

<b>Document Title</b>	Specification of Time Synchronization over FlexRay
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	675

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.3.1

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
2017-12-08	4.3.1	AUTOSAR Release Management	Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Offset message formats changed</li> <li>• Immediate Time Synchronization message transmission</li> <li>• Various enhancements and corrections</li> </ul>
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Error code FRTSYN_E_INVALID_PDU_SDU_ID replaced by FRTSYN_E_INVALID_PDUID</li> <li>• FlexRay communication state handling simplified (Frlf_GetPOCStatus replaced by Frlf_GetState)</li> </ul>
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

<b>Table of Contents</b> .....	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 .....	13
7 Functional specification .....	15
7.1 Overview .....	15
7.2 Module Handling .....	15
7.2.1 Initialization .....	15
7.2.2 FlexRay Interface.....	16
7.3 Message Format.....	16
7.3.1 SYNC message .....	17
7.3.2 OFS message.....	17
7.4 Acting as Time Master.....	18
7.4.1 SYNC message processing .....	19
7.4.2 OFS message processing.....	20
7.4.3 Transmission mode.....	20
7.4.4 Debounce Time.....	21
7.4.5 Immediate Time Synchronization.....	21
7.4.6 Calculation and Assembling of Time Synchronization Messages .....	22
7.5 Acting as Time Slave.....	25
7.5.1 SYNC message processing .....	25
7.5.2 OFS message processing.....	25
7.5.3 Validation and Disassembling of Time Synchronization Messages .....	26
7.6 Global Time Measurement Support.....	29
7.7 Error Classification .....	29
7.7.1 Development Errors .....	29
7.7.2 Runtime Errors.....	29
7.7.3 Transient Faults .....	29
7.7.4 Production Errors.....	30
7.7.5 Extended Production Errors.....	30

8	API specification.....	31
8.1	API.....	31
8.1.1	Imported types .....	31
8.1.2	Type definitions.....	31
8.1.3	Function definitions .....	32
8.1.4	Call-back notifications .....	33
8.1.5	Scheduled functions.....	35
8.1.6	Expected Interfaces .....	36
9	Sequence diagrams .....	37
9.1	StbM_GetCurrentTime <Master FR SYNC>.....	37
9.2	StbM_BusSetGlobalTime <Slave FR SYNC> .....	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 .....	44
10.2.6	FrTSynGlobalTimeSyncDataIDListElement.....	45
10.2.7	FrTSynGlobalTimeOfsDataIDList .....	46
10.2.8	FrTSynGlobalTimeOfsDataIDListElement .....	47
10.2.9	FrTSynGlobalTimeMaster.....	48
10.2.10	FrTSynGlobalTimeMasterPdu .....	51
10.2.11	FrTSynGlobalTimeSlave.....	52
10.2.12	FrTSynGlobalTimeSlavePdu .....	54
10.3	Published Information.....	56

## 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.

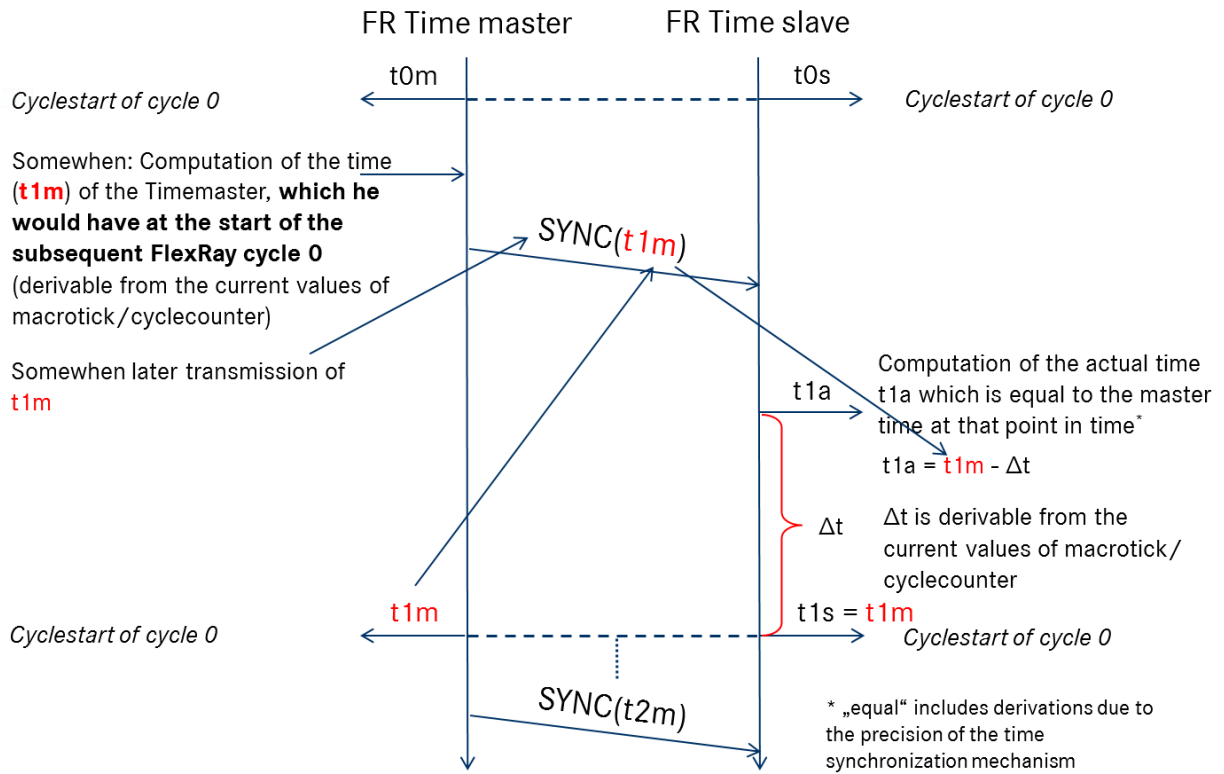
Both, Time Master and Time Slaves have the same view on the FlexRay global time. It is therefore just necessary to define the same point in (FlexRay) time and to transmit the time information, which will be valid at that point in (FlexRay) time.

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 10 $\mu$ s.

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 (Frlf) and the Default Error Tracer (DET).

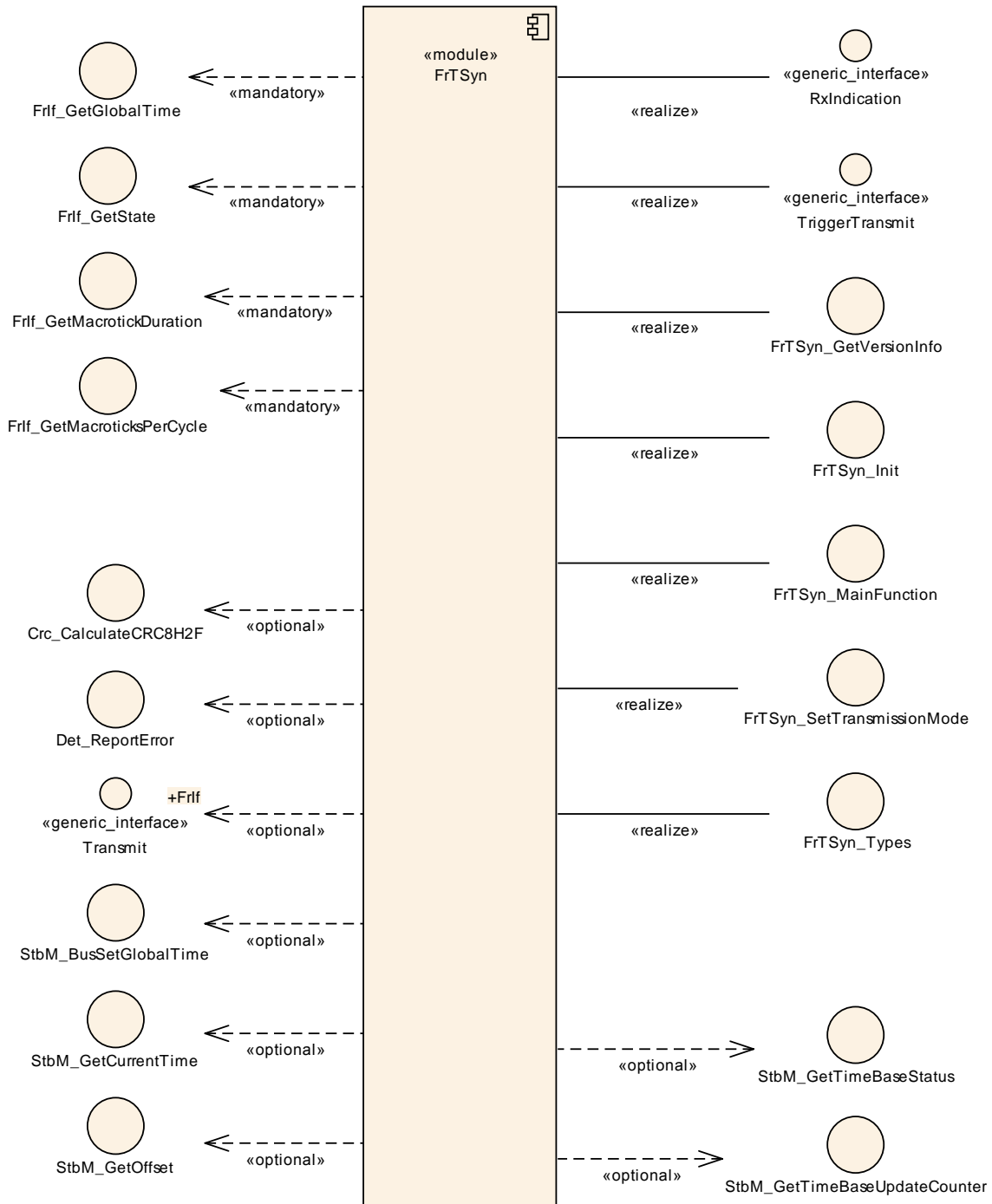


Figure 2: Module dependencies of the FrTSyn module

- 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].

