EB tresos® AutoCore Generic 8 FlexRay Stack documentation

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Table of Contents

EB tresos® AutoCore Generic 8 FlexRay Stack documentation
Chapter 1. Overview of EB tresos AutoCore Generic 8 FlexRay Stack
documentation

1. Overview of EB tresos AutoCore Generic 8 FlexRay Stack documentation

Welcome to the EB tresos AutoCore Generic 8 FlexRay Stack (ACG8 FlexRay Stack) product documentation.

This document provides:

► Chapter 2, "Supported features": list of features supported by the ACG8 FlexRay Stack

► Chapter 3, "ACG8 FlexRay Stack release notes": release notes for the ACG8 FlexRay Stack modules

► Chapter 4, "ACG8 FlexRay Stack user's guide": containing background information and instructions

► Chapter 5, "ACG8 FlexRay Stack module references": information about configuration parameters and the application programming interface

2. Supported features

2.1. Supported FrNm features

▶ **Support AUTOSAR network management coordination algorithm**: Support of transmission of periodic NM Vote messages as long as the bus communication is requested and support of detection of NM Vote messages signaling that other nodes request bus-communication.

▶ **Support operational modes**: Support of operational modes Network Mode (with internal states Repeat Message State, Normal Operation State, Ready Sleep State), Synchronize Mode, and Bus-Sleep Mode according to AUTOSAR specifications.

▶ **Support configurable structure of NM messages**: Support for configurable structure of NM messages according to AUTOSAR specification.

▶ **Support for communication startup**: Support for interface to the upper layer to initiate transmission of NM Vote messages due to any user(s) requesting communication.

▶ **Support for communication shutdown**: Support for interface to the upper layer to stop transmission of NM Vote messages due to any user(s) not requesting communication.

▶ **Support for communication passive wakeup**: Support interface to the upper layer to initiate communication capabilities due to a wakeup event network start or network restart indication.

▶ **Support for passive mode**: Support for nodes with transmission of NM Vote messages disabled.

▶ **Support for detection of remote sleep**: Support for detecting if all other nodes are ready to sleep.

▶ **Support for state change notification**: Support for notification function for `Nm` when `FrNm` state changes.

▶ **Support for car wakeup**: Support of car wakeup bit as part of the network management PDU and car wakeup callout function.

▶ **Support for bus-load reduction mechanism**: Support mechanism to reduce the number of transmitted NM messages for realizing network management algorithm.

▶ **Support for user data in NM messages**: Support for updating user data in NM messages either via `FrNM` interfaces or the communication stack by collecting the data from an I-PDU.

▶ **Support for communication control**: Support for interfaces to enable/disable transmission of NM messages.

▶ **Support for partial networking**: Support for updating and filtering partial network information as part of the NM messages.

▶ **Support of RepeateMsgInd|NodeDetection|NodeIdEnabled channel-based configurable**: Support per channel configuration of parameters `FrNmNodeDetectionEnabled`, `FrNmSourceNodeIdentifierEnabled`.

► **Support for post-build**: Support for handling post-build loadable and selectable configuration.

2.2. Supported features of FrTp, FrIf, FrSm, FrArTp

In addition to the functionality specified in AUTOSAR 4.0.3 (FrArTp: AUTOSAR 4.2.2), the following features are supported:

► **Configurable limit of parallel active connections in FrTp**: The number of active transmissions can be limited on a remote address basis and additional transmission requests can be buffered.

► **Pre-compile parameters to support code optimization**: Various pre-compile parameters are available to remove unnecessary code and reduce ROM consumption and execution time.

► **Device health status**: Return the controller's health status (several error status bits) in an abstract way.

► **Extended IRQ (interrupt request) control**: Additional APIs are provided to control further interrupts, e. g. cycle start or start of dynamic segment (if supported by hardware).

► **Relative timer control**: Additional APIs are provided to control the relative timer.

► **MTS (Media Access Test Symbol) control**: Additional APIs are provided to control transmit MTS and check whether an MTS was received.

► **Virtual PDU support**: A PDU-specific configuration parameter is provided that enables support for virtual PDUs. Virtual PDUs may overlap original PDUs and are indicated in addition to the original PDUs.

► **Support for post-build**:

  ► Support for handling post-build loadable and selectable configuration in FrIf.

  ► With FrIf post-build selectable: Selection of different FlexRay schedules at initialization time.

  ► Support for post-build loadable configuration in FrTp and FrArTp.

3. ACG8 FlexRay Stack release notes

3.1. Overview

This chapter provides the ACG8 FlexRay Stack product specific release notes. General release notes that are applicable to all products are provided in the EB tresos AutoCore Generic documentation. Refer to the general release notes in addition to the product release notes documented here.

3.2. Scope of the release

3.2.1. Configuration tool

Your release of EB tresos AutoCore is compatible with the release of the EB tresos Studio configuration tool:

▶ EB tresos Studio: 28.2.0 b211016-0103

3.2.2. AUTOSAR modules

The following table lists the AUTOSAR modules that are part of this ACG8 FlexRay Stack release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
FrArTp	4.2.2 []	4.2.2 [0000]	1.0.9	Elektrobit Automotive GmbH
FrIf	4.0.3 []	3.3.0 [0000]	5.3.26	Elektrobit Automotive GmbH
FrNm	4.0.3 []	4.2.0 [0003]	5.16.8	Elektrobit Automotive GmbH
FrSM	4.0.3 []	2.2.0 [0000]	5.3.19	Elektrobit Automotive GmbH
FrTp	4.0.3 []	4.0.0 [0000]	4.4.27	Elektrobit Automotive GmbH

Table 3.1. Hardware-Independent Modules specified by the AUTOSAR standard

3.2.3. EB (Elektrobit) modules

The following table lists all modules which are part of this release but are not specified by the AUTOSAR standard. These modules include tooling developed by EB or they may hold files shared by all other modules.

Module name	Module version	Supplier
FrAs	1.0.34	Elektrobit Automotive GmbH

Table 3.2. Modules not specified by the AUTOSAR standard

3.2.4. MCAL modules and EB tresos AutoCore OS

For information about MCAL modules and OS, refer to the respective documentation, which is available as PDF at `$TRESOS_BASE/doc/3.0_EB_tresos_AutoCore_OS` and `$TRESOS_BASE/doc/5.0_MCAL_-modules`[1]. It is also available in the online help in EB tresos Studio. Browse to the folders `EB tresos AutoCore OS` and `MCAL modules`.

3.3. Module release notes

3.3.1. FrArTp module release notes

► AUTOSAR R4.2 Rev 2

► AUTOSAR SWS document version: 4.2.2

► Module version: 1.0.9.B466224

► Supplier: Elektrobit Automotive GmbH

3.3.1.1. Change log

This chapter lists the changes between different versions.

Module version 1.0.9
2021-10-08

► Internal module improvement. This module version update does not affect module functionality.

---

[1] `$TRESOS_BASE` is the location at which you installed EB tresos Studio.

Module version 1.0.8

2021-06-25

▶ ASCFRARTP-368: Added support for TxConfirmation() to request another transmission of the same PDU in TP modules. This functionality is required to use the PduR TP-gateway-queuing feature on FlexRay buses.

Module version 1.0.7

2021-03-05

▶ ASCFRARTP-359 Fixed known issue: FrArTp does not schedule N-PDUs after acknowledgement timeout.

▶ ASCFRARTP-354 Fixed known issue: Possible Out-of-bounds read access during reception of First-Frame and First-Frame-Extended.

▶ Internal module improvement. This module version update does not affect module functionality.

Module version 1.0.6

2020-10-23

▶ ASCFRARTP-194 Add configuration check to limit PduLength linked to a FrArTpPdu.

▶ ASCFRARTP-302: Add runtime checks to consider PduLengthType UINT16 and UINT32.

▶ . ASCFRARTP-268 Add support for 1:n communication.

▶ ASCFRARTP-224 Add a configuration check that address pairs of connections are unique within pool.

▶ ASCFRARTP-305 Fixed known issue: FrArTp_RxIndication signature is missing const qualifier for parameter.

▶ ASCFRARTP-325 Fixed known issue: Trace headers are not complete.

▶ ASCFRARTP-322 Add protection of runtime data.

▶ ASCFRARTP-312 Fixed known issue: PbcfgM Support.

Module version 1.0.5

2020-06-19

▶ Internal module improvement. This module version update does not affect module functionality.

Module version 1.0.4

2020-02-21

▶ ASCFRARTP-136 Fixed known issue: SF-I decoding length checks.

▶ ASCFRARTP-188 Fixed known issue: FrArTp_ChangeParameter changing parameter when called with wrong parameter.

▶ ASCFRARTP-183 Fixed known issue: Reception of SF-I resets currently ongoing segmented transmission.

▶ ASCFRARTP-186 Fixed known issue: XDM Check for FrArTpPduIds missing.

▶ ASCFRARTP-184 Fixed known issue: Handle Id Wizard for FrArTpPduId causes XDM error.

▶ ASCFRARTP-162 Fixed known issue: Wrong pointer an memory classes used.

▶ ASCFRARTP-158 Fixed known issue: FF-I decoding length checks.

▶ ASCFRARTP-161 Fixed known issue: Number of used frames in a block overflows if maxBs is set to 255, causing infinite block.

▶ ASCFRARTP-141 Fixed known issue: Block Size upper limit check is off by 1 from channel defined maxBs.

▶ ASCFRARTP-118 Support for BlockSize=0 functionality.

Module version 1.0.3

2018-06-22

▶ ASCFRARTP-87 Fixed known issue: actConnIdx is given as a parameter to the function FrArTp_SduMng_copyRxData() instead of connIdx.

▶ ASCFRARTP-90 Fixed known issue: Wrong checking of the Buffer before transition into the CTS.WaitPdu state.

▶ ASCFRARTP-86 Fixed known issue: Wrong checking of the size of the received SF-E frame in FrArTp.

Module version 1.0.2

2018-01-04

▶ Correct reference in adjacent layer properties file for PduR.

Module version 1.0.1

2017-03-31

▶ Initial development production version.

Module version 1.0.0

2016-11-04

▶ Initial prototype production version (limited feature set).

3.3.1.2. New features

► No new features have been added since the last release.

3.3.1.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

► This module provides no EB-specific enhancements.

3.3.1.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

► Retry functionality is not implemented

Description:

The current implementation of Flexray AUTOSAR Transport Layer does not support the retry mechanism. Thus in case of communication errors, the connection will be terminated immediately instead of recovering by retransmissions.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00009, SWS_FrArTp_00060, SWS_FrArTp_00063, SWS_FrArTp_00082, SWS_FrArTp_-00106, SWS_FrArTp_00111, SWS_FrArTp_00140, SWS_FrArTp_00232, SWS_FrArTp_00233, SWS_-FrArTp_00234, SWS_FrArTp_00239, SWS_FrArTp_00244, SWS_FrArTp_00283, SWS_FrArTp_00284, SWS_FrArTp_00300

► Acknowledge functionality is not implemented

Description:

The current implementation of Flexray AUTOSAR Transport Layer does not support the acknowledge mechanism.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00009, SWS_FrArTp_00028, SWS_FrArTp_00039, SWS_FrArTp_00057, SWS_FrArTp_-
00063, SWS_FrArTp_00068, SWS_FrArTp_00072, SWS_FrArTp_00073, SWS_FrArTp_00074, SWS_-
FrArTp_00075, SWS_FrArTp_00076, SWS_FrArTp_00077, SWS_FrArTp_00078, SWS_FrArTp_00082,
SWS_FrArTp_00087, SWS_FrArTp_00117, SWS_FrArTp_00120, SWS_FrArTp_00227, SWS_FrArTp_-
00241, SWS_FrArTp_00244, SWS_FrArTp_00246, SWS_FrArTp_00247, SWS_FrArTp_00248, SWS_-
FrArTp_00249, SWS_FrArTp_00250, SWS_FrArTp_00251, SWS_FrArTp_00252, SWS_FrArTp_00253,
SWS_FrArTp_00266, SWS_FrArTp_00267, SWS_FrArTp_00268, SWS_FrArTp_00269, SWS_FrArTp_-
00270, SWS_FrArTp_00271, SWS_FrArTp_00272, SWS_FrArTp_00280, SWS_FrArTp_00281, SWS_-
FrArTp_00285, SWS_FrArTp_00286, SWS_FrArTp_00287

► Segmented 1:n communication is not implemented

Description:

The current implementation of Flexray AUTOSAR Transport Layer does not support segmented 1:n com-
munication.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00009, SWS_FrArTp_00036, SWS_FrArTp_00059, SWS_FrArTp_00086, SWS_FrArTp_-
00091, SWS_FrArTp_00104, SWS_FrArTp_00283

► Retry of transmissions in case of local timeout is not performed

Description:

Transmission in case of local TimeoutAR or TimeoutAS is not repeated. Thus configuration parameters
FrArTpMaxAr and FrArTpMaxAs are not supported. If there is a TimeoutAR or TimeoutAS, then immedi-
ately E_NOT_OK is reported to the upper layer, aborting the connection.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00263, SWS_FrArTp_00282

► Timeout after reception of one block containing CF with invalid sequence number is not supported

Description:

FrArTp does not support timeout after reception of one block containing CF with invalid sequence number.
Instead, it immediately aborts the reception by calling PduR_FrArTpRxIndication() with E_NOT_OK.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00265

► Initialization check in `FrArTp_MainFunction()`

Description:

If the `FrArTp_MainFunction()` is called while the module is not yet initialized, the `FrArTp_Main-Function()` returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independently of the development error detection setting.

Rationale:

The SchM module may schedule the function `FrArTp_MainFunction()` before the module is initialized. This would result in lots of Det errors during startup. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

Requirements:

SWS_FrArTp_00292

► There is no consistency check between code files and header files

Description:

According to the FrArTp SWS, the FrArTp module shall perform inter-module version checks. This implementation does not perform inter-module version checks.

Rationale:

The module consistency check is not within the responsibility of the basic software but part of the configuration management and delivery process.

Requirements:

SWS_FrArTp_00201

► Prioritized transmit Pdus are not supported

Description:

The prioritization of transmit Pdus per connection is currently not supported. Thus each connection has the same priority.

Requirements:

SWS_FrArTp_00258, SWS_FrArTp_00278

► Immediate transmission not supported

Description:

FrArTp only supports decoupled transmission mode of FrIf. Thus, all transmit Pdus in FrIf used by FrArTp must be configured for decoupled transmission.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00187

► 6 Byte ISO transmission not supported

Description:

FrArTp does not support FRARTP_ISO6 for config parameter FrArTpLm.

Rationale:

This feature is not required by current customers.

Requirements:

SWS_FrArTp_00025, SWS_FrArTp_00028, SWS_FrArTp_00034, SWS_FrArTp_00091

► No padding of N-PDUs in FrArTp

Description:

The remaining space (bits) in the N-PDU are not set to 0.

Rationale:

The FrIf/Fr will add padding bytes to the frame with a configured value (FrIfUnusedBitValue). See FrIf.-ASR40.FrIf05723.

Requirements:

SWS_FrArTp_00255

► Some DET errors not supported

Description:

Module FrArTp does not support the following development errors:

▶ `FRARTP_E_INIT_FAILED`

Requirements:

SWS_FrArTp_00179

▶ `FrArTp_Shutdown()` has not been implemented.

Description:

The API function `FrArTp_Shutdown()` has not been implemented.

Rationale:

There is no AUTOSAR internal user for the API function `FrArTp_Shutdown()` and the behavior and operating constraints are not clearly specified in the AUTOSAR SWS. Using the function might be risky since expectations and actual behavior might differ, so it was decided to skip the function implementation.

Requirements:

SWS_FrArTp_00148

▶ Timing constraints not considered

Description:

The following timing constraints are not considered.

$VE + FrArTpTimeBr + (FrArTpTimeoutAr * FrArTpMaxAr) + VS < FrArTpTimeoutBs$

$Vs + FrArTpTimeCs + (FrArTpTimeoutAs * FrArTpMaxAs) + VE < FrArTpTimeoutCr$

Rationale:

It is up to the user, the lower layer modules and the underlying hardware platform to fulfill these constraints.

Requirements:

SWS_FrArTp_00242, SWS_FrArTp_00243, AUDI_ASR_FrTP_034, AUDI_ASR_FrTP_035

▶ SymbolicNameValues are not implemented.

Description:

SymbolicNameValues are not implemented.

Rationale:

This feature is not required by current customers.

Requirements:

N/A

► StartOfReception return value BUFREQ_E_NOT_OK not supported.

Description:

StartOfReception return value BUFREQ_E_NOT_OK not supported.

Rationale:

PduR does not support BUFREQ_E_NOT_OK as a return value for PduR_FrArTpStartOfReception.

Requirements:

AUDI_ASR_FrTP_014

► PduInfoPtr parameter of FrArTp_RxIndication() is not defined as pointer to const.

Description:

The parameter `PduInfoPtr` of the function `FrArTp_RxIndication` is defined as a pointer to a variable (`PduInfoType* PduInfoPtr`), not a pointer to a constant (`const PduInfoType* PduInfoPtr`).

Rationale:

To maintain compatibility with FlexRay Interface (FrIf) based on AUTOSAR 4.0, which expects a pointer to a variable parameter.

Requirements:

SWS_FrArTp_00152

► FrArTpMaxWft configuration parameter is not optional.

Description:

The configuration parameter `FrArTpMaxWft` in the `FrArTpChannel` container is not optional. It follows that the current implementation does not support `FrArTpMaxWft` equal to infinity by leaving it unset in the configuration.

Rationale:

This feature is not required by current customers.

Requirements:

ECUC_FrArTp_00059

3.3.1.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

▶ Limitation on number of `FrArTpChannel` containers

Description:

The number of configured `FrArTpChannel` containers is limited to 254.

▶ Limitation on number of `FrArTpConnection` containers

Description:

The total number of configured `FrArTpConnection` containers is limited to 254.

▶ Limitation on number of concurrent connections

Description:

The total number of configured concurrent connections, which is the sum of all `FrArTpConcurrentConnections` of all `FrArTpChannel` containers, is limited to 255.

▶ Limitation on number of RxPdus

Description:

The total number of configured of configured RxPdus (`FrArTpPdu` with `FrArTpPduDirection` configured to FRARTP_RX) is limited to 255.

▶ Limitation on number of TxPdus

Description:

The total number of configured of configured TxPdus (`FrArTpPdu` with `FrArTpPduDirection` configured to FRARTP_TX) is limited to 255.

▶ Limitation on Low Level Routing transmission functionality

Description:

`<LLR>_FrIf_FrTp_TriggerTransmit` is allowed to return `E_NOT_OK` (`FrArTp_TriggerTransmit` is bypassed) only if `<LLR>_FrTp_FrIf_Transmit` returns `E_NOT_OK` as well (`FrIf_Transmit` is not processed).

3.3.1.6. Open-source software

FrArTp does not use open-source software.

3.3.2. FrAs module release notes

► Module version: 1.0.34.B466224

► Supplier: Elektrobit Automotive GmbH

3.3.2.1. Change log

This chapter lists the changes between different versions.

Module version 1.0.34
2021-03-05

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.33
2021-02-12

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.32
2021-01-22

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.31
2020-12-18

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.30
2020-09-25

► ASCFRAS-156 Fixed known issue: Job Trigger Number is not saved in FrIf JobList Assignment AutoCon-
figure Wizard

Module version 1.0.29

2020-06-19

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.28

2020-05-22

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.27

2020-04-27

► Provided Redzone mechanism that enables the user to reduce latencies between job execution and frame transmission

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.26

2020-02-21

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.25

2019-06-14

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.24

2019-04-18

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.23

2019-02-15

▶ Internal module improvement. This module version update does not affect module functionality

Module version 1.0.22

2018-09-28

▶ Internal module improvement. This module version update does not affect module functionality

Module version 1.0.21

2018-08-24

▶ ASCFRAS-116 Fixed known issue: FrAs orders PREPARE_LPDU and RECEIVE_AND_INDICATE incorrectly for 3rd party Fr modules

Module version 1.0.20

2018-07-27

▶ Added the command line access to the FrAs JobList creation function via GuidedConfigWizard

Module version 1.0.19

2018-05-25

▶ Internal module improvement. This module version update does not affect module functionality

Module version 1.0.18

2018-02-16

▶ Internal module improvement. This module version update does not affect module functionality

Module version 1.0.17

2017-12-15

Module version 1.0.16

2016-07-01

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.15
2016-05-25

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.14
2016-04-29

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.13
2016-04-01

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.12
2015-06-19

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.11
2015-02-20

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.10
2014-10-03

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.9
2014-04-25

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.8

2013-10-11

► Added support for reading in and writing out `FrIfReconfigurable`

► Added support for dedicated Rx/Tx jobs in auto assignment

► Added support for reassigning communication operations to jobs

Module version 1.0.7

2013-06-14

► Added support for AUTOSAR 4.0 configurations

Module version 1.0.6

2013-05-08

► Removed support for AUTOSAR 3.x configurations which became obsolete

Module version 1.0.5

2013-02-08

► Removed support for AUTOSAR 2.1 configurations which became obsolete

Module version 1.0.4

2012-10-12

► FrAs Release for ACG 6.3

Module version 1.0.3

2012-06-15

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.2

2012-03-16

► Internal module improvement. This module version update does not affect module functionality

Module version 1.0.1

2012-01-20

► FrAs Release for ACG 6.2

Module version 1.0.0

2011-09-30

► Initial AUTOSAR 4.0 version

3.3.2.2. New features

► No new features have been added since the last release.

3.3.2.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

3.3.2.4. Deviations

This module is not part of the AUTOSAR specification.

3.3.2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

► For this module no limitations are known.

3.3.2.6. Open-source software

FrAs does not use open-source software.

3.3.3. FrIf module release notes

▶ AUTOSAR R4.0 Rev 3

▶ AUTOSAR SWS document version: 3.3.0

▶ Module version: 5.3.26.B466224

▶ Supplier: Elektrobit Automotive GmbH

3.3.3.1. Change log

This chapter lists the changes between different versions.

Module version 5.3.26
2021-10-08

▶ ASCFRIF-1007 Fixed known issue: PDU transmission request can get lost if FrIf_Transmit() preempts FrIf_JoblistExec()

Module version 5.3.25
2021-06-25

▶ Implemented support of Mirror module. Note: This feature requires EB Fr Module ACM-8.8.3, or Fr Module supporting Autosar 4.4.0 or higher.

Module version 5.3.24
2021-03-05

▶ ASCFRIF-980 Fixed known issue: Out-Of-Bound access can occur when Fr_TransmitTxLPdu is called for decoupled transmission

▶ Implemented AR-87698 to extend shortening of L-Sdu if FrIfAllowDynamicLSduLength is set to TRUE and L-Sdu is transmitted in the dynamic segment

Module version 5.3.23

2020-10-23

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.22

2020-09-25

► Added support for ASR43 conform UL_TxConfirmation

Module version 5.3.21

2020-06-19

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.20

2020-02-21

► Added buffer reconfiguration support for quaranteed dynamic Tx segment Note: This feature requires EB Fr module version 5.2.11

Module version 5.3.19

2019-10-11

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.18

2019-07-23

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.17

2019-06-14

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.16

2019-02-15

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.15

2018-10-26

► Added Linux support of FrIf MCG

Module version 5.3.14

2018-06-22

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.13

2017-09-22

► Added memory mapping section SEC_VAR_INIT_8 for module initialization status. Consider updating the MemMap configuration.

► Changed code and comments to Comply to MISRA-C:2012.

Module version 5.3.12

2017-03-31

► Changed config parameter FrIfReadCCConfigApi to editable.

Module version 5.3.11

2016-11-04

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.10

2016-10-07

► Adapted resource file for the scheduling of main functions to the split of `IpduM_MainFunction()` into `IpduM_MainFunctionRx()` and `IpduM_MainFunctionTx()`.

Module version 5.3.9
2016-05-25

► ASCFRIF-809 Fixed known issue: Nesting memory section may lead to compile error

Module version 5.3.8
2016-02-05

► Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`

Module version 5.3.7
2015-06-19

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.6
2015-02-20

► Added support for configurable debouncing of Dem events

► Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro FRIF_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined

► Added support for variant specifific configuration generation enabling selectable post-build configurations

Module version 5.3.5
2014-10-03

► ASCFRIF-774 Fixed known issue: FrIf MCG reports an error in combination with non EB Fr driver if `FrFifo` is configured

Module version 5.3.4
2014-04-25

► ASCFRIF-739 Fixed known issue: Configuration parameters `FrIfUserTxUL` and `FrIfUserRxIndica-tionUL` are not editable although other modules might depend on them

► ASCFRIF-746 Fixed known issue: FrIf configuration generation might abort with error message if `PRE-PARE_LPDU` communication action is used for RX_LPDUs

► ASCFRIF-742 Fixed known issue: Transmission of payload containing update bits might be truncated if dynamic payload length is enabled

► Added configuration checks to ensure that each FrIfPdu is assigned to a `FrIfFrameTriggering`

► ASCFRIF-758 Fixed known issue: No `FrIf_PBcfg.c` file is generated on certain computers

► ASCFRIF-760 Fixed known issue: Build error due to missing file `FrIf_PBcfg.c` if code generation for FrIf is disabled and only post-build configuration is compiled

► Increased error detection timeout value for FrIf config generator

► Changed default value of configuration parameter `FrIfConfirm` to true

Module version 5.3.3
2013-10-11

► Removed superfluous `NULL_PTR` Det-check from API function `FrIf_GetNmVector()`

► Updated symbolic name value naming schema according to AUTOSAR 4.0 Rev 3

► ASCFRIF-723 Fixed known issue: TxCounter is not decreased if `<UL>_TriggerTransmit()` returns a different value than `E_OK`

► Extended MCG to generate XML code for Binary Code Generation

Module version 5.3.2
2013-06-24

► Added missing AUTOSAR compiler abstraction macros

► Added checking of configuration and platform-specific signature to prevent loading of incompatible post-build configuration.

► Updated FrIf to use AutoCore generic `TS_MemSet()` function instead of FrIf-specific implementation

► ASCFRIF-639 Fixed known issue: `TxConfirmation` might be called for IPdus that are not transmitted if `FrIfNoneMode` is enabled

► ASCFRIF-681 Fixed known issue: An illegal memory access may occur if multiple transceiver driver support is enabled but no transceiver configured

► ASCFRIF-690 Fixed known issue: `FrIf_Transmit()` does not return `E_NOT_OK` if FrIf is in state `FRIF_-STATE_OFFLINE`

► ASCFRIF-698 Fixed known issue: FrIf may generate an invalid configuration if relocatable config is enabled

► ASCFRIF-707 Fixed known issue: FrIf generator aborts if FrNm TX user data PDU is configured

Module version 5.3.1

2013-02-07

► ASCFRIF-630 Fixed known issue: If `<UL>_TriggerTransmit()` returns `E_NOT_OK` for a PDU, the up-date-bits for all PDUs in this frame might be randomized

► Added relocatability to post build configuration

► Added copyright information to generated file `FrIf_PBcfg_Initializer.c.txt`

Module version 5.3.0

2012-10-12

► ASCFRIF-588 Fixed known issue: `Post-Build` initializer file is not compatible with the PbcfgM

► ASCFRIF-594 Fixed known issue: The FrIf generator may incorrectly report an error that the interval between two jobs is less than FrIfMaxISRDelay

► Implemented ASR 4.0 HandleId policy

► ASCFRIF-615 Fixed known issue: Configuration generation aborts if an upper layer uses HandleId value `255` for Rx-IPdus

► `FrIf_PBCfg.c` shall be empty if PBcfgM is used

► Improved description of user defined upper module configuration

Module version 5.2.0

2012-06-20

► Updated configuration for basic post-build support

► ASCFRIF-462 Fixed known issue: If the parameter `ComOpCycleFilterEnable` is set to true, then generated post build configuration output is not deterministic

Module version 5.1.1

2012-04-20

► ASCFRIF-562 Fixed known issue: Wrong PDU length might be indicated to the upper layer by `<UL>_-RxIndication()` if a single PDU is configured for `FrIfFrameTriggering`

▶ ASCFRIF-564 Fixed known issue: The disabling and reconfiguration of L-PDUs does not work

Module version 5.1.0

2012-03-16

▶ ASCFRIF-512 Fixed known issue: The transmission of simple FlexRay frames cannot be canceled with the return value of callout `<UL>_TriggerTransmit()`

▶ Updated naming scheme for #defines for symbolic name values to AUTOSAR 4.0 Rev 3 naming scheme

▶ ASCFRIF-523 Fixed known issue: FrIf-generated code will not compile if the configuration contains multiple controllers and also single driver optimization enabled

▶ Removed range check for unused configuration parameters

▶ Implemented that `FrIf_PduInfoPtr` in `FrIf_Transmit` shall be DET-checked only for immediate PDUs

▶ Added additional MCG checks for immediate PDU configurations

▶ Added generation of BSWMD

Module version 5.0.1

2011-11-11

▶ Implemented extended receive FIFO support

▶ Implemented Channel Status Evaluation and Dem-/Det-Reporting

▶ ASCFRIF-502 Fixed known issue: The Service Needs Calculator does not add FrIf related Dem event to the Dem configuration

▶ ASCFRIF-503 Fixed known issue: The Service Needs Calculator assigns identical `SchMExclu-siveAreaId` for all exclusive areas of the FrIf

▶ Removed potential compiler warnings

Module version 5.0.0

2011-09-30

▶ Initial AUTOSAR 4.0 version

3.3.3.2. New features

▶ No new features have been added since the last release.

3.3.3.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

► Extended receive FIFO support

Description:

A new communication operation `CONSUME_RXFIFO` is implemented, which supports consuming elements out of a FlexRay controller's receive FIFO. The received element is mapped to a configured FrameTriggering and the contained PDUs are indicated in case of proper reception. Not configured FrameTriggerings which were received by the FIFO are filtered out by software.

Please note that this feature requires an enhanced FlexRay driver which implements the enhanced API functions and configuration parameters.

► Pre-compile time single cluster optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that reduces code size and runtime overhead in case only a single FlexRay cluster is required.

► Pre-compile time single controller optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that reduces code size and runtime overhead in case only a single FlexRay controller is required.

► Pre-compile time single FlexRay Driver optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that reduces code size and runtime overhead in case only a single FlexRay Driver is used (single type of FlexRay controller).

► Pre-compile time FrIf wrapper reduction optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that reduces code size and runtime overhead in case the FrIf is used in combination with an EB FlexRay Driver module.

► Pre-compile time single FlexRay Transceiver Driver optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that reduces code size and runtime overhead in case only a single FlexRay Transceiver Driver is used (single type of FlexRay transceiver).

▶ Pre-compile time API deactivation (extension to AUTOSAR specification)

Description:

The following pre-compile time configuration parameters are provided to disable certain APIs at compile time:

- ▶ `FrIf_SendWUP()`, `FrIf_SetWakeupChannel()`
- ▶ `FrIf_GetAbsoluteTimerIRQStatus()`
- ▶ `FrIf_DisableAbsoluteTimerIRQ()`
- ▶ `FrIf_GetTransceiverMode()`
- ▶ `FrIf_GetTransceiverWUReason()`
- ▶ `FrIf_ClearTransceiverWakeup()`
- ▶ `FrIf_Check_WakeupByTransceiver()`
- ▶ `FrIf_AllowColdstart()`

▶ Device health status (extension to AUTOSAR specification)

Description:

An additional API is provided that returns the controller's health status (several error status-bits) in an abstracted way. This API function can be disabled by a pre-compile time configuration parameter.

▶ Extended IRQ control (extension to AUTOSAR specification)

Description:

Additional APIs are provided to control further interrupts (if supported by hardware) like cycle start, start of dynamic segment. These API functions can be disabled by a pre-compile time configuration parameter.

▶ Relative timer control (extension to AUTOSAR specification)

Description:

Additional APIs are provided to control the relative timer. These API functions can be disabled by a pre-compile time configuration parameter.

▶ MTS control (extension to AUTOSAR specification)

Description:

Additional APIs are provided to control transmit MTS and check whether an MTS has been received. These API functions can be disabled by a pre-compile time configuration parameter.

▶ Pre-compile time delay check deactivation (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that disables the delay check of the joblist execution function which finally reduces code size and runtime overhead.

▶ Pre-compile time communication action `PREPARE_LPDU` deactivation (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that disables statically scheduled buffer reconfiguration which reduces code size and runtime overhead.

