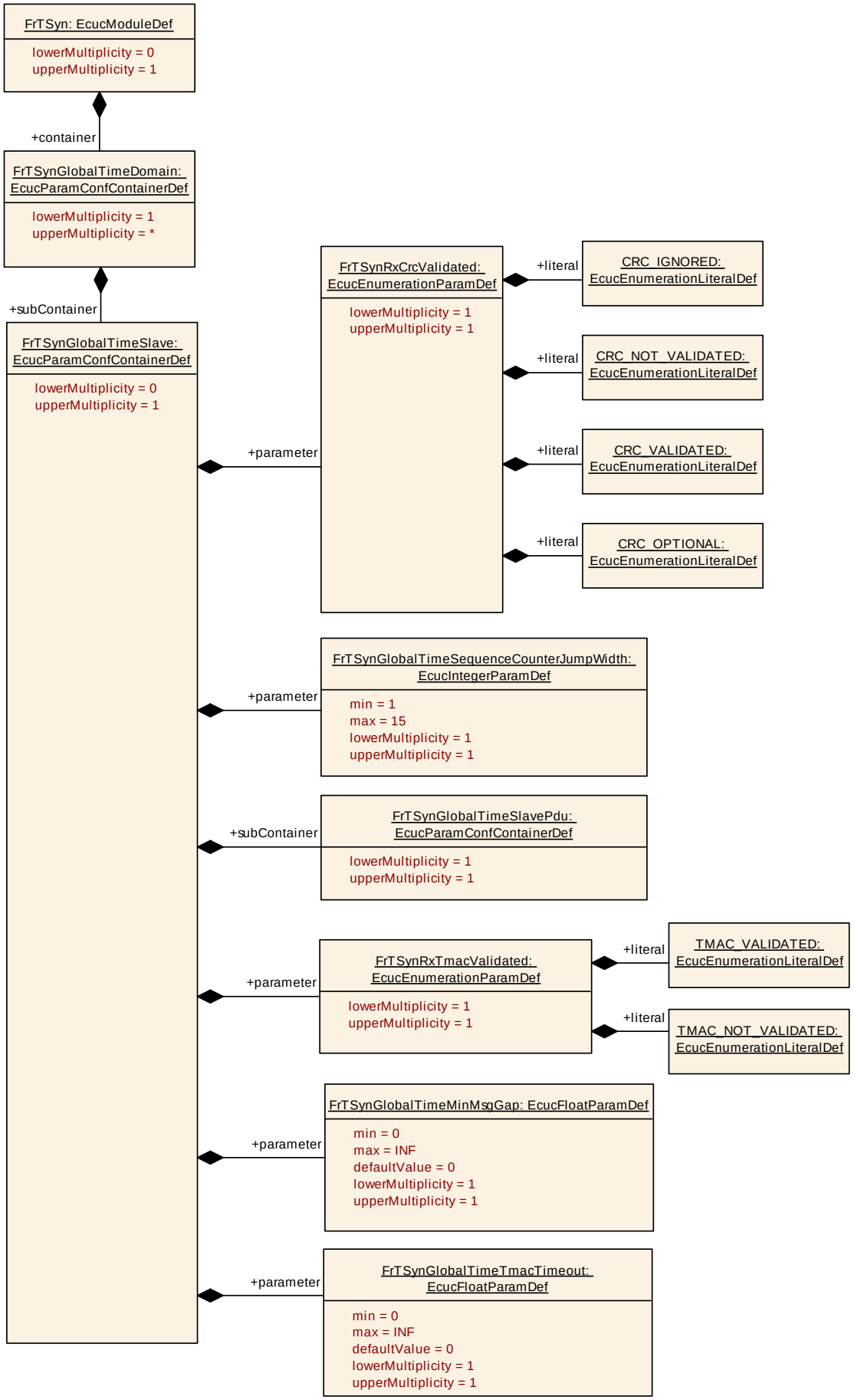
SWS Item	ECUC_FrTSyn_00017 :
Name	FrTSynRxCrcValidated
Parent Container	FrTSynGlobalTimeSlave

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		

SWS Item	**ECUC_FrTSyn_00037 :**		
Name	FrTSynRxTmacValidated		
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.


FrTSyn: EcucModuleDef
lowerMultiplicity = 0
upperMultiplicity = 1
+container
FrTSynGlobalTimeDomain: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+subContainer
FrTSynGlobalTimeSlave: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
FrTSynRxCrcValidated: EcucEnumerationParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
+literal
CRC_IGNORED: EcucEnumerationLiteralDef
+literal
CRC_NOT_VALIDATED: EcucEnumerationLiteralDef
+literal
CRC_VALIDATED: EcucEnumerationLiteralDef
+literal
CRC_OPTIONAL: EcucEnumerationLiteralDef
+parameter
FrTSynGlobalTimeSequenceCounterJumpWidth: EcucIntegerParamDef
min = 1
max = 15
lowerMultiplicity = 1
upperMultiplicity = 1
+subContainer
FrTSynGlobalTimeSlavePdu: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
FrTSynRxTmacValidated: EcucEnumerationParamDef
lowerMultiplicity = 1
upperMultiplicity = 1
+literal
TMAC_VALIDATED: EcucEnumerationLiteralDef
+literal
TMAC_NOT_VALIDATED: EcucEnumerationLiteralDef
+parameter
FrTSynGlobalTimeMinMsgGap: EcucFloatParamDef
min = 0
max = INF
defaultValue = 0
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
FrTSynGlobalTimeTmacTimeout: EcucFloatParamDef
min = 0
max = INF
defaultValue = 0
lowerMultiplicity = 1
upperMultiplicity = 1