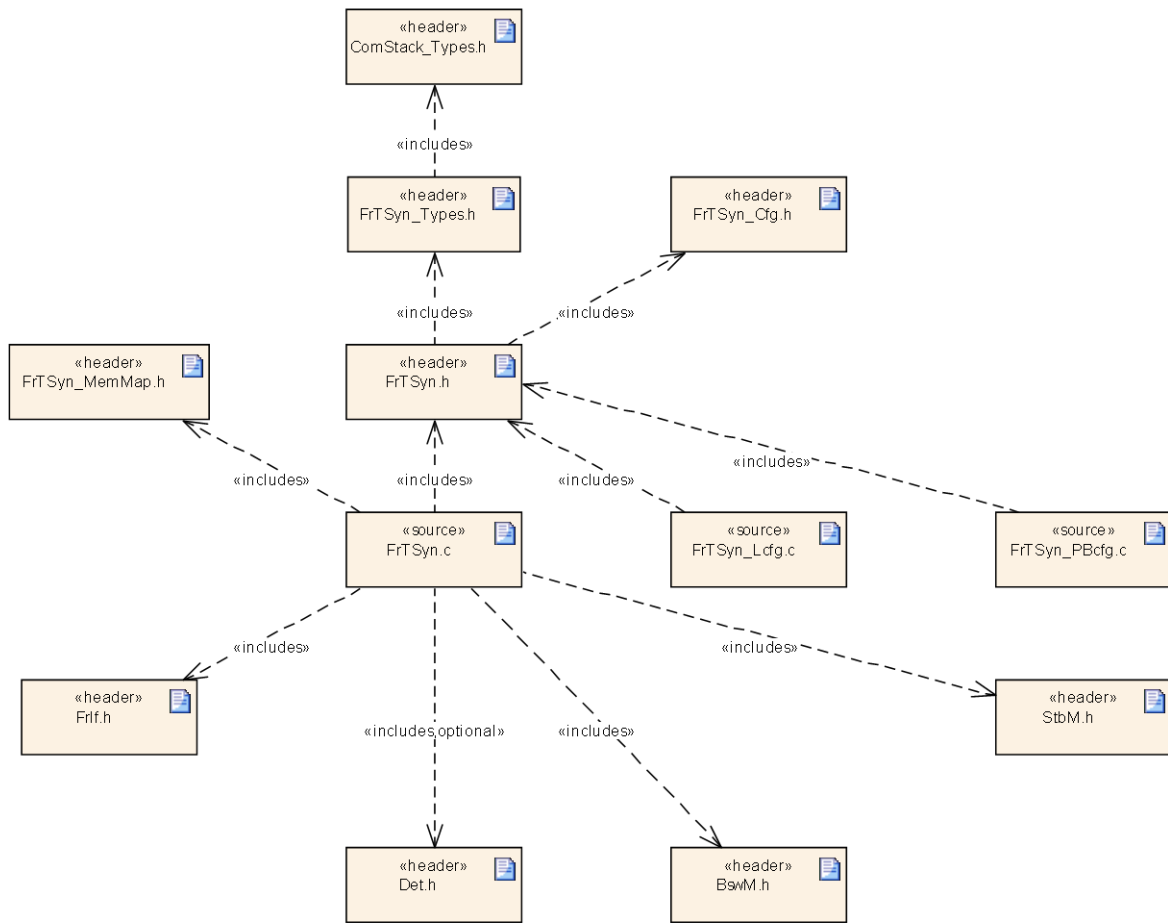
### 5.1.2 Header file structure

Besides the files defined in section 5.1.7 "Header file structure" of the SWS BSW General [3], the Time Synchronization over FlexRay needs to include the files defined below.

#### **[SWS\_FrTSyn\_00002]**

The implementation header files shall include *ComStack\_Types.h*.  
(SRS\_BSW\_00301, SRS\_BSW\_00456)

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



**Figure 3: File structure of FrTSyn**

## 6 Requirements traceability

Requirement	Description	Satisfied by
SRS_BSW_00301	All AUTOSAR Basic Software Modules shall only import the necessary information	SWS_FrTSyn_00002
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_00004, SWS_FrTSyn_00067, SWS_FrTSyn_00070, SWS_FrTSyn_00095
SRS_BSW_00385	List possible error notifications	SWS_FrTSyn_00059
SRS_BSW_00456	- A Header file shall be defined in order to harmonize BSW Modules	SWS_FrTSyn_00002
SRS_StbM_20018	The StbM shall initialize the Local Time Base with 0 at startup if configured as Time Slave	SWS_FrTSyn_00003, SWS_FrTSyn_00005
SRS_StbM_20019	The StbM shall initialize the Global Time Base with a configurable startup value if configured as Time Master	SWS_FrTSyn_00003, SWS_FrTSyn_00005
SRS_StbM_20039	The FlexRay Timesync 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
SRS_StbM_20040	The FlexRay Timesync 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
SRS_StbM_20041	The FlexRay Timesync 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
SRS_StbM_20042	The FlexRay Timesync 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
SRS_StbM_20043	The FlexRay Timesync 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
SRS_StbM_20044	The FlexRay Timesync 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
SRS_StbM_20045	The FlexRay Timesync 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
SRS_StbM_20046	The FlexRay Timesync configuration shall allow the FrTSyn to support different roles for a Time Base	SWS_FrTSyn_00077
SRS_StbM_20057	The StbM shall provide measurement data to the application	SWS_FrTSyn_00092

## 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.  
J(SRS\_StbM\_20018, SRS\_StbM\_20019)

##### [SWS\_FrTSyn\_00004]

When DET reporting is enabled (see `FrTSynDevErrorDetect`), the Time Synchronization over FlexRay shall call `Det_ReportError()` with the error code `FRTSYN_E_UNINIT` when any API other than `FrTSyn_GetVersionInfo()` or `FrTSyn_Init()` is called in uninitialized state.  
J(SRS\_BSW\_00337)

##### [SWS\_FrTSyn\_00005]

When `FrTSyn_Init()` is called in initialized state, the Time Synchronization over FlexRay shall re-initialize its internal variables.  
J(SRS\_StbM\_20018, SRS\_StbM\_20019)

##### [SWS\_FrTSyn\_00006]

The Sequence Counter (SC) shall be initialized with 0.  
J(SRS\_StbM\_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.

](SRS\_StbM\_20040, SRS\_StbM\_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”.

](SRS\_StbM\_20043, SRS\_StbM\_20044)

### [SWS\_FrTSyn\_00009]

The PayloadLength is 16.

](SRS\_StbM\_20043, SRS\_StbM\_20044)

### [SWS\_FrTSyn\_00010]

Time Synchronization messages contain User Data according to the given message format.

](SRS\_StbM\_20043, SRS\_StbM\_20044, SRS\_StbM\_20045)

### [SWS\_FrTSyn\_00011]

User Data shall be read consistently from the incoming Time Synchronization messages.

](SRS\_StbM\_20045)

### [SWS\_FrTSyn\_00012]

User Data shall be written consistently to outgoing Time Synchronization messages.

](SRS\_StbM\_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.

](SRS\_StbM\_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

](SRS\_StbM\_20041, SRS\_StbM\_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

](SRS\_StbM\_20041, SRS\_StbM\_20042, SRS\_StbM\_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  
J(SRS\_StbM\_20041, SRS\_StbM\_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  
J(SRS\_StbM\_20041, SRS\_StbM\_20042, SRS\_StbM\_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.