▶ Pre-compile time immediate Tx-deactivation (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that disables immediate transmission which reduces code size and runtime overhead.

▶ Pre-compile time communication action `RECEIVE_AND_STORE` and `RX_INDICATION` deactivation (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that disables decoupled reception which reduces code size and runtime overhead.

▶ Pre-compile time limitation of `FrIfCounterLimit` to the value `1` (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that fixes the allowed value of `FrIfCounterLimit` to the value 1 which reduces code size and runtime overhead.

▶ Joblist IRQ multiplexing feature (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter is provided that enables demultiplexing of the FlexRay interface joblist execution function. This enables to share multiple FlexRay controller interrupts via a single interrupt line.

▶ Virtual PDU support (extension to AUTOSAR specification)

Description:

A PDU specific configuration parameter is provided that enables support for virtual PDUs. Virtual PDUs may overlap original PDUs and will be indicated in addition to the original PDUs.

▶ Enhanced production error reporting

Description:

An enhanced production error reporting mechanism has been introduced. This allows to configure the following options independently for each Dem event:

▶ Report production errors to the `Diagnostics Event Manager` (`Dem`).

▶ Report production errors to the `Development Error Tracer` (`Det`) as development errors.

▶ Do not report production errors at all.

If a production error is redirected towards the `Det`, you may configure the reported `Det` error-ID.

Rationale:

This enhancement implements the HIS requirements concerning fault operation and error detection: His-Gen0007, HisGen0008 and HisGen0009.

▶ Buffer reconfiguration for Tx LPdus in guaranteed dynamic Segment(extension to AUTOSAR specification)

Description:

The additional config parameter `FrIfGLastGuaranteedDynamicSlotId` is given in FrIf Module to specify last guaranteed SlotId in dynamic segment. This guaranteed dynamic segment is used for buffer reconfiguration of dynamic Tx LPdus by Fr MCG if Fr config parameter `DynamicGuaranteedTxRecon-figMsgBufOptEnable` is enabled.

3.3.3.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

▶ The API function `FrIf_Transmit()` performs a state check

Description:

The API function `FrIf_Transmit()` checks the state of the FrIf state machine for the related cluster, and performs the transmission only if at least a single cluster of the FrIf is in the state `FRIF_STATE_ONLINE`. Otherwise the API function `FrIf_Transmit()` does not transmit and returns `E_NOT_OK`.

Rationale:

Provide immediate feedback to the user if transmission can be performed or not. Achive consistent behavior with CanIf module (see http://www.autosar.org/bugzilla/show_bug.cgi?id=59369).

▶ No Tx-confirmation if no Tx-request and `AlwaysTransmit` are enabled (reference to product description: ASCPD-8)

Description:

In case the configuration parameter `FrIfAlwaysTransmit` is enabled (writing message buffer without data update) for a frame triggering, the pending bit will not be set if the transmission confirmation is enabled

for the contained PDUs. This means that an Tx-confirmation will be provided only if it was preceded by a transmit request (`FrIf_Transmit()`).

Rationale:

There is no detailed description in the SWS. According to the sequence-diagrams a Tx-confirmation would be provided for `FrIfAlwaysTransmit` L-PDUs no matter whether they were transmitted before or not. The current implementation follows logical behavior and satisfies the regular use-case.

► Non-compliant deviations in the vendor-specific module definition file

Description:

The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:

The following configuration parameters are optional:

► `FrIfFrameTriggering`

► `FrIfLPdu`

► `FrIfTransceiver`

► `FrIfJobList`

► `FrIfFrameStructure`

► `FrIfPdu`

Within the AUTOSAR specification these parameters are mandatory.

Rationale:

Optional configuration parameters `FrIfFrameTriggering`, `FrIfLPdu`, `FrIfTransceiver`, `FrIfJobList`, `FrIfFrameStructure`, `FrIfPdu` allow to configure a FlexRay controller that does not transmit/receive data. This is useful to supply only startup/sync frames to the cluster.

► Initialization check in `FrIf_MainFunction()`

Description:

If the `FrIf_MainFunction()` is called while the module is not yet initialized, the `FrIf_MainFunction()` returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

Rationale:

The SchM module may schedule the module `FrIf_MainFunction()` before the module is initialized. This would result in lots of Det errors during startup. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

► The configuration parameter `FrIfRxComOpMaxLoop` is not used

Description:

Instead of the communication operation specific configuration parameter `FrIfRxComOpMaxLoop`, a global parameter `FrIfGlobalRxMaxLoop` is used to limit the maximal number of L-PDUs that can be received in each communication operation `RECEIVE_AND_INDICATE`.

Rationale:

One `FrIfRxComOpMaxLoop` value for each communication operation is highly redundant and not essential. The definition as a global parameter reduces ROM consumption and execution time of the module code.

Requirements:

FrIf00007_Conf

▶ The JobList synchronization starts after the FlexRay controller is synchronized to global time

Description:

In contradiction to AUTOSAR FrIf SWS 4.0 requirement FrIf05120a, this implementation starts and synchronizes the JobList as soon as the FlexRay controller is synchronized to the FlexRay global time (even in the state `FRIF_STATE_OFFLINE`).

Rationale:

Early synchronization of the JobList speeds up FlexRay startup time when switching the state to `FRIF_-STATE_ONLINE`. Certain communication operations (e.g. `PREPARE_LPDU`) must also be executed in the state `FRIF_STATE_OFFLINE`. This works only if the JobList is already synchronized in the state `FRIF_-STATE_OFFLINE`.

Requirements:

FrIf05120a

▶ Actually unused bits are filled with the configurable value `FrIfUnusedBitValue`

Description:

According to requirement FrIf05723, only unused bits according to the frame construction plan shall be filled with the configurable default value `FrIfUnusedBitValue`. Instead, the implementation additionally fills the currently non-transmitted PDU payload area with the `FrIfUnusedBitValue`.

Rationale:

According to the specification, the payload area of non-transmitted PDUs would be undefined otherwise.

▶ Det error `FRIF_E_JLE_SYNC` is called asynchronous to the `FrIf_JobListExec()`

Description:

The Det Error `FRIF_E_JLE_SYNC` is called from the `FrIf_MainFunction()` after the invocation of the function `FrIf_JobListExec()` has been out of the allowed execution time window `FrIfMaxISRDelay`. It is also called if the execution of the function `FrIf_JobListExec()` has been missed completely.

Rationale:

The detection of a missing function call `FrIf_JobListExec()` cannot be performed within the function `FrIf_JobListExec()`. Thus the monitoring of the correct `FrIf_JobListExec()` execution is implemented in the function `FrIf_MainFunction()`.

Requirements:

FrIf05138

▶ The pointer argument DET-checks in `FrIf_Transmit()` are only performed for immediate PDUs

Description:

In the API function `FrIf_Transmit()` the DET checks for valid pointers are only performed for immediate PDUs (`FrIfImmediate` = true). For decoupled PDUs, these checks are not performed.

Rationale:

Checks are performed only if the pointer is effectively used. For decoupled frames the pointers are not dereferenced, thus `NULL_PTR` may be passed as valid value.

▶ The configuration parameter `FrIfNoneMode` is not configurable per FrIfPdu

Description:

The configuration `FrIfNoneMode` is not configurable per FrIfPdu but it is configurable per `FrIf-FrameTriggering`. The value configured for the `FrIfFrameTriggering` applies to all contained `FrIfPdus`.

Requirements:

FrIf06050_Conf

▶ `FREE_OP_A` and `FREE_OP_B` are not implemented

Description:

The communication operations `FREE_OP_A` and `FREE_OP_B` have not been implemented in this release.

Requirements:

FrIf06067_Conf

▶ The API function `FrIf_CancelTransmit()` is not supported (reference to product description: ASCPD-24)

Description:

The API function `FrIf_CancelTransmit()` has not been implemented in this release.

Requirements:

FrIf05070, FrIf05703, FrIf05704, FrIf05705

▶ Dynamic frame length is supported only for `FrIfFrameTriggering` which contains a single `FrIfPdu`

Description:

The transmission of FlexRay frames with dynamic length is supported only for `FrIfFrameTriggering` which contains a single `FrIfPdu` only. Thus, the configuration parameter `FrIfAllowDynamicLSdu-Length` must be set to false if multiple elements of `FrIfPdu` are contained in a `FrIfFrameTriggering`.

Requirements:

FrIf05093

▶ No AUTOSAR Debugging support

Description:

The requirements associated with AUTOSAR Debugging are not supported. This comprises all requirements mentioned within the section *Debugging*.

Rationale:

EB tresos Debug & Trace is intended to be used.

Requirements:

FrIf05400, FrIf05401, FrIf05402, FrIf05403

▶ User-configurable functions are not supported

Description:

The user-configurable functions `User_TxConfirmation()`, `User_TriggerTransmit()`, and `User_-RxIndication()` have not been implemented in this release.

Rationale:

This feature is implemented as an EB-specific configuration parameter. Use configuration parameter `FrIfUserUpperLayerConfig` instead.

Requirements:

FrIf00014_Conf, FrIf00015_Conf, FrIf00016_Conf, FrIf00017_Conf, FrIf06084_Conf, FrIf06116_Conf

► No payload padding on receive side

Description:

According to the requirement FrIf05724_1, unused bits according to the frame construction plan shall be filled with the configurable default value `FrIfUnusedBitValue` after the FlexRay Driver has copied the payload into the FrIf temporary buffer. However, this implementation does not fill unreceived bits with default values.

Rationale:

On reception side, payload padding makes no sense, as only actually received payload is indicated.

Requirements:

FrIf05724

► There is no consistency check between code files and header files

Description:

According to the FrIf SWS, the FrIf module shall perform inter-module version checks. This implementation does not perform inter-module version checks.

Rationale:

The module consistency check is not within the responsibility of the basic software but part of the config-uration management and delivery process.

Requirements:

FrIf05301

► Only post-build configuration is supported

Description:

The FrIf module only supports the configuration variant `VARIANT-POST-BUILD`. `VARIANT-PRE-COM-PILE` (FrIf05282) and `VARIANT-LINK-TIME` (FrIf05286) are not supported.

Requirements:

FrIf05282, FrIf05286, FrIf05069

► Configuration parameter `FrIfDetectNITError` is not used

Description:

The configuration parameter `FrIfDetectNITError` is not used by this FrIf implementation.

Rationale:

This configuration parameter is redundant to config parameters `FRIF_E_NIT_CH_A` and `FRIF_E_NIT_-CH_B`.

Requirements:

FrIf00003_Conf

► Max FrIfCounterLimit is limited to 127

Description:

The FrIf module only supports a maximum `FrIfCounterLimit` of `127`.

Rationale:

To reduce memory consumption only one byte is used to store the FrIfCounter and to indicate if it is an decoupled or immediate PDU.

► `Bit 0-6: FrIfCounter`

► `Bit 7: Pending TxConfirmation`

Requirements:

FrIf06076_Conf

► Transmitted immediate PDUs support only dynamic LSdu length.

Description:

The transmission of immediate PDUs supports only dynamic LSdu length. The parameter `SduLength` parameter passed to Driver's API `Fr_TransmitTxLPdu()` is always set to the `SduLength` passed via `FrIf_Transmit()`.

Requirements:

FrIf05296

3.3.3.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

► For this module no limitations are known.

3.3.3.6. Open-source software

The software that is delivered with EB tresos AutoCore Generic can be classified into the following two categories:

► Software that is executed on the electronic control unit (ECU).

► Software that is used for the development infrastructure (configuration, generation, building) and thus executed on the development platform.

All license text files are located in your module delivery in the sub-folder

```
<installedmodulelocation>\impl\licenses\
```

3.3.3.6.1. Open-source software in software used for the development infrastructure

The following list of open-source software that is used in development is delivered with FrIf:

► Xerces-C
Version 2.0
http://www.apache.org/licenses/LICENSE-2.0

List of licenses:

► Apache License, Version 2.0
APACHE_LICENSE-2.0.txt (as stored in the folder mentioned above)

List of copyrights:

► Copyright 2018 Elektrobit Automotive GmbH

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

► Getopt

List of licenses:

► Getopt License, Version 2.0
GETOPT_LICENSE.txt (as stored in the folder mentioned above)

List of copyrights:

► Copyright (c)2002-2003 Mark K. Kim

► MinGW
Version 6.1
http://mingw.org/license

List of licenses:

► MinGW License, Version 6.1
MINGW_LICENSE-6.1.txt (as stored in the folder mentioned above)

List of copyrights:

► Copyright (c) 2012 MinGW.org project

3.3.4. FrNm module release notes

▶ AUTOSAR R4.0 Rev 3

▶ AUTOSAR SWS document version: 4.2.0

▶ Module version: 5.16.8.B466224

▶ Supplier: Elektrobit Automotive GmbH

3.3.4.1. Change log

This chapter lists the changes between different versions.

Module version 5.16.8
2021-10-27

▶ Internal module improvement. This module version update does not affect module functionality.

Module version 5.16.7
2021-06-25

▶ ASCFRNM-1183 Fixed known issue: FrNm might not compile if memory mapping is used

Module version 5.16.6
2021-03-05

▶ Internal module improvement. This module version update does not affect module functionality.

Module version 5.16.5
2020-10-23

► Improved Active Wakeup Bit functionality

Module version 5.16.4

2020-06-19

► Improved behavior of FrNm in RepeatMessage state when FrNmRepeatMessageTime = 0

Module version 5.16.3

2020-02-21

► Internal module improvement. This module version update does not affect module functionality.

Module version 5.16.2

2019-10-11

► ASCFRNM-1114 Fixed known issue: FrNm cannot be selected in the ImportEcuConfig wizard

► Changed maximum value for FrNmPNInfoOffset to 31 and minimum value for FrNmPNInfoLength to 1

► ASCFRNM-1127 Fixed known issue: Linker errors are reported due to incorrect memory mapping

Module version 5.16.1

2019-06-14

► Internal module improvement. This module version update does not affect module functionality.

Module version 5.16.0

2019-02-15

► Improved robustness check for references, optional parameters property and enable parameters property

► Added support for post-build selectable

Module version 5.15.0

2018-10-26

► Implemented Multi-core support

► ASCFRNM-1077 Fixed known issue: FrNm generates an invalid basic software module description if no configuration set is provided

► Changed FrNmReadySleepCnt according to AUTOSAR 4.3

Module version 5.14.3

2018-06-22

► Internal module improvement. This module version update does not affect module functionality.

Module version 5.14.2

2018-02-16

► Added support for PDU Length greater than 8 Bytes

► Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros

Module version 5.14.1

2017-09-22

► Implemented support for Car Wake Up

► Added FrNmNodeDetectionEnabled and FrNmSourceNodeIdentifierEnabled as per channel configurable.

► ASCFRNM-965 Fixed known issue: Compilation error occurs if all FrNmPnFilterMaskByteValues are set to zero

► ASCFRNM-966 Fixed known issue: Out of bounds access if at post-build more PNCs are configured than at precompile time

► ASCFRNM-977 Fixed known issue: Out of bounds access if more then one Rx PDU is configured for at least one channel

► ASCFRNM-975 Fixed known issue: State machine events may not be treated

► ASCFRNM-969 Fixed known issue: Compilation error occurs if FrNmPnEraCalcEnabled is set to true and FrNmPnEiraCalcEnabled is set to false

► ASCFRNM-971 Fixed known issue: Compiler error for FrNmPnEraCalcEnabled and invalid reference inside PduRRoutingTable

► ASCFRNM-983 Fixed known issue: User data has to be configured for all configured channels even if only one channel uses partial networking

Module version 5.14.0

2017-03-31

► Added missing memory sections and corrected compiler abstraction defines

► ASCFRNM-911 Fixed known issue: Incorrect consistency check of FrNmComUserDataSupport against FrNmUserDataEnabled

► Improved user data handling

► Implemented reception of NM Vote in dynamic segment requires vote bit to be set

► Updated dependency for FrNmControlBitVectorEnabled as specified by RfC number 73597

► ASCFRNM-942 Fixed known issue: Inconsistent partial networking information could be provided to ComM if channel specific FrNm main functions could preempt each other

Module version 5.13.0

2016-10-31

► ASCFRNM-902 Fixed known issue: NM Pdu could be corrupted in case multiple FlexRay channels are used

► Changed default value of FrNmMainFunctionPeriod to 5ms

► ASCFRNM-906 Fixed known issue: If the user requests repeat message state to FrNm, the request could be lost

Module version 5.12.0

2016-05-24

► Implemented Synchronization loss handling based on RFC 52552

Module version 5.11.0

2016-02-10

► ASCFRNM-876 Fixed known issue: FrNm lockup in Synchronized state after transient bus error

► Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`

Module version 5.10.0

2015-11-06

► Implement Flexray NM Active Wakeup Bit Support functionality

► ASCFRNM-854 Fixed known issue: Incorrect handling of ControlBitVector if other schedule variants than 1 or 2 are used

► ASCFRNM-853 Fixed known issue: CBV do not remain set to 0x00 if FrNmControlBitVectorActive is FALSE

Module version 5.9.0

2015-06-24

► Fixed description of the FrNmScheduleVariant2VoteBitValue parameter

► ASCFRNM-754 Fixed known issue: Null pointer dereferentiation in FrNm_SetSleepReadyBit if passive mode is enabled

Module version 5.8.0

2015-02-20

► "routing path not configured" error message is missleading if PduRDestPduHandleId is disabled

Module version 5.7.0

2015-01-07

► Implement FrNmAllNmMessagesKeepAwake functionality

► Updated multiplicity of FrNmRxPdu elements according to resolution in bugzilla 54555

Module version 5.6.0

2014-10-02

► ASCFRNM-762 Fixed known issue: Unreachable DEBUG macro

► ASCFRNM-776 Fixed known issue: Fix error checking in `FrNm_MainFunction`

► ASCFRNM-758 Fixed known issue: Incomplete initialization of `EiraTimer` array

► Implemented configurable value for the vote bit in case of scheduling variant 2

► Implemented `VARIANT-POST-BUILD` support

► ASCFRNM-759 Fixed known issue: FrNm passes cluster index to `FrIf_GetNmVector()` and `FrIf_-GetGlobalTime()` instead of controller index

► Implement support for the aggregation of external requested partial networks (ERA)

Module version 5.5.0

2014-04-25

► ASCFRNM-733 Fixed known issue: User data is initialized incorrectly to `0x00U` instead of `0xFFU` when partial networking is enabled

► ASCFRNM-741 Fixed known issue: FrIf uses `FrNm_TriggerTransmit` and `FrNm_TxConfirmation` although FrNm does not transmit PDUs

► ASCFRNM-738 Fixed known issue: `NULL_PTR` exception can occur if NM-passive is configured

► Added Misra deviation for empty version of `FrNm_TriggerTransmit`

► ASCFRNM-746 Fixed known issue: Configuration parameters available (editable) when there is no channel configured with `FrNmPnEnabled = true`

▶ ASCFRNM-740 Fixed known issue: Differently guarded declaration and definition of `FrNm_Eira-TimerMap` variable generates uncompilable code

▶ Fixed compiler warnings

▶ ASCFRNM-757 Fixed known issue: Incorrect origin of `FrNmRxUserDataPduId` and `FrNmRxUserDataPduRef` parameters in `FrNm.xdm.m4`

▶ Investigated unreachable code coverage issue for `FrNm_TriggerTransmit` and `FrNm_TxConfirmation`

Module version 5.4.1

2013-10-18

▶ Added symbolic names according to AUTOSAR 4.0 Rev 3 naming schema for configuration elements `FrNmUserDataTxPdu` and `FrNmUserDataRxPdu`

▶ ASCFRNM-706 Fixed known issue: A compiler error occurs when `FrNmDevErrorDetect` is disabled, `FrNmComUserDataSupport` is enabled and only one channel is configured

▶ ASCFRNM-713 Fixed known issue: `FrNm_Init()` when called in *Synchronize* mode does not result in a transition to *Bus-Sleep Mode*

▶ ASCFRNM-721 Fixed known issue: A vote received through the NM-Vector hardware service is discarded

Module version 5.4.0

2013-06-26

▶ ASCFRNM-661 Fixed known issue: `PduR_FrNmTxConfirmation()` and `PduR_FrNmTriggerTransmit()` are called with the wrong handle ID

▶ Improved the robustness of the finite state machine design by revising the event handling

▶ ASCFRNM-636 Fixed known issue: Entry to *Normal Operation State* or *Ready Sleep State* from *Repeat Message State* is delayed by one Repetition Cycle time

▶ ASCFRNM-681 Fixed known issue: EIRA changes may be reported incorrectly if `FrNm_Init()` is preempted by `FrNm_MainFunction()`

Module version 5.3.0

2013-02-15

▶ ASCFRNM-639 Fixed known issue: Compiler errors when symbolic names according to AUTOSAR 4.-0.3 are used

▶ ASCFRNM-637 Fixed known issue: PN-related parameters throw errors even when `FrNmPnEnabled` is deactivated

▶ ASCFRNM-638 Fixed known issue: The default value of the parameter `FrNmCoordinatorSyncSupport` causes an error to be reported

▶ Changed the reference path of `ComMChannel` in parameter `FrNmComMNetworkHandleRef` to `/AUTOSAR/EcucDefs/ComM/ComMConfigSet/ComMChannel`

▶ Implemented memory allocation keywords in compliance to AUTOSAR 4.0.3

Module version 5.2.0

2012-10-12

▶ ASCFRNM-598 Fixed known issue: Out-of-bounds memory access when "PN Info" is configured to be out of the limits of Rx-PDU

▶ AASCFRNM-604 Fixed known issue: Return type of API functions is not consistent with AUTOSAR 4.0.3

▶ Adapted the handle ID policy to AUTOSAR 4.0

▶ Changed the top-level structure of the software-component description in the ARXML files from `/AUTOSAR/FrNm` to `/AUTOSAR_FrNm`

▶ ASCFRNM-601 Fixed known issue: Nm user data cannot be received via the PDU Router

Module version 5.1.1

2012-06-27

▶ ASCFRNM-583 Fixed known issue: EIRA contains PN requests which are not relevant for the ECU

Module version 5.1.0

2012-03-29

▶ ASCFRNM-504 Fixed known issue: Vote is changed in the last repetition cycle before the Ready Sleep Counter expires regardless of the setting of parameter `FrNmVoteInhibitionEnabled`

▶ ASCFRNM-511 Fixed known issue: Remote Sleep Indication is invoked even when FrNmRemoteSleepIndTime is configured as 0

▶ ASCFRNM-510 Fixed known issue: Node doesn't stay in Repeat Message State for the configured time

▶ ASCFRNM-554 Fixed known issue: FrNm throws compilation error when the number of the configured channels is not equal to the number of the configured clusters

▶ ASCFRNM-555 Fixed known issue: Out of bound array access results in undefined behavior

▶ Updated to AUTOSAR 4.0.3 version

▶ ASCFRNM-564 Fixed known issue: Service Needs calculator created wrong CRITICAL SECTIONS

Module version 5.0.0

2011-10-14

► Initial AUTOSAR 4.0 version

3.3.4.2. New features

► No new features have been added since the last release.

3.3.4.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

► `FrNmVoteBitValue`

Description:

In case `FrNmPduScheduleVariant` is configured to `FRNM_PDU_SCHEDULE_VARIANT_2` or `FRNM_-PDU_SCHEDULE_VARIANT_6` the vote and the data are part of the same Data PDU so the presence of the Data PDU indicates a positive vote while its absence indicates a negative vote.

The new optional `FrNmVoteBitValue` parameter can be used to force the vote bit to a certain value in case `FRNM_PDU_SCHEDULE_VARIANT_2` and `FRNM_PDU_SCHEDULE_VARIANT_6` is selected.

In order to preserve backward compatibility, the value of `FrNmVoteBitValue` is ignored and the actual value of the vote bit is the same as in the previous releases if one of the following conditions is met:

►   `FrNmVoteBitValue` is disabled.

►   `FrNmVoteBitValue` is enabled, but `FrNmPduScheduleVariant` is set to a different value than `FRNM_PDU_SCHEDULE_VARIANT_2` or `FRNM_PDU_SCHEDULE_VARIANT_6`.

Note:

In case `FRNM_PDU_SCHEDULE_VARIANT_2` is selected, the actual value of the vote bit contained in the Data PDU should not be used to determine a positive or a negative vote. For example, if `FrNmVoteBit-Value` is enabled and set to `0` and `FRNM_PDU_SCHEDULE_VARIANT_2` is selected, any received Data PDU (which in consequence have the vote bit set to 0) actually indicates a positive vote.

► `New parameter FrNmAllNmMessagesKeepAwake`

Description:

This EB specific parameter has been introduced to control the behaviour of the ECU going to sleep depending on the values of EIRA/ERA and the vote bit.

This parameter is only relevant in case the PN functionality is enabled (`FrNmPnSupported` is TRUE).

If `FrNmAllNmMessagesKeepAwake` is set to TRUE then the ECU will stay awake as long as the vote bit is set (in case of `FRNM_PDU_SCHEDULE_VARIANT_2`, the existence of the PDU counts as a positive vote) without taking into account the value of EIRA/ERA.

In case PN is enabled and the message does not contain PN info Bit in the CBV or all PN information is zero the recieved CBV is not updated. This affects:

► Received Vote (Mixed PDUs only)

► Received Repeat Message Request is not handled (Mixed and Data PDUs)

► Outdated CBV when FrNm_GetPduData is called (Mixed and Data PDUs)

If `FrNmAllNmMessagesKeepAwake` is set to FALSE then the ECU will stay awake as long as the EIRA/ERA bit for the respective PNC is set.

The previous behaviour corresponds to `FrNmAllNmMessagesKeepAwake` being set to FALSE.

► Description of parameter FrNmSynchErrExtended

false : BusOff handling is done as described in AUTOSAR 4.0.2

true : BusOff handling is done as described in AUTOSAR 4.2.1

► Extended the range for the offset of the PN request information in the NM message

Description:

Starting from AUTOSAR requirement FrNm0062_Conf, FrNmPnInfoOffset range has changed from 1..-7 to 1..31.

Rationale:

More flexibility and freedom of configuration for user is achieved.

3.3.4.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

► Description:

Parameter FrNmPduScheduleVariant does not support values: FRNM_PDU_SCHEDULE_VARIANT_3 and FRNM_PDU_SCHEDULE_VARIANT_5.

Requirements:

FrNm0022_SchV_3_Refine, FrNm0022_SchV_5_Refine

► New configuration dependency to FrNmCoordinatorSyncSupport

Description:

FrNmCoordinatorSyncSupport has to be set to FALSE if FrNmPassiveModeEnabled is set to TRUE.

Rationale:

After decision from: https://www.autosar.org/bugzilla/show_bug.cgi?id=64141 .

Requirements:

FrNm0081_Conf

▶ Return value of APIs as `E_NOT_OK` instead of `NM_E_NOT_OK`

Description:

The requirements FRNM186, FRNM056, FRNM057, FRNM050, FRNM051, FRNM260, FRNM319, FRNM388, FRNM389, FRNM391, and FRNM392 specify the return value `NM_E_NOT_OK` for some APIs.

In contrast to this, the respective APIs return `E_NOT_OK`.

Similarly, the requirement FRNM174 specifies the return value `NM_E_OK` for the API `FrNm_Request-BusSynchronization`.

In contrast to this, `FrNm_RequestBusSynchronization` always returns `E_OK`.

Rationale:

The requirements mentioned above contradict the descriptions of the respective APIs. This is a defect in the requirement specification and has been reported in AUTOSAR Bugzilla. See http://www.autosar.org/bugzilla/show_bug.cgi?id=56329 .

Requirements:

FRNM186, FRNM056, FRNM057, FRNM050, FRNM051, FRNM260, FRNM319, FRNM174, FRNM388, FRNM389, FRNM391, FRNM392

▶ Changes in symbolic name references

Description:

If the attribute `SHORT-NAME` is not specifed for the container `FrNmTxUserDataPdu`, the symbolic name macros for `FrNmTxUserDataPduId` are generated not according to requirement ecuc_sws_2108, but according to the naming pattern `FrNmConf_<ChannelName>_FrNmUserDataTxPdu`, where `<ChannelName>` is the name of the channel that contains `FrNmTxUserDataPdu`.

Rationale:

If no short-name is specified, EB tresos Studio assumes the name of the corresponding schema node as a default. Thus, the symbolic name macros generated according to requirement [ecuc_sws_2108] are not unique.

▶ No support for multiple module configurations (reference to product description: ASCPD-77)

Description:

The FrNm supports only a single instance of the `FrNmChannelConfig` configuration container, although the AUTOSAR specification allows the module to contain multiple containers for post-build time configuration.

Requirements:

FrNm0002_Conf

▶ *Remote Sleep Indication* is not invoked at the expiry of `FrNmRemoteSleepIndTime`

Description:

Requirement FRNM181 demands `Nm_RemoteSleepIndication` to be called when no positive vote is received in the *Normal Operation State* for the duration of `FrNmRemoteSleepIndTime`. In contrast to this, `Nm_RemoteSleepIndication` is invoked only at the completion of a Repetition Cycle after the expiry of `FrNmRemoteSleepIndTime`.

Rationale:

`FrNmRepetitionCycle` is the time frame by which a node changes its NM vote. Therefore, it is better that remote sleep indication shall only be reported at a repetition cycle boundary if no remote keep-awake votes have been received at least for one full repetition cycle. See http://www.autosar.org/bugzilla/show_-bug.cgi?id=52392 .

Requirements:

FRNM181

▶ Asynchronous reaction to `FrNm_StartupError()`

Description:

Requirement FRNM336 demands that FlexRay Network Management must react (execute) synchronously on reception of the indication `FrNm_StartupError()` from the FlexRay State Manager even when the `FrNm_Mainfunction()` is no longer executing. In contrast to this, the reaction is done only when `FrNm_-Mainfunction()` is executed.

Rationale:

FrSM and FrNm are normally scheduled from the same entity and FrSM needs to be scheduled even if synchronization is lost because it is the task of FrSM to organize the re-synchronization of the FlexRay channel. Therefore, it is highly probable that the FrNm will also be scheduled if the synchronization is lost and hence a synchronous reaction is not necessary. This deviation allows a more efficient implementation.

Requirements:

FRNM336

▶ Signature of function `FrNm_Transmit()`

Description:

Requirement FRNM359 demands that the function `FrNm_Transmit()` has the signature `Std_Return- Type FrNm_Transmit(void)`. However, the following signature is being used: `Std_ReturnType FrNm_Transmit(PduIdType TxPduId, PduInfoType PduInfoPtr)`.

Rationale:

The function `FrNm_Transmit()` can be referred by the PDU Router. Therefore the signature of `FrNm_- Transmit` has to match the signature expected by the PDU Router to avoid compilation errors. See http:// www.autosar.org/bugzilla/show_bug.cgi?id=51682 .

Requirements:

FRNM359

▶ Invoking `Nm_NetworkStartIndication()`

Description:

Requirement FRNM175 demands that the function `Nm_NetworkStartIndication()` shall be called if the FrNm module successfully receives a NM message when it is in the bus-sleep mode. In contrast to this, function `Nm_NetworkStartIndication()` is called only if the FrNm module receives a NM message that contains a positive NM-vote bit when it is in the bus-sleep mode.

Rationale:

If NM-messages are sent in the static segment, these messages would be present in the FlexRay bus until FrSM puts the bus offline. This would mean that during the network shutdown phase, NM messages with no NM vote are on the bus. In the network shutdown phase some nodes have entered `BusSleepState` and other nodes are about to leave *Ready Sleep State*). On reception of such NM-messages, nodes which have already entered `BusSleepState` would call `Nm_NetworkStartIndication()` causing the bus communication to be restarted. Consequently the network would never shutdown. See http://www.- autosar.org/bugzilla/show_bug.cgi?id=59206 .

Requirements:

FRNM175

► Signature of function `FrNm_Init()`

Description:

Requirement FRNM236 demands that the function `FrNm_Init()` has the signature `void FrNm_-
Init(FrNm_ConfigType* const nmConfigPtr)`. However, the following signature is being used:
`void FrNm_Init( const FrNm_ConfigType* const nmConfigPtr )`

Rationale:

Signature has been changed to avoid compiler errors or warnings because of the qualifier mismatch if
`FrNm_Init()` is called with a pointer to a constant structure.