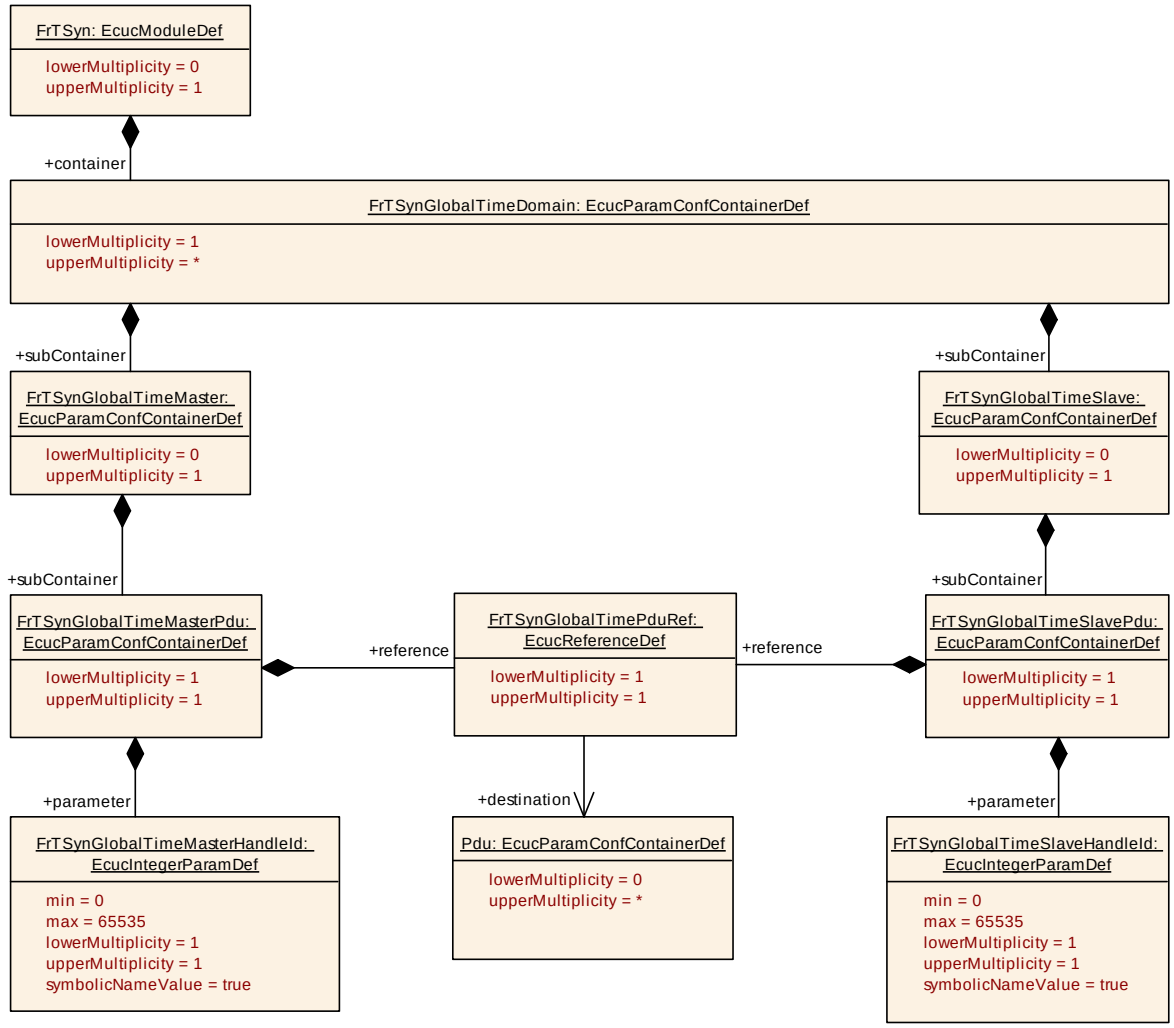
10.2.12 FrTSynGlobalTimeSlavePdu

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

SWS Item	**ECUC_FrTSyn_00011 :**		
Name	FrTSynGlobalTimeSlaveHandleId		
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	true		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_FrTSyn_00021 :**		
Name	FrTSynGlobalTimePduRef		
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



10.3 Published Information

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