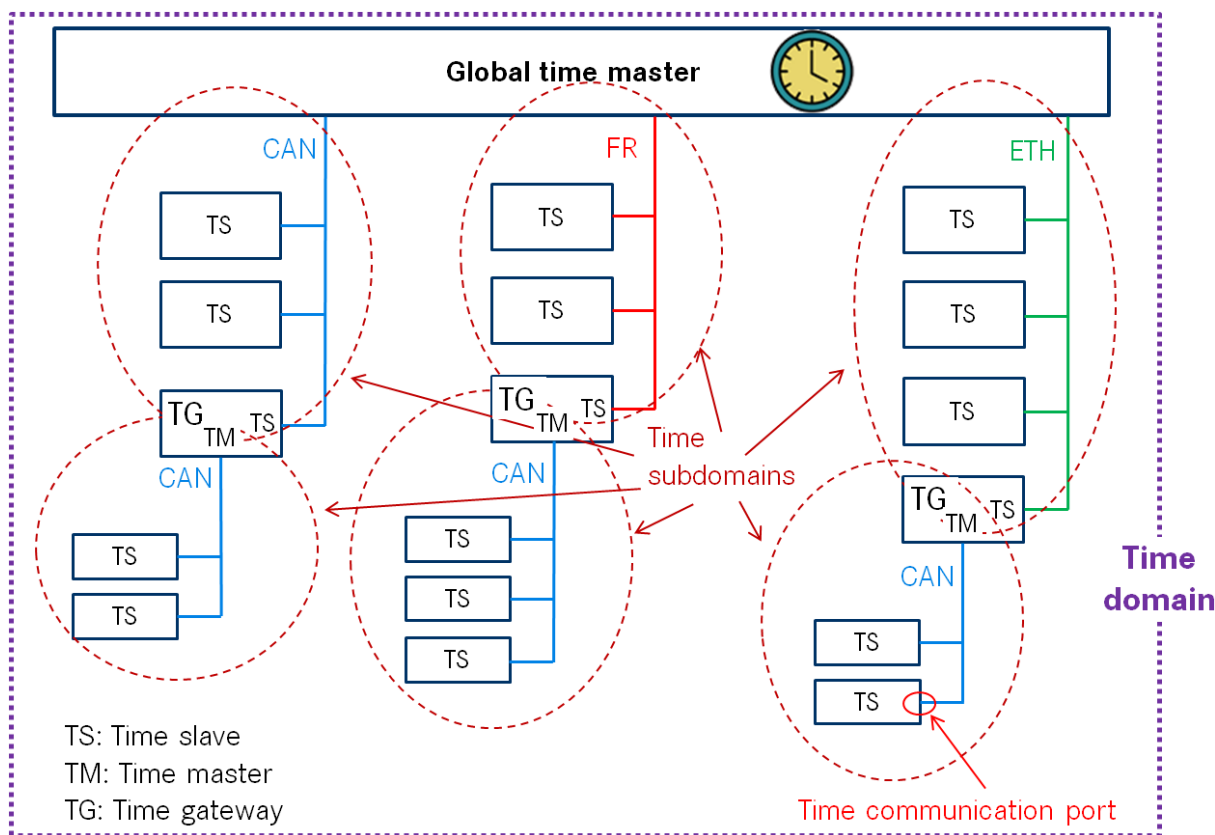


Figure 4: Terminology Example

### 7.4.1 SYNC message processing

#### [SWS\_FrTSyn\_00018]

A Time Synchronization message sequence consists of a SYNC message per Time Domain.

](SRS\_StbM\_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 5: Master FR SYNC) 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).

](SRS\_StbM\_20039, SRS\_StbM\_20043)

#### [SWS\_FrTSyn\_00021]

Depending on `FrTsynGlobalTimeTxCrcSecured` (**ECUC\_FrTsyn\_00013** : ) the SYNC message shall be of type:

<code>FrTsynGlobalTimeTxCrcSecured</code>	SYNC
<code>CRC_NOT_SUPPORTED</code>	0x10 SYNC not CRC secured message
<code>CRC_SUPPORTED</code>	0x20 SYNC CRC secured message

](SRS\_StbM\_20041, SRS\_StbM\_20043)

## 7.4.2 OFS message processing

### [SWS\_FrTsyn\_00022]

An offset message sequence consists of an OFS message per Time Domain.

](SRS\_StbM\_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).

](SRS\_StbM\_20039, SRS\_StbM\_20044)

### [SWS\_FrTsyn\_00025]

Depending on `FrTsynGlobalTimeTxCrcSecured` (**ECUC\_FrTsyn\_00013** : ) the OFS message shall be of type:

<code>FrTsynGlobalTimeTxCrcSecured</code>	OFS
<code>CRC_NOT_SUPPORTED</code>	0x34 OFS not CRC secured message
<code>CRC_SUPPORTED</code>	0x44 OFS CRC secured message

](SRS\_StbM\_20041, SRS\_StbM\_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.

](SRS\_StbM\_20039, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20039, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_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.

](SRS\_StbM\_20039)

##### [SWS\_FrTSyn\_00086]

A new Timesync PDU shall only be sent if the corresponding `debounceCounter` has a value equal or less than zero.

](SRS\_StbM\_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.

](SRS\_StbM\_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.

J(SRS\_StbM\_20039)

**Note:** `timeBaseStatus` can be obtained by `StbM_GetTimeBaseStatus()` or `StbM_GetCurrentTime()`.

**Note:** The `debounceTimer` 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.

J(SRS\_StbM\_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.

J(SRS\_StbM\_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.

J(SRS\_StbM\_20039)

**[SWS\_FrTSyn\_00093]**

If the `cyclicMsgResumeCounter` is started, `FrTSyn` shall stop cyclic `Timesync` message transmission.

J(SRS\_StbM\_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<sup>st</sup> step) and how the message will be assembled (2<sup>nd</sup> 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. Get `currentCycle` and `currentMacroticks` via `FrIf_GetGlobalTime()`

2. Retrieve current Synchronized Time Base value as  $T_{\text{SYNC}}$  via  
`StbM_GetCurrentTime()`
3. Calculate the (future) time value of the Time Base at the start of the next FlexRay cycle by  
$$T_0 = T_{\text{SYNC}} + (\text{MacroticksPerCycle} * (64 - \text{currentCycle}) - \text{currentMacroticks}) * \text{MacrotickDuration}$$
4. Calculate *SyncTimeSec* (second portion of  $T_0$ ) and *SyncTimeNSec* (nanosecond portion of  $T_0$ )

](SRS\_StbM\_20043)

**Note:** `MacroticksPerCycle` and `MacrotickDuration` are given statically by configuration.

#### [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*

](SRS\_StbM\_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*.

](SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20041, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20041, SRS\_StbM\_20043, SRS\_StbM\_20044)

**[SWS\_FrTSyn\_00035]**

The `DataID` shall be calculated as  $\text{DataID} = \text{DataIDList}[\text{SC}]$ , where `DataIDList` (**ECUC\_FrTSyn\_00023 : ECUC\_FrTSyn\_00024** : ) is given by configuration for each message *Type*.

](SRS\_StbM\_20041, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20041, SRS\_StbM\_20043, SRS\_StbM\_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)

](SRS\_StbM\_20043, SRS\_StbM\_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.  
J(SRS\_StbM\_20042, SRS\_StbM\_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.  
J(SRS\_StbM\_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.  
J(SRS\_StbM\_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.  
J(SRS\_StbM\_20042, SRS\_StbM\_20043)

#### [SWS\_FrTSyn\_00041]

For valid SYNC messages a new Global Time value shall be calculated and forwarded to the StbM module via StbM\_BusSetGlobalTime() (see Figure 6: Slave FR SYNC).  
J(SRS\_StbM\_20040, SRS\_StbM\_20042, SRS\_StbM\_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.  
J(SRS\_StbM\_20042, SRS\_StbM\_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.  
J(SRS\_StbM\_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.  
J(SRS\_StbM\_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.  
J(SRS\_StbM\_20042, SRS\_StbM\_20044)

**[SWS\_FrTSyn\_00045]**

For valid OFS messages a new Offset Time value shall be calculated (according **[SWS\_FrTSyn\_00047]**) and forwarded to the StbM module via StbM\_BusSetGlobalTime().  
J(SRS\_StbM\_20040, SRS\_StbM\_20042, SRS\_StbM\_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 (1<sup>st</sup> step) and how the message will be disassembled (2<sup>nd</sup> 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  $T_0$  (`SyncTimeSec`, `SyncTimeNSec`)
2. Get `currentCycle` and `currentMacroticks` via `FrIf_GetGlobalTime()`
3. Calculate  $T_1$  to update the Time Slave's Local Time Base as:  

$$T_1 = T_0 + (((\text{MacroticksPerCycle} * \text{currentCycle}) + \text{currentMacroticks}) * \text{MacrotickDuration})$$
4. If `currentCycle` is greater or equal than the retrieved FCNT value from the transmitter (Time Master), then the calculated value  $T_1$  shall be subtracted by 64 times the FR cycle duration:  

$$T_1 = T_1 - (\text{MacroticksPerCycle} * 64 * \text{MacrotickDuration})$$

J(SRS\_StbM\_20043)

**Note:** `MacroticksPerCycle` and `MacrotickDuration` are given statically by configuration.

#### [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`

](SRS\_StbM\_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`.

](SRS\_StbM\_20040, SRS\_StbM\_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.  
](SRS\_StbM\_20042, SRS\_StbM\_20043, SRS\_StbM\_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 1<sup>st</sup> received `SYNC` (or `OFS`) message per Time Domain against the defined Sequence Counter Jump Width.

](SRS\_StbM\_20042, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20042, SRS\_StbM\_20043, SRS\_StbM\_20044)

**[SWS\_FrTSyn\_00054]**

The `DataID` shall be calculated as `DataID = DataIDList[SC]`, where `DataIDList` is given by configuration for each message *Type*.

](SRS\_StbM\_20042, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20042, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20043, SRS\_StbM\_20044)

**[SWS\_FrTSyn\_00057]**

For each received Time Synchronization message the `FrTSyn` shall disassemble the message after successful validation **[SWS\_FrTSyn\_00056]**.

](SRS\_StbM\_20042, SRS\_StbM\_20043, SRS\_StbM\_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.

](SRS\_StbM\_20057)

## 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.

](SRS\_StbM\_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:

<i>Type or error</i>	<i>Related error code</i>	<i>Value [hex]</i>
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 files are listed:

##### [SWS\_FrTSyn\_00060] [

<i>Module</i>	<i>Imported Type</i>
ComStack_Types	PduIdType
	PduInfoType
FrIf	FrIf_StateType
StbM	StbM_MeasurementType
	StbM_SynchronizedTimeBaseType
	StbM_TimeBaseStatusType
	StbM_TimeStampType
	StbM_UserDataType
Std_Types	Std_ReturnType
	Std_VersionInfoType

] (SRS\_StbM\_20043)

#### 8.1.2 Type definitions

##### 8.1.2.1 FrTSyn\_ConfigType

##### [SWS\_FrTSyn\_00061] [

<b>Name:</b>	FrTSyn_ConfigType		
<b>Type:</b>	Structure		
<b>Element:</b>	void	implementation specific	--
<b>Description:</b>	<p>This is the base type for the configuration of the Time Synchronization over FlexRay.</p> <p>A pointer to an instance of this structure will be used in the initialization of the Time Synchronization over FlexRay.</p> <p>The content of this structure is defined in chapter 10 Configuration specification.</p>		

] (SRS\_StbM\_20043)

##### 8.1.2.2 FrTSyn\_TransmissionModeType

##### [SWS\_FrTSyn\_00062] [

<b>Name:</b>	FrTSyn_TransmissionModeType		
<b>Type:</b>	Enumeration		
<b>Range:</b>	FRTSYN_TX_OFF	--	Transmission Disabled

	FRTSYN_TX_ON	--	Transmission Enabled
<b>Description:</b>	Handles the enabling and disabling of the transmission mode		

] (SRS\_StbM\_20043)

### 8.1.3 Function definitions

#### 8.1.3.1 FrTSyn\_Init

[SWS\_FrTSyn\_00063] [

<b>Service name:</b>	FrTSyn_Init		
<b>Syntax:</b>	<pre>void FrTSyn_Init(     const FrTSyn_ConfigType* configPtr )</pre>		
<b>Service ID[hex]:</b>	0x01		
<b>Sync/Async:</b>	Synchronous		
<b>Reentrancy:</b>	Non Reentrant		
<b>Parameters (in):</b>	configPtr		Pointer to selected configuration structure
<b>Parameters (inout):</b>	None		
<b>Parameters (out):</b>	None		
<b>Return value:</b>	None		
<b>Description:</b>	This function initializes the Time Synchronization over FlexRay.		