Requirements:

FRNM236

► Reaction on synchronization loss in state *Ready Sleep* (reference to product description: ASCPD-89)

Description:

According to requirements FRNM378, FRNM379, FRNM380, the *Bus-Sleep* state must be entered on
expiry of `FrNm_ReadySleepCnt` in *Ready Sleep* state if `FRNM_CYCLE_COUNTER_EMULATION` is set to
true and synchronization to the FlexRay schedule has been lost. In contrast to this, the synchronize state
is entered if a synchronous error is detected.

Rationale:

The definition of the behavior within the SWS document as to AUTOSAR 4.0.3 is not clear. For example
the SWS mentions an OS timer for handling the counters, but does not specify a reference to it within
the module configuration. As for AUTOSAR 4.1.1, the concerned requirements are thoroughly revised as
you can see in http://www.autosar.org/bugzilla/show_bug.cgi?id=52552 in order to overcome a couple of
shortcomings. See http://www.autosar.org/bugzilla/show_bug.cgi?id=52552 and http://www.autosar.org/
bugzilla/show_bug.cgi?id=53059 .

Requirements:

FRNM378, FRNM379, FRNM380, FrNm0079_Conf

► Parameter `FrNmVotingNextToLastRepetitionCycleDisable` is not implemented

Description:

Requirement FRNM357 states that vote changes in the next to last repetition cycle shall be statically con-
figurable at pre-compile time by the configuration parameter `FrNmVotingNextToLastRepetitionCy-
cleDisable`.

The description of requirement FrNm0073_Conf states that `FrNmVotingNextToLastRepetitionCycleDisable` is a "Pre-processor switch for disabling vote changes in the last two repetition cycles before the Ready Sleep Counter expires".

However, the parameter `FrNmVotingNextToLastRepetitionCycleDisable` is not supported in this implementation.

Rationale:

The description of parameter `FrNmVotingNextToLastRepetitionCycleDisable` is very much similar to that of parameter `FrNmVoteInhibitionEnabled`. Therefore, it seems that this configuration parameter is superfluous. See http://www.autosar.org/bugzilla/show_bug.cgi?id=57767 .

Requirements:

FRNM357, FrNm0073_Conf

► No AUTOSAR Debugging support

Description:

FrNm is not instrumented for the usage with AUTOSAR Debugging.

Requirements:

FRNM345, FRNM346, FRNM347, FRNM348, FRNM349

► Service ID of API `FrNm_StartupError()`

Description:

Requirement FRNM393 specifies the Service ID of the API `FrNm_StartupError()` as `0x01` which is in conflict with the Service ID of `FrNm_Init()`. Therefore the Service ID of `FrNm_StartupError()` has been changed to `0x10`.

See http://www.autosar.org/bugzilla/show_bug.cgi?id=51575.

Requirements:

FRNM393

► Encoding of `FrNmChannelHandle` in the `FrNmRxPduId` parameter

Description:

Requirement FrNm0013_Conf specifies that the parameter `FrNmChannelHandle` shall be encoded in the `FrNmRxPduId` parameter which is passed to the `FrNm_RxIndication()` function called by the FrIf.

In contrast to this, the parameters `FrNmChannelHandle` and `FrNmRxPduId` can be configured independently. Also FrNm does not require the parameter `FrNmChannelHandle` to be encoded in the `FrNmRxPduId` parameter passed to the `FrNm_RxIndication()` function.

Rationale:

The description of requirement FrNm0013_Conf seems to be wrong or incomplete because it does not tell how the parameter `FrNmChannelHandle` shall be encoded in the `FrNmRxPduId` parameter.

Requirements:

FrNm0013_Conf

▶ Configuration class of parameters `FrNmTxUserDataPduRef` and `FrNmPnFilterMaskByteValue` is now `PostBuild`.

Description:

According to `AUTOSAR_SWS_FlexRayNetworkManagement.pdf`, V4.2.0, the configuration class of the `FrNmTxUserDataPduRef` and `FrNmPnFilterMaskByteValue` parameters should be `LinkTime` in case `VARIANT-POST-BUILD` is selected. However, in the current implementation, these parameters are actually `PostBuild`.

Rationale:

Allow ECUs to be removed from partial network clusters during post-build.

FrNm0057_Conf, FrNm0066_Conf

▶ Configuration class of parameter `FrNmComMNetworkHandleRef` is now `PostBuild`.

Description:

According to `AUTOSAR_SWS_FlexRayNetworkManagement.pdf`, V4.2.0, the configuration class of the `FrNmComMNetworkHandleRef` parameter should be `LinkTime` in case `VARIANT-POST-BUILD` is selected. However, in the current implementation, this parameter is actually `PostBuild`.

Rationale:

Allow ECUs to be removed from partial network clusters during post-build.

FrNm0014_Conf

▶ No support for Link-time configuration parameters

Description:

The following parameters are treated as pre-compile time parameters instead of as link time parameters:

    ▶ `FrNmCarWakeUpBitPosition`

▶ `FrNmCarWakeUpBytePosition`

▶ `FrNmCarWakeUpFilterEnabled`

▶ `FrNmCarWakeUpFilterNodeId`

▶ `FrNmCarWakeUpRxEnabled`

▶ `FrNmControlBitVectorActive`

▶ `FrNmPduScheduleVariant`

▶ `FrNmPnEraCalcEnabled`

▶ `FrNmRepeatMessageBitActive`

▶ `FrNmSynchronizationPointEnabled`

▶ `FrNmPnEraRxNSduRef`

▶ `FrNmTxConfirmationPduId`

▶ `FrNmTxUserDataPduId`

▶ `FrNmDataCycle`

▶ `FrNmMainFunctionPeriod`

▶ `FrNmReadySleepCnt`

▶ `FrNmRemoteSleepIndTime`

▶ `FrNmRepeatMessageTime`

▶ `FrNmRepetitionCycle`

▶ `FrNmVotingCycle`

▶ `FrNmNumberOfClusters`

▶ `FrNmPnEiraCalcEnabled`

▶ `FrNmPnResetTime`

▶ `FrNmPnEiraRxNSduRef`

▶ `FrNmPnInfoLength`

▶ `FrNmPnInfoOffset`

▶ `FrNmPnFilterMaskByteIndex`

▶ `FrNmMainAcrossFrCycle`

▶ `FrNmRxPduId`

▶ `FrNmChannelHandle`

▶ `FrNmComMNetworkHandleRef`

Requirements:

FRNM064

► Partial Post-Build support.

Description:

Only the following parameters have been implemented so that they are configurable at post-build time:

- ► `FrNmNodeId`
- ► `FrNmPnEnabled`
- ► `FrNmRxPduContainsData`
- ► `FrNmRxPduContainsVote`
- ► `FrNmRxPduRef`
- ► `FrNmTxPduContainsData`
- ► `FrNmTxPduContainsVote`
- ► `FrNmTxPduRef`
- ► `FrNmTxUserDataPduRef`
- ► `FrNmMsgTimeoutTime`
- ► `FrNmSyncLossTimer`
- ► `FrNmPnFilterMaskByteValue`

The rest of the parameters are implemented to be configurable at pre-compile time.

Requirements:

FRNM290, FRNM292, FRNM018

► FrNmRxPdu multiplicity

Description:

The multiplicity for FrNmTxPdu is not 0..2 as specified by AUTOSAR 4.0.3 but is 0..4 as specified by AUTOSAR 4.1.3

Deviation has been introduced due to bugzilla ticket #54555.

Requirements:

FrNm0009_Conf

► FrNmRxPdu multiplicity

Description:

The multiplicity for FrNmTxPdu is not 1..2 as specified by AUTOSAR 4.0.3 but is 1..* as specified by AUTOSAR 4.1.3

Deviation has been introduced due to bugzilla ticket #54555.

Requirements:

FrNm0010_Conf

► FrNmPnEraRxNSduRef multiplicity

Description:

The multiplicity for FrNmPnEraRxNSduRef is not 1 as specified by AUTOSAR 4.0.3 but is 0..1.

Requirements:

FrNm0070_Conf

3.3.4.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

► No support for link time configuration

Description:

Link time configuration is not supported.

Requirements:

FRNM064, FRNM291

3.3.4.6. Open-source software

FrNm does not use open-source software.

3.3.5. FrSM module release notes

► AUTOSAR R4.0 Rev 3

► AUTOSAR SWS document version: 2.2.0

► Module version: 5.3.19.B466224

► Supplier: Elektrobit Automotive GmbH

3.3.5.1. Change log

This chapter lists the changes between different versions.

Module version 5.3.19

2021-10-08

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.18

2021-06-25

► Implemented PostBuildVariant support.

Module version 5.3.17

2021-03-05

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.16

2020-10-23

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.15

2020-06-19

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.14

2019-10-11

► Added xdm check to guarantee that FrIfGetWakeUpRxStatusSupport is enabled as this is a mandatory interface for FrSM in AUTOSAR 4.0.3

Module version 5.3.13

2019-06-14

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.12

2019-02-15

► ASCFRSM-378 Fixed known issue: CONST variables are not defined in an appropriate MemMap section

Module version 5.3.11

2018-10-26

► Implemented BSW distribution support.

Module version 5.3.10

2018-06-22

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.9

2017-09-22

► Renamed memory mapping section for module initialization status from SEC_VAR_FAST_INIT_-UNSPECIFIED to SEC_VAR_INIT_8. Consider updating the MemMap configuration.

Module version 5.3.8

2017-03-31

► Reorganize exclusive areas in order to avoid locking while calling external functions

Module version 5.3.7

2016-11-04

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.6

2016-10-07

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.5

2016-02-05

► Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`

► Changed type FrSM_BswM_StateType from enumeration to uint8

Module version 5.3.4

2014-10-03

► Added support for configuration importer

Module version 5.3.3

2014-04-25

► Changed FrSM to silently ignore `COMM_SILENT_COMMUNICATION` (harmonization with other <Net>SM modules)

Module version 5.3.2

2013-10-11

► Internal module improvement. This module version update does not affect module functionality

Module version 5.3.1

2013-06-14

► Changed upper range of parameters `FrSMDurationT1`, `FrSMDurationT2` and `FrSMDurationT3` to depend on main function period

▶ ASCFRSM-253 Fixed known issue: Asynchronous `FrSm_RequestComMode(COMM_NO_COMMUNI-CATION)` is delayed until after the next `FrSM_Mainfunction()` invocation when in states `FRSM_START-UP`, `FRSM_WAKEUP` or `FRSM_ONLINE_PASSIVE`

▶ Implemented AUTOSAR Bugzilla Rfc 52244 - correction of State machine inconsistencies

Module version 5.3.0

2013-02-07

▶ Updated reference paths of FrSM-`ComMChannel` reference for the introduction of `ComMConfigSet` container in ComM

▶ Implemented functionality for `SyncLoss` Dem-reporting

▶ Implemented configurable interface `<Cdd>_SyncLossErrorIndication`

Module version 5.2.0

2012-10-17

▶ Changed the top-level structure of the software-component description in the ARXML files from `/AU-TOSAR/FrSm` to `/AUTOSAR_FrSm`

Module version 5.1.1

2012-06-20

▶ Changed range for config parameter `FrSMClusterStartupReportToDemDetErrorId` to 5 - 255

Module version 5.1.0

2012-03-16

▶ Improved coding style, removed MISRA-C:2004 violations where possible

▶ Disabled `KeySlotOnlyMode` as default

▶ Added generation of BSWMD

Module version 5.0.0

2011-10-05

▶ Initial AUTOSAR 4.0 version

3.3.5.2. New features

► No new features have been added since the last release.

3.3.5.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

► Pre-compile time single cluster optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter `FrSM/FrSMGeneral/FrSMSingleClstOptEnable` is provided that reduces the code size and runtime overhead in case only a single FlexRay cluster is required.

► Pre-compile time FlexRay transceiver control optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter `FrSM/FrSMGeneral/FrSMFrTrcvControlEnable` is provided that reduces the code size and runtime overhead in case FlexRay transceivers are not controlled by FrSM. Additionally this simplifies the integration if no FrTrcv module is used.

► Pre-compile time FlexRay optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter `FrSM/FrSMGeneral/FrSMSetEcuPassiveEnable` is provided that reduces the code size in case the API function `FrSM_SetEcuPassive` is not used.

► Pre-compile time ComM indication control optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter `FrSM/FrSMGeneral/FrSMComMIndicationEnable` is provided that reduces the code size and runtime overhead in case FrSM shall not indicate state changes to ComM. Additionally this simplifies the integration if no ComM module is used.

► Pre-compile time FrNm startup error indication optimization (extension to AUTOSAR specification)

Description:

A pre-compile time configuration parameter `FrSM/FrSMGeneral/FrSMFrNmStartupErrorEnable` is provided that reduces the code size and runtime overhead in case FrSM shall not indicate startup errors to `FrNm`.

Additionally this simplifies the integration if no `FrNm` module is used.

► Enhanced production error reporting

Description:

An enhanced production error reporting mechanism has been introduced. This allows to configure the following options independently for each Dem event:

▶ Report production errors to the `Diagnostics Event Manager` (Dem).

▶ Report production errors to the `Development Error Tracer` (Det) as development errors.

▶ Do not report production errors at all.

If a production error is redirected towards the `Det`, you may configure the reported `Det` error-ID.

Rationale:

This enhancement implements the HIS requirements concerning fault operation and error detection: His-Gen0007, HisGen0008 and HisGen0009.

▶ Disable BswM-Indication

Description:

A pre-compile time configuration parameter `FrSM/FrSMGeneral/FrSMReportToBswMEnable` has been introduced. This allows to enable/disable state transition indications to BswM.

Rationale:

In case BswM-Indication is not required, disabling this parameter allows to reduce code size and runtime overhead.

3.3.5.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

▶ Only post-build configuration is supported

Description:

The FrSM module supports the configuration variant `VARIANT-POST-BUILD` only. The FrSM implementation does not support the configuration variants `VARIANT-PRE-COMPILE` (FrSm098) and `VARIANT-LINK-TIME` (FrSm099).

Requirements:

FrSm098, FrSm099

▶ Service ID for `FrSM_RequestComMode()` is `0x02` instead of `0x03`

Description:

According to the requirement FrSm020, the service ID of the API function `FrSM_RequestComMode()` shall be `0x03`. The FrSM implementation uses the service ID `0x02` for the API function `FrSM_Request-ComMode()` instead.

Rationale:

The service ID `0x03` is already used for `FrSM_GetCurrentComMode()`. In order to avoid duplicate usage of the service ID `0x03`, service ID `0x02` is used instead for the API function `FrSM_RequestComMode()`.

Requirements:

FrSm020

▶ If `FrSMDurationT1 = 0`, `FrIf_AllowColdstart()` is not always called immediately after `FrIf_-StartCommunication()`

Description:

FrSm142 requires that if the configuration parameter `FrSMDurationT1` is set to `0`, the call to the API function `FrIf_AllowColdstart()` shall immediately follow the call to the API function `FrIf_StartCommunication()`. Contrary to this, in the outlined configuration the FrSM module does not call `FrIf_Start-Communication()` on a transition from the states `FRSM_ONLINE`, `FRSM_KEYSLOT_ONLY`, or `FRSM_-ONLINE_PASSIVE` to the state `FRSM_STARTUP`.

Rationale:

In the transitions from the states `FRSM_ONLINE`, `FRSM_KEYSLOT_ONLY`, or `FRSM_ONLINE_PASSIVE` to the state `FRSM_STARTUP`, the timer t1 does not run during the transition. The timer t1 is also not started during the transition.

Therefore, there is no reason to call the API function `FrIf_AllowColdstart()` immediately after the API function `FrIf_StartCommunication()`.

Requirements:

FrSm142

▶ Single channel configuration: Always perform `FE_ALLOW_COLDSTART` if the cold start inhibit timer t1 expires

Description:

According to the requirement FrSm075, transition T04 (a) shall only trigger if `wakeupType = Sin-gleChannelWakeup`. Contrary to this, the FrSM implementation does not depend on the `wakeupType`.

Rationale:

Transition T04 (a) does not cover the case where timer t1 expires if the state `FRSM_STARTUP` has been reached via transition T02 (b). This behavior revises the version 2.2.0 of the FrSM SWS (AUTOSAR 4.0 Rev 3). Refer to http://www.autosar.org/bugzilla/show_bug.cgi?id=52244.

Requirements:

FrSm075

► Always start the timer for startup repetition when going into the state `FRSM_STARTUP`

Description:

According to the actions listed for transition T03 (c), the FrSM shall not start the timer t2 (startup repetition) during this transition into the state `FRSM_STARTUP`. Contrary to this, the FrSM implementation starts the timer t2 also in this case.

Rationale:

The timer t2 shall always run in the state `FRSM_STARTUP`. Otherwise, no startup repetition is performed if the startup of the FlexRay cluster is not successful.

Requirements:

FrSm184

► On the POC-state `POC:HALT`, execute transition back to the state `FRSM_STARTUP` also if `FrSMIsWake-upEcu` is true

Description:

According to the FrSM SWS, the transitions T10a, T10b and T17 shall only execute if the configuration parameter `FrSMCheckWakeupReason` is true.

Contrary to this, the FrSM implementation executes the transitions T10a, T10b and T17 also if the configuration parameter `FrSMIsWakeupEcu` is false.

Rationale:

Without this change, the FrSM would not perform recovery actions if the POC-state halts in the state `FRSM_ONLINE/FRSM_KEYSLOT_ONLY/FRSM_ONLINE_PASSIVE` and the configuration parameter `FrSMIsWakeupEcu` is false and the configuration parameter `FrSMCheckWakeupReason` is false.

Requirements:

FrSm081 FrSm117

► Transition T03 (b) takes precedence over transition T03 (c)

Description:

The conditions for transitions T03 (b) and T03 (c) are not mutually exclusive.

The conditions for both transitions T03 (b) and T03 (c) can be fulfilled if `wakeupTransmitted = false` and `t1_IsActive = false`. In this case, the transition T03 (b) will take precedence over the transition T03 (c).

Rationale:

The conditions for transitions T03 (b) and T03 (c) are not mutually exclusive.

Requirements:

FrSm184

► No consistency check between code files and header files

Description:

According to the FrSM SWS, the FrSM module shall perform file and inter-module version checks. This implementation does not perform file and inter-module version checks.

Rationale:

The module consistency check is not within the responsibility of the basic software, but part of the configuration management and delivery process.

Requirements:

FrSm182, FrSm140

► No AUTOSAR Debugging support

Description:

The requirements associated with AUTOSAR Debugging are not supported. This comprises all requirements mentioned within the section *Debugging*.

Rationale:

EB tresos Debug & Trace is intended to be used.

Requirements:

FrSm133, FrSm134, FrSm135, FrSm136, FrSm137

► State `FRSM_LOW_NUMBER_OF_COLDSTARTERS` not implemented

Description:

The FrSM does not support the state machine state `FRSM_LOW_NUMBER_OF_COLDSTARTERS` and the associated state machine transitions.

Requirements:

FrSm032, FrSm180, FrSm187, FrSm188, FrSm_BswM_StateType, FrSm026, FrSm192, FrSm168_Conf

► Timer for coldstart inhibit not started when going into the state `FRSM_WAKEUP`

Description:

According to the actions listed for transition T01 (b) and T01 (c), the FrSM shall start the timer t1 (coldstart inhibit) during this transition into the state `FRSM_WAKEUP`. Contrary to this, the FrSM implementation does not start the timer t1 also in this case.

Rationale:

This behavior revises the version 2.2.0 of the FrSM SWS (AUTOSAR 4.0 rev. 3). Refer to http://www.-autosar.org/bugzilla/show_bug.cgi?id=52244.

Requirements:

FrSm151 FrSm152

► Always stop the timer for coldstart inhibit when going into the state `FRSM_READY`

Description:

According to the requirement FrSm085, transition T13 shall only cancel timer t3. Contrary to this, the FrSM implementation also cancels timer t1.

Rationale:

This behavior revises the version 2.2.0 of the FrSM SWS (AUTOSAR 4.0 rev. 3). Refer to http://www.-autosar.org/bugzilla/show_bug.cgi?id=52244.

Requirements:

FrSm085

► Some configuration parameters have a lower config variant than specified

Description:

The following configuration parameters are specified to have implemented config variant link time but actually implement config variant pre-compile:

   ► `FrSMSyncLossErrorIndicationName`

► Extended trigger conditions in case `POC!Status.Freeze = true` for certain transitions

Description:

The following transitions are fired if the hardware enters `POC!Status.Freeze = true` in addition to the specified transition trigger condition:

► T04(b)

► T05

► T06

Rationale:

Transitions T04(b), T05 and T06 shall be triggered if the FlexRay controller stops communication. This happens either if `POC!Status.State` leaves the communication state or if the FlexRay controller sets `POC!Status.Freeze = true` (while obtaining the old `POC!Status.State`). This modification is required for a defined FlexRay controller operation.

Requirements:

FrSm155 FrSm076 FrSm077

► `FrSM_RequestComMode()` silently ignores `COMM_SILENT_COMMUNICATION` request

Description:

Instead of returning `E_NOT_OK` and reporting a Det error, `FrSM_RequestComMode()` silently ignores a communication request to ComM mode `COMM_SILENT_COMMUNICATION` and returns `E_OK`.

Rationale:

Streamlines behavior for all `<Net>Sm` modules and thus makes special treatment of FrSm in ComM superfluous.

Requirements:

FrSm141

3.3.5.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

► Single controller per cluster supported (reference to product description: ASCPD-9)

Description:

This implementation supports a single FlexRay controller per FlexRay cluster.

Rationale:

The FrSM SWS does not describe the behavior if multiple controllers are connected to a single cluster.

► Implementation-specific parameter range limitations

Description:

Parameter `FrSMStartupRepetitions`: range limited from 0..65535 to 0..65534.

Rationale:

This limitation allows for more efficient implementation.

► Implementation-specific parameter range limitations

Description:

Parameter `FrSMDurationT1`: range limited from 0..Infinity to 0 up to 65535 times the main function period (this equates to a time of 327.675 sec if the main function period is set to 5ms).

Rationale:

This limitation allows for more efficient implementation.

► Implementation-specific parameter range limitations

Description:

Parameter `FrSMDurationT2`: range limited from 0..Infinity to 0 up to 65535 times the main function period (this equates to a time of 327.675 sec if the main function period is set to 5ms).

Rationale:

This limitation allows for more efficient implementation.

► Implementation specific parameter range limitations

Description:

Parameter `FrSMDurationT3`: range limited from 0..Infinity to 0 up to 65535 times the main function period (this equates to a time of 327.675 sec if the main function period is set to 5ms).

Rationale:

This limitation allows for more efficient implementation.

► Implementation specific parameter range limitations

Description:

Parameter `FrSMNumWakeupPatterns`: range limited from 0..65535 to 1 up to 255.

Rationale:

This limitation allows for more efficient implementation.

3.3.5.6. Open-source software

FrSm does not use open-source software.

3.3.6. FrTp module release notes

► AUTOSAR R4.0 Rev 3

► AUTOSAR SWS document version: 4.0.0

► Module version: 4.4.27.B466224

► Supplier: Elektrobit Automotive GmbH

3.3.6.1. Change log

This chapter lists the changes between different versions.

Module version 4.4.27
2021-10-08

► ASCFRTP-967 Fixed known issue: Frame reception interrupting transmit request leads to undefined behavior in full-duplex configuration

Module version 4.4.26
2021-06-25

► Internal module improvement. This module version update does not affect module functionality.

Module version 4.4.25
2021-03-05

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.24

2020-10-23

► ASCFRTP-947 Fixed known issue: Reception of corrupted frame leads to discarding of subsequent frames

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.23

2020-06-19

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.22

2020-02-21

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.21

2019-10-11

► ASCFRTP-935 Fixed known issue: Transmit requests are not processed if FrTpLimitNumberOfConnections is active.

Module version 4.4.20

2019-07-23

► Added performance improvement: FrTp does not send CFEOB on PduR buffer underrun anymore. Note: This could lead to the termination of a segmented transmission if PduR does not indicate new data within FrTpMaxBufReq, whereas before this change PduR had time to indicate new data within FrTpMaxBufReq starting from the arrival of a new CTS from the receiver.

Module version 4.4.19

2019-06-14

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.18

2019-02-15

► Added support to limit the number of active connections per remote address and buffering additional data transmission requests.

Module version 4.4.17

2018-10-26

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.16

2018-09-28

► ASCFRTP-898 Fixed known issue: FrTp generation error in case if N:1 routing is used in PduR

Module version 4.4.15

2018-06-22

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.14

2018-04-09

► Added support for TxConfirmation() to request another transmission of the same PDU. This is a prerequisite for the PduR TP-gateway-queueing functionality.

Module version 4.4.13

2018-01-19

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.12

2017-10-03

► Compiler abstraction and memory mapping corrections. Renamed memory mapping section from VAR_-INIT_UNSPECIFIED to VAR_NO_INIT_UNSPECIFIED. Added memory section VAR_INIT_8. Consider updating the MemMap configuration.

► ASCFRTP-843 Fixed known issue: Wrong payload data might be transmitted in case of a retransmission.

Module version 4.4.11

2017-03-31

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.10

2016-11-04

► Internal module improvement. This module version update does not affect module functionality

Module version 4.4.9

2016-10-07

► Adapted resource file for the scheduling of main functions to the split of `IpduM_MainFunction()` into `IpduM_MainFunctionRx()` and `IpduM_MainFunctionTx()`.

Module version 4.4.8

2016-05-25

► ASCFRTP-805 Fixed known issue: Limited or no FrTp communication is possible in case of cancelled communication

► ASCFRTP-808 Fixed known issue: Frames with PCI.FPL value 0 are not discarded

Module version 4.4.7

2016-02-05

► Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`

Module version 4.4.6

2015-06-19

► Changed FC.CTS Bfs parameter to 0 if receive buffer is large enough for outstanding payload to receive.

Module version 4.4.5

2015-02-20

► Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro FRTP_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined

► Added configuration check for maximum Sdu length

► ASCFRTP-785 Fixed known issue: If FrTp_MainFunction() preempts FrTp_Transmit() the FrTp might be left in an unresponsive state

Module version 4.4.4

2014-10-03

► Added missing memory mapping for external function declarations

► Added check of PDU length for received data

Module version 4.4.3

2014-04-25

► Removed unnecessarily checks of configuration parameters `FrIfUserTxUL` and `FrIfUserRxIndicationUL`

► ASCFRTP-747 Fixed known issue: FrTp reports an error message if value of `FrTpRxPduRef` is not unique Note: Support for non-unique `FrTpRxPxduRef` is not defined by AUTOSAR and thus may result in troubles if a third-party FrIf module is used

► ASCFRTP-751 Fixed known issue: FrTp sends CTS on reception of segmented or acknowledged message even if `MultipleReceiverCon` is set to true

► ASCFRTP-767 Fixed known issue: Build error due to missing file `FrTp_PBcfg.c` if code generation for FrTp is disabled and only post-build configuration is compiled

Module version 4.4.2

2013-10-11

► ASCFRTP-708 Fixed known issue: FrTp passes not specified enum literal `TP_NORETRY` to `PduR_FrTpCopyTxData()` in case of unacknowledged transmission

► ASCFRTP-719 Fixed known issue: FrTp violates Bfs if remaining message fits into a single TX-Pdu

► ASCFRTP-705 Fixed known issue: If FrTp receives a `BUFREQ_E_BUSY` from PduR for an unacknowledged single-frame, then FrTp does not repeat the operation

► Extended MCG to generate XML code for Binary Code Generation

Module version 4.4.1

2013-06-14

► Added checking of configuration and platform-specific signature to prevent loading of incompatible post-build configuration

► ASCFRTP-665 Fixed known issue: Bandwidth control processing is not restarted after receiving a new flow control clear to send (FC.CTS)

► Updated the code to send RETRY frame with BP set to 0 when `PduR_FrTpCopyRxData` returned `BUFREQ_E_BUSY` for `STF`

► ASCFRTP-672 Fixed known issue: Adding TxPdus at post-build time results in invalid memory access

► ASCFRTP-671 Fixed known issue: If PduR returns `BUFREQ_E_NOT_OK` when FrTp expects available Tx-data, an allocated Tx-PDU is not released

► Added checking of published information signature to prevent loading of incompatible post-build configuration

► Updated the code to copy the STF payload to the local buffer in case `PduR_FrTpStartOfReception` does not provide the necessary buffer and if the config parameter `FrTpCopyToLocalBuffer` was enabled

► ASCFRTP-691 Fixed known issue: Generation aborts if configuration parameter `FrIfUserTxUL` is not set to `FR_TP`

► ASCFRTP-697 Fixed known issue: FrTp post-build time configuration does not compile if used with PbcfgM

► Updated the code to respect Time Br also before sending a RETRY frame after STF frame

Module version 4.4.0

2013-02-08

► Added relocatability to post-build configuration

► ASCFRTP-569 Fixed known issue: Indefinite number of flow control WAIT frames may be transmitted when `FrTpMaxFCWait` is configured to `0`

► ASCFRTP-752 Fixed known issue: Bandwidth control is not respected if `FrTp_MainFunction` cycle time is different from the FlexRay communication cycle time

► ASCFRTP-611 Fixed known issue: The module does not release the PDU used in `FrTp_TriggerTransmit`, if the call is not successful due to the error during a `PduR_FrTp CopyTxData` request

► ASCFRTP-612 Fixed known issue: FrTp accesses invalid memory if `FrTp_Transmit()` is called and no Tx-PDU is available for the transmission of the startframe (STF)

► ASCFRTP-620 Fixed known issue: If full duplex is disabled, the allocated Tx-Pdu is not released if an AR timeout occurs

► Updated the BSW to resend the STF if FC RETRY is received with `BP 0` for an unsegmented acknowledged transmission

► ASCFRTP-467 Fixed known issue: FrTp does not retry calling `PduR_FrTpCopyTxData()` when `PduR_FrTpCopyTxData()` returns `BUFREQ_E_BUSY` resulting in a connection timeout

► ASCFRTP-478 Fixed known issue: FrTp does not copy payload of start frame to PduR if Rx buffer is not available at start of reception

► Implemented configurable performance time Br

► Added configuration parameter `MaxBufReq` to limit the number of calls to `PduR_FrTpCopyTxData()` in case no buffer is provided

▶ Set bandwidth control byte in FC CTS frame according to configured value

▶ ASCFRTP-578 Fixed known issue: Incorrect configuration check causes code generation failure if `FrTp-MultipleReceiverCon` is set to false and no RxSdu is configured

Module version 4.3.0
2012-10-16

▶ Updated to AUTOSAR 4.0 TP API

▶ Implemented AUTOSAR 4.0 `HandleId` policy

▶ Changed the top-level structure of the software-component description in the ARXML files from `/AUTOSAR/FrTp` to `/AUTOSAR_FrTp`

Module version 4.2.0
2012-06-20

▶ ASCFRTP-492 Fixed known issue: FrTp might retransmit data without RETRY flow control frame

Module version 4.1.0
2012-03-20

▶ ASCFRTP-433 Fixed known issue: FrTp does not compile if the `sizeof() PduLengthType` is larger than `sizeof() uint16`

▶ ASCFRTP-435 Fixed known issue: No LF is transmitted after a CFEOB that contains the last data bytes to be transferred

▶ Supported special reduced configurations (connections handle only Rx or Tx)

▶ Updated naming scheme for #defines for symbolic name values to AUTOSAR 4.0 Rev 3 naming scheme

▶ ASCFRTP-446 Fixed known issue: Redefinition of `TS_RELOCATABLE_CFG_ENABLE` might result in compiler error messages

▶ ASCFRTP-448 Fixed known issue: FrTp bandwidth control is decreased to the next even number of PDUs per cycle

▶ ASCFRTP-449 Fixed known issue: FrTp accepts start frames with wrong PCI values

▶ ASCFRTP-451 Fixed known issue: FrTp accepts last frames with invalid PCI values consequently closing the connection unexpectedly

▶ ASCFRTP-450 Fixed known issue: FrTp might sent a last frame with an amount of payload that exceeds the receivers buffer size

▶ ASCFRTP-454 Fixed known issue: FrTp might send a consecutive frame end of block (CFEOB) instead of a consecutive frame (CF)

► Added generation of BSWMD

► Implemented setting of default values and disabling unused config parameters

Module version 4.0.0

2011-10-05

► Initial AUTOSAR 4.0 version

3.3.6.2. New features

► No new features have been added since the last release.

3.3.6.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

► Optional support for copy of STF to local buffer

Description:

A received FrTp startframe is copied to a local buffer only if `FrTpCopyToLocalBuffer` is enabled. Thus if in the upper layer no buffer is immediately available and `FrTpCopyToLocalBuffer` is disabled, the following happens:

► For unacknowledged reception the connection is rejected sending a FC-OVFLW frame instead of a FC-WAIT frame.

► For acknowledged reception the data is re-requested sending a FC-RETRY frame instead of a FC-WAIT frame.

This enhancement reduces RAM and execution time consumption.

► Optional support to limit active transmissions and buffer transmission requests

Description:

The number of active transmissions can be limited on a remote address basis and additional transmission requests occurring via `FrTp_Transmit` can be buffered if `FrTpLimitNumberOfConnections` is enabled.

The maximum number of active connections per remote address can be configured by setting the remote address `FrTpRa` and the respective connection limit `FrTpConnectionLimit` in `FrTpConnectionLimitConfig`. Adequate buffer sizes can be chosen by setting `FrTpConnectionBufferSize` in every `FrTpConnectionLimitConfig` and `FrTpMaxNumberOfBufferedConnections` for unlimited connections.

This feature can be used to realize a scenario where only a small number of transmission connections are active at the same time and therefore have a greater bandwidth per connection. It avoids having a large number of active transmission connections, where each connection has a small bandwidth.

This enhancement increases RAM, ROM and execution time consumption.

3.3.6.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

► An invalid `FrTpRxPduId` in function `FrTp_RxIndication()` is not detected

Description:

This implementation does not evaluate the parameter `FrTpRxPduId` passed to `FrTp_RxIndication()`. No error checking of this parameter is done and no action depending on this value is taken.

Rationale:

Errors in parameter `FrTpRxPduId` can only occur due to an incorrect system configuration. Transmission errors are still detected by address information and sequence number contained in PCI.

Requirements:

FRTP1147, FRTP1069, FRTP1070, FRTP1074, FRTP1075

► Synchronous state machine processing

Description:

The FrTp state machine is processed within the callback functions `FrTp_RxIndication()`, `FrTp_-TriggerTransmit()`, and `FrTp_TxConfirmation()` as far as possible. Data is immediately processed and forwarded to upper layers.

Rationale:

This synchronous implementation reduces the overall RAM and execution time requirements as no extra data copy operations are required. Additionally thread concurrency is reduced that results in a less complex implementation.

Requirements:

FRTP1123

► `FrTp_Shutdown()` has not been implemented (reference to product description: ASCPD-96)

Description:

The API function `FrTp_Shutdown()` has not been implemented.

Rationale:

There is no AUTOSAR internal user for the API function `FrTp_Shutdown()` and the behavior and operating constraints are not clearly specified in the AUTOSAR SWS. Using the function might be risky since expectations and actual behavior might differ, so it was decided to skip the function implementation.

Requirements:

FRTP1036, FRTP148

▶ Functionality of `COUNTER_AR` and `COUNTER_AS` have not been implemented

Description:

`COUNTER_AR/AS` counts the retry attempts of calling `FrIf_Transmit()` if a timeout AR/AS occurs. This counter is not implemented. After a timeout AR/AS, an error is reported to the upper layer and the communication is stopped.

Rationale:

As communication is stopped caused by a AR/AS timeout, there is no reason to count the number of timeouts and restart transmission, as the whole connection has been closed.

Requirements:

FRTP012_Conf, FRTP013_Conf, FRTP1114

▶ Immediate transmission has not been implemented

Description:

Support for FrIf immediate transmission has not been implemented in this release. The lower layer module must call `FrTp_TriggerTransmit` to fetch data ready for transmission.

Requirements:

FRTP1084

▶ Receive/transmit cancellation have not been implemented

Description:

The API functions `FrTp_CancelReceive()` and `FrTp_CancelTransmit()` have not been implemented in this release.

Requirements:

FRTP1120, FRTP422, FRTP423, FRTP1104, FRTP424, FRTP1180, FRTP1181, FRTP1182, FRTP1183, FRTP1097, FRTP384, FRTP1116, FRTP385, FRTP386, FRTP150, FRTP1141, FRTP1172, FRTP036_-Conf

► `FrTp_ChangeParameter()` has not been implemented

Description:

`FrTp_ChangeParameter()` has not been implemented in this release.

Requirements:

FRTP242, FRTP1143, FRTP151, FRTP1115, FRTP1144, FRTP1156, FRTP052_Conf

► Unknown message length has not been implemented

Description:

Transmission and reception of messages with unknown message length has not been implemented in this release.

Requirements:

FRTP1011, FRTP1012, FRTP1101, FRTP1102, FRTP1043, FRTP1044, FRTP1134, FRTP1124, FRTP1062, FRTP1063, FRTP1064, FRTP1065, FRTP1066, FRTP1067, FRTP1184, FRTP044_Conf

► `FrTpTimeFrIf` is not supported

Description:

In case of `FrIf_Transmit` returning an error, the time until the next call to `FrIf_Transmit` is not configurable.

Requirements:

FRTP031_Conf

► `FrTpTimeBuffer` is not supported

Description:

In case `PduR_FrTpCopyTxData` or `PduR_FrTpCopyRxData` returns `BUSY`, the call will be retried in the next main function invocation. `FrTpTimeBuffer` will not be respected.

Requirements:

FRTP030_Conf

► `FrIf_CancelTransmit` is not called (reference to product description: ASCPD-24)

Description:

In case of timeout As or timeout Ar `FrIf_CancelTransmit` is not called.

Requirements:

FRTP578, FRTP1100

▶ Support of configuration variant post-build (reference to product description: ASCPD-77)

Description:

The FrTp module only supports the configuration variant post-build

Requirements:

FRTP1001, FRTP1131

▶ No support of configuration parameter `FrTpMaxBufferSize`

Description:

The configuration parameter `FrTpMaxBufferSize` is not evaluated.

Rationale:

There is no use case for this configuration parameter. A buffer is never requested, instead FrTp retrieves the available PduR buffer size and uses it to allocate TxPdus.

Requirements:

FRTP015_Conf

▶ No support of configuration parameter `FrTpAckRt`

Description:

The configuration parameter `FrTpAckRt` is not evaluated. It is not possible to reduce code size by disabling the acknowledge and retry mechanism.

Requirements:

FRTP002_Conf, FRTP598

▶ No AUTOSAR Debugging support

Description:

The requirements associated with AUTOSAR Debugging are not supported. This comprises all requirements mentioned within the section *Debugging*.

Rationale:

EB tresos Debug & Trace is intended to be used.

Requirements:

FRTP212, FRTP1159

► Contradicting requirements

Description:

Requirements which are contradicting itself or the ISO10681-2 Specification are not implemented.

Requirements:

FRTP200, FRTP1074, FRTP1076, FRTP1186, FRTP218, FRTP1110

► Intermodule version checking is not implemented

Description:

Intermodule version checking is not implemented.

Requirements:

FRTP1192, FRTP1158, FRTP1189, FRTP1190, FRTP1191

3.3.6.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

► The maximum value allowed for `FrTpTxConfirmationPduId` is 255.

Description:

The maximum value allowed for `FrTpTxConfirmationPduId` is 255. This means that at most 256 Tx PDUs must be configured.

Rationale:

This limitation reduces the configuration size.

► FrTp supports non-unique values in `FrTpRxPxduRef`

Description:

FrTp allows configurations where a single PDU is used multiple times within the FrTp configuration by providing the same value to configuration parameter `FrTpRxPxduRef` several times. AUTOSAR does

not define how the FrIf module shall handle such a configuration, thus third-party FrIf modules may fail to generate code. However, the EB FrIf module properly handles such a configuration.

Rationale:

OEMs want to reuse a single RX-PDU in multiple connections (with different RxPools).

► The maximum number of allowed connections is 254.

Description:

The maximum number of allowed connections is 254. This means that no more than 254 `FrTpConnection` parameter containers can be configured.

Rationale:

This limitation reduces the configuration size.

3.3.6.6. Open-source software

FrTp does not use open-source software.

4. ACG8 FlexRay Stack user's guide

4.1. Overview

The ACG8 FlexRay Stack user's guide provides information about the concepts of the FlexRay stack in the AUTOSAR context. Before you read this user's guide, read the general concepts about communication stacks in AUTOSAR that are described in the EB tresos AutoCore Generic documentation.

► Section 4.2, "Background information" describes the concept of FlexRay communication in the AUTOSAR context.

► Section 4.3, "FlexRay communication stack dependencies" describes the FlexRay stack module dependencies that differ from the general communication stack module dependencies as described in the EB tresos AutoCore Generic documentation.

► Section 4.5, "Generating jobs for the FrIf joblist" describes the FrIf Joblist Editor tool designed to help you configure the Flexray joblist and L-PDUs in the `FrIf` module.

► Section 4.6, "Extended RxFifo support in FrIf and Fr" describes a special Elektrobit (EB) optimization feature that works with an EB `Fr` driver for Freescale FlexRay controllers.

4.2. Background information

This chapter provides general information about the FlexRay communication concepts in the AUTOSAR context. If you are not familiar with the general concepts of communication in AUTOSAR, read the general information provided in the EB tresos AutoCore Generic documentation first.

4.2.1. Communication in AUTOSAR FlexRay

The FlexRay communication stack supports a many-to-one mapping between I/N-PDUs and L-PDUs. This means that multiple I/N-PDUs can be packed into a single FlexRay frame at the sending ECU and have to be extracted again by the receiving ECU. This packing and unpacking is performed by the `FrIf` module.

As far as the temporal properties of the transmission and the reception of L-PDUs are concerned, the FlexRay communication controllers operate on the basis of a time-division multiple access (TDMA) scheme. This fact is agreed upon among all communication controllers of a FlexRay cluster and part of each communication controller's configuration.

The `FrIf` contains a *job list* consisting of *jobs*. Hereby each job itself can comprise multiple *communication operations*. Each job is assigned a defined temporal offset from the start of the job list. The `FrIf_JobLis-`

`tExec()` function takes care of invoking the jobs of the job list at the appropriate point in time. In this way, the job list of the `FrIf` is very similar to the schedule table of the `LinIf` module. The main difference is the fact that the `FrIf`'s job list is executed synchronously to the autonomous communication schedule on the FlexRay network. This is done by means of the synchronously called `FrIf_JobListExec()` function.

The job list of the `FrIf` contains entries

▶ to transmit L-PDUs,

▶ to receive L-PDUs,

▶ to enable a transmission confirmation,

▶ and to enable a reception indication.

The `FrIf` module distinguishes between *direct transmission* and *decoupled transmission*.

In *direct* transmission, the upper layer's call to `FrIf_Transmit()` has to be synchronous to the FlexRay communication schedule and causes an immediate call to the `Fr` module's `Fr_TransmitTxLPdu()` function for the transmission of the frame.

In *decoupled* transmission, the request is simply stored in the `FrIf` for later processing.

1. As soon as the transmission job of the frame containing the I/N-PDU is due, the `FrIf` requests the data for all I/N-PDUs in this frame from the upper layer by calling, for example, `PduR_FrIfTriggerTransmit()`.

2. Once this data has been retrieved for all I/N-PDUs, the frame content is assembled and passed to the `Fr` module for transmission by calling the function `Fr_TransmitTxLPdu()`.

3. After the transmission has taken place, the `FrIf`'s job list contains a job with a transmission confirmation communication operation.

   This communication operation queries whether the preceding transmission has been successful by calling the `Fr` module's function `Fr_CheckTxLPduStatus()`.

4. If the transmission was successful, the `FrIf` module calls the upper layer's `TxConfirmation()` function for each I/N-PDU in the frame for which the upper layer has issued a transmission request.

For receiving L-PDUs, the `FrIf`'s job list has to contain a job with a communication operation for receiving a frame.

1. This communication operation causes the `Fr` module's `Fr_ReceiveRxLPdu()` function to be called.

   This retrieves the received data from the FlexRay communication controller's hardware buffers.

   Thus the job containing a communication operation for receiving a frame has to be scheduled *after* the reception of the frame has been completed by the FlexRay communication controller.

2. In case the `Fr_ReceiveRxLPdu()` indicates that the frame was received successfully, the `FrIf`'s receive communication operation generates a reception indication for each N-PDU in the frame by calling the `RxIndication()` function of the upper layer, e.g. `PduR_FrIfRxIndication()`.

Depending on whether these indications are triggered immediately by the receive communication operation or by a dedicated reception indication communication operation in a job scheduled at a later point in time, the `FrIf` distinguishes between *immediate reception* or *decoupled reception*.

4.3. FlexRay communication stack dependencies

This section describes issues, in which the functionality and/or the module dependencies of the FlexRay communication stack modules differ from the description provided in the EB tresos AutoCore Generic documentation.

4.3.1. Module dependencies

In the current version of EB tresos AutoCore, the interface between the `FrTrcv` and the FlexRay transceiver is restricted to digital I/Os (`Dio`).

4.4. Network management in AUTOSAR FlexRay stack

The network and state management is described in the EB tresos AutoCore Generic documentation concept chapter "Network management and state management stack". You find information about the concepts of the network and state management in AUTOSAR. You also learn how to configure the stack.

4.5. Generating jobs for the FrIf joblist

4.5.1. Overview

In this chapter you are going to learn how to generate a joblist and L-PDUS for an existing `FrIf` configuration with the help of a design wizard, the FrIf Joblist Editor.

The FrIf Joblist Editor is a EB tresos Studio plug-in, for manually or automatically generating a joblist and L-PDUs for an existing `FrIf` configuration.

TIP	**The FrIf Joblist Editor simplifies creating a `FrIf` joblist**
	Configuring the `FrIf` joblist without assistance of the FrIf Joblist Editor is a complex and time-consuming task. The FrIf Joblist Editor simplifies this task.

▶ Section 4.5.2.1, "Preconditions: Necessary configuration parameters" outlines which kind of parameters the FrIf Joblist Editor depends on.

▶ Section 4.5.2.3, "Automatic creation and assignment of job triggers" explains the mechanism of creating a joblist in detail and addresses advanced users and all those interested in the underlying algorithm.

▶ Section 4.5.3, "Using the FrIf Joblist Editor" explains the GUI elements and can be used together with Section 4.5.4, "Creating a job trigger" as guidance to create or edit a joblist.

4.5.2. Background information

Required knowledge

▶ To understand the following sections, you need knowledge about EB tresos Studio and the AUTOSAR modules `Fr` and `FrIf`.

▶ If you have never worked with EB tresos Studio before, you are strongly recommended to read the `EB tresos Studio documentation`, chapter `EB tresos Studio user's guide` before you continue to read this chapter.

▶ Moreover, if you are configuring the `Fr` and `FrIf` modules for the first time, be sure to have read the chapters about the FlexRay communication stack in Section 4.3, "FlexRay communication stack dependencies" before continuing here.

4.5.2.1. Preconditions: Necessary configuration parameters

Before you start the FrIf Joblist Editor, you need to configure the parameters in the following list.

The parameters in the list are `FrIf` parameters that are part of an ECU extract of a system configuration. These parameters are usually obtained by importing data from an external source, e.g. by importing from an AUTOSAR system description or Fibex file.

`FrIf/FrIfConfig/FrIfConfig/FrIfCluster`
  The `FrIfCluster` must contain at least one `FrIfController` subcontainer. The `FrIfCluster` to which the `FrIfJobList` belongs, must already contain valid values for the following configuration parameters:

  ▶ `FrIfGdCycle`

  ▶ `FrIfGdMacrotick`

- ▶ `FrIfGdStaticSlot`

- ▶ `FrIfGNumberOfStaticSlots`

- ▶ `FrIfGdNit`

`FrIf/FrIfConfig/FrIfConfig/FrIfCluster/FrIfController`

- ▶ The `FrIfController` must contain one or more `FrIfFrameTriggering` subcontainers.

- ▶ The container in the `Fr` module configuration referenced via `FrIfFrCtrlRef` must contain at least one `FrAbsoluteTimer`.

`FrIf/FrIfConfig/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering`

The `FrIfFrameTriggering` subcontainers of each `FrIfController` must contain values for the following configuration parameters:

- ▶ `FrIfBaseCycle`

- ▶ `FrIfChannel`

- ▶ `FrIfCycleRepetition`

- ▶ `FrIfSlotId`

- ▶ `FrIfFrameStructureRef`

4.5.2.2. Jobs and communication operations

The FrIf Joblist Editor is used to configure all communication operations for sending and receiving FlexRay frames (via communication operations). In ECU configurations, a set of operations are grouped to a *job*, which defines the absolute point in global FlexRay time from the beginning of a FlexRay communication cycle when the communication operations are actually being executed.

Jobs and communication operations are modelled using job trigger and communication operation objects.

4.5.2.3. Automatic creation and assignment of job triggers

The process of automatic job trigger assignment lets you define the maximum number of job triggers to be created. With the option **AUTO JT**, a set of job triggers is generated automatically. Job trigger offsets are calculated automatically. For instructions, see Section 4.5.4, "Creating a job trigger".

The process of automatic assignation automatically assigns communication operations to the existing job triggers. Automatic assignation is done with the option **AUTO Assign**. For instructions, see Section 4.5.4, "Creating a job trigger".

In Section 4.5.2.3.1, "Simple Auto JT algorithm", the algorithm for the simple automatic job trigger (Simple Auto JT) is outlined. This automatic job trigger uses a maximum number of two job triggers. The advanced automatic

job trigger (Advanced Auto JT) uses three up to nine job triggers. The Advanced Auto JT is outlined in section Section 4.5.2.3.2, "Advanced Auto JT algorithm"

4.5.2.3.1. Simple Auto JT algorithm

The Simple Auto JT algorithm creates at most two job triggers and is sufficient for applications that tolerate processing latencies of sent and/or received frame triggers of about half a communication cycle within the `FrIf`.

4.5.2.3.1.1. Job trigger offset calculation

The `FrIfMacrotick` parameter of the two created job triggers is calculated by dividing the communication cycle into two parts of approximately the same length, so that the end of the first part lies at the end of a static communication slot. If the dynamic part is larger than half of the communication cycle, the two parts are divided at the beginning of the cycle's dynamic part.

The offset of the job triggers is set to the midpoint of each part.

4.5.2.3.1.2. Assignment of communication operations to job triggers

Communication operations for frame triggers located in the first part are assigned to the job trigger that is executed during the second part and vice versa.

An example is outlined in Figure 4.1, "Simple Auto JT": The whole length of the communication cycle is `5000MT`. The first part ends at slot m at `2500MT`. The `FrIfMacrotick` parameter for both job triggers is therefore chosen to be `1250MT` and `3750MT`. Job 2 handles all communication operations for the slots with a slot number larger than m. Job 1 handles all slots with a slot number equal or lower than m.
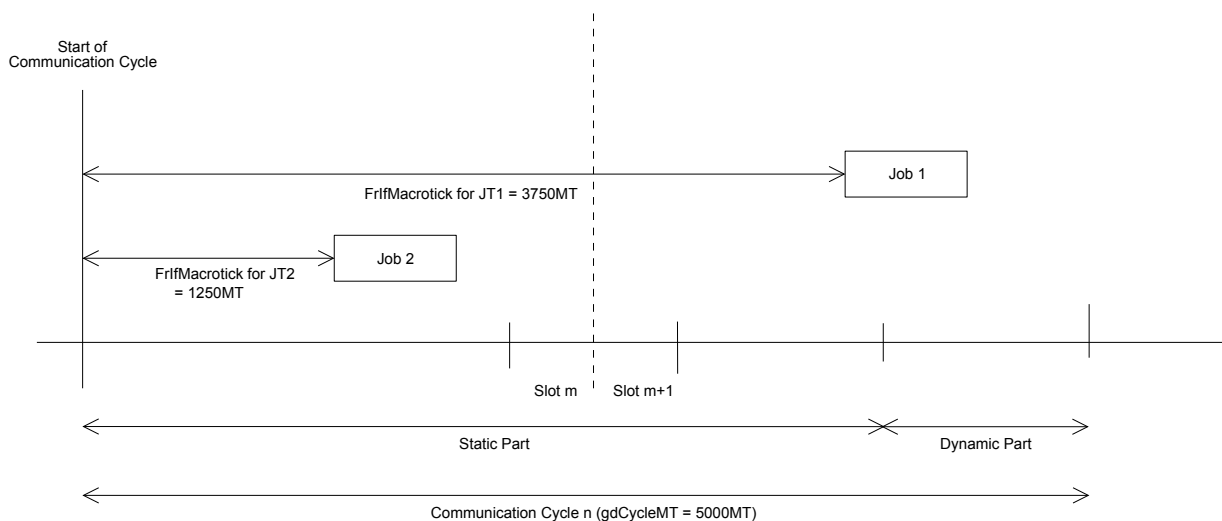
Figure 4.1. Simple Auto JT

4.5.2.3.2. Advanced Auto JT algorithm

The Advanced Auto JT algorithm creates up to nine job triggers and can be used for applications that require small processing latencies of sent or received frame triggers, e.g. gateway applications.

4.5.2.3.2.1. Job trigger offset calculation

The `FrIfMacrotick` parameter of the created job triggers is calculated by defining n job triggers in such a way that the interval between any two succeeding job triggers is the same and that one job trigger has its `FrIfMacrotick` parameter set to the end of the static part plus `GdStaticSlot`.

The algorithm requires that at least one job trigger exists with a `FrIfMacrotick` value that is larger than `GdStaticSlot` and smaller than the length of the static part minus `GdStaticSlot`. In the rare cases in which no job trigger satisfies this condition, the simple Auto JT algorithm is applied.

4.5.2.3.2.2. Assignment of communication operations to job triggers

For the Advanced Auto JT algorithm, the assignment depends on whether the communication operation for a frame trigger is to be executed before the beginning of the frame trigger slot or after the end of the slot, and on whether the slot resides in the static or in the dynamic part of the communication cycle.

If the communication operation must be executed before the beginning of the frame trigger slot (`DECOU-PLED_TRANSMISSION`, `PREP_LPDU_DEC_TX`, `PREPARE_LPDU`), and the slot resides in the dynamic part, the communication operation is assigned to the job trigger that starts immediately before the dynamic part begins, which is `Job 3` in Figure 4.2, "Advanced Auto JT".

If the slot resides in the static part, the communication operation is assigned to the job trigger preceding the job trigger that lies immediately before the start of the slot. In the example Figure 4.2, "Advanced Auto JT", `Job 3` is the job trigger that starts immediately before slot 10. The predecessor of `Job 3` is `Job 2`, hence it is the job trigger to which `DECOUPLED_TRANSMISSION`, `PREP_LPDU_DEC_TX`, and `PREPARE_LPDU` communication operations for slot 10 are assigned. This assignment strategy always allows the communication operations to finish in time, i.e. before the slot starts. The downside of that strategy is the period between the start time of the job trigger and the actual transmission time of the slot. All data that arrives at `FrIf` for sending in this period is delayed until the next instance of the job trigger. Section 4.5.2.3.2.3, "Assignment of communication operations to job triggers using Redzones" describes a more aggressive assignment strategy by defining *Redzones*.

If the communication operation must be executed after the end of the slot (`RECEIVE_AND_INDICATE`, `TX_-CONFIRMATION`), and the job trigger after the slot starts at least `GdStaticSlot` macroticks after the slot end, the communication operation is assigned to this job trigger. If the job trigger starts earlier, the communication operation is assigned to the subsequent job trigger.

In the example Figure 4.2, "Advanced Auto JT", `Job 3` is assigned `RECEIVE_AND_INDICATE`, `TX_CONFIR-MATION` communication operations for slot 5. Because `Job 3` starts too close to the end of slot 6, the suc-

cessor of `Job 3` (i.e. `Job 1`) is assigned `RECEIVE_AND_INDICATE`, `TX_CONFIRMATION` communication operations for slot 6.
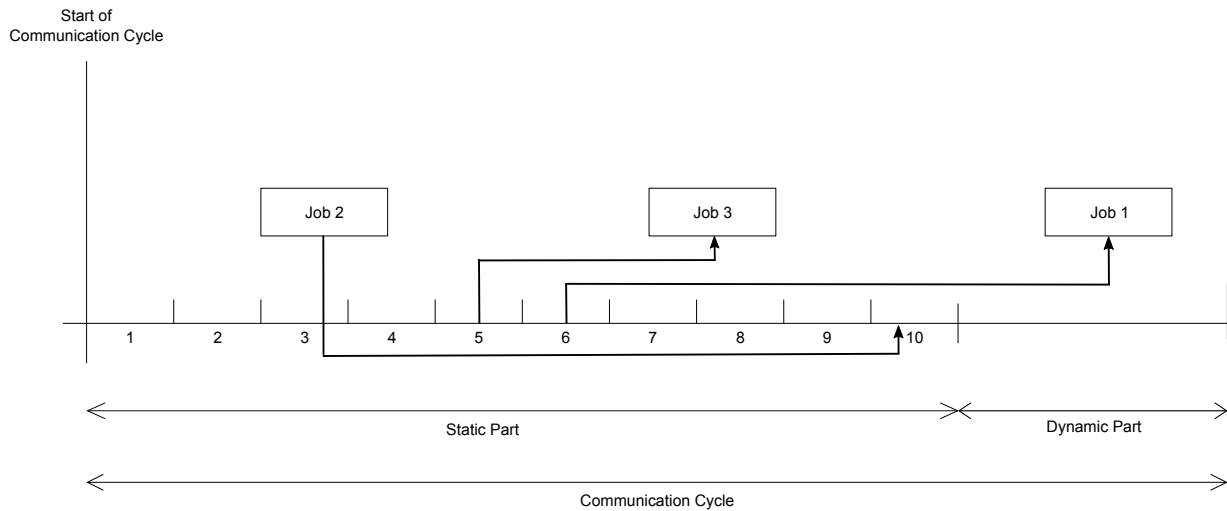
Figure 4.2. Advanced Auto JT

4.5.2.3.2.3. Assignment of communication operations to job triggers using Redzones

Figure 4.3, "Advanced Auto JT using Redzones" shows that the ECU uses slots 8, 9, and 10 for sending frames. Hence, `DECOUPLED_TRANSMISSION` and `PREP_LPDU_DEC_TX` communication operations for these slots are assigned to `Job 2`. Any send requests for PDUs that are supposed to be sent in these slots are delayed to the next instance of `Job 2` if the requests arrive after the start time of `Job 2`.

In order to reduce this latency, a *Redzone* can be defined for `Job 3`. The *Redzone* specifies a time period in macroticks beginning with the start time of `Job 3`. Any static slot that starts after this *Redzone* is processed by `DECOUPLED_TRANSMISSION`, `PREP_LPDU_DEC_TX`, `PREPARE_LPDU` communication operations of `Job 3`, whereas these communication operations for any slot that starts within the *Redzone* are assigned to the predecessor of `Job 3`.

As Figure 4.3, "Advanced Auto JT using Redzones" indicates, slot 10 is now processed by `Job 3` because this slot lies outside its *Redzone*. The slots 8 and 9 lie within the *Redzone*, so they are processed by the predecessor of `Job 3`, which is `Job 2`.

TIP	**Measure the finishing times of your job triggers**
	Providing a *Redzone* requires knowledge about what point in time a job trigger is always finished. This needs to be measured in a running system. Sent data is delayed sporadically or even systematically if the *Redzone* of a job trigger is too short.
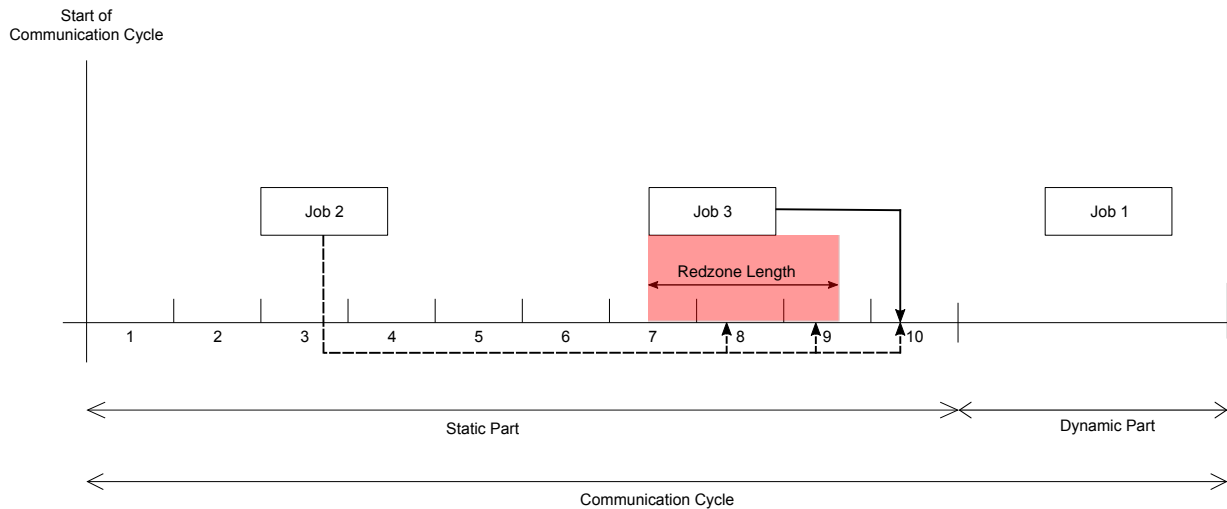
Figure 4.3. Advanced Auto JT using Redzones

4.5.2.3.3. Calculation of communication operation parameters

For every frame trigger sent or received that has its active field set to `true`, one or more communication operations are created and assigned to the related job triggers.

For Rx frame triggers, if the buffer sharing field is set to `forbidden`, one communication operation of type `RECEIVE_AND_INDICATE` is created. If the buffer sharing field is set to `allowed`, one additional communication operation of type `PREPARE_LPDU` is created. The priority value of the `RECEIVE_AND_INDICATE` communication operation is set to 20. The priority value of the `PREPARE_LPDU` communication operation is set to 25.

For Tx frame triggers that have the Tx mode field set to `decoupled`, if the buffer sharing field is set to `forbidden`, one communication operation of type `DECOUPLED_TRANSMISSION` is created. If the buffer sharing field is set to `allowed`, one communication operation of type `PREP_LPDU_DEC_TX` is created. In both cases, the priority value is set to 30. If the sent frame contains at least one PDU that has its `FrIfConfirm` bit set to `true`, one additional communication operation of type `TX_CONFIRMATION` with priority value of 10 is created.

The cycle repetition (CR) parameter of a communication operation is set to the value of the CR parameter of the associated frame trigger.

The base cycle (BC) parameter for the communication operations depends on:

► The `FrIfCommunicationAction` types can be grouped according to their execution before or after the slot in which the frame is transferred.

► If the `FrIfCommunicationAction` is of type `DECOUPLED_TRANSMISSION`, or `PREPARE_LPDU`, it must be executed before the slot.

► If the `FrIfCommunicationAction` is of type `TX_CONFIRMATION`, `RECEIVE_AND_INDICATE`, `RECEIVE_AND_STORE`, or `RX_INDICATION` (must also be preceded by a `RECEIVE_AND_STORE` operation), it must be executed after the slot.

To set the BC of a communication operation to the BC parameter of the frame trigger, the following conditions apply:

► `FrIfCommunicationAction`

    ► The `FrIfCommunicationAction` must be executed before the frame transmission and the `FrIfCommunicationAction` is executed in the same communication cycle before the frame transmission. This means the job trigger's `FrIfMacrotick` parameter is lower than the frame trigger slot start time.

    OR

    ► The `FrIfCommunicationAction` must be executed after the frame transmission and the `FrIfCommunicationAction` is executed in the same communication cycle after the frame transmission. This means the job trigger's `FrIfMacrotick` parameter is higher than the frame trigger slot end time.

► The BC of a communication operation is set to the BC value of the frame trigger plus one if the `FrIfCommunicationAction` must be executed after the frame transmission and the `FrIfCommunicationAction` is executed in the same communication cycle before the frame transmission. This means the job trigger's `FrIfMacrotick` is lower than the slot end time.

► The BC of a communication operation is set to the BC value of the frame trigger minus one if the `FrIfCommunicationAction` must be executed before the frame transmission and the `FrIfCommunicationAction` is executed in the same communication cycle after the frame transmission. This means the job trigger's `FrIfMacrotick` is higher than the slot start time.