] (SRS\_StbM\_20043)

See section 7.2.1 for details.

#### 8.1.3.2 FrTSyn\_GetVersionInfo

[SWS\_FrTSyn\_00064] [

<b>Service name:</b>	FrTSyn_GetVersionInfo		
<b>Syntax:</b>	<pre>void FrTSyn_GetVersionInfo(     Std_VersionInfoType* versioninfo )</pre>		
<b>Service ID[hex]:</b>	0x02		
<b>Sync/Async:</b>	Synchronous		
<b>Reentrancy:</b>	Non Reentrant		
<b>Parameters (in):</b>	None		
<b>Parameters (inout):</b>	None		
<b>Parameters (out):</b>	versioninfo		Pointer to where to store the version information of this module.
<b>Return value:</b>	None		
<b>Description:</b>	Returns the version information of this module.		

] (SRS\_StbM\_20043)

#### 8.1.3.3 FrTSyn\_SetTransmissionMode

[SWS\_FrTSyn\_00065] [

<b>Service name:</b>	FrTSyn_SetTransmissionMode		
<b>Syntax:</b>	<pre>void FrTSyn_SetTransmissionMode(     uint8 CtrlIdx,     FrTSyn_TransmissionModeType Mode )</pre>		



<b>Service ID[hex]:</b>	0x03	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	CtrlIdx	Index of the FlexRay channel
	Mode	FRTSYN_TX_OFF FRTSYN_TX_ON
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	This API is used to turn on and off the TX capabilities of the FrTSyn.	

] (SRS\_StbM\_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. The function prototypes of the callback functions shall be provided in the file `FrTSyn_Cbk.h`.

### 8.1.4.1 FrTSyn\_RxIndication

#### [SWS\_FrTSyn\_00066] [

<b>Service name:</b>	FrTSyn_RxIndication	
<b>Syntax:</b>	<pre>void FrTSyn_RxIndication(     PduIdType RxPduId,     const PduInfoType* PduInfoPtr )</pre>	
<b>Service ID[hex]:</b>	0x42	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different Pduls. Non reentrant for the same Pdul.	
<b>Parameters (in):</b>	RxPdul	ID of the received PDU.
	PdulInfoPtr	Contains the length ( <code>SduLength</code> ) of the received PDU, a pointer to a buffer ( <code>SduDataPtr</code> ) containing the PDU, and the <code>MetaData</code> related to this PDU.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Indication of a received PDU from a lower layer communication interface module.	

] (SRS\_StbM\_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]** [

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

](SRS\_StbM\_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] [

<b>Service name:</b>	FrTSyn_MainFunction
<b>Syntax:</b>	<code>void FrTSyn_MainFunction(     void )</code>
<b>Service ID[hex]:</b>	0x04
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	None
<b>Parameters (inout):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Main function for cyclic call / resp. Timesync message transmission

] (SRS\_StbM\_20043)

[SWS\_FrTSyn\_00072][

The frequency of invocations of `FrTSyn_MainFunction()` is determined by the configuration parameter `FrTSynMainFunctionPeriod` (refer to **ECUC\_FrTSyn\_00016** : ).

](SRS\_StbM\_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] [

<i>API function</i>	<i>Description</i>
FrIf_GetGlobalTime	Wraps the FlexRay Driver API function Fr_GetGlobalTime().
FrIf_GetMacrotickDuration	Retrieves the Duration of a Macrotick in ns
FrIf_GetMacroticksPerCycle	Retrieves the amount of Macroticks per Cycle
FrIf_GetState	Get current FrIf state.

] (SRS\_StbM\_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] [

<i>API function</i>	<i>Description</i>
Crc_CalculateCRC8H2F	This service makes a CRC8 calculation with the Polynomial 0x2F on Crc_Length
Det_ReportError	Service to report development errors.
FrIf_Transmit	Requests transmission of a PDU.
StbM_BusSetGlobalTime	Allows the Time Base Provider Modules to forward a new Global Time value to the StbM, which has been received from a bus.
StbM_GetCurrentTime	Returns a time value (Local Time Base derived from Global Time Base) in standard format.
StbM_GetOffset	Allows the Timesync Modules to get the current Offset Time and User Data.
StbM_GetTimeBaseStatus	Returns the detailed status of the Time Base. For Offset Time Bases the status of the Offset Time Base itself and the status of the underlying Synchronized Time Base is returned.
StbM_GetTimeBaseUpdateCounter	Allows the Timesync Modules to detect, whether a Time Base should be transmitted immediately in the subsequent <Bus>TSyn_MainFunction() cycle.

] (SRS\_StbM\_20043)

## 9 Sequence diagrams

### 9.1 StbM\_GetCurrentTime <Master FR SYNC>

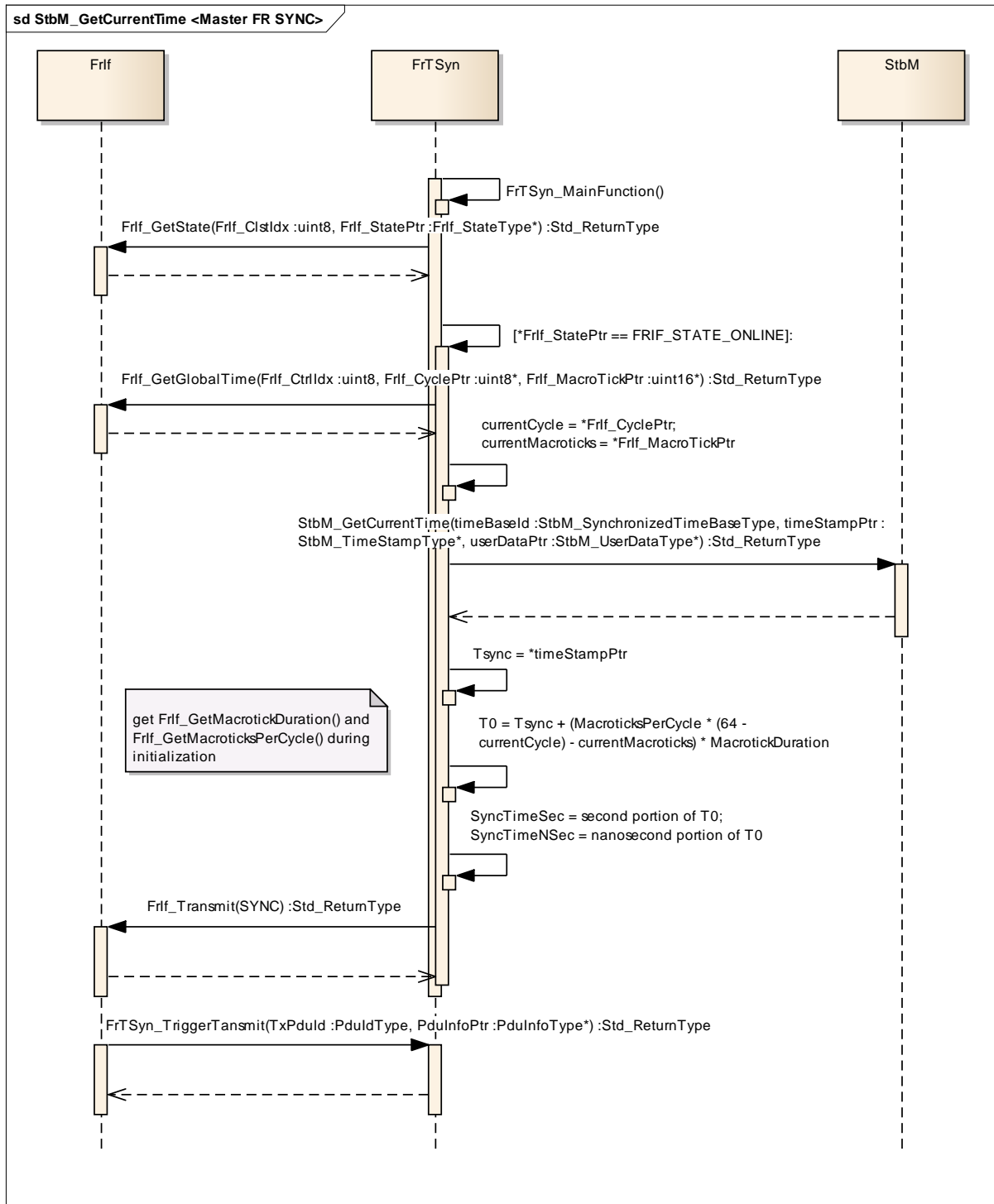


Figure 5: Master FR SYNC

## 9.2 StbM\_BusSetGlobalTime <Slave FR SYNC>

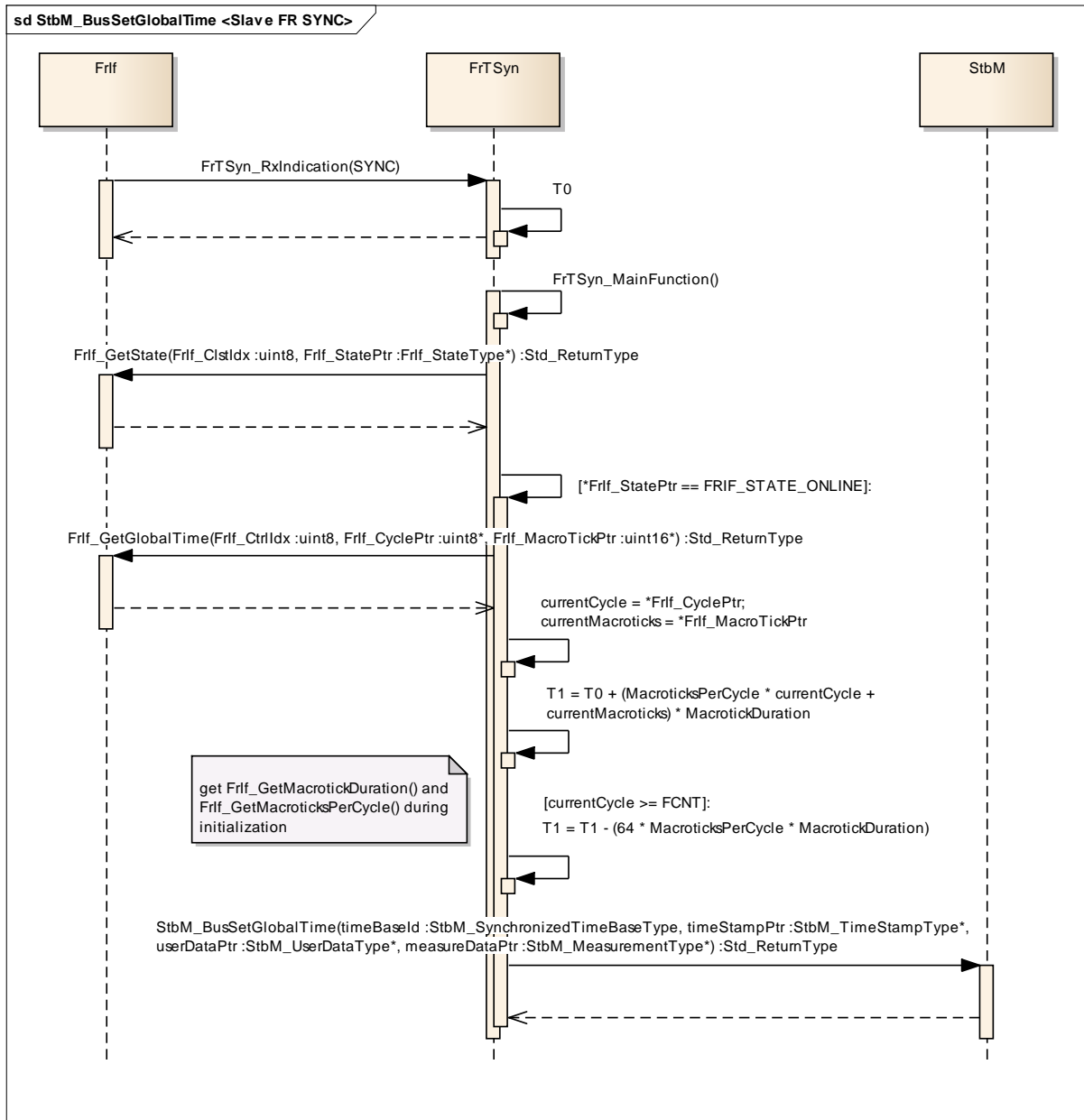


Figure 6: Slave FR SYNC

## 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.

](SRS\_StbM\_20046)

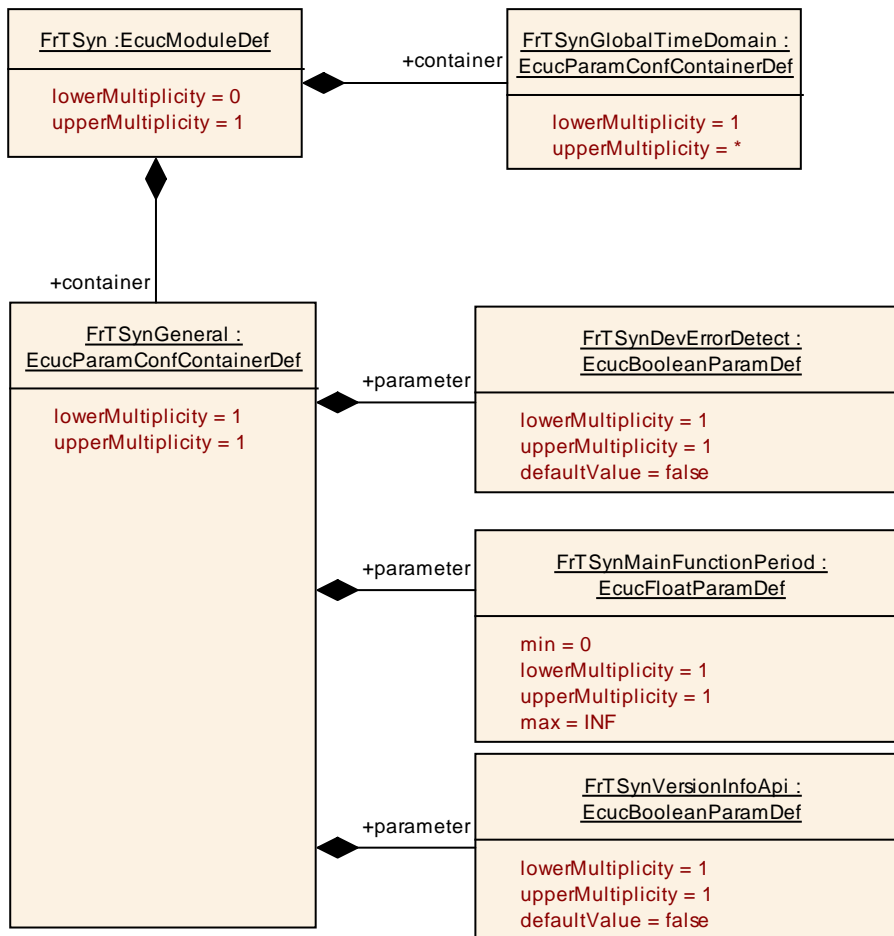
The module supports different post-build variants (previously known as post-build selectable configuration sets), but not post-build loadable configuration.

### 10.2.2 FrTSyn

<b>SWS Item</b>	<b>ECUC_FrTSyn_00001 :</b>
<b>Module Name</b>	<i>FrTSyn</i>
<b>Module Description</b>	This represents the specific configuration variant for the TSyn on Flexray.
<b>Post-Build Variant Support</b>	true
<b>Supported Config Variants</b>	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
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

<b>SWS Item</b>	<b>ECUC_FrTSyn_00003 :</b>
<b>Container Name</b>	FrTSynGeneral
<b>Description</b>	This container holds the general parameters of the Flexray-specific Synchronized Time-base Manager
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_FrTSyn_00002 :</b>		
<b>Name</b>	FrTSynDevErrorDetect		
<b>Parent Container</b>	FrTSynGeneral		
<b>Description</b>	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> <li>• true: detection and notification is enabled.</li> <li>• false: detection and notification is disabled.</li> </ul>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_FrTSyn_00016 :</b>		
<b>Name</b>	FrTSynMainFunctionPeriod		
<b>Parent Container</b>	FrTSynGeneral		
<b>Description</b>	Schedule period of the main function FrTSyn_MainFunction. Unit: [s].		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF[		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