4.5.3. Using the FrIf Joblist Editor

1. Before you start the FrIf Joblist Editor, you need to load a `FrIf` configuration that contains a `FrIfCluster`.

2. To start the FrIf Joblist Editor, locate and press the tab **ECU** in the **Sidebar** view as shown in Figure 4.4, "Locate the FrIf Joblist Editor".
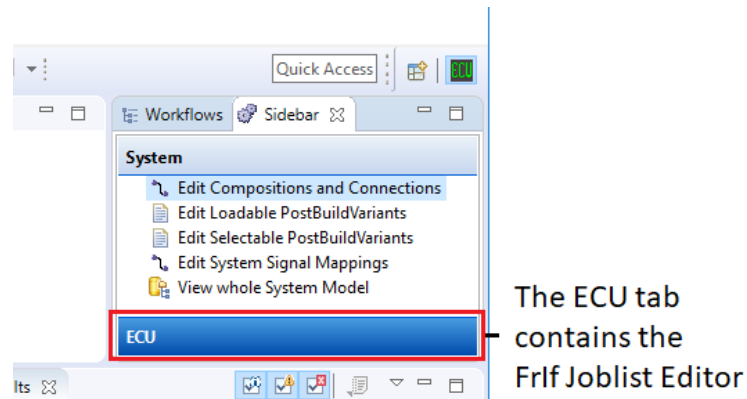
Figure 4.4. Locate the FrIf Joblist Editor

3. Double-click **Edit FrIf Joblist** in the **Sidebar** view as shown in Figure 4.5, "Start the FrIf Joblist Editor".
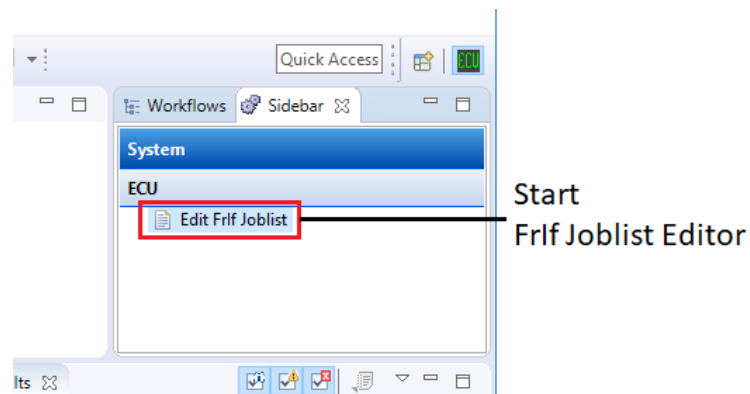
Figure 4.5. Start the FrIf Joblist Editor

4. Select a **FrIf Cluster** and press the **Next** button as shown in Figure 4.6, "Select a FrIf Cluster".
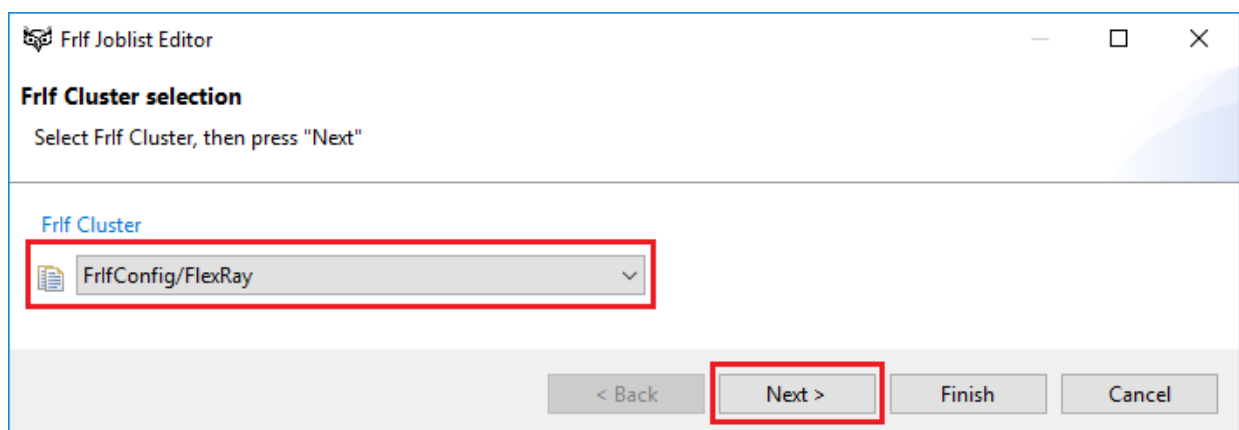
Figure 4.6. Select a FrIf Cluster

5. The FrIf Joblist Editor opens in a new window.
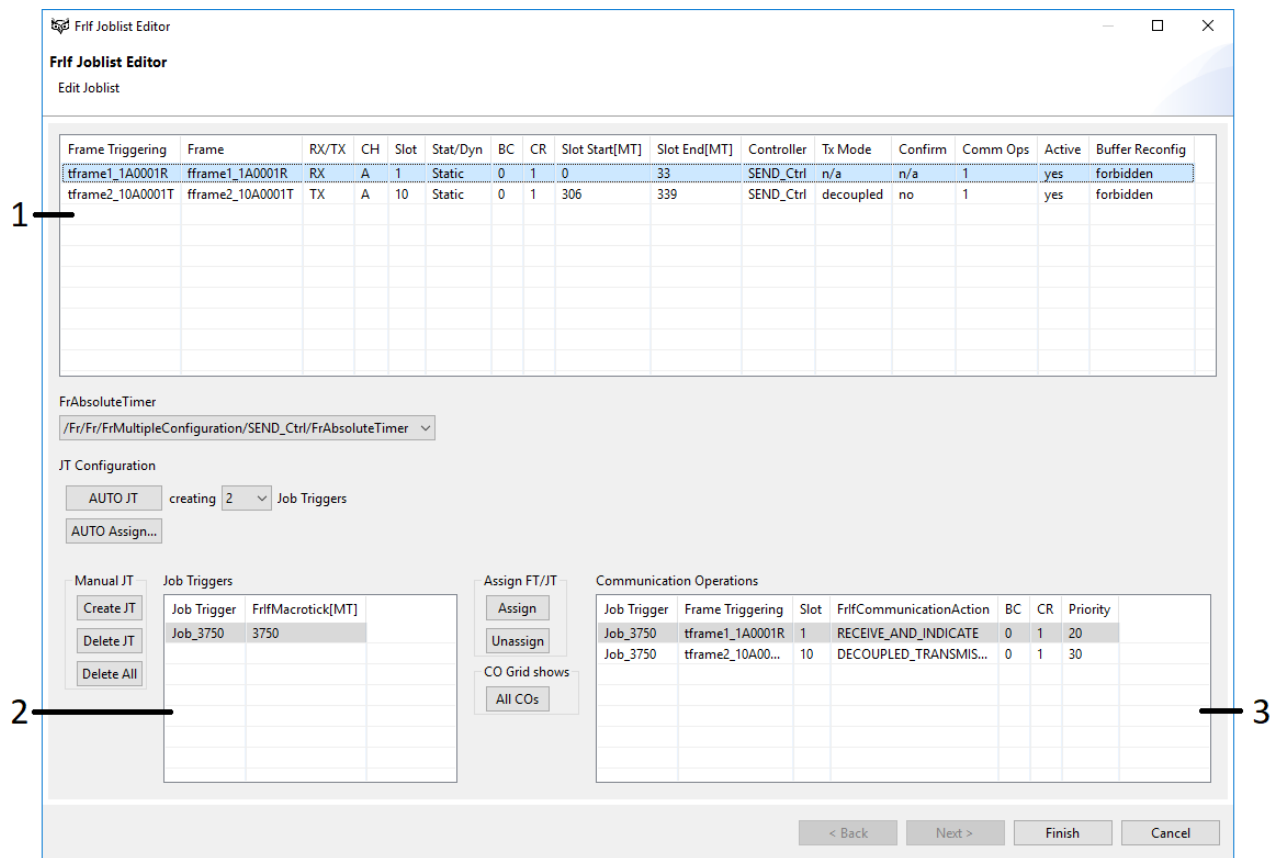
Figure 4.7. The FrIf Joblist Editor

Nr	Description
1	The Frame Triggering grid. The Frame Triggering grid displays all frame triggers which are either sent or received by the ECU. For a detailed description, see Section 4.5.3.1, "Frame Triggering grid".
2	The Job Triggers grid displays all job triggers that are defined for the `FrIfCluster`. For a detailed description, see Section 4.5.3.2, "Job Triggers grid".
3	The Communications Operations grid displays all communication operations that are executed if the checkbox **Filter COs of JT** is unchecked. For a detailed description, see Section 4.5.3.3, "Communication Operations grid".

Table 4.1. Overview of the FrIf Joblist Editor

4.5.3.1. Frame Triggering grid

The Frame Triggering grid at the top of the window shows all frame triggers which are either sent or received by the ECU. The columns of the Frame Triggering grid are either read-only or can be edited to influence the Auto JT algorithm.

Figure 4.8. Frame Triggering grid

Nr	Description
1	Name of the frame trigger.
2	Name of the frame (L-SDU) assigned to the frame trigger.
3	Indicates whether the frame trigger is sent (TX) or received (RX).
4	FlexRay channel (A, B or AB).
5	FlexRay slot number.
6	Indicates whether the FlexRay slot is a dynamic or a static one.
7	Base cycle parameter of the frame trigger.
8	Cycle repetition parameter of the frame trigger.
9	Starting time of the frame on the FlexRay bus. The starting time is provided in units of macroticks.
10	Ending time of the frame on the FlexRay bus. The ending time is provided in units of macroticks.
11	Communication controller that sends or receives the frame trigger.
12	This value is only defined for Tx frame triggers. The value indicates whether the frame trigger is sent in decoupled or in immediate mode.
13	This value is only defined for Tx frame triggers. The value indicates whether at least one PDU sent via this frame trigger has its `FrIfConfirm` flag set to true.
14	Indicates the number of communication operations that are currently assigned to the frame trigger.
15	Indicates whether the frame trigger is to be taken into account for the Simple/Advanced Auto JT algorithm. This column is editable. If the value is set to `no`, no communication operations are assigned to this frame trigger during Auto JT and no L-PDU object is created.
16	This column is editable. If you set the buffer sharing field to `allowed`, the Auto JT algorithm inserts additional `PREPARE_LPDU` communication operations. With these communication operations multiple frame triggers can be assigned to the same FlexRay communication controller buffer, thus allowing the FlexRay communication controller to process more frame triggers than buffers are

Nr	Description
	available. For details of the two fields, see Section 4.5.2.3, "Automatic creation and assignment of job triggers".

<div align="center">Table 4.2. Overview of the Frame Triggering grid</div>

NOTE	Timing of dynamic slots
(i)	Since no start and end time is known for dynamic slots, the beginning of the dynamic part and the end of the communication cycle are indicated as start and end time. The NIT (network idle time) is not taken into account.

TIP	Using multiple selection modification
💡	To change the **Active** and **Buffer Sharing** columns of multiple frame triggers, select the frame trigger, then set the field of one frame trigger to the desired value.

4.5.3.2. Job Triggers grid

The **Job Triggers** grid displays all job triggers that are defined for the `FrIfCluster`.

<div align="center">Figure 4.9. Job Triggers grid</div>

Nr	Description
1	Name of the job trigger.
2	Defines the offset from the beginning of the communication cycle to the point in time when the execution of the associated Communication Operations starts. The offset is defined in macroticks. This column is editable.

<div align="center">Table 4.3. Overview of the Job Triggers grid</div>

4.5.3.3. Communication Operations grid

The **Communication Operations** grid displays all communication operations. To display all communication operations, keep the checkbox **Filter COs of JT** unchecked. To display communication operations of the selected job triggers only, check the **Filter COs of JT**. The columns of the **Communication Operations** grid are either read-only or can be edited.

Figure 4.10. Communication Operations grid

Nr	Description
1	Name of the job trigger. This column is editable.
2	Name of the frame trigger.
3	FlexRay slot number.
4	Indicates the type of operation that is to be performed. This column is editable. For Tx frame triggers, the following operations are defined: ▶ `DECOUPLED_TRANSMISSION`: Transmission of an outgoing frame. ▶ `TX_CONFIRMATION`: Check whether an outgoing frame has been sent successfully; indicate this event to the upper modules of the PDUs contained with a turned on `FrIfConfirm` flag. ▶ `PREPARE_LPDU`: Prepare transmission of an L-PDU, usually by reconfiguring a transmit buffer shared by two or more Tx L-PDUs. ▶ `PREP_LPDU_DEC_TX`: Execution of one `PREPARE_LPDU` operation, immediately followed by a `DECOUPLED_TRANSMISSION` operation. For Rx frame triggers, the following operations are defined: ▶ `RECEIVE_AND_INDICATE`: Receive an incoming frame; indicate the reception to the upper modules of successfully received PDUs.

Nr	Description
	► `RECEIVE_AND_STORE`: Receive an incoming frame; store the payload of the successfully received PDUs in a FrIf buffer without indicating the reception to the owner modules.
	► `RX_INDICATION`: Indicate the reception of successfully received PDUs of a frame to the PDU owner modules. This operation must be preceded by a `RECEIVE_AND_STORE` operation.
	► `PREPARE_LPDU`: Prepare the reception of an L-PDU, usually by reconfiguring a receive buffer shared by two or more incoming L-PDUs.
	► `CONSUME_RXFIFO`: If defined in the module description .xdm file.
	Other `FrIfCommunicationAction` values as defined in the module description .xdm file are available for both Rx and Tx frame triggers.
5	`BC`: The fields BC (base cycle) and CR (cycle repetition) indicate the communication cycles during which the communication operation is to be performed. This column is editable. $$\text{Cycle Number} = (B + n * 2^R)_{mod64}$$ with exactly one tuple of values for `B` and 2^R, where: ► Base Cycle `B` is an element of the set [0 … 63] ► Cycle Repetition 2^R; R is an element of the set [0 … 6] ► Variable `n` = 0 … 63 ► $B < 2^R$
6	The fields CR (cycle repetition) and BC (base cycle) indicate the communication cycles during which the communication operation is to be performed.
7	You may use the priority to enforce a specific order of execution of communication operations within the same job. A lower value means a higher priority - thus earlier execution. If the execution order is not relevant, you may set two communication operations to the same priority.

Table 4.4. Overview of the Job Triggers grid

TIP **Using multiple selection modification**

To change the **FrIfCommunicationAction**, **BC**, **CR**, and **Priority** fields of multiple communication operations, select the communication operations, then set the field of one communication operation to the desired value.

4.5.4. Creating a job trigger

You may create job triggers and communication operations either automatically or manually.
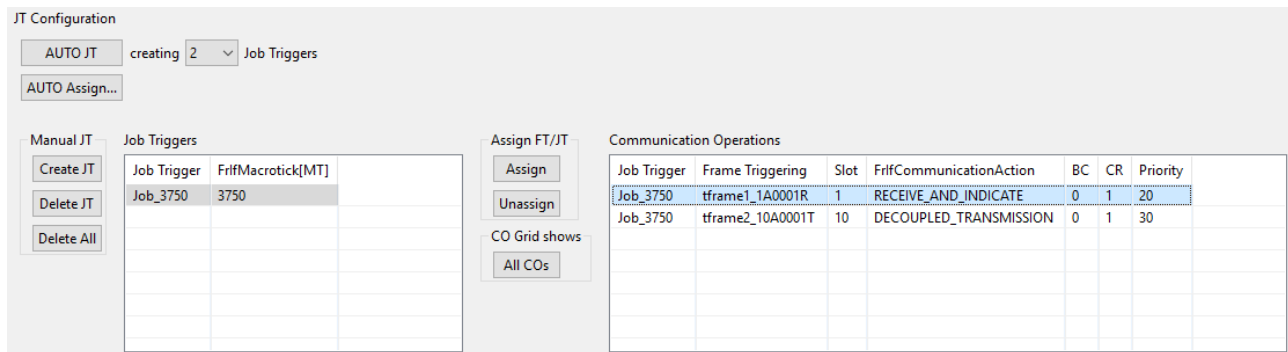
Figure 4.11. FrIf Joblist Editor buttons

Aim	How to
Automatically create job triggers	► To create job triggers for the currently selected MCU, press the **AUTO JT** button (stands for automatic job trigger). ► To create communication operations for all frame triggers and to assign them to job triggers, press **AUTO Assign...**. For a detailed description of the automatic job trigger, refer to Section 4.5.2.3, "Automatic creation and assignment of job triggers". For a detailed description of the Auto Assign FTs to Jobs dialog, refer to Section 4.5.5, "Auto Assign FTs to Jobs dialog".
Manually create job triggers	► To create a new job trigger, press the **Create JT** button. ► To remove the currently selected job trigger and its associated communication operations, press the **Delete JT** button.
Delete job triggers	► To delete all job triggers and the assigned communication operations, press the **Delete All** button.
Assign and remove communication operations	► To display only those communication operations that are assigned to the currently selected job trigger, check the checkbox **Filter COs of JT**. To display all communication operations, keep the checkbox unchecked. ► To create and assign one communication operation for the currently selected job trigger and for every frame trigger currently selected in the Frame Triggering grid, press the **Assign** button in the **Assign FT/JT** section. Selected as default `FrIfCommunicationAction` is: For Tx frame trigger `DECOUPLED_TRANSMISSION`, for Rx frame trigger `RECEIVE_AND_INDICATE`.

Aim	How to
	▶ To remove all currently selected communication operations from the assigned job trigger, press the **Unassign** button.
Configure a set of BC, CR parameter values automatically or manually for the communication operation	Depends upon: ▶ The direction of the frame trigger ▶ The BC, CR, and slot number parameters of the frame trigger ▶ The `FrIfMacrotick` parameter of the job trigger For a description on how the FrIf Joblist Editor automatically configures BC and CR, see section Section 4.5.2.3, "Automatic creation and assignment of job triggers".
Write job list to data model or discard job list configuration	▶ To write the `FrIfJobList` job list back to into the data model of EB tresos Studio, press the **OK** button. ▶ To discard the currently configured joblist configuration, press **Cancel**.

Table 4.5. Steps to create a job trigger

4.5.5. Auto Assign FTs to Jobs dialog

In this dialog, you can select for each job whether it should process Rx frame triggers, Tx frame triggers, or both. Moreover, you can define a *Redzone* for each job trigger to reduce buffer access latencies. Section 4.5.2.3.2.3, "Assignment of communication operations to job triggers using Redzones" describes this concept in detail.
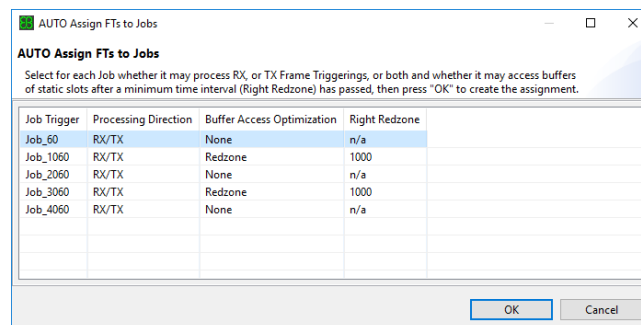
Figure 4.12. The Auto Assign FTs to Jobs dialog

To select the processing direction of a job trigger, click the cell in the **Processing Direction** column. To start the auto assignment, click OK.

4.6. Extended RxFifo support in FrIf and Fr

NOTE	**This feature requires a special license**
$\mathbf{i}$	This optimization feature only works with an EB Fr driver for Freescale FlexRay controllers. It requires the license FR_RXFIFO_OPT to be installed in EB tresos Studio

4.6.1. Overview

FlexRay communication is performed by hardware devices called FlexRay communication controllers. The number of FlexRay message buffers available on a FlexRay communication controller limits the number of messages that a particular FlexRay communication controller can receive and transmit. Therefore, the number of message buffers is a scarce resource which limits the possible FlexRay communication schedule.

The extended RxFifo overcomes this message buffer limitation for receive messages. Receive messages assigned to the extended RxFifo do not occupy message buffers, thus reducing the total number of required message buffers or increasing the total number of FlexRay messages a FlexRay controller is able to process.
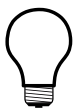
TIP	**Use the extended RxFifo feature if there are insufficient message buffers on the FlexRay communication controller**
	If you run out of FlexRay message buffers during `Fr` module configuration generation, use the extended RxFifo feature to optimize the use of receive buffers.

▶ Section 4.6.2, "Background information" describes more details of the functionality of this feature.

▶ Section 4.6.3, "Configuration of extended RxFifo Support in `FrIf` and EB `Fr` driver" explains the configuration parameters which must be configured to use this feature.

4.6.2. Background information

This feature extends the functionality and the API interface of AUTOSAR modules `Fr` and `FrIf` with the following EB specific extensions:

▶ `Fr` is extended by an API function `Fr_ReceiveRxFIFO()` which removes the oldest element from a FlexRay controller's receive hardware FIFO. This API function returns frame-ID, cycle, channel, length, status information and payload of the received FlexRay message.

▶ `FrIf` is extended to call `Fr_ReceiveRxFIFO()` and map the returned message to a configured `FrIf-FrameTriggering` based on Frame-ID, channel and cycle information. If this mapping is successful, the PDUs received are indicated to the upper layers.

4.6.3. Configuration of extended RxFifo Support in `FrIf` and EB `Fr` driver

Enable usage of the extended RxFifo by setting the following configuration parameter values:

▶ Set configuration parameter `RxFIFOEnable` to `true` and set the configuration parameter `RxFIFOMode` to `FR_RXFIFO_COMPATIBLE`.

Add communication operations into `FrIfJob` with `FrIfCommunicationAction` set to `CONSUME_RXFIFO` and adapt configuration parameter `FrIfGlobalRxMaxLoop` accordingly.

▶ Set configuration parameter `RxFIFOMode` to `FR_RXFIFO_EXTENDED`.

Configure `FrFifo` configuration parameters to cover all related FlexRay receive messages.

4.7. Automatic creation of Job Triggers via GuidedConfigWizard

The functionality for automatic creation of Job Triggers like described in [Section 4.5.2.3, "Automatic creation and assignment of job triggers"](#) can also be executed via command line or Unattended Wizard in the EB tresos Studio.

4.7.1. Command Line

You must provide the given parameters to run the Unattended Wizard via the command line. For details, see the example:

```
tresos_cmd.bat -DFrAsMaxJTNumber=5 autoconfigure Demo FrAs.AutoConfigure
```

The Unattended Wizard for the project named `Demo` is executed and the maximum number of Job Triggers is set to 5.

`-DFrAsMaxJTNumber="<parameter>"`
    The parameter defines the maximum number of allowed Job Triggers. Please define a value in the range of [2..9].

`autoconfigure`
    This parameter indicates that an Unattended Wizard shall be executed.

`<project name>`
    The name of the project on which the Unattended Wizard shall perform its operation.

`FrAs.AutoConfigure`
    The ID of the Unattended Wizard.

4.7.2. Unattended Wizard in EB tresos Studio

To configure the maximum number of Job Triggers:

1.   Select the menu item **Unattended wizard configuration**.

2.   Select the sub item **Autoconfigure FrIf Joblist(FrAs.AutoConfigure)**.

3.   Configure the maximum number of Job Triggers in the window that appears, see Figure 4.14, "Configuring the maximum number of Job Triggers".

4.   Select the menu item **Autoconfigure FrIf Joblist(FrAs.AutoConfigure)** as depicted in Figure 4.13, "Configuring and Launching the Unattended Wizard" if you want to launch the automatic creation of Job Triggers for the selected project.
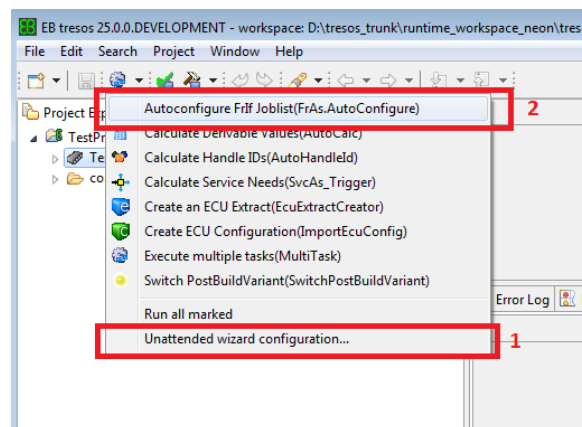
Figure 4.13. Configuring and Launching the Unattended Wizard
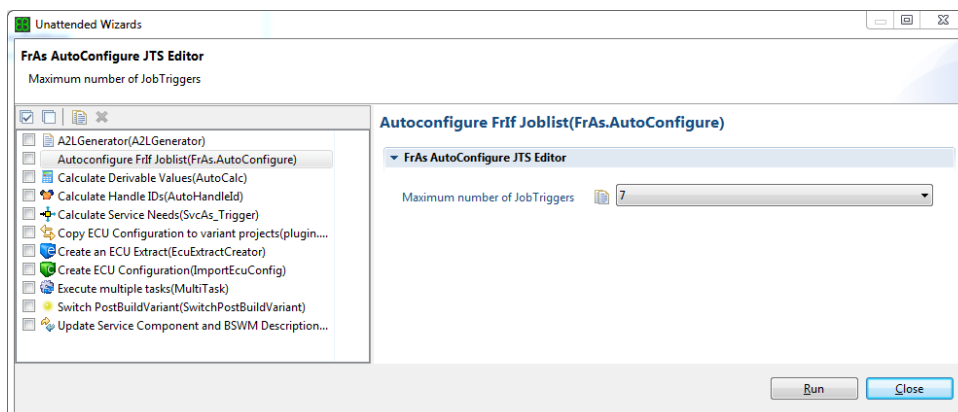
Figure 4.14. Configuring the maximum number of Job Triggers

5. ACG8 FlexRay Stack module references

5.1. Overview

This chapter provides module references for the ACG8 FlexRay Stack product modules. These include a detailed description of all configuration parameters. Furthermore this chapter lists the application programming interface with all data types, constants and functions.

The content of the sections is sorted alphabetically according the EB tresos AutoCore Generic module names.

For further information on the functional behavior of these modules, refer to the chapter ACG8 FlexRay Stack user's guide.

5.1.1. Notation in EB module references

EB notation may differ from the AUTOSAR standard notation in the software specification documents (SWS). This section describes the notation of *default value* and *range* fields in the EB module references.

5.1.1.1. Default value of configuration parameters

If there is no default value specified for a parameter, the default value field is omitted to prevent ambiguity with parameters that have `--` as default values.

Example: The parameter `BswMCompuConstText` of the `BswM` module of EB tresos AutoCore Generic 8 Mode Management has no default value field, therefore it is omitted.

5.1.1.2. Range information of configuration parameters

The range of a configuration parameter contains an upper and a lower boundary. However, in special cases the range of allowed values can be computed by means of an XPath function that is evaluated at configuration time. An XPath function can either be a standard `xpath:<function>()` or a custom `cxpath:<function>()` function. The range of a configuration parameter may be computed based on other configuration parameters that are referenced from the XPath function. For more information on custom XPath functions, see section *Custom XPath Functions API* of the EB tresos Studio developer's guide.

Example: The parameter `BswMCompuConstText` of the `BswM` module of EB tresos AutoCore Generic 8 Mode Management has the custom XPath function `cxpath:getCompuMethodsVT()` in the range field which provides the allowed values.

5.2. FrArTp

5.2.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
FrArTpDefensiveProgramming	1..1	**Label:** Defensive Programming Options Parameters for defensive programming
FrArTpGeneral	1..1	This container contains the general configuration (parameters) of the FlexRay TP.
FrArTpMultipleConfig	1..1	This container contains the configuration parameters and sub containers of the AUTOSAR FrArTp module.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

Parameters included	
Parameter name	**Multiplicity**
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild

Range	VariantPostBuild	
Configuration class	**VariantPostBuild:**	VariantPostBuild

5.2.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion	
Label	AUTOSAR Major Version	
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.	
Multiplicity	1..1	
Type	INTEGER_LABEL	
Default value	4	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

Parameter Name	ArMinorVersion
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL

Default value	2
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ArPatchVersion**
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMajorVersion**
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMinorVersion**
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwPatchVersion**

Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	9
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ModuleId**
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	38
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**VendorId**
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**Release**
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	**PublishedInformation:**

Origin	Elektrobit Automotive GmbH

5.2.1.2. FrArTpDefensiveProgramming

Parameters included	
Parameter name	**Multiplicity**
FrArTpDefProgEnabled	1..1
FrArTpPrecondAssertEnabled	1..1
FrArTpPostcondAssertEnabled	1..1
FrArTpStaticAssertEnabled	1..1
FrArTpUnreachAssertEnabled	1..1
FrArTpInvariantAssertEnabled	1..1

Parameter Name	**FrArTpDefProgEnabled**	
Label	Enable Defensive Programming	
Description	Enables or disables the defensive programming feature for the module FrArTp. Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows: 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrArTpPrecondAssertEnabled**
Label	Enable Precondition Assertions
Description	Enables handling of precondition assertion checks reported from the module FrArTp. Dependency on parameter(s):

	► Enable Development Error Detection (`FrArTpDevErrorDetect`): must be enabled
	► Enable Defensive Programming (`FrArTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrArTpPostcondAssertEnabled**
Label	Enable Postcondition Assertions
Description	Enables handling of postcondition assertion checks reported from the module FrArTp.
	Dependency on parameter(s):
	► Enable Development Error Detection (`FrArTpDevErrorDetect`): must be enabled
	► Enable Defensive Programming (`FrArTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrArTpStaticAssertEnabled**
Label	Enable Static Assertions
Description	Enables handling of static assertion checks reported from the module FrArTp.
	Dependency on parameter(s):
	► Enable Development Error Detection (`FrArTpDevErrorDetect`): must be enabled
	► Enable Defensive Programming (`FrArTpDefProgEnabled`): must be enabled
Multiplicity	1..1

Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrArTpUnreachAssertEnabled**
Label	Enable Unreachable Code Assertions
Description	Enables handling of unreachable code assertion checks reported from the module FrArTp. Dependency on parameter(s): ► Enable Development Error Detection (`FrArTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrArTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrArTpInvariantAssertEnabled**
Label	Enable Invariant Assertions
Description	Enables handling of invariant assertion checks reported from functions of the module FrArTp. Dependency on parameter(s): ► Enable Development Error Detection (`FrArTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrArTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild

Origin	Elektrobit Automotive GmbH

5.2.1.3. FrArTpGeneral

Parameters included	
Parameter name	**Multiplicity**
FrArTpDevErrorDetect	1..1
FrArTpHaveAckRt	1..1
FrArTpHaveGrpSeg	1..1
FrArTpHaveLm	1..1
FrArTpHaveTc	1..1
FrArTpMainFuncCycle	1..1
FrArTpVersionInfoApi	1..1
FrArTpRelocatablePbcfgEnable	1..1
FrArTpMaxConnections	1..1
FrArTpMaxActiveConnections	1..1
FrArTpMaxTxPdus	1..1
SupportLowLevelRouting	1..1
LowLevelRoutingPraefix	1..1

Parameter Name	FrArTpDevErrorDetect	
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpHaveAckRt
Description	Preprocessor switch for enabling the Acknowledgement and retry mechanisms. NOTE: This feature is not supported by the current implementation.
Multiplicity	1..1
Type	BOOLEAN
Default value	false

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpHaveGrpSeg	
Description	Preprocessor switch for enabling segmentation of 1:n messages. NOTE: This feature is not supported by the current implementation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpHaveLm	
Description	Preprocessor switch for enabling the mechanism for message longer than allowed by ISO 15765-2. NOTE: This feature is always active (FrArTpHaveLm = true) in the current implementation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpHaveTc	
Description	Preprocessor switch for enabling Transmit Cancellation and Receive Cancellation. If it is checked the preprocessor CANCEL_API will be generated with STD_ON, otherwise will be STD_OFF	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpMainFuncCycle	
Description	This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.	
Multiplicity	1..1	

Type	FLOAT
Range	<=1.024
	>=0.001
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrArTpVersionInfoApi**
Description	Preprocessor switch for enabling the Version info API.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrArTpRelocatablePbcfgEnable**
Description	Enables/disable support for relocatable postbuild configuration. ► True: Postbuild configuration relocatable in memory. ► False: Postbuild configuration not relocatable in memory.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrArTpMaxConnections**
Description	This parameter defines the absolute (i.e. among all channels) number of connections that can be configured.
Multiplicity	1..1
Type	INTEGER
Default value	10
Range	<=255
	>=1
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	Elektrobit Automotive GmbH	
Parameter Name	**FrArTpMaxActiveConnections**	
Description	This parameter defines the absolute (i.e. among all channels) number of connections that can be active at the same time.	
Multiplicity	1..1	
Type	INTEGER	
Default value	5	
Range	<=255	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrArTpMaxTxPdus**	
Description	This parameter defines the absolute (i.e. among all channels) number of transmit N-PDUs that can be configured.	
Multiplicity	1..1	
Type	INTEGER	
Default value	5	
Range	<=255	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**SupportLowLevelRouting**	
Description	This parameter defines wether low-level routing is activated (true) or deactivated (false).	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**LowLevelRoutingPraefix**
Description	This parameter defines the prefix of the low-level routing module.