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

**No Included Containers**

### 10.2.4 FrTSynGlobalTimeDomain

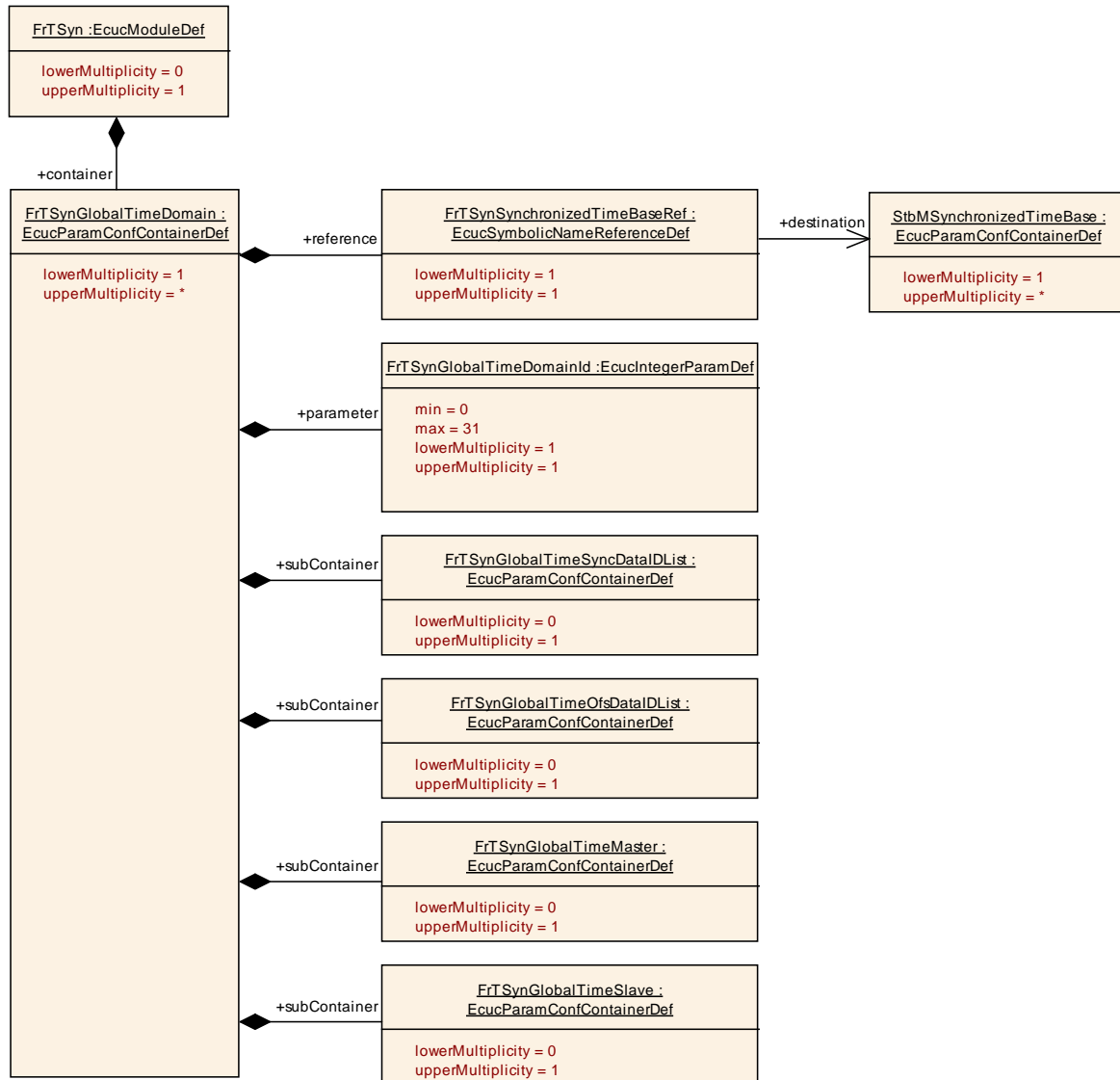
<b>SWS Item</b>	<b>ECUC_FrTSyn_00004 :</b>		
<b>Container Name</b>	FrTSynGlobalTimeDomain		
<b>Description</b>	<p>This represents the existence of a global time domain on Flexray. The FrTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains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

<b>SWS Item</b>	<b>ECUC_FrTSyn_00005 :</b>		
<b>Name</b>	FrTSynGlobalTimeDomainId		
<b>Parent Container</b>	FrTSynGlobalTimeDomain		
<b>Description</b>	The global time domain ID.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 31		
<b>Default value</b>	--		

<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_FrTSyn_00018 :</b>		
<b>Name</b>	FrTSynSynchronizedTimeBaseRef		
<b>Parent Container</b>	FrTSynGlobalTimeDomain		
<b>Description</b>	Mandatory reference to the required synchronized time-base.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to [ StbMSynchronizedTimeBase ]		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

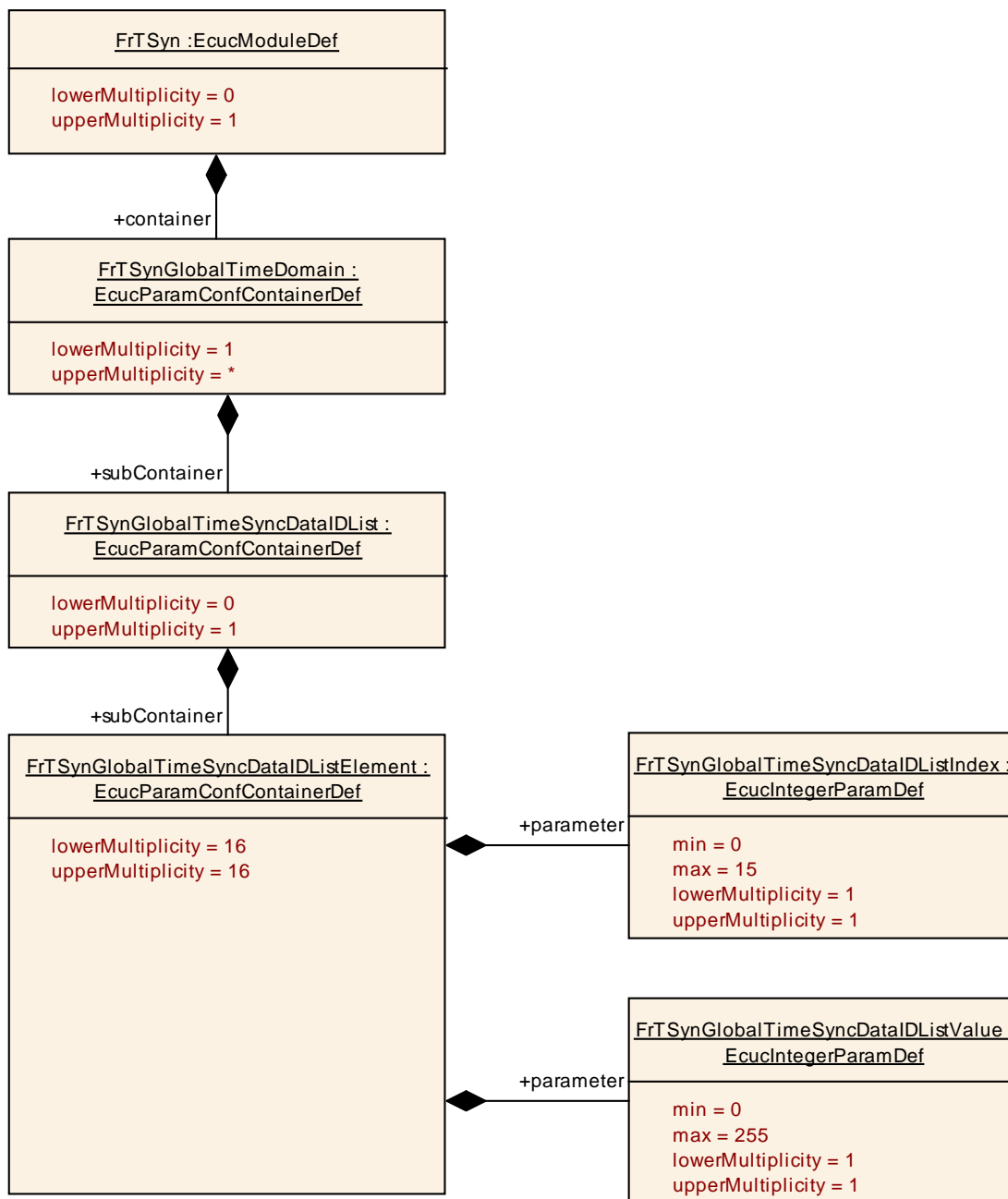
<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
FrTSynGlobalTimeMaster	0..1	Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.
FrTSynGlobalTimeOfsDataIDList	0..1	The DataIDList for OFS messages ensures the identification of data elements due to CRC calculation 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 process.



### 10.2.5 FrTSynGlobalTimeSyncDataIDList

<b>SWS Item</b>	<b>ECUC_FrTSyn_00023 :</b>		
<b>Container Name</b>	FrTSynGlobalTimeSyncDataIDList		
<b>Description</b>	The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation process.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
FrTSynGlobalTimeSyncDataIDListElement	16	Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation process.



### 10.2.6 FrTSynGlobalTimeSyncDataIDListElement

<b>SWS Item</b>	<b>ECUC_FrTSyn_00025 :</b>
<b>Container Name</b>	FrTSynGlobalTimeSyncDataIDListElement
<b>Description</b>	Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation process.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_FrTSyn_00026 :</b>
<b>Name</b>	FrTSynGlobalTimeSyncDataIDListIndex

<b>Parent Container</b>	FrTSynGlobalTimeSyncDataIDLListElement		
<b>Description</b>	Index of the DataIDLList for SYNC messages ensures the identification of data elements due to CRC calculation process.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 15		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