Multiplicity	1..1	
Type	STRING	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.4. FrArTpMultipleConfig

Containers included		
Container name	**Multiplicity**	**Description**
FrArTpChannel	1..255	This container contains the configuration (parameters) of one FlexRay TP channel.

5.2.1.5. FrArTpChannel

Containers included		
Container name	**Multiplicity**	**Description**
FrArTpConnection	1..n	This container contains the configuration (parameters) of one FlexRay TP connection.
FrArTpPdu	1..n	Container to hold the PDU parameters.

Parameters included	
Parameter name	**Multiplicity**
FrArTpAckType	1..1
FrArTpAdrType	1..1
FrArTpConcurrentConnections	0..1
FrArTpGrpSeg	1..1
FrArTpLm	1..1
FrArTpMaxAr	1..1
FrArTpMaxAs	1..1
FrArTpMaxBs	1..1
FrArTpMaxRn	1..1
FrArTpMaxWft	1..1
FrArTpStMin	1..1

Parameters included	
FrArTpStMinGrpSeg	0..1
FrArTpTc	1..1
FrArTpTimeBr	1..1
FrArTpTimeCs	1..1
FrArTpTimeoutAr	1..1
FrArTpTimeoutAs	1..1
FrArTpTimeoutBs	1..1
FrArTpTimeoutCr	1..1

Parameter Name	FrArTpAckType	
Description	This parameter defines the type of acknowledgement which is used for the specific channel. NOTE: This feature is not supported by the current implementation.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	FRARTP_NO	
Range	FRARTP_NO	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpAdrType	
Description	This parameter states the addressing type this connection has. The meanings of the values are one byte and two byte.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	FRARTP_OB	
Range	FRARTP_OB	
	FRARTP_TB	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpConcurrentConnections
Description	This parameter defines the number of connections that can be active at the same time. If set to 0, all configured connections can be active at the same time.

Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpGrpSeg**	
Description	Here can be specified, whether segmentation within a 1:n connection is allowed or not. NOTE: This feature is not supported by the current implementation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpLm**	
Description	This specifies the maximum message length for the particular channel.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	FRARTP_ISO	
Range	FRARTP_ISO	
	FRARTP_L4G	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpMaxAr**	
Description	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs. NOTE: This feature is not supported by the current implementation.	
Multiplicity	1..1	
Type	INTEGER	

Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpMaxAs**	
Description	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs. NOTE: This feature is not supported by the current implementation.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpMaxBs**	
Description	This parameter defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpMaxRn**	
Description	This parameter defines the maximum number of retries (if retry is configured for the particular channel). NOTE: This feature is not supported by the current implementation.	
Multiplicity	1..1	

Type	INTEGER
Default value	0
Range	<=255
	>=0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrArTpMaxWft**
Description	This parameter defines the maximal number of wait frames to be sent for a pending connection.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=255
	>=0
Origin	AUTOSAR_ECUC

Parameter Name	**FrArTpStMin**
Description	This parameter defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100?s, 200?s .. 900?s, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Range	<=0.127
	>=0.0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrArTpStMinGrpSeg**
Description	This parameter defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100?s, 200?s ... 900?s, 1ms, 2ms .. 127ms. The value can be changed at runtime using

	the FrArTp_ChangeParameter interface. NOTE: This feature is not supported by the current implementation.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Range	<=0.127	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTc**	
Description	With this switch Transmit Cancellation and Receive Cancellation can be turned on or off for this channel. NOTE: This feature is always active (FrArTpTc = true) in the current implementation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTimeBr**	
Description	This parameter defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Range	<=65.535	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTimeCs**
Description	This parameter defines the time in seconds between the sending of two consecutive CFs or between reception of an FC or AF and sending of the next CF .

Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Range	<=65.535	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTimeoutAr**	
Description	This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=65.535	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTimeoutAs**	
Description	This parameter states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=65.535	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTimeoutBs**

Description	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=65.535	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTimeoutCr**	
Description	This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=65.535	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.6. FrArTpConnection

Containers included		
Container name	**Multiplicity**	**Description**
FrArTpRxSdu	0..1	Describes the Rx SDU
FrArTpTxSdu	0..1	Describes the Tx SDU

Parameters included	
Parameter name	**Multiplicity**
FrArTpConPrioPdus	0..1
FrArTpLa	1..1

Parameters included	
FrArTpMultRec	1..1
FrArTpRa	1..1

Parameter Name	FrArTpConPrioPdus	
Description	This parameter defines the number of TxNPdus to which this connection has prioritized access. It must be ensured that the number of prioritized PDUs of all connections is smaller than the total number of TxNPdus in the associated PDU pool. NOTE: This feature is not supported by the current implementation.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpLa	
Description	This parameter defines the Local Address for the respective connection.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpMultRec	
Description	This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	FrArTpRa	
Description	This parameter defines the Remote Address for the respective connection.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.7. FrArTpRxSdu

Parameters included	
Parameter name	**Multiplicity**
FrArTpSduRxId	1..1
FrArTpRxSduRef	1..1

Parameter Name	FrArTpSduRxId	
Description	This is a unique identifier for a received message. This Id is used in the Cancel-Receive and ChangeParameter API call.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrArTpRxSduRef	
Description	Reference to a PDU in the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.8. FrArTpTxSdu

Parameters included	
Parameter name	**Multiplicity**
FrArTpSduTxId	1..1
FrArTpTxSduRef	1..1

Parameter Name	**FrArTpSduTxId**	
Description	This is a unique identifier for a received or a to be transmitted message. With this (and by means of e.g. a lookup table) the PDU Router can route the message appropriately without dealing with the particularities of the Transport Layer. This parameter can also be seen as the identifier of a connection.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpTxSduRef**	
Description	Reference to a PDU in the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.9. FrArTpPdu

Parameters included	
Parameter name	**Multiplicity**
FrArTpPduDirection	1..1
FrArTpPduId	1..1
FrArTpPduRef	1..1

Parameter Name	**FrArTpPduDirection**

Description	This parameter defines the direction of the PDU.
Multiplicity	1..1
Type	ENUMERATION
Range	FRARTP_RX
	FRARTP_TX
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrArTpPduId**	
Description	This is the identifier of the FlexRay Interface PDUs (Fr N-PDU, Fr L-SDU) in which the Transport Layer Frames of this channel should be transmitted.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrArTpPduRef**	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.10. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	**PbcfgMSupport**
Label	PbcfgM support
Description	Specifies whether or not the FrArTp can use the PbcfgM module for post-build support.

Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.2.2. Application programming interface (API)

5.2.2.1. Macro constants

5.2.2.1.1. FRARTP_CANCELRECEIVE_SERVICE_ID

Purpose	Service identifiers for FrArTp_CancelReceive API function.
Value	0x08U

5.2.2.1.2. FRARTP_CANCELTRANSMIT_SERVICE_ID

Purpose	Service identifiers for FrArTp_CancelTransmit API function.
Value	0x03U

5.2.2.1.3. FRARTP_CHANGEPARAMETER_SERVICE_ID

Purpose	Service identifiers for FrArTp_ChangeParameter API.
Value	0x04U

5.2.2.1.4. FRARTP_DET_GENERIC_INSTANCE_ID

Purpose	
Value	0x0U

5.2.2.1.5. FRARTP_E_INIT_FAILED

Purpose	Det error Id FRARTP_E_INIT_FAILED.
Value	0x04U

5.2.2.1.6. FRARTP_E_INVALID_PARAMETER

Purpose	Det error Id FRARTP_E_INVALID_PARAMETER.
Value	0x05U

5.2.2.1.7. FRARTP_E_INVALID_PDU_SDU_ID

Purpose	Det error Id FRARTP_E_INVALID_PDU_SDU_ID.
Value	0x03U

5.2.2.1.8. FRARTP_E_PARAM_POINTER

Purpose	Det error Id FRARTP_E_PARAM_POINTER.
Value	0x02U

5.2.2.1.9. FRARTP_E_UNINIT

Purpose	Det error Id FRARTP_E_UNINIT.
Value	0x01U

5.2.2.1.10. FRARTP_GETVERSIONINFO_SERVICE_ID

Purpose	Service identifiers for FrArTp_GetVersionInfo API function.
Value	0x27U

5.2.2.1.11. FRARTP_INIT_SERVICE_ID

Purpose	Service identifiers for FrArTp_Init API function.
Value	0x00U

5.2.2.1.12. FRARTP_MAINFUNCTION_SERVICE_ID

Purpose	Service identifiers for FrArTp_MainFunction API function.
Value	0x10U

5.2.2.1.13. FRARTP_RXINDICATION_SERVICE_ID

Purpose	Service identifiers for FrArTp_RxIndication API function.
Value	0x42U

5.2.2.1.14. FRARTP_TRANSMIT_SERVICE_ID

Purpose	Service identifiers for FrArTp_Transmit API function.
Value	0x02U

5.2.2.1.15. FRARTP_TRIGGERTRANSMIT_SERVICE_ID

Purpose	Service identifiers for FrArTp_TriggerTransmit API function.
Value	0x41U

5.2.2.1.16. FRARTP_TXCONFIRMATION_SERVICE_ID

Purpose	Service identifiers for FrArTp_TxConfirmation API function.
Value	0x40U

5.2.2.2. Functions

5.2.2.2.1. FrArTp_CancelReceive

Purpose	By calling this API with the corresponding RxSduId the currently ongoing data reception is terminated immediately. When the function returns, no reception is in progress anymore with the given N-SDU identifier.
Synopsis	`Std_ReturnType` **`FrArTp_CancelReceive`** `( PduIdType FrArTpRxSduId );`
Service ID	0x08
Sync/Async	Synchronous

Reentrancy	Non Reentrant	
Parameters (in)	`FrArTpRxSduId`	SDU-Id of currently ongoing reception.
Return Value	E_OK Reception was terminated sucessfully. E_NOT_OK Reception was not terminated.	

5.2.2.2.2. FrArTp_CancelTransmit

Purpose	This service primitive is used to cancel the transmission of pending Fr N-SDUs. The connection is identified by FrArTpTxSduId.	
Synopsis	`Std_ReturnType` **`FrArTp_CancelTransmit`** `( PduIdType FrArTpTxSduId );`	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	`FrArTpTxSduId`	Contains the unique identifier of the transmitted Fr N-SDU which has to be canceled.
Return Value	E_OK Cancellation request of the transmission of the specified Fr N-SDU is accepted. E_NOT_OK Cancellation request of the transmission of the specified Fr N-SDU is rejected.	
Description	When the function returns, no transmission is in progress anymore with the given N-SDU identifier.	

5.2.2.2.3. FrArTp_ChangeParameter

Purpose	This service is used to request the change of the reception parameters STmin and BS for the specified N-SDU.	
Synopsis	`Std_ReturnType` **`FrArTp_ChangeParameter`** `( PduIdType id , TPParameterType parameter , uint16 value );`	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	`id`	Identifies the affected N-SDU.

	parameter	Specifies the parameter which shall be changed.
	value	The new value of the parameter.
Return Value	E_OK request is accepted E_NOT_OK request is not accepted	

5.2.2.2.4. FrArTp_GetVersionInfo

Purpose	Get version information of the FrArTp module.
Synopsis	void **FrArTp_GetVersionInfo** (Std_VersionInfoType * versioninfo);
Service ID	0x02
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (out)	versioninfo — Pointer to where to store the version information of this module.
Description	This service returns the version information of this module. The version information includes: ► Module Id ► Vendor Id ► Vendor specific version numbers

5.2.2.2.5. FrArTp_Init

Purpose	Initializes the FrArTp stack.
Synopsis	void **FrArTp_Init** (const FrArTp_ConfigType * ConfigPtr);
Service ID	0x01
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	ConfigPtr — Address of the post-build time configuration data of the FrArTp module.
Description	This service initializes the TCP/IP Stack. The call of this service is mandatory before using the FrArTp instance for further processing.

5.2.2.2.6. FrArTp_IsValidConfig

Purpose	Checks compatibility of the post-build-time configuration.	
Synopsis	`Std_ReturnType` **`FrArTp_IsValidConfig`** `( const void * voidConfigP-tr );`	
Service ID	N/A	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`voidConfigPtr`	Pointer to the configuration data of the FrArTp module.
Return Value	Result of compatibility check	
	`E_OK`	Provided configuration is compatible.
	`E_NOT_OK`	Provided configuration is notcompatible.
Description	This service checks the compatibility of the post-build-time configuration against the source code.	

5.2.2.2.7. FrArTp_MainFunction

Purpose	The main function for scheduling the TP (Entry point for scheduling).
Synopsis	`void` **`FrArTp_MainFunction`** `( void );`
Service ID	0x10

5.2.2.2.8. FrArTp_RxIndication

Purpose	Indication of a received I-PDU from a lower layer communication interface module.	
Synopsis	`void` **`FrArTp_RxIndication`** `( PduIdType RxPduId , PduInfoType * PduInfoPtr );`	
Service ID	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	`RxPduId`	ID of the received I-PDU.
	`PduInfoPtr`	Contains the length (SduLength) of the received I-PDU and a pointer to a buffer (SduDataPtr) containing the I-PDU.

5.2.2.2.9. FrArTp_Transmit

Purpose	Request the transmission of data.
Synopsis	`Std_ReturnType` **`FrArTp_Transmit`** `( PduIdType FrArTpTxSduId , const PduInfoType * FrArTpTxSduInfoPtr );`
Service ID	0x02
Sync/Async	Asynchronous
Reentrancy	Reentrant
Parameters (in)	`FrArTpTxSduId`	This parameter contains the unique identifier of the FrArTp N-SDU to be transmitted.
	`FrArTpTxSduInfoPtr`	Tx N-SDU Information Structure which contains a) pointer to the FrArTp Tx N-SDU b) the length of the FrArTp Tx N-SDU
Return Value	E_OK The request has been accepted E_NOT_OK The request has not been accepted, e. g. because the parameter check has failed (invalid N-SDU ID or size), the corresponding connection is still occupied, or no state machine is free (FrArTpConcurrentConnections).
Description	This service is utilized to request the transmission of data. This function has to be called with FrArTp's SDU-Id, i.e. the upper layer has to translate its own PDU-Id into the FrArTp's SDU-ID for the corresponding message. Within the provided FrArTpSduInfoPtr only SduLength is valid (no data)! If this function returns E_OK then there will arise an call of PduR_FrArTpCopyTxData in order to get data for sending

5.2.2.2.10. FrArTp_TriggerTransmit

Purpose	This function is called by the FlexRay Interface for sending out a FlexRay frame.
Synopsis	`Std_ReturnType` **`FrArTp_TriggerTransmit`** `( PduIdType TxPduId , PduInfoType * PduInfoPtr );`
Service ID	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 (out)	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.
Return Value	Std_ReturnType	
	E_OK	SDU has been copied and SduLength indicates the number of copied bytes.
	E_NOT_OK	No SDU 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.	

5.2.2.2.11. FrArTp_TxConfirmation

Purpose	The lower layer communication interface module confirms the transmission of an I-PDU.
Synopsis	void **FrArTp_TxConfirmation** (PduIdType TxPduId);
Service ID	0x40
Sync/Async	Synchronous
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.
Parameters (in)	TxPduId	ID of the I-PDU that has been transmitted.

5.2.3. Integration notes

5.2.3.1. Exclusive areas

This section describes the exclusive areas used by the `FrArTp` module.

5.2.3.1.1. SCHM_FRARTP_EXCLUSIVE_AREA_0

Protected data structures	All shared data that shall be protected from mutual access.
Recommended locking mechanism	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the `EB tresos AutoCore Generic documentation`. Refer to the section `Mapping exclusive areas in the basic software modules` in the `Integration notes` section for details.

5.2.3.2. Production errors

Production errors are not reported by the `FrArTp` module.

5.2.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
VAR_CLEARED_UNSPECIFIED
CODE
VAR_INIT_32
CONFIG_DATA_UNSPECIFIED
CONST_UNSPECIFIED
VAR_INIT_8
VAR_CLEARED_16
CONST_8
CONST_16
CONST_32
VAR_CLEARED_32
VAR_CLEARED_8
VAR_INIT_UNSPECIFIED

VAR_INIT_16

5.2.3.4. Integration requirements

WARNING	Integration requirements list is not exhaustive
⚠️	The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

5.2.3.4.1. lim.FrArTp.EB_INTREQ_FrArTp_0001

Description	The integrator must assure that the following functions do not interrupt each other or themselves: FrArTp_RxIndication, FrArTp_TxConfirmation, FrArTp_TriggerTransmit, FrArTp_MainFunction;
Rationale	This limitation reduces code size and execution time.

5.2.3.4.2. lim.FrArTp.EB_INTREQ_FrArTp_0002

Description	The integrator must assure that the following functions are not preempted by any other function: FrArTp_RxIndication, FrArTp_TxConfirmation, FrArTp_TriggerTransmit;
Rationale	This limitation reduces code size and execution time.

5.2.3.4.3. lim.FrArTp.EB_INTREQ_FrArTp_0003

Description	The integrator must assure that the following functions are not preempted and do not preempt any other function: FrArTp_Init;
Rationale	This limitation reduces code size and execution time.

5.2.3.4.4. lim.FrArTp.EB_INTREQ_FrArTp_0004

Description	The integrator must assure that only the last transmit PDU (in temporal order) of each PDU pool has a TxConfirmation in FrIf configured. It follows that all other PDUs must not have a TxConfirmation in FrIf configured. The temporal order corresponds to the FrArTpPduId, see lim.FrArTp.EB_INTREQ_FrArTp_0005.

Rationale	This limitation originates from the FrArTp SWS: SWS_FrArTp_00182.

5.2.3.4.5. lim.FrArTp.EB_INTREQ_FrArTp_0005

Description	The integrator must assure that the FrIf transmits N-PDUs of a pool in temporal order according to ascending FrArTpPduId from FrArTp. Additionally, all PDUs of a pool shall be arranged such that they are always received in the same order in which they have been transmitted, independent of the current cycle in the FlexRay communication round.
Rationale	This limitation originates from the FrArTp SWS: SWS_FrArTp_00174.

5.3. FrIf

5.3.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
FrIfConfig	1..1	Configuration of the FlexRay Interface. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.
FrIfDefensiveProgramming	1..1	**Label:** Defensive Programming Options Parameters for defensive programming
FrIfUserUpperLayerConfig	0..16	User upper layer configuration for FrIf. If you use an user upper layer above FrIf (other than PduR, FrNm, Iso-FrTp) you have to use these configuration parameters below this container. This means that if you want to use e.g. Xcp, you must add an FrIfUserUpperLayerConfig entry. **As soon as you use**

Containers included		
		FrIfUserUpperLayerConfig at all, then all upper modules you use (including the standard modules PduR, FrNm, FrTp) must be added to this FrIfUserUpperLayerConfig configuration. If a particular Pdu shall be assigned to a user specific upper FrIf module, then this must be configured at the configuration parameters **FrIfUpperLayerRef** and **FrIfUpperLayerHandleId** within the FrIfPdu configuration container.
FrIfGeneral	1..1	This container contains the general configuration parameters of the FlexRay Interface.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

Parameters included	
Parameter name	**Multiplicity**
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.3.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1

Parameters included	
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
Label	AUTOSAR Major Version
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ArMinorVersion
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	5
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	26
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL

Default value	61	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

Parameter Name	**VendorId**	
Label	Vendor ID	
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list	
Multiplicity	1..1	
Type	INTEGER_LABEL	
Default value	1	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

Parameter Name	**Release**	
Label	Release Information	
Multiplicity	1..1	
Type	STRING_LABEL	
Default value		
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

5.3.1.2. FrIfConfig

Containers included		
Container name	**Multiplicity**	**Description**
FrIfCluster	1..n	This container specifies a FrIf Cluster and all related data which is required to enable communication of the Cluster. A Cluster may consist of more than one Controller.
FrIfFrameStructure	0..n	The Frame structure specifies a Construction Plan how a Frame is assembled with PDUs and their respective Update-Bits.
FrIfPdu	0..n	Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.

5.3.1.3. FrIfCluster

Containers included		
Container name	**Multiplicity**	**Description**
FrIfClusterDemEventParame-terRefs	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.
FrIfController	1..n	This container contains the configuration of FlexRay CC.
FrIfJobList	0..1	This container specifies a list of all FlexRay Jobs of the Cluster to be performed by FrIf_JobListExec_<ClstIdx>().

Parameters included	
Parameter name	**Multiplicity**
FrIfClstIdx	1..1
FrIfDetectNITError	1..1
FrIfGChannels	1..1
FrIfGColdStartAttempts	1..1
FrIfGCycleCountMax	1..1
FrIfGListenNoise	1..1
FrIfGMacroPerCycle	1..1
FrIfGMaxWithoutClockCorrectFatal	1..1
FrIfGMaxWithoutClockCorrectPassive	1..1
FrIfGNetworkManagementVectorLength	1..1
FrIfGNumberOfMinislots	1..1
FrIfGNumberOfStaticSlots	1..1
FrIfGPayloadLengthStatic	1..1
FrIfGSyncFrameIDCountMax	1..1
FrIfGdActionPointOffset	1..1
FrIfGdBit	1..1
FrIfGdCasRxLowMax	1..1
FrIfGdCycle	1..1

Parameters included	
FrIfGdDynamicSlotIdlePhase	1..1
FrIfGdIgnoreAfterTx	1..1
FrIfGdMacrotick	1..1
FrIfGdMiniSlotActionPointOffset	1..1
FrIfGdMinislot	1..1
FrIfGdNit	1..1
FrIfGdSampleClockPeriod	1..1
FrIfGdStaticSlot	1..1
FrIfGdSymbolWindow	1..1
FrIfGdSymbolWindowActionPointOffset	1..1
FrIfGdTSSTransmitter	1..1
FrIfGdWakeupRxIdle	1..1
FrIfGdWakeupRxLow	1..1
FrIfGdWakeupRxWindow	1..1
FrIfGdWakeupTxActive	1..1
FrIfGdWakeupTxIdle	1..1
FrIfMainFunctionPeriod	1..1
FrIfMaxIsrDelay	1..1
FrIfSafetyMargin	1..1
FrIfGLastGuaranteedDynamicSlotId	1..1

Parameter Name	FrIfClstIdx	
Description	This parameter provides a zero-based consecutive index of the FlexRay Clusters. Upper layer BSW modules and the FrIf itself use this index to identify a FlexRay Cluster.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=63	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfDetectNITError	
Description	Indicates whether NIT error status of each cluster shall be detected or not. *Note: This functionality is achieved by config parameters FRIF_E_NIT_CH_A and FRIF_E_NIT_CH_B.*	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGChannels	
Description	The channels that are used by the cluster.Implementation Type: Fr_Channel-Type	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	FR_CHANNEL_AB	
Range	FR_CHANNEL_A	
	FR_CHANNEL_AB	
	FR_CHANNEL_B	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGColdStartAttempts	
Description	Maximum number of times a node in the cluster is permitted to attempt to start the cluster by initiating schedule synchronization.	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<=31	
	>=2	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGCycleCountMax

Description	Maximum cycle counter value in a given cluster.
	Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.
	Note: This configuration parameter is disabled since it is not required for hardware configuration.
Multiplicity	1..1
Type	INTEGER
Default value	63
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGListenNoise**
Description	Upper limit for the start up listen timeout and wake up listen timeout in the presence of noise. It is used as a multiplier of the node. parameter pdListenTimeout.
Multiplicity	1..1
Type	INTEGER
Default value	2
Range	<=16
	>=2
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGMacroPerCycle**
Description	Number of macroticks in a communication cycle.
	Note: Lower limit 10 for FlexRay Protocol 2.1 Rev. A compliance
Multiplicity	1..1
Type	INTEGER
Default value	2000
Range	<=16000
	>=10
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGMaxWithoutClockCorrectFatal**

Description	Threshold used for testing the vClockCorrectionFailed counter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state. [Even/odd cycle pairs].	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Range	<=15	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGMaxWithoutClockCorrectPassive**	
Description	Threshold used for testing the vClockCorrectionFailed counter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state. [Even/Odd cycle pairs].	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Range	<=15	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGNetworkManagementVectorLength**	
Description	Length of the Network Management vector in a cluster [bytes].	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Range	<=12	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGNumberOfMinislots	
Description	Number of minislots in the dynamic segment. Remark: Upper limit 7986 for FlexRay Protocol 2.1 Rev. A compliance	
Multiplicity	1..1	
Type	INTEGER	
Default value	97	
Range	<=7986	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGNumberOfStaticSlots	
Description	Number of static slots in the static segment.	
Multiplicity	1..1	
Type	INTEGER	
Default value	30	
Range	<=1023	
	>=2	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGPayloadLengthStatic	
Description	Payload length of a static frame [16 bit words].	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<=127	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGSyncFrameIDCountMax

Description	Maximum number of distinct syncframe identifiers present in a given cluster. Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.
Multiplicity	1..1
Type	INTEGER
Default value	15
Range	<=15
	>=2
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGdActionPointOffset**
Description	Number of macroticks the action point is offset from the beginning of a static slot.
Multiplicity	1..1
Type	INTEGER
Default value	6
Range	<=63
	>=1
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGdBit**
Description	Nominal bit time in seconds.
Multiplicity	1..1
Type	ENUMERATION
Default value	T100NS
Range	T100NS
	T200NS
	T400NS
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGdCasRxLowMax**

Description	Upper limit of the CAS acceptance windows [gdBit].
	Remark: Range 67 to 99 for FlexRay Protocol 2.1 Rev. A compliance
Multiplicity	1..1
Type	INTEGER
Default value	81
Range	<=99
	>=67
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGdCycle**	
Description	Length of the cycle, expressed in [s].	
	Remark: Lower limit 0.000024 for FlexRay Protocol 3.0 compliance.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.002	
Range	<=0.016	
	>=2.4E-5	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdDynamicSlotIdlePhase**	
Description	Duration of the idle phase within a dynamic slot [Minislots].	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=2	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdIgnoreAfterTx**

Description	Duration for which the bitstrobing is paused after transmission [gdBit].
	Remark: Set to 0 for FlexRay Protocol 2.1 Rev. A compliance.
	Note: This configuration parameter is disabled since it is not required for hardware configuration.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	**VariantPostBuild:** · VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGdMacrotick**	
Description	Duration of the cluster wide nominal macrotick, expressed in s.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.000001	
Range	<=6E-6	
	>=1E-6	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdMiniSlotActionPointOffset**	
Description	Number of Macroticks the Minislot action point is offset from the beginning of a Minislot [Macroticks].	
Multiplicity	1..1	
Type	INTEGER	
Default value	3	
Range	<=31	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdMinislot**

Description	Duration of a minislot [Macroticks].	
Multiplicity	1..1	
Type	INTEGER	
Default value	6	
Range	<=63	
	>=2	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdNit**	
Description	Duration of the Network Idle Time [Macroticks]. Remark: Upper limit 805 for FlexRay Protocol 2.1 Rev. A compliance.	
Multiplicity	1..1	
Type	INTEGER	
Default value	125	
Range	<=805	
	>=2	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdSampleClockPeriod**	
Description	Sample clock period.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	T12_5NS	
Range	T12_5NS	
	T25NS	
	T50NS	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdStaticSlot**	

Description	Duration of a static slot [Macroticks].
	Remark: Range 4-661 for FlexRay Protocol 2.1 Rev. A compliance.
Multiplicity	1..1
Type	INTEGER
Default value	43
Range	<=661
	>=4
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGdSymbolWindow**	
Description	Duration of the symbol window [Macroticks].	
	Remark: Range 0-142 for FlexRay Protocol 2.1 Rev. A compliance.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=142	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdSymbolWindowActionPointOffset**	
Description	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].	
	Remark: Set to GdActionPointOffset for FlexRay Protocol 2.1 Rev. A compliance.	
	Note: This configuration parameter is disabled since it is not required for hardware configuration.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	FrIfGdTSSTransmitter	
Description	Number of bits in the Transmission Start Sequence [gdBits]. Remark: Lower limit 3 for FlexRay Protocol 2.1 Rev. A compliance.	
Multiplicity	1..1	
Type	INTEGER	
Default value	6	
Range	<=15	
	>=3	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGdWakeupRxIdle	
Description	Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle. Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance.	
Multiplicity	1..1	
Type	INTEGER	
Default value	59	
Range	<=59	
	>=14	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGdWakeupRxLow
Description	Number of bits used by the node to test the duration of the LOW phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow. Lower limit 11 for FlexRay Protocol 2.1 Rev. A compliance.

Multiplicity	1..1	
Type	INTEGER	
Default value	55	
Range	<=59	
	>=11	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdWakeupRxWindow**	
Description	The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow. Upper limit 301 for FlexRay Protocol 2.1 Rev. A compliance.	
Multiplicity	1..1	
Type	INTEGER	
Default value	301	
Range	<=301	
	>=76	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGdWakeupTxActive**	
Description	Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxLow.	
Multiplicity	1..1	
Type	INTEGER	
Default value	60	
Range	<=60	
	>=15	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGdWakeupTxIdle	
Description	Number of bits used by the node to transmit the 'idle' part of a wakeup symbol [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxIdle.	
Multiplicity	1..1	
Type	INTEGER	
Default value	180	
Range	<=180	
	>=45	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfMainFunctionPeriod	
Description	The execution cycle of the FrIf_MainFunction_<cluster>() in seconds. The BSW scheduler uses this information in order to plan its tasks. The FrIf uses this information to perform the execution monitoring of the FrIf_JobListExec() from within the FrIf_MainFunction().	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.005	
Range	<=1	
	>=0.0001	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfMaxIsrDelay	
Description	The maximum delay in macroticks the FrIf_JoblistExec_<cluster>() function is processed after the absolute timer interrupt was triggered.	
Multiplicity	1..1	
Type	INTEGER	
Default value	100	
Range	<=10240000	
	>=0	

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfSafetyMargin	
Description	Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has be resynchronized.	
Multiplicity	1..1	
Type	INTEGER	
Default value	100	
Range	<=16000	
	>=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGLastGuaranteedDynamicSlotId	
Description	If DynamicGuaranteedTxReconfigMsgBufOptEnable is set to true the last guaranteed Slot Id in dynamic segment has to be set. Each Tx frame with a Slot Id higher than FrIfGNumberOfStaticSlots and lower or equal than FrIfGLastGuaranteedDynamicSlotId will be considered for dynamic buffer reconfiguration *Note: It has to be ensured by the OEM that all Tx frames within the quaranteed dynamic segment can be sent in every cycle under any circumstances.*	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.4. FrIfClusterDemEventParameterRefs

Parameters included	
Parameter name	Multiplicity
FRIF_E_ACS_CH_A	0..1

Parameters included	
FRIF_E_ACS_CH_B	0..1
FRIF_E_NIT_CH_A	0..1
FRIF_E_NIT_CH_B	0..1
FRIF_E_SW_CH_A	0..1
FRIF_E_SW_CH_B	0..1
FRIF_E_JLE_SYNC	0..1

Parameter Name	FRIF_E_ACS_CH_A	
Description	Reference to the DemEventParameter that shall be issued when an error in ACS on channel A is detected. Dependency on parameter(s): ▶ `FrIfAggregatedStatusAReportToDem`: Select DEM to enable the reporting of `FRIF_E_ACS_CH_A`. Further notes: ▶ Activation: This error is reported if an aggregated channel error on channel A is detected. ▶ Healing: This error is healed as soon as no aggregated channel error on channel A is detected. ▶ Trigger debounce: None. The error is reported on first occurrence. ▶ Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FRIF_E_ACS_CH_B
Description	Reference to the DemEventParameter that shall be issued when an error in ACS on channel B is detected. Dependency on parameter(s): ▶ `FrIfAggregatedStatusBReportToDem`: Select DEM to enable the reporting of `FRIF_E_ACS_CH_B`. Further notes:

	► Activation: This error is reported if an aggregated channel error on channel B is detected.
	► Healing: This error is healed as soon as no aggregated channel error on channel B is detected.
	► Trigger debounce: None. The error is reported on first occurrence.
	► Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**PostBuild:** · VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FRIF_E_NIT_CH_A**
Description	Reference to the DemEventParameter that shall be issued when an error in NIT on channel A is detected.
	Dependency on parameter(s):
	► `FrIfNITStatusAReportToDem`: Select DEM to enable the reporting of `FRIF_E_NIT_CH_A`.
	Further notes:
	► Activation: This error is reported if a network idle time error on channel A is detected.
	► Healing: This error is healed as soon as no network idle time error on channel A is detected.
	► Trigger debounce: None. The error is reported on first occurrence.
	► Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**PostBuild:** · VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FRIF_E_NIT_CH_B**
Description	Reference to the DemEventParameter that shall be issued when an error in NIT on channel B is detected.
	Dependency on parameter(s):

	► `FrIfNITStatusBReportToDem`: Select DEM to enable the reporting of `FRIF_E_NIT_CH_B`. Further notes: ► Activation: This error is reported if a network idle time error on channel B is detected. ► Healing: This error is healed as soon as no network idle time error on channel B is detected. ► Trigger debounce: None. The error is reported on first occurrence. ► Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FRIF_E_SW_CH_A**
Description	Reference to the DemEventParameter that shall be issued when an error in SW on channel A is detected. Dependency on parameter(s): ► `FrIfSymbolWindowStatusAReportToDem`: Select DEM to enable the reporting of `FRIF_E_SW_CH_A`. Further notes: ► Activation: This error is reported if a symbol window error on channel A is detected. ► Healing: This error is healed as soon as no symbol window error on channel A is detected. ► Trigger debounce: None. The error is reported on first occurrence. ► Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FRIF_E_SW_CH_B**

Description	Reference to the DemEventParameter that shall be issued when an error in SW on channel B is detected. Dependency on parameter(s): ► `FrIfSymbolWindowStatusBReportToDem`: Select DEM to enable the reporting of `FRIF_E_SW_CH_B`. Further notes: ► Activation: This error is reported if a symbol window error on channel B is detected. ► Healing: This error is healed as soon as no symbol window error on channel B is detected. ► Trigger debounce: None. The error is reported on first occurrence. ► Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**PostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FRIF_E_JLE_SYNC**
Description	Reference to the DemEventParameter that shall be issued when the following error occurred: Job List Execution lost synchronization to the FlexRay Global Time. Dependency on parameter(s): ► `FrIfJoblistSyncReportToDem`: Select DEM to enable the reporting of `FRIF_E_JLE_SYNC`. Further notes: ► Activation: This error is reported if the job list is not executed properly in time any more. ► Healing: This error is healed as soon as the joblist is executed properly again. ► Trigger debounce: None. The error is reported on first occurrence. ► Rate of diagnostic checks: Checked on every FrIf_MainFunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE

Configuration class	PostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.5. FrIfController

Containers included		
Container name	**Multiplicity**	**Description**
FrIfFrameTriggering	0..n	A Frame triggering contains the communication parameters of the FlexRay Frame as well as a reference to the Frame Construction Plan.
FrIfLPdu	0..n	Reference to a L-PDU index.
FrIfTransceiver	0..2	Up to two FlexRay Transceivers may connect a Controller to a Cluster. This container realizes a Controller-Transceiver assignment.

Parameters included	
Parameter name	**Multiplicity**
FrIfCtrlIdx	1..1
FrIfFrCtrlRef	1..1

Parameter Name	FrIfCtrlIdx	
Description	This parameter provides a zero-based consecutive index of the FlexRay Communication Controllers. Upper layer BSW modules and the FrIf itself use this index to identify a FlexRay CC.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=31	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfFrCtrlRef
Description	Reference to a Controller, which is handled by a specific Driver. This reference is unique for the ECU.

Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.6. FrIfFrameTriggering

Containers included		
Container name	**Multiplicity**	**Description**
FrIfFrameTriggeringDemEventParameterRefs	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.

Parameters included	
Parameter name	**Multiplicity**
FrIfAllowDynamicLSduLength	1..1
FrIfAlwaysTransmit	1..1
FrIfBaseCycle	1..1
FrIfChannel	1..1
FrIfCycleRepetition	1..1
FrIfLSduLength	1..1
FrIfMessageId	0..1
FrIfPayloadPreamble	1..1
FrIfSlotId	1..1
FrIfFrameStructureRef	1..1
FrIfNoneMode	1..1

Parameter Name	FrIfAllowDynamicLSduLength
Description	Allows L-PDU length reduction ('FrIfLSduLength' defines max. length) and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU.

Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfAlwaysTransmit**	
Description	Defines wether the driver's API function Fr_TransmitTxLPdu() shall always be called for this L-PDU.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfBaseCycle**	
Description	This parameter contains the FlexRay Base Cycle used to transmit this FlexRay Frame.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=63	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfChannel**	
Description	This parameter contains the FlexRay Channel used to transmit this FlexRay Frame.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	FRIF_CHANNEL_A	
	FRIF_CHANNEL_AB	
	FRIF_CHANNEL_B	

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfCycleRepetition	
Description	This parameter contains the FlexRay Cycle Repetition used to transmit this FlexRay Frame..possible Values: 1,2,4,8,16,32,64	
Multiplicity	1..1	
Type	INTEGER	
Range	<=64	
	>=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfLSduLength	
Description	The payload length of the Frame is given here. This parameter is required for validation if configured PDUs and update information fits into the Frame at configuration time [bytes].	
Multiplicity	1..1	
Type	INTEGER	
Range	<=254	
	>=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfMessageId	
Description	The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID. *Note: This feature is currently not supported.*	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	PostBuild:	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	**FrIfPayloadPreamble**	
Description	Switching the Payload Preamble bit.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfSlotId**	
Description	This parameter contains the FlexRay Slot ID used to transmit this FlexRay Frame.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfFrameStructureRef**	
Description	Reference to the Construction Plan of the FlexRay Frame.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfNoneMode**	
Description	If FrIfNoneMode is set to "true", the FlexRay Interface fetches Data from the upper layer for transmission even if FrIf_Transmit() was not called before. All Pdus contained in the FrameTriggering are affected by this FrameTriggering. *Note:This behaviour works only for decoupled transmission.*	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.7. FrIfFrameTriggeringDemEventParameterRefs

Parameters included	
Parameter name	**Multiplicity**
FrIfDemFTSlotStatusRef	0..1

Parameter Name	FrIfDemFTSlotStatusRef	
Description	Reference to DEM event Id that is reported when FlexRay driver module detects slot errors. If this parameter is not configured, no event reporting happens. Note: This parameter is not supported in the current implementation.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.8. FrIfLPdu

Parameters included	
Parameter name	**Multiplicity**
FrIfLPduIdx	1..1
FrIfReconfigurable	1..1
FrIfVBTriggeringRef	1..1

Parameter Name	FrIfLPduIdx	
Description	This parameter identifies the L-PDU in the interaction between FlexRay Interface and FlexRay Driver.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=4095	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfReconfigurable	

Description	This parameter specifies that this LPdu is reconfigurable using FrIf_ReconfigLP-du. This means that this LPdu can be assigned to a different FrameTriggering at runtime. However, this reconfiguration is limited by hardware constraints. The direction of the LPdu cannot be reconfigured.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfVBTriggeringRef**	
Description	Reference to the assigned Frame triggering.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.9. FrIfTransceiver

Parameters included	
Parameter name	**Multiplicity**
[FrIfClusterChannel](#)	1..1
[FrIfFrTrcvChannelRef](#)	1..1

Parameter Name	**FrIfClusterChannel**	
Description	This parameter identifies to which one of the two Channels (A, B, A and B) of the Cluster the Transceiver is connected. FrIfClusterChannel shall map to Fr_ChannelType: FRIF_CHANNEL_A == FR_CHANNEL_A FRIF_CHANNEL_B == FR_-CHANNEL_B FR_CHANNEL_AB shall not be used.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	FRIF_CHANNEL_A	
	FRIF_CHANNEL_B	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	**FrIfFrTrcvChannelRef**	
Description	Reference to a Transceiver Driver Channel. This reference is unique for the ECU.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.10. FrIfJobList

Containers included		
Container name	**Multiplicity**	**Description**
FrIfJob	1..n	A job may contain more than one operation that are executed at a specific point in time.

Parameters included	
Parameter name	**Multiplicity**
FrIfAbsTimerRef	1..1

Parameter Name	**FrIfAbsTimerRef**	
Description	Reference to the absolute timer to be used to trigger the interrupt whose ISR contains the FrIf_JobListExec_<ClstIdx>() function.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.11. FrIfJob

Containers included		
Container name	**Multiplicity**	**Description**
FrIfCommunicationOperation	1..n	A separate operation which is part of a FlexRay Job and defines what type of action is executed.

Parameters included	
Parameter name	**Multiplicity**
FrIfCycle	1..1
FrIfMacrotick	1..1

Parameter Name	**FrIfCycle**	
Description	The FlexRay Cycle in which the communication operation will execute this job.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=63	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfMacrotick**	
Description	Macrotick offset in the Cycle [Macrotick].	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=10240000	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.12. FrIfCommunicationOperation

Parameters included	
Parameter name	**Multiplicity**
FrIfCommunicationAction	1..1
FrIfCommunicationOperationIdx	1..1
FrIfRxComOpMaxLoop	1..1
FrIfLPduIdxRef	1..1

Parameter Name	FrIfCommunicationAction
Description	The type of operation to be performed in this communication operation. Choices: ▶ DECOUPLED_TRANSMISSION = Fetch data for I-PDUs contained in the L-PDU and transmit the L-PDU. ▶ RECEIVE_AND_INDICATE = Read a L-PDU from the Fr and indicate contained I-PDUs to the upper layer. ▶ RECEIVE_AND_STORE = Read a L-PDU from the Fr and store it in a local buffer. ▶ RX_INDICATION = Read a L-PDU out of a local cluster and indicate contained I-PDUs to the upper layer. ▶ TX_CONFIRMATION = Check if a previously transmitted L-PDU and provide an indication to upper layer. ▶ PREPARE_LPDU = Prepare for transmission/reception of a L-PDU. This enables the Fr to share a singele message buffer over multiple L-PDUs. ▶ CONSUME_RXFIFO = Consume (at maximum FrIfGlobalRxMaxLoop) LP-dus from the extended FlexRay driver RxFIFO (EB extension) and indicate expected Pdus to the upper layers.
Multiplicity	1..1
Type	ENUMERATION
Default value	DECOUPLED_TRANSMISSION
Range	DECOUPLED_TRANSMISSION
	PREPARE_LPDU
	RECEIVE_AND_INDICATE
	RECEIVE_AND_STORE
	RX_INDICATION
	TX_CONFIRMATION
	CONSUME_RXFIFO
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrIfCommunicationOperationIdx
Description	For each FlexRay Communication Job, this index spans a range of zero-based consecutive values and thus defines the order of the FlexRay Communication Operation in the respective FlexRay Communication Job.

Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfRxComOpMaxLoop**	
Description	Defines the maximum number of loops for the receive RECEIVE_AND_INDI-CATE (Use case: emptying a FIFO). *Note: This configuration parameter is not used. The maximum number of loops for receiving operations is defined with the global configuration parameter FrIf-GlobalRxMaxLoop.*	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfLPduIdxRef**	
Description	Reference to a L-PDu index.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.13. FrIfFrameStructure

Containers included		
Container name	**Multiplicity**	**Description**
FrIfPdusInFrame	1..n	This container holds all the information about a PDU in a FlexRay Frame.

Parameters included	
Parameter name	**Multiplicity**
FrIfByteOrder	1..1

Parameter Name	FrIfByteOrder	
Description	This parameter defines the ByteOrder of all Pdus that are mapped into the Frame.The absolute position of a Pdu in the Frame is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the FrIfPduOffset indicates the position of the most significant bit in the Frame. If LITTLE_ENDIAN is specified, the FrIfPduOffset indicates the position of the least significant bit in the Frame. *This configuration parameter is not used. The byte position is given in units of bytes, thus is distinct and needs no further information. The position of the Update-bit is always interpreted as BIG_ENDIAN notation.*	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	BIG_ENDIAN	
Range	BIG_ENDIAN	
	LITTLE_ENDIAN	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.14. FrIfPdusInFrame

Parameters included	
Parameter name	**Multiplicity**
FrIfPduOffset	1..1
FrIfPduUpdateBitOffset	0..1
FrIfPduRef	1..1

Parameter Name	FrIfPduOffset
Description	The value specifies the offset of the PDU within the Frame [bytes].
Multiplicity	1..1
Type	INTEGER
Default value	0

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfPduUpdateBitOffset	
Description	This value specifies where the PDU's Update-Bit is stored in the Frame (bit location of PDU's Update-Bit in the FlexRay Frame).	
Multiplicity	0..1	
Type	INTEGER	
Range	<=2031	
	>=0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfPduRef	
Description	This is the reference to the local definition of a PDU.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.15. FrIfPdu

Containers included		
Container name	Multiplicity	Description
FrIfPduDirection	1..1	A PDU is either transmit or receive.
VendorSpecific	0..1	EB Vendor specific parameters for FrIfPdu.

5.3.1.16. FrIfPduDirection

Containers included		
Container name	Multiplicity	Description
FrIfRxPdu	1..1	Receive PDU.
FrIfTxPdu	1..1	This container specifies transmission PDUs.

5.3.1.17. FrIfRxPdu

Parameters included	
Parameter name	**Multiplicity**
[FrIfRxIndicationName](#)	0..1
[FrIfUserRxIndicationUL](#)	1..1
[FrIfRxPduRef](#)	1..1

Parameter Name	FrIfRxIndicationName	
Description	This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter FRIF_USERRXINDICATION_UL. If FRIF_USER-RXINDICATION_UL equals FR_TP, FR_NM, PDUR or XCP, the name of the <User_RxIndication> is fixed. If FRIF_USERRXINDICATION_UL equals CDD, the name of the <User_RxIndication> is selectable. **This configuration parameter is currently not used. Please see configuration container FrIfUserUpperLayerConfig instead.**	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfUserRxIndicationUL		
Description	This parameter defines the upper layer (UL) module to which the indication of the successfully received FRIFRXPDU has to be routed via <User_RxIndica-tion>. This <User_RxIndication> has to be invoked when the indication of the configured FRIFRXPDU will be received by a Rx indication event from the FR Driver module. If no upper layer (UL) module is configured, no <User_RxIndica-tion> has to be called in case of a Rx indication event of the FRIFRXPDU from the FR Driver module. **This configuration parameter is currently not used. Please see configuration container FrIfUserUpperLayerConfig instead.**		
Multiplicity	1..1		
Type	ENUMERATION		
Default value	PDUR		
Range	CDD		
	FR_NM		
	FR_TP		
	PDUR		

	XCP	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfRxPduRef**	
Description	Reference to the external PDU definition.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.18. FrIfTxPdu

Parameters included	
Parameter name	**Multiplicity**
FrIfConfirm	1..1
FrIfCounterLimit	1..1
FrIfImmediate	1..1
FrIfNoneMode	1..1
FrIfTxConfirmationName	0..1
FrIfTxPduId	1..1
FrIfUserTriggerTransmitName	0..1
FrIfUserTxUL	1..1
FrIfTxPduRef	1..1

Parameter Name	**FrIfConfirm**	
Description	Defines whether the transmission of a PDU should be checked and confirmed to the PDU owning BSW module.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfCounterLimit	
Description	This value states the maximum number of indication of ready PDU data to the FrIf (i.e. maximum number of invocations of FrIf_Transmit) without an intermediate transmission of the PDU.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfImmediate	
Description	Defines whether the PDU is transmitted immediate or decoupled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfNoneMode	
Description	Using the "None-Mode" which means that there is no API FrIf_Transmit call of the upper layer for this PDU. *Note: This configuration parameter is disabled. Use the configuration parameter FrIfNoneMode located in FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering/FrIfNoneMode.*	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfTxConfirmationName
Description	This parameter defines the name of the <User_TxConfirmation>. This parameter depends on the parameter FrIfUserTxUL. If FrIfUserTxUL equals FR_TP, FR_NM, PDUR or XCP, the name of the <User_TxConfirmation> is fixed. If FrIfUserTxUL equals CDD, the name of the <User_TxConfirmation> is selectable.

	This configuration parameter is currently not used. Please see configuration container FrIfUserUpperLayerConfig instead.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	**PostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfTxPduId**	
Label	FrIfTxPduId (Tx)	
Description	The global PDU identifier, which has to be used by the upper layer BSW module. The identifier has to be zero based and consecutive.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfUserTriggerTransmitName**	
Description	This parameter defines the name of the <User_TriggerTransmit>. This parameter depends on the parameter FrIfUserTxUL. If FrIfUserTxUL equals FR_TP, FR_NM, PDUR or XCP, the name of the <User_TriggerTransmit> is fixed. If FrIfUserTxUL equals CDD, the name of the <User_TriggerTransmit> is selectable. **This configuration parameter is currently not used. Please see configuration container FrIfUserUpperLayerConfig instead.**	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfUserTxUL**
Description	This parameter defines the upper layer (UL) module to which the trigger of the to be transmitted FRIFTXPDUID (via the <User_TriggerTransmit>) or the confirmation of the successfully transmitted FRIFTXPDUID has to be routed (via the <User_TxConfirmation>). **This configuration parameter is currently not**

	used. Please see configuration container FrIfUserUpperLayerConfig instead.
Multiplicity	1..1
Type	ENUMERATION
Default value	PDUR
Range	CDD
	FR_NM
	FR_TP
	PDUR
	XCP
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfTxPduRef**
Description	Reference to the external PDU definition.
Multiplicity	1..1
Type	REFERENCE
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

5.3.1.19. VendorSpecific

Parameters included	
Parameter name	**Multiplicity**
FrIfUpperLayerHandleId	0..1
FrIfUpperLayerRef	0..1
FrIfIsVirtualPdu	0..1

Parameter Name	**FrIfUpperLayerHandleId**
Description	Id of the I-PDU passed to the upper layer within functions:
	▶ [FrIfUpperLayerFunctionPrefix]TxConfirmation.
	▶ [FrIfUpperLayerFunctionPrefix]TriggerTransmit.
	▶ [FrIfUpperLayerFunctionPrefix]RxIndication.

Multiplicity	0..1	
Type	INTEGER	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfUpperLayerRef**	
Description	Reference to upper layer configuration item within FrIf that transmits/receives this I-PDU.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfIsVirtualPdu**	
Description	If a I-PDU is marked as virtual the code generator doesn't provide an error if I-PDUs overlapp within an L-PDU.	
Multiplicity	0..1	
Type	BOOLEAN	
Configuration class	**PostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.20. FrIfDefensiveProgramming

Parameters included	
Parameter name	**Multiplicity**
FrIfDefProgEnabled	1..1
FrIfPrecondAssertEnabled	1..1
FrIfPostcondAssertEnabled	1..1
FrIfStaticAssertEnabled	1..1
FrIfUnreachAssertEnabled	1..1
FrIfInvariantAssertEnabled	1..1

Parameter Name	**FrIfDefProgEnabled**

Label	Enable Defensive Programming
Description	Enables or disables the defensive programming feature for the module FrIf. Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows: 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfPrecondAssertEnabled**
Label	Enable Precondition Assertions
Description	Enables handling of precondition assertion checks reported from the module FrIf. Dependency on parameter(s): ► Enable Development Error Detection (`FrIfDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrIfDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfPostcondAssertEnabled**
Label	Enable Postcondition Assertions
Description	Enables handling of postcondition assertion checks reported from the module FrIf. Dependency on parameter(s):

	► Enable Development Error Detection (`FrIfDevErrorDetect`): must be enabled
	► Enable Defensive Programming (`FrIfDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfStaticAssertEnabled**
Label	Enable Static Assertions
Description	Enables handling of static assertion checks reported from the module FrIf.
	Dependency on parameter(s):
	► Enable Development Error Detection (`FrIfDevErrorDetect`): must be enabled
	► Enable Defensive Programming (`FrIfDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfUnreachAssertEnabled**
Label	Enable Unreachable Code Assertions
Description	Enables handling of unreachable code assertion checks reported from the module FrIf.
	Dependency on parameter(s):
	► Enable Development Error Detection (`FrIfDevErrorDetect`): must be enabled
	► Enable Defensive Programming (`FrIfDefProgEnabled`): must be enabled
Multiplicity	1..1

Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfInvariantAssertEnabled**	
Label	Enable Invariant Assertions	
Description	Enables handling of invariant assertion checks reported from functions of the module FrIf. Dependency on parameter(s): ► Enable Development Error Detection (`FrIfDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrIfDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.21. FrIfUserUpperLayerConfig

Parameters included	
Parameter name	**Multiplicity**
FrIfUpperLayerIncludeFile	1..1
FrIfUpperLayerFunctionPrefix	1..1
FrIfUpperLayerASR43TxConfirmSupport	1..1

Parameter Name	**FrIfUpperLayerIncludeFile**
Description	Name of upper layer include file which contains the callback function declarations for the following functions ► [FrIfUpperLayerFunctionPrefix]TxConfirmation. ► [FrIfUpperLayerFunctionPrefix]TriggerTransmit. ► [FrIfUpperLayerFunctionPrefix]RxIndication.

	The following list gives the required header-files for the standard upper modules: ► PduR: PduR_FrIf.h ► FrNm: FrNm_Cbk.h ► FrTp: FrTp_Cbk.h
Multiplicity	1..1
Type	STRING
Configuration class	**VariantPostBuild:** · VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfUpperLayerFunctionPrefix**
Description	Prefix-string of upper module to be used by FrIf for Pdu transmission/reception. This string is concatenated with a fixed name for the services ► [FrIfUpperLayerFunctionPrefix]TriggerTransmit ► [FrIfUpperLayerFunctionPrefix]TxConfirmation ► [FrIfUpperLayerFunctionPrefix]RxIndication The following list gives the required strings for the standard modules: ► PduR: PduR_FrIf ► FrNm: FrNm_ ► FrTp: FrTp_
Multiplicity	1..1
Type	STRING
Configuration class	**VariantPostBuild:** · VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfUpperLayerASR43TxConfirmSupport**
Description	Defines if ASR 4.3.0 TxConfirmation API shall be used for UL. Note: Only supported for CDDs
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** · VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.3.1.22. FrIfGeneral

Containers included		
Container name	**Multiplicity**	**Description**
ReportToDem	1..1	**Label:** Production error handling Production error handling
VendorSpecific	1..1	

Parameters included	
Parameter name	**Multiplicity**
FrIfAbsTimerIdx	1..1
FrIfAllSlotsSupport	1..1
FrIfCancelTransmitSupport	1..1
FrIfDevErrorDetect	1..1
FrIfDisableLPduSupport	1..1
FrIfDisableTransceiverBranchSupport	1..1
FrIfEnableTransceiverBranchSupport	1..1
FrIfGetClockCorrectionSupport	1..1
FrIfGetChannelStatusSupport	1..1
FrIfGetGetChannelStatusSupport	1..1
FrIfGetNmVectorSupport	1..1
FrIfGetNumOfStartupFramesSupport	1..1
FrIfGetSyncFrameListSupport	1..1
FrIfGetTransceiverErrorSupport	1..1
FrIfGetWakeupRxStatusSupport	1..1
FrIfNumClstSupported	1..1
FrIfNumCtrlSupported	1..1
FrIfPublicCddHeaderFile	1..1
FrIfReadCCConfigApi	1..1
FrIfReconfigLPduSupport	1..1
FrIfUnusedBitValue	0..1
FrIfVersionInfoApi	1..1
FrIfBusMirroringSupport	1..1
FrIfDataMemSize	0..1

Parameters included	
FrIfGlobalRxMaxLoop	1..1

Parameter Name	FrIfAbsTimerIdx	
Description	Maximum number of supported absolut timers. *Note: This configuration parameter is disabled since it is not required for hardware configuration.*	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfAllSlotsSupport	
Description	Switches the API service `FrIf_AllSlots()` on or off. ▶ `true`: `FrIf_AllSlots()` API service is enabled. ▶ `false`: `FrIf_AllSlots()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfCancelTransmitSupport	
Description	Configuration parameter to enable/disable FrIf support to request the cancellation of the I-PDU transmission to FrDrv.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	FrIfDevErrorDetect
Description	Switches the Development Error Detection and Notification on or off. ▶ `true`: Development error detection and development error reporting is enabled. ▶ `false`: Development error detection and development error reporting is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrIfDisableLPduSupport
Description	Switches the API service `FrIf_DisableLPdu()` on or off. ▶ `true`: `FrIf_DisableLPdu()` API service is enabled. ▶ `false`: `FrIf_DisableLPdu()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrIfDisableTransceiverBranchSupport

Description	Configuration parameter to enable/disable FrIf support to disable branches of an active star.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** | VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfEnableTransceiverBranchSupport**
Description	Configuration parameter to enable/disable FrIf support to enable branches of an active star.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** | VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGetClockCorrectionSupport**
Description	Switches the API service `FrIf_GetClockCorrection()` on or off. ▶ `true`: `FrIf_GetClockCorrection()` API service is enabled. ▶ `false`: `FrIf_GetClockCorrection()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** | VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGetChannelStatusSupport**
Description	Switches the API service `FrIf_GetChannelStatus()` on or off. ▶ `true`: `FrIf_GetChannelStatus()` API service is enabled.

▶ false: `FrIf_GetChannelStatus()` API service is disabled.

Optimization Effect:

▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.

Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGetGetChannelStatusSupport**
Description	Instead of the configuration parameter FrIfGetGetChannelStatusSupport, FrIfGetChannelStatusSupport is used to enable/disable the API service `FrIf_-GetChannelStatus()`.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrIfGetNmVectorSupport**
Description	Switches the API service `FrIf_GetNmVector()` on or off. ▶ true: `FrIf_GetNmVector()` API service is enabled. ▶ false: `FrIf_GetNmVector()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrIfGetNumOfStartupFramesSupport	
Description	Configuration parameter to enable/disable FrIf support to enable/disable of polling the FlexRay Driver for the actual number of received startup frames on the bus.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGetSyncFrameListSupport	
Description	Switches the API service `FrIf_GetSyncFrameList()` on or off. ▶ `true`: `FrIf_GetSyncFrameList()` API service is enabled. ▶ `false`: `FrIf_GetSyncFrameList()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfGetTransceiverErrorSupport	
Description	Configuration parameter to enable/disable FrIf support to get the FlexRay Transceiver errors by calling the FlexRay Transceiver module. ▶ `true`: `FrIf_GetTransceiverError()` API service is enabled. ▶ `false`: `FrIf_GetTransceiverError()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	

Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfGetWakeupRxStatusSupport**	
Description	Switches the API service `FrIf_GetWakeupRxStatus()` on or off. ▶ `true`: `FrIf_GetWakeupRxStatus()` API service is enabled. ▶ `false`: `FrIf_GetWakeupRxStatus()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfNumClstSupported**	
Description	Maximum number of FlexRay Clusters that the FlexRay Interface supports.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=15	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfNumCtrlSupported**	
Description	Maximum number of FlexRay CCs that the FlexRay Interface supports.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=15	

	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfPublicCddHeaderFile**	
Description	Defines header files for callback functions which shall be included in case of CDDs. **This config parameter is currently not used.**	
Multiplicity	1..1	
Type	STRING	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfReadCCConfigApi**	
Description	Configuration parameter to enable/disable the optional FrIf_ReadCCConfig API.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfReconfigLPduSupport**	
Description	Switches the API service `FrIf_ReconfigLPdu()` on or off. ▶ `true`: `FrIf_ReconfigLPdu()` API service is enabled. ▶ `false`: `FrIf_ReconfigLPdu()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfUnusedBitValue**	

Description	Set unused bits to a defined value.	
Multiplicity	0..1	
Type	INTEGER	
Range	<=1	
	>=0	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfVersionInfoApi	
Description	Switches the API service `FrIf_GetVersionInfo()` on or off. ► `true`: `FrIf_GetVersionInfo()` is implemented as function. ► `false`: `FrIf_GetVersionInfo()` is implemented as preprocessor macro. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrIfBusMirroringSupport	
Description	Configuration parameter to enable/disable FrIf support to enable/disable reporting received/transmitted frames to the Bus Mirroring module. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ► **RAM reduction (config):** Disabling this parameter reduces the RAM consumption of the module configuration. ► **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	

Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrIfDataMemSize**	
Description	Size of internal FrIf data in units of bytes (static memory allocation) - Memory required by post-build configuration must be smaller than this constant. If the parameter is disabled, the MCG calculates the required amount of memory itself.	
Multiplicity	0..1	
Type	INTEGER	
Range	<=65535	
	>=1	
Configuration class	**Link:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfGlobalRxMaxLoop**	
Description	Defines the maximum number of loops for for all communication operations with communication action RECEIVE_AND_INDICATE (Use case: emptying a FIFO).	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=512	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.23. ReportToDem

Parameters included	
Parameter name	**Multiplicity**
FrIfJoblistSyncReportToDem	1..1
FrIfJoblistSyncDebounceMethod	1..1

Parameters included	
FrIfJoblistSyncReportToDemDetErrorId	1..1
FrIfNITStatusAReportToDem	1..1
FrIfNITStatusADebounceMethod	1..1
FrIfNITStatusAReportToDemDetErrorId	1..1
FrIfNITStatusBReportToDem	1..1
FrIfNITStatusBDebounceMethod	1..1
FrIfNITStatusBReportToDemDetErrorId	1..1
FrIfSymbolWindowStatusAReportToDem	1..1
FrIfSymbolWindowStatusADebounceMethod	1..1
FrIfSymbolWindowStatusAReportToDemDetErrorId	1..1
FrIfSymbolWindowStatusBReportToDem	1..1
FrIfSymbolWindowStatusBDebounceMethod	1..1
FrIfSymbolWindowStatusBReportToDemDetErrorId	1..1
FrIfAggregatedStatusAReportToDem	1..1
FrIfAggregatedStatusADebounceMethod	1..1
FrIfAggregatedStatusAReportToDemDetErrorId	1..1
FrIfAggregatedStatusBReportToDem	1..1
FrIfAggregatedStatusBDebounceMethod	1..1
FrIfAggregatedStatusBReportToDemDetErrorId	1..1

Parameter Name	FrIfJoblistSyncReportToDem
Label	FRIF_E_JLE_SYNC report to
Description	Selects the handling of the production error FRIF_E_JLE_SYNC. ► DEM: The error is reported to the Diagnostics Event Manager (Dem). ► DET: The error is reported to the Development Error Tracer (Det) if enabled. ► DISABLE: The error is not reported at all. **Optimization Effect:** ► **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.
Multiplicity	1..1

Type	ENUMERATION	
Default value	DET	
Range	DEM	
	DET	
	DISABLE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfJoblistSyncDebounceMethod**	
Label	FRIF_E_JLE_SYNC Dem Debouncing method	
Description	If a production error is reported towards the Dem, FrIfJoblistSyncDe-bounceMethod defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL). In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus(). In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus().	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	INTERNAL	
Range	DEM	
	INTERNAL	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfJoblistSyncReportToDemDetErrorId**
Label	FRIF_E_JLE_SYNC Dem To Det error ID
Description	If a production error is reported towards the Det, FrIfJoblistSyncReport-ToDemDetErrorId defines the error ID which is reported towards the Det.
Multiplicity	1..1
Type	INTEGER
Default value	9
Range	<=255

	>=9	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfNITStatusAReportToDem**	
Label	FRIF_E_NIT_CH_A report to	
Description	Selects the handling of the production error FRIF_E_NIT_CH_A. ► DEM: The error is reported to the Diagnostics Event Manager (Dem). ► DET: The error is reported to the Development Error Tracer (Det) if enabled. ► DISABLE: The error is not reported at all. **Optimization Effect:** ► **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DISABLE	
Range	DEM	
	DET	
	DISABLE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfNITStatusADebounceMethod**	
Label	FRIF_E_NIT_CH_A Dem Debouncing method	
Description	If a production error is reported towards the Dem, FrIfNITStatusAReportToDem-Method defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL). In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus(). In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus().	