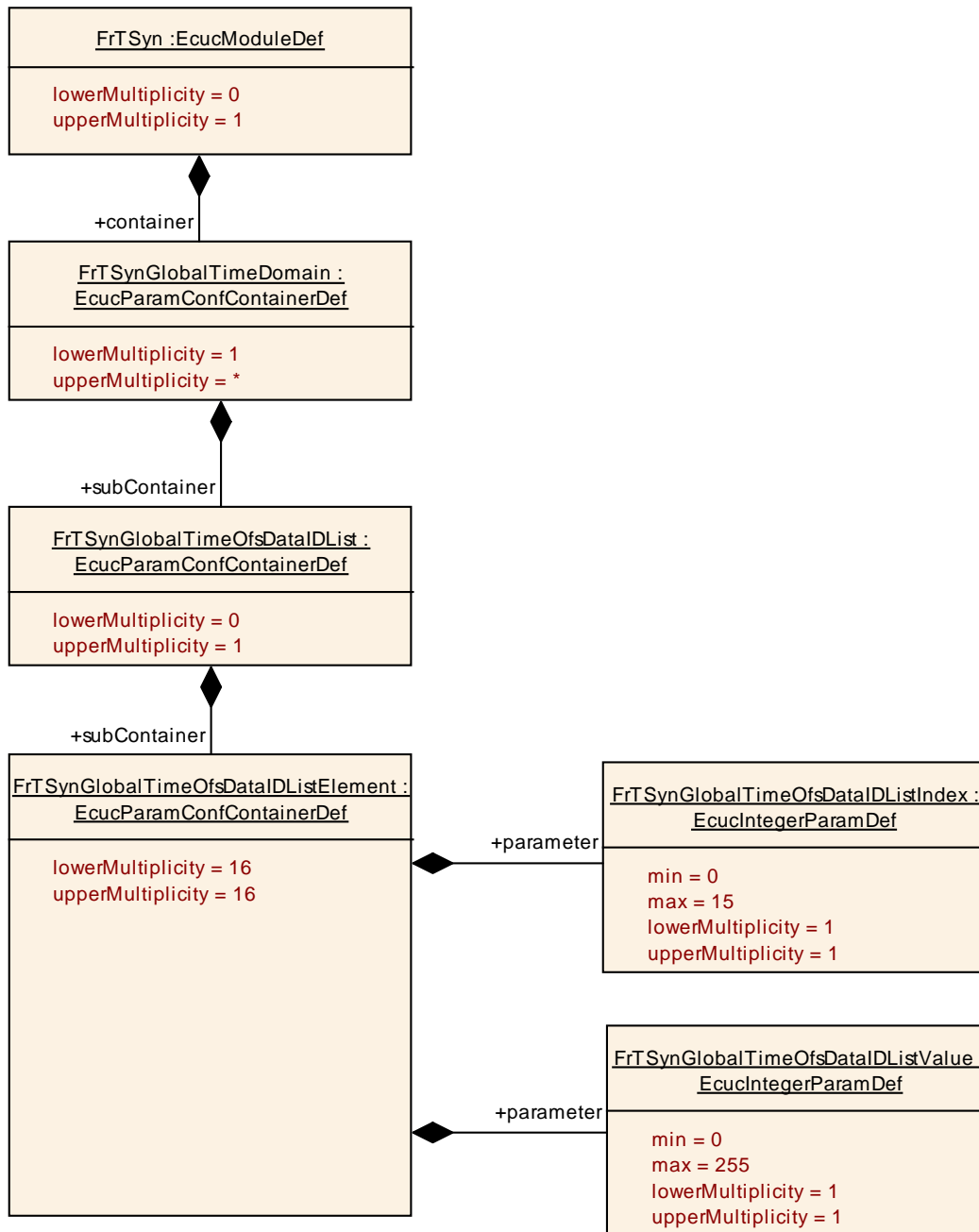
<b>SWS Item</b>	<b>ECUC_FrTSyn_00027 :</b>		
<b>Name</b>	FrTSynGlobalTimeSyncDataIDLListValue		
<b>Parent Container</b>	FrTSynGlobalTimeSyncDataIDLListElement		
<b>Description</b>	Value of the DataIDLList for SYNC messages ensures the identification of data elements due to CRC calculation process.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.2.7 FrTSynGlobalTimeOfsDataIDLList

<b>SWS Item</b>	<b>ECUC_FrTSyn_00024 :</b>		
<b>Container Name</b>	FrTSynGlobalTimeOfsDataIDLList		
<b>Description</b>	The DataIDLList for OFS messages ensures the identification of data elements due to CRC calculation process.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

<b>Included Containers</b>			
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>	
FrTSynGlobalTimeOfsDataIDLListElement	16	Element of the DataIDLList for OFS messages ensures the identification of data elements due to CRC calculation process.	



### 10.2.8 FrTSynGlobalTimeOfsDataIDListElement

<b>SWS Item</b>	<b>ECUC_FrTSyn_00028 :</b>
<b>Container Name</b>	FrTSynGlobalTimeOfsDataIDListElement
<b>Description</b>	Element of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation process.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_FrTSyn_00029 :</b>
<b>Name</b>	FrTSynGlobalTimeOfsDataIDListIndex
<b>Parent Container</b>	FrTSynGlobalTimeOfsDataIDListElement
<b>Description</b>	Index of the DataIDList for OFS messages ensures the identification of

	data elements due to CRC calculation process.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 15		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

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

**No Included Containers**

### 10.2.9 FrTSynGlobalTimeMaster

<b>SWS Item</b>	<b>ECUC_FrTSyn_00006 :</b>		
<b>Container Name</b>	FrTSynGlobalTimeMaster		
<b>Description</b>	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.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_FrTSyn_00032 :</b>		
<b>Name</b>	FrTSynCyclicMsgResumeTime		
<b>Parent Container</b>	FrTSynGlobalTimeMaster		
<b>Description</b>	Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. INF]		
<b>Default value</b>	--		



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

<b>SWS Item</b>	<b>ECUC_FrTSyn_00033 :</b>		
<b>Name</b>	FrTSynGlobalTimeDebounceTime		
<b>Parent Container</b>	FrTSynGlobalTimeMaster		
<b>Description</b>	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		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. INF[		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

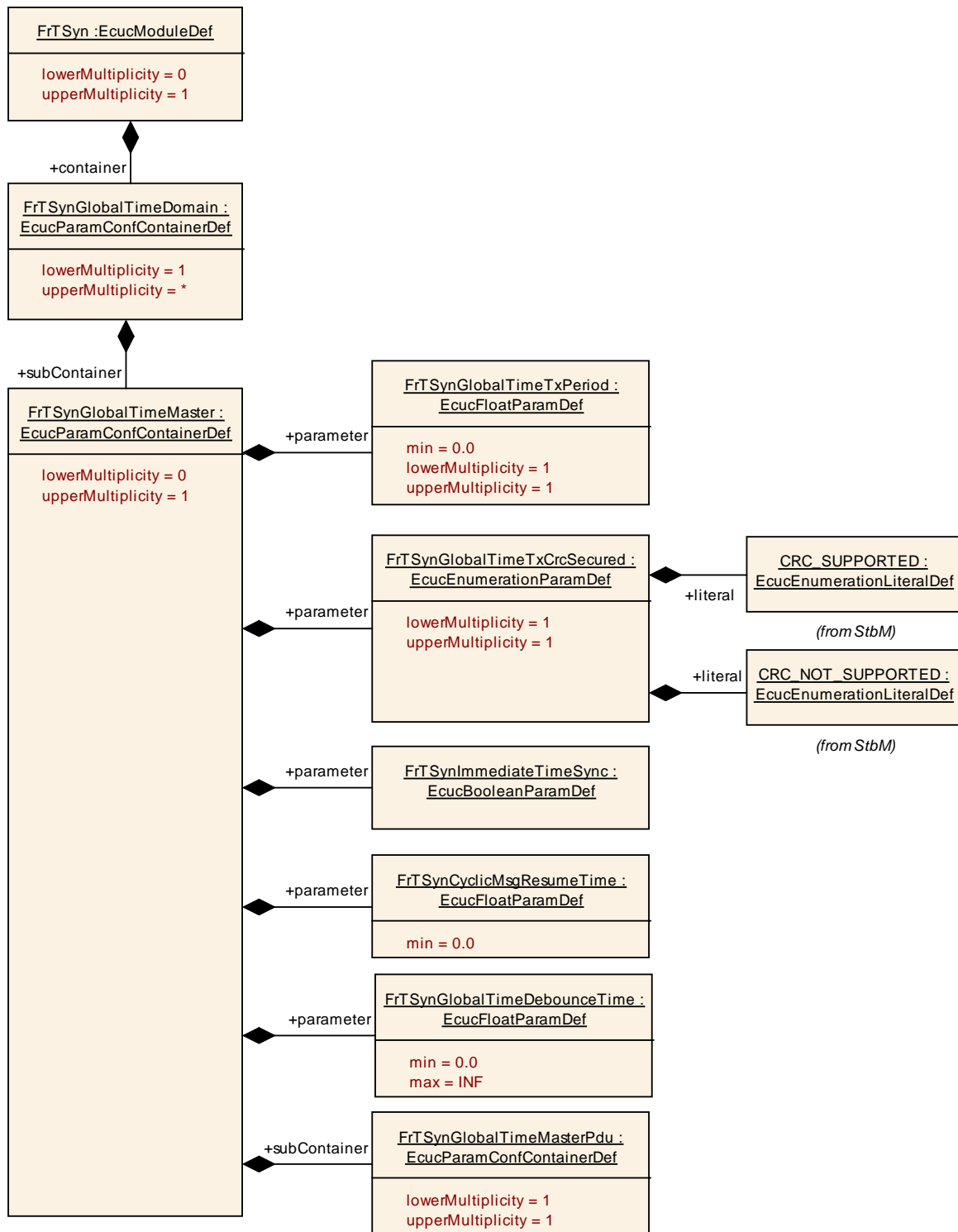
<b>SWS Item</b>	<b>ECUC_FrTSyn_00013 :</b>		
<b>Name</b>	FrTSynGlobalTimeTxCrcSecured		
<b>Parent Container</b>	FrTSynGlobalTimeMaster		
<b>Description</b>	This represents the configuration of whether or not CRC is supported.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>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

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

<b>SWS Item</b>	<b>ECUC_FrTSyn_00031 :</b>		
-----------------	----------------------------	--	--

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

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
FrTSynGlobalTimeMasterPdu	1	This container carries all properties required to configure the PDU sent by the global time master for the given global time domain.