Multiplicity	1..1
Type	ENUMERATION
Default value	INTERNAL
Range	DEM
	INTERNAL
Configuration class	**PreCompile:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfNITStatusAReportToDemDetErrorId**
Label	FRIF_E_NIT_CH_A Dem To Det error ID
Description	If a production error is reported towards the Det, FrIfNITStatusAReport-ToDemDetErrorId defines the error ID which is reported towards the Det.
Multiplicity	1..1
Type	INTEGER
Default value	101
Range	<=255
	>=9
Configuration class	**PreCompile:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfNITStatusBReportToDem**
Label	FRIF_E_NIT_CH_B report to
Description	Selects the handling of the production error FRIF_E_NIT_CH_B.
	► DEM: The error is reported to the Diagnostics Event Manager (Dem).
	► DET: The error is reported to the Development Error Tracer (Det) if enabled.
	► DISABLE: The error is not reported at all.
	Optimization Effect:
	► **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code.
	► **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.
Multiplicity	1..1

Type	ENUMERATION	
Default value	DISABLE	
Range	DEM	
	DET	
	DISABLE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfNITStatusBDebounceMethod**	
Label	FRIF_E_NIT_CH_B Dem Debouncing method	
Description	If a production error is reported towards the Dem, FrIfNITStatusBReportToDem-Method defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL). In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus(). In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus().	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	INTERNAL	
Range	DEM	
	INTERNAL	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfNITStatusBReportToDemDetErrorId**	
Label	FRIF_E_NIT_CH_B Dem To Det error ID	
Description	If a production error is reported towards the Det, FrIfNITStatusBReport-ToDemDetErrorId defines the error ID which is reported towards the Det.	
Multiplicity	1..1	
Type	INTEGER	
Default value	102	
Range	<=255	

	>=9	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrIfSymbolWindowStatusAReportToDem
Label	FRIF_E_SW_CH_A report to
Description	Selects the handling of the production error FRIF_E_SW_CH_A.

- ► DEM: The error is reported to the Diagnostics Event Manager (Dem).
- ► DET: The error is reported to the Development Error Tracer (Det) if enabled.
- ► DISABLE: The error is not reported at all.

Optimization Effect:

- ► **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code.
- ► **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.

Multiplicity	1..1
Type	ENUMERATION
Default value	DISABLE
Range	DEM
	DET
	DISABLE

Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrIfSymbolWindowStatusADebounceMethod
Label	FRIF_E_SW_CH_A Dem Debouncing method
Description	If a production error is reported towards the Dem, FrIfSymbolWindowStatusAReportToDemMethod defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL).

In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus().

In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus(). |

Multiplicity	1..1	
Type	ENUMERATION	
Default value	INTERNAL	
Range	DEM	
	INTERNAL	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfSymbolWindowStatusAReportToDemDetErrorId**	
Label	FRIF_E_SW_CH_A Dem To Det error ID	
Description	If a production error is reported towards the Det, FrIfSymbolWindowStatusAReportToDemDetErrorId defines the error ID which is reported towards the Det.	
Multiplicity	1..1	
Type	INTEGER	
Default value	103	
Range	<=255	
	>=9	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfSymbolWindowStatusBReportToDem**
Label	FRIF_E_SW_CH_B report to
Description	Selects the handling of the production error FRIF_E_SW_CH_B. ▶ DEM: The error is reported to the Diagnostics Event Manager (Dem). ▶ DET: The error is reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: The error is not reported at all. **Optimization Effect:** ▶ **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.
Multiplicity	1..1

Type	ENUMERATION
Default value	DISABLE
Range	DEM
	DET
	DISABLE
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfSymbolWindowStatusBDebounceMethod**	
Label	FRIF_E_SW_CH_B Dem Debouncing method	
Description	If a production error is reported towards the Dem, FrIfSymbolWindowStatusBReportToDemMethod defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL). In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus(). In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus().	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	INTERNAL	
Range	DEM	
	INTERNAL	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfSymbolWindowStatusBReportToDemDetErrorId**
Label	FRIF_E_SW_CH_B Dem To Det error ID
Description	If a production error is reported towards the Det, FrIfSymbolWindowStatusBReportToDemDetErrorId defines the error ID which is reported towards the Det.
Multiplicity	1..1
Type	INTEGER
Default value	104
Range	<=255

	>=9	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfAggregatedStatusAReportToDem**	
Label	FRIF_E_ACS_CH_A report to	
Description	Selects the handling of the production error FRIF_E_ACS_CH_A. ► DEM: The error is reported to the Diagnostics Event Manager (Dem). ► DET: The error is reported to the Development Error Tracer (Det) if enabled. ► DISABLE: The error is not reported at all. **Optimization Effect:** ► **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DISABLE	
Range	DEM	
	DET	
	DISABLE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfAggregatedStatusADebounceMethod**	
Label	FRIF_E_ACS_CH_A Dem Debouncing method	
Description	If a production error is reported towards the Dem, FrIfAggregatedStatusAReportToDemMethod defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL). In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus(). In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus().	

Multiplicity	1..1
Type	ENUMERATION
Default value	INTERNAL
Range	DEM
	INTERNAL
Configuration class	**PreCompile:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrIfAggregatedStatusAReportToDemDetErrorId**	
Label	FRIF_E_ACS_CH_A Dem To Det error ID	
Description	If a production error is reported towards the Det, FrIfAggregatedStatusAReport-ToDemDetErrorId defines the error ID which is reported towards the Det.	
Multiplicity	1..1	
Type	INTEGER	
Default value	105	
Range	<=255	
	>=9	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfAggregatedStatusBReportToDem**
Label	FRIF_E_ACS_CH_B report to
Description	Selects the handling of the production error FRIF_E_ACS_CH_B. ► DEM: The error is reported to the Diagnostics Event Manager (Dem). ► DET: The error is reported to the Development Error Tracer (Det) if enabled. ► DISABLE: The error is not reported at all. **Optimization Effect:** ► **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.
Multiplicity	1..1

Type	ENUMERATION	
Default value	DISABLE	
Range	DEM	
	DET	
	DISABLE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfAggregatedStatusBDebounceMethod**	
Label	FRIF_E_ACS_CH_B Dem Debouncing method	
Description	If a production error is reported towards the Dem, FrIfAggregatedStatusBReportToDemMethod defines the whether Event debouncing is performed in Dem (DEM) or not at all (INTERNAL). In case 'DEM' is selected, FrIf always reports status PRE-PASSED/PRE-FAILED to Dem _ReportErrorStatus(). In case 'INTERNAL' is selected, FrIf always reports status PASSED/FAILED to Dem _ReportErrorStatus().	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	INTERNAL	
Range	DEM	
	INTERNAL	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfAggregatedStatusBReportToDemDetErrorId**
Label	FRIF_E_ACS_CH_B Dem To Det error ID
Description	If a production error is reported towards the Det, FrIfAggregatedStatusBReportToDemDetErrorId defines the error ID which is reported towards the Det.
Multiplicity	1..1
Type	INTEGER
Default value	106
Range	<=255

	>=9	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.24. VendorSpecific

Parameters included	
Parameter name	**Multiplicity**
SingleClstOptEnable	1..1
SingleCtrlOptEnable	1..1
SingleFrOptEnable	1..1
FrIntegrationEnable	1..1
SingleFrTrcvOptEnable	1..1
ReportToDetEnable	1..1
JoblistIRQMuxEnable	1..1
JoblistIRQDelayCheckEnable	1..1
JoblistPrepareLPduEnable	1..1
RelativeTimerApiEnable	1..1
SetExtSyncApiEnable	1..1
GetSyncStateApiEnable	1..1
MtsApiEnable	1..1
WakeupApiEnable	1..1
GetIrqStatusApiEnable	1..1
DisableIrqApiEnable	1..1
AllowColdstartApiEnable	1..1
GetCtrlErrorStatusApiEnable	1..1
ExtIRQServicesApiEnable	1..1
GetTransceiverModeApiEnable	1..1
GetTransceiverWUReasonApiEnable	1..1
WakeupControlApiEnable	1..1
CheckWakeupByTransceiverApiEnable	1..1
DynamicPayloadLengthEnable	1..1

Parameters included	
ImmediateTxEnable	1..1
DecoupledRxEnable	1..1
TransmitQueueEnable	1..1
ComOpCycleFilterEnable	1..1
ExtendedRxFIFOEnable	1..1
LPduIdxSize	1..1
PduIdxSize	1..1
ComOpIdxSize	1..1
FrDriverAutosarVersion	1..1
FrIfRelocatablePbcfgEnable	1..1

Parameter Name	SingleClstOptEnable	
Description	Optimization for a configuration consisting of a single FlexRay cluster. ► `true`: Enables optimization but limits the configuration to a single FlexRay cluster. ► `false`: Disables optimization but allows to configure more than one FlexRay cluster. **Optimization Effect:** ► **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration. ► **ROM reduction (code):** Enabling this parameter reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Enabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SingleCtrlOptEnable
Description	Optimization for a configuration consisting of a single FlexRay controller.

	► `true`: Enables optimization but limits the configuration to a single FlexRay controller.	
	► `false`: Disables optimization but allows to configure more than one FlexRay controller.	
	Optimization Effect:	
	► **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration.	
	► **ROM reduction (code):** Enabling this parameter reduces the ROM consumption of the module code.	
	► **Execution time reduction (code):** Enabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**SingleFrOptEnable**	
Description	Optimizations for a configuration consisting of a single FlexRay driver.	
	► `true`: Enables optimization but limits the configuration to a single FlexRay driver.	
	► `false`: Disables optimization but allows to configure more than one FlexRay driver.	
	Optimization Effect:	
	► **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration.	
	► **ROM reduction (code):** Enabling this parameter reduces the ROM consumption of the module code.	
	► **Execution time reduction (code):** Enabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	Elektrobit Automotive GmbH

Parameter Name	FrIntegrationEnable
Description	Enable Fr integration of FrIf. This breaks the interface contracts but increases performance. ▶ `true`: Enables integration of FrIf and Fr (optimzed). ▶ `false`: Disables integration of FrIf and Fr (standard). This feature shall be enabled only if: ▶ `SingleFrOptEnable` is set to `true`. ▶ Fr module has `FrIfIntegrationEnable` set to `true` (Elektrobit Fr modules only). **Optimization Effect:** ▶ **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration. ▶ **ROM reduction (code):** Enabling this parameter reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Enabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SingleFrTrcvOptEnable
Description	Optimization for a configuration consisting of a single FlexRay transceiver driver. ▶ `true`: Enables optimization but limits the configuration to a single FlexRay transceiver driver. ▶ `false`: Disables optimization but allows to configure more than one FlexRay transceiver driver. **Optimization Effect:** ▶ **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration.

	► **ROM reduction (code):** Enabling this parameter reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Enabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	ReportToDetEnable	
Description	Switches the Development Error Notification on or off. ► `true`: Development error reporting is enabled. ► `false`: Development error reporting is disabled. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	JoblistIRQMuxEnable
Description	Switching demultiplexing of absolute timer interrupt for joblist execution on or off. ► `true`: The Joblist interrupt is multiplexed with other interrupts and the FrIf performs interrupt demultiplexing. ► `false`: The FrIf is the one and only user of the ISR. No interrupt demultiplexing is performed. **Optimization Effect:**

	▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
	▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	JoblistIRQDelayCheckEnable
Description	Switching invocation delay check for joblist execution on or off.
	▶ `true`: The joblist invocation delay is checked against `MaxIsrDelay`.
	▶ `false`: The joblist invocation delay is not checked.
	Optimization Effect:
	▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
	▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	JoblistPrepareLPduEnable
Description	Switching support for the communication operation `PREPARE_LPDU` on and off.
	▶ `true`: The communication operation `PREPARE_LPDU` is enabled.
	▶ `false`: The communication operation `PREPARE_LPDU` is disabled.
	Optimization Effect:
	▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1

Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	RelativeTimerApiEnable	
Description	Switches the API services `FrIf_SetRelativeTimer()`, `FrIf_CancelRelativeTimer()`, `FrIf_EnableRelativeTimerIRQ()`, `FrIf_DisableRelativeTimerIRQ()`, `FrIf_GetRelativeTimerIRQStatus()` and `FrIf_AckRelativeTimerIRQStatus()` on or off. ▶ `true`: Relative timer API services are enabled. ▶ `false`: Relative timer API services are disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SetExtSyncApiEnable	
Description	Switches the API service `FrIf_SetExtSync()` on or off. ▶ `true`: API service `FrIf_SetExtSync()` is enabled. ▶ `false`: API service `FrIf_SetExtSync()` is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	Elektrobit Automotive GmbH

Parameter Name	GetSyncStateApiEnable
Description	Switches the API service `FrIf_GetSyncState()` on or off. ► `true`: API service `FrIf_GetSyncState()` is enabled. ► `false`: API service `FrIf_GetSyncState()` is disabled. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** | VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	MtsApiEnable
Description	Switches the API services `FrIf_SendMTS()`, `FrIf_StopMTS()` and `FrIf_-CheckMTS()` on or off. ► `true`: MTS API services are enabled. ► `false`: MTS API services are disabled. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** | VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	WakeupApiEnable
Description	Switches the API services `FrIf_SendWUP()` and `FrIf_SetWakeupChannel()` on or off. ► `true`: Wakeup API services are enabled.

	► `false`: Wakeup API services are disabled.
	Optimization Effect:
	► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**GetIrqStatusApiEnable**	
Description	Switches the API services `FrIf_GetAbsoluteTimerIRQStatus()` and `FrIf_GetRelativeTimerIRQStatus()` on or off.	
	► `true`: IRQ status API services are enabled.	
	► `false`: IRQ status API services are disabled.	
	Optimization Effect:	
	► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**DisableIrqApiEnable**	
Description	Switches the API services `FrIf_DisableAbsoluteTimerIRQ()` and `FrIf_-DisableRelativeTimerIRQStatus()` on or off.	
	► `true`: IRQ disable API services are enabled.	
	► `false`: IRQ disable API services are disabled.	
	Optimization Effect:	
	► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	

Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**AllowColdstartApiEnable**	
Description	Switches the API service `FrIf_AllowColdstart()` on or off. ▶ `true`: `FrIf_AllowColdstart()` API service is enabled. ▶ `false`: `FrIf_AllowColdstart()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**GetCtrlErrorStatusApiEnable**	
Description	Switches the API service `FrIf_GetControllerErrorStatus()` on or off. ▶ `true`: `FrIf_GetControllerErrorStatus()` API service is enabled. ▶ `false`: `FrIf_GetControllerErrorStatus()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	ExtIRQServicesApiEnable
Description	Switches the API services `FrIf_AckIRQ()`, `FrIf_EnableIRQ()`, `FrIf_DisableIRQ()` and `FrIf_GetIRQStatus()` on or off. ► `true`: Extended IRQ API services are enabled. ► `false`: Extended IRQ API services are disabled. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	GetTransceiverModeApiEnable
Description	Switches the API service `FrIf_GetTransceiverMode()` on or off. ► `true`: `FrIf_GetTransceiverMode()` API service is enabled. ► `false`: `FrIf_GetTransceiverMode()` API service is disabled. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	GetTransceiverWUReasonApiEnable
Description	Switches the API service `FrIf_GetTransceiverWUReason()` on or off. ► `true`: `FrIf_GetTransceiverWUReason()` API service is enabled. ► `false`: `FrIf_GetTransceiverWUReason()` API service is disabled. **Optimization Effect:**

	▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**WakeupControlApiEnable**
Description	Switches the API services `FrIf_ClearTransceiverWakeup()` on or off. ▶ `true`: FrIf_ClearTransceiverWakeup() is enabled. ▶ `false`: FrIf_ClearTransceiverWakeup() is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**CheckWakeupByTransceiverApiEnable**
Description	Switches the API service `FrIf_CheckWakeupByTransceiver()` on or off. ▶ `true`: `FrIf_CheckWakeupByTransceiver()` API service is enabled. ▶ `false`: `FrIf_CheckWakeupByTransceiver()` API service is disabled. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild

Origin	Elektrobit Automotive GmbH

Parameter Name	DynamicPayloadLengthEnable	
Description	Switches support for dynamic payload length on or off. ▶ `true`: Support for dynamic payload length is enabled. ▶ `false`: Support for dynamic payload length is disabled. **Optimization Effect:** ▶ **ROM reduction (config):** Disabling this parameter reduces the ROM consumption of the module configuration. ▶ **RAM reduction (config):** Disabling this parameter reduces the RAM consumption of the module configuration. ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	ImmediateTxEnable
Description	Switches support for immediate transmission on or off. ▶ `true`: Support for immediate transmission is enabled. ▶ `false`: Support for immediate transmission is disabled. **Optimization Effect:** ▶ **ROM reduction (config):** Disabling this parameter reduces the ROM consumption of the module configuration. ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1

Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DecoupledRxEnable	
Description	Switches support for decoupled reception (communication operations RE-CEIVE_INDICATION and RECEIVE_AND_STORE) on or off. ▶ `true`: Support for decoupled reception is enabled. ▶ `false`: Support for decoupled reception is disabled. **Optimization Effect:** ▶ **ROM reduction (config):** Disabling this parameter reduces the ROM consumption of the module configuration. ▶ **RAM reduction (config):** Disabling this parameter reduces the RAM consumption of the module configuration. ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TransmitQueueEnable	
Description	Switches support for transmit request queueing (`FrIfCounterLimit` > 1) on or off. ▶ `true`: Transmit request queueing is enabled. ▶ `false`: Transmit request queueing is disabled. **Optimization Effect:** ▶ **ROM reduction (config):** Disabling this parameter reduces the ROM consumption of the module configuration. ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	

Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**ComOpCycleFilterEnable**	
Description	Changes the way cycle multiplexing is coded into the generated configuration. **This optimization is currently not supported. The setting of this switch is ignored.** ▶ `true`: All jobs at the same macrotick offset are merged to a single job, with repetition 1. The cycle filter is attached to each communication operation instead. Depending on the joblist this can significantly reduce the configuration memory footprint. ▶ `false`: Cycle filtering is performed at job-level. This can lead to larger configurations but sligthtly better performance. **Optimization Effect:** ▶ **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration. ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**ExtendedRxFIFOEnable**
Description	If set to "true" support for demultiplexing received LPdus from the extended RxFIFO (EB FlexRay driver feature) is enabled. If set to "false" support for demultiplexing received LPdus from the extended RxFIFO (EB FlexRay driver feature) is disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	LPduIdxSize
Description	Defines the size which is used to internally store LPduIds. Choices: ► **SIZE_8_BIT:** A maximum of 255 LPdus is supported for a particular FlexRay Controller. ► **SIZE_16_BIT:** A maximum of 65535 LPdus is supported for a particular FlexRay Controller. **Optimization Effect:** ► **ROM reduction (config):** Setting this parameter to SIZE_8_BIT reduces the ROM consumption of the module configuration.
Multiplicity	1..1
Type	ENUMERATION
Default value	SIZE_8_BIT
Range	SIZE_8_BIT
	SIZE_16_BIT
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	PduIdxSize
Description	Defines the size which is used to internally store PduIds for Rx- and Tx-IPDus. Choices: ► **SIZE_8_BIT:** A maximum of 255 Tx-Pdus and a Rx-PduId range from 0-255 is supported. ► **SIZE_16_BIT:** A maximum of 65535 Tx-Pdus and a Rx-PduId range from 0-65535 is supported. **Optimization Effect:** ► **ROM reduction (config):** Setting this parameter to SIZE_8_BIT reduces the ROM consumption of the module configuration.
Multiplicity	1..1
Type	ENUMERATION

Default value	SIZE_8_BIT	
Range	SIZE_8_BIT	
	SIZE_16_BIT	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**ComOpIdxSize**	
Description	Defines the size which is used to internally store the number of communication operations per job.	
	Choices:	
	► **SIZE_8_BIT:** A maximum of 255 communication operations per communication job is supported.	
	► **SIZE_16_BIT:** A maximum of 65535 communication operations per communication job is supported.	
	Optimization Effect:	
	► **ROM reduction (config):** Setting this parameter to SIZE_8_BIT reduces the ROM consumption of the module configuration.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	SIZE_8_BIT	
Range	SIZE_8_BIT	
	SIZE_16_BIT	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrDriverAutosarVersion**
Description	Defines which Autosar version is supported by used Fr driver.
	Choices:
	► **ASR_40:** FlexRay driver supports Autosar version 4.0.3.
	► **ASR_44:** FlexRay driver supports Autosar version 4.4.0.
Multiplicity	1..1
Type	ENUMERATION
Default value	ASR_40

Range	ASR_40	
	ASR_44	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrIfRelocatablePbcfgEnable**	
Description	Enables/disable support for relocatable postbuild configuration.	
	► True: Postbuild configuration relocatable in memory.	
	► False: Postbuild configuration not relocatable in memory.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.25. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	**PbcfgMSupport**	
Label	PbcfgM support	
Description	Specifies whether or not the FrIf can use the PbcfgM module for post-build support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

5.3.2. Application programming interface (API)

5.3.2.1. Type definitions

5.3.2.1.1. FrIf_ClstRuntimeDataType

Purpose		
Type	struct	
Members	`uint16 NextJob`	
	`uint8 JobListCycleBase`	
	`uint8 JobListTimeoutCounter`	
	`uint8 FrIf_SyncState`	
	`uint8 FrIf_State`	
	`boolean BusmirroringEnabled`	

5.3.2.1.2. FrIf_FrFunctionType

Purpose		
Type	struct	
Members	`Std_ReturnType(* Fr_ControllerInit`	
	`Std_ReturnType(* Fr_SendMTS`	
	`Std_ReturnType(* Fr_StopMTS`	
	`Std_ReturnType(* Fr_CheckMTS`	
	`Std_ReturnType(* Fr_StartCommunication`	
	`Std_ReturnType(* Fr_HaltCommunication`	
	`Std_ReturnType(* Fr_AbortCommunication`	
	`Std_ReturnType(* Fr_SetWakeupChannel`	
	`Std_ReturnType(* Fr_SendWUP`	
	`Std_ReturnType(* Fr_SetExtSync`	
	`Std_ReturnType(* Fr_GetSyncState`	

	`Std_ReturnType(* Fr_GetPOCSta-tus`	
	`Std_ReturnType(* Fr_Transmit-TxLPdu`	
	`Std_ReturnType(* Fr_Re-ceiveRxLPdu`	
	`Std_ReturnType(* Fr_CheckTxLP-duStatus`	
	`Std_ReturnType(* Fr_PrepareLPdu`	
	`Std_ReturnType(* Fr_GetGlobal-Time`	
	`Std_ReturnType(* Fr_SetAb-soluteTimer`	
	`Std_ReturnType(* Fr_CancelAb-soluteTimer`	
	`Std_ReturnType(* Fr_EnableAb-soluteTimerIRQ`	
	`Std_ReturnType(* Fr_AckAb-soluteTimerIRQ`	
	`Std_ReturnType(* Fr_DisableAb-soluteTimerIRQ`	
	`Std_ReturnType(* Fr_GetAb-soluteTimerIRQStatus`	
	`Std_ReturnType(* Fr_SetRela-tiveTimer`	
	`Std_ReturnType(* Fr_CancelRela-tiveTimer`	
	`Std_ReturnType(* Fr_EnableRela-tiveTimerIRQ`	
	`Std_ReturnType(* Fr_AckRela-tiveTimerIRQ`	
	`Std_ReturnType(* Fr_DisableRel-ativeTimerIRQ`	
	`Std_ReturnType(* Fr_GetRela-tiveTimerIRQStatus`	
	`Std_ReturnType(* Fr_GetNmVector`	

	`Std_ReturnType(* Fr_AllowCold-start`	
	`Std_ReturnType(* Fr_GetChan-nelStatus`	
	`Std_ReturnType(* Fr_GetCon-trollerErrorStatus`	
	`Std_ReturnType(* Fr_AllSlots`	
	`Std_ReturnType(* Fr_ReconfigLP-du`	
	`Std_ReturnType(* Fr_DisableLPdu`	
	`Std_ReturnType(* Fr_GetWake-upRxStatus`	
	`Std_ReturnType(* Fr_AckIRQ`	
	`Std_ReturnType(* Fr_EnableIRQ`	
	`Std_ReturnType(* Fr_DisableIRQ`	
	`Std_ReturnType(* Fr_GetIRQSta-tus`	
	`Std_ReturnType(* Fr_GetClock-Correction`	
	`Std_ReturnType(* Fr_-GetSyncFrameList`	
	`Std_ReturnType(* Fr_GetNumOfS-tartupFrames`	
	`Std_ReturnType(* Fr_ReadCC-Config`	
	`Std_ReturnType(* Fr_Re-ceiveRxFIFO`	
	`Std_ReturnType(* Fr_FlushRxFIFO`	

5.3.2.1.3. FrIf_PduOwnerFunctionASR40Type

Purpose		
Type	struct	
Members	`Std_ReturnType(* FrIfTrigger-Transmit`	

`void(* FrIfTxConfirmation`	
`void(* FrIfRxIndication`	

5.3.2.1.4. FrIf_PduOwnerFunctionASR43Type

Purpose	
Type	struct
Members	`Std_ReturnType(* FrIfTrigger-Transmit`	
	`void(* FrIfTxConfirmation`	
	`void(* FrIfRxIndication`	

5.3.2.1.5. FrIf_TrcvFunctionType

Purpose	
Type	struct
Members	`Std_ReturnType(* FrTrcv_Set-TransceiverMode`	
	`Std_ReturnType(* FrTrcv_Get-TransceiverMode`	
	`Std_ReturnType(* FrTrcv_Get-TransceiverWUReason`	
	`Std_ReturnType(* FrTr-cv_ClearTransceiverWakeup`	
	`void(* FrTrcv_CheckWakeupBy-Transceiver`	
	`Std_ReturnType(* FrTrcv_Get-TransceiverError`	
	`Std_ReturnType(* FrTrcv_Dis-ableTransceiverBranch`	
	`Std_ReturnType(* FrTrcv_Enable-TransceiverBranch`	

5.3.2.2. Macro constants

5.3.2.2.1. FRIF_ABORTCOMMUNICATION_SERVICE_ID

Purpose	
Value	(0x06U)

5.3.2.2.2. FRIF_ACKABSOLUTETIMERIRQ_SERVICE_ID

Purpose	
Value	(0x21U)

5.3.2.2.3. FRIF_ACKIRQ_SERVICE_ID

Purpose	
Value	(0x86U)

5.3.2.2.4. FRIF_ACKRELATIVETIMERIRQ_SERVICE_ID

Purpose	
Value	(0x97U)

5.3.2.2.5. FRIF_ALLOWCOLDSTART_SERVICE_ID

Purpose	
Value	(0x10U)

5.3.2.2.6. FRIF_ALLSLOTS_SERVICE_ID

Purpose	
Value	(0x33U)

5.3.2.2.7. FRIF_ALWAYS_TRANSMIT_FRAME

Purpose	
Value	(0x01U)

5.3.2.2.8. FRIF_ALWAYS_TRANSMIT_SIMPLE_FRAME

Purpose	
Value	(0x0AU)

5.3.2.2.9. FRIF_ARRAY_SIZE

Purpose	
Value	(sizeof(arrayname)/sizeof((arrayname)[0]))

5.3.2.2.10. FRIF_CANCELABSOLUTETIMER_SERVICE_ID

Purpose	
Value	(0x1BU)

5.3.2.2.11. FRIF_CANCELRELATIVETIMER_SERVICE_ID

Purpose	
Value	(0x94U)

5.3.2.2.12. FRIF_CHECKMTS_SERVICE_ID

Purpose	
Value	(0x92U)

5.3.2.2.13. FRIF_CHECKWAKEUPBYTRANSCEIVER_SERVICE_ID

Purpose	

Value	(0x39U)

5.3.2.2.14. FRIF_CLEARTRANSCEIVERWAKEUP_SERVICE_ID

Purpose	
Value	(0x18U)

5.3.2.2.15. FRIF_CONFIRM_DEC_FRAME

Purpose	
Value	(0x04U)

5.3.2.2.16. FRIF_CONFIRM_DEC_SIMPLE_FRAME

Purpose	
Value	FRIF_CONFIRM_IMM_SIMPLE_FRAME
Description	communication operation 'tx confirmation' for decoupled transmitted frames with one pdu

5.3.2.2.17. FRIF_CONFIRM_IMM_FRAME

Purpose	
Value	(0x03U)

5.3.2.2.18. FRIF_CONFIRM_IMM_SIMPLE_FRAME

Purpose	
Value	FRIF_CONFIRM_IMM_FRAME

5.3.2.2.19. FRIF_CONTROLLERINIT_SERVICE_ID

Purpose	
Value	(0x03U)

5.3.2.2.20. FRIF_DISABLEABSOLUTETIMERIRQ_SERVICE_ID

Purpose	
Value	(0x23U)

5.3.2.2.21. FRIF_DISABLEIRQ_SERVICE_ID

Purpose	
Value	(0x88U)

5.3.2.2.22. FRIF_DISABLELPDU_SERVICE_ID

Purpose	
Value	(0x28U)

5.3.2.2.23. FRIF_DISABLERELATIVETIMERIRQ_SERVICE_ID

Purpose	
Value	(0x98U)

5.3.2.2.24. FRIF_DISABLETRANSCEIVERBRANCH_SERVICE_ID

Purpose	
Value	(0x37U)

5.3.2.2.25. FRIF_ENABLEABSOLUTETIMERIRQ_SERVICE_ID

Purpose	
Value	(0x1DU)

5.3.2.2.26. FRIF_ENABLEBUSMIRRORING_SERVICE_ID

Purpose	
Value	(0x4BU)

5.3.2.2.27. FRIF_ENABLEIRQ_SERVICE_ID

Purpose	
Value	(0x87U)

5.3.2.2.28. FRIF_ENABLERELATIVETIMERIRQ_SERVICE_ID

Purpose	
Value	(0x95U)

5.3.2.2.29. FRIF_ENABLETRANSCEIVERBRANCH_SERVICE_ID

Purpose	
Value	(0x36U)

5.3.2.2.30. FRIF_E_INV_CHNL_IDX

Purpose	DET error code.
Value	(0x04U)
Description	Invalid channel index passed to service.

5.3.2.2.31. FRIF_E_INV_CLST_IDX

Purpose	DET error code.
Value	(0x03U)
Description	Invalid cluster index passed to service.

5.3.2.2.32. FRIF_E_INV_CTRL_IDX

Purpose	DET error code.
Value	(0x02U)
Description	Invalid controller index passed to service.

5.3.2.2.33. FRIF_E_INV_CYCLE

Purpose	DET error code.
Value	(0x81U)
Description	Invalid cycle vale passed to service.

5.3.2.2.34. FRIF_E_INV_JOB_IDX

Purpose	DET error code.
Value	(0x80U)
Description	Invalid job index passed to service.

5.3.2.2.35. FRIF_E_INV_LPDU_IDX

Purpose	DET error code.
Value	(0x07U)
Description	Invalid LPdu index passed to service. (Specified but not used)

5.3.2.2.36. FRIF_E_INV_POINTER

Purpose	DET error code.
Value	(0x01U)
Description	Invalid pointer (NULL_PTR) passed to service.

5.3.2.2.37. FRIF_E_INV_TIMER_IDX

Purpose	DET error code.
Value	(0x05U)
Description	Invalid timer index offset passed to service.

5.3.2.2.38. FRIF_E_INV_TXPDUID

Purpose	DET error code.

Value	(0x06U)
Description	Invalid tx pdu index offset passed to service.

5.3.2.2.39. FRIF_E_NOT_INITIALIZED

Purpose	DET error code.
Value	(0x08U)
Description	Service was called although module was not initialized.

5.3.2.2.40. FRIF_GETABSOLUTETIMERIRQSTATUS_SERVICE_ID

Purpose	
Value	(0x1FU)

5.3.2.2.41. FRIF_GETCHANNELSTATUS_SERVICE_ID

Purpose	
Value	(0x26U)

5.3.2.2.42. FRIF_GETCLOCKCORRECTION_SERVICE_ID

Purpose	
Value	(0x29U)

5.3.2.2.43. FRIF_GETCONTROLLERERRORSTATUS_SERVICE_ID

Purpose	
Value	(0x83U)

5.3.2.2.44. FRIF_GETCYCLELENGTH_SERVICE_ID

Purpose	