### 10.2.10 FrTSynGlobalTimeMasterPdu

<b>SWS Item</b>	<b>ECUC_FrTSyn_00008 :</b>
<b>Container Name</b>	FrTSynGlobalTimeMasterPdu
<b>Description</b>	This container carries all properties required to configure the PDU sent by the global time master for the given global time domain.
<b>Configuration Parameters</b>	

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

<b>SWS Item</b>	<b>ECUC_FrTSyn_00020 :</b>		
<b>Name</b>	FrTSynGlobalTimePduRef		
<b>Parent Container</b>	FrTSynGlobalTimeMasterPdu		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ Pdu ]		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.2.11 FrTSynGlobalTimeSlave

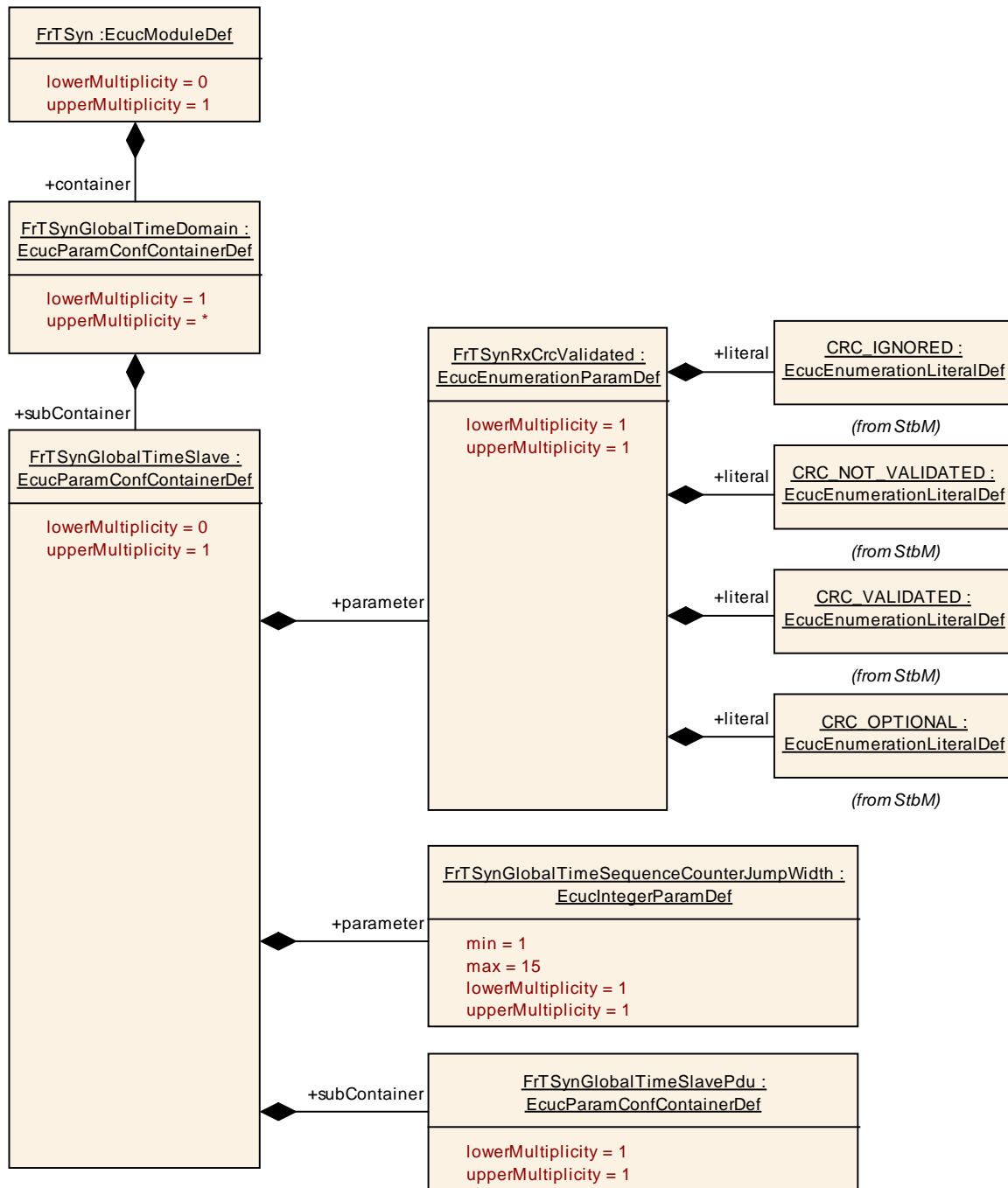
<b>SWS Item</b>	<b>ECUC_FrTSyn_00010 :</b>		
<b>Container Name</b>	FrTSynGlobalTimeSlave		
<b>Description</b>	This represents the time slave for the enclosing global time domain.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_FrTSyn_00022 :</b>		
<b>Name</b>	FrTSynGlobalTimeSequenceCounterJumpWidth		
<b>Parent Container</b>	FrTSynGlobalTimeSlave		
<b>Description</b>	The SequenceCounterJumpWidth specifies the maximum allowed gap of the Sequence Counter between two SYNC resp. two OFS messages.		
<b>Multiplicity</b>	1		

<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 15		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_FrTSyn_00017 :</b>		
<b>Name</b>	FrTSynRxCrcValidated		
<b>Parent Container</b>	FrTSynGlobalTimeSlave		
<b>Description</b>	This parameter controls whether or not CRC validation shall be supported.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	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.	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
FrTSynGlobalTimeSlavePdu	1	This container carries all properties required to configure the PDU received by the time slave for the given global time domain.



### 10.2.12 FrTSynGlobalTimeSlavePdu

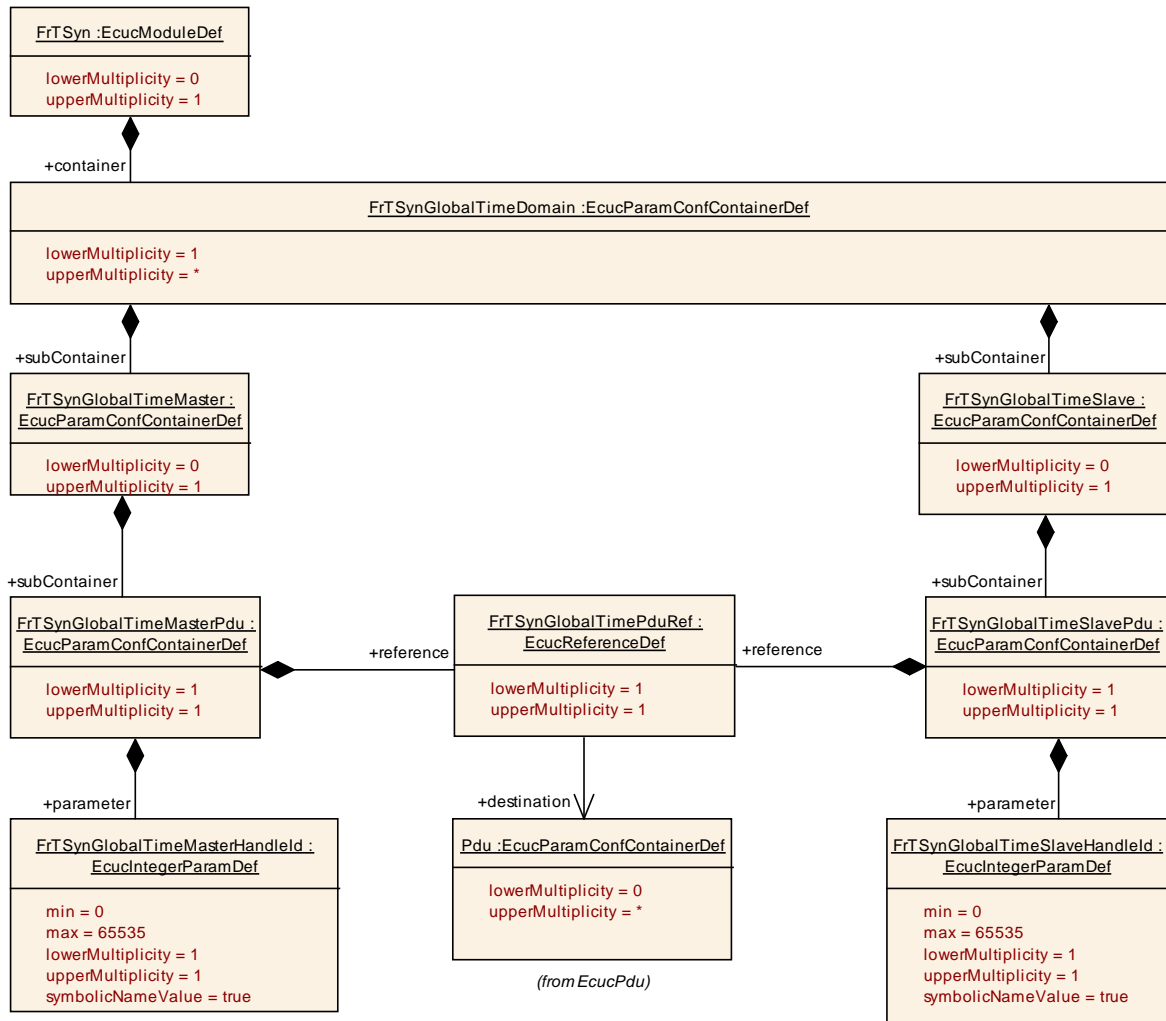
<b>SWS Item</b>	<b>ECUC_FrTSyn_00012 :</b>
<b>Container Name</b>	FrTSynGlobalTimeSlavePdu
<b>Description</b>	This container carries all properties required to configure the PDU received by the time slave for the given global time domain.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_FrTSyn_00011 :</b>
<b>Name</b>	FrTSynGlobalTimeSlaveHandleId

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

<b>SWS Item</b>	<b>ECUC_FrTSyn_00021 :</b>		
<b>Name</b>	FrTSynGlobalTimePduRef		
<b>Parent Container</b>	FrTSynGlobalTimeSlavePdu		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ Pdu ]		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**



### 10.3 Published Information

For details, refer to the chapter 10.3 “Published Information” in *SWS\_BSWGeneral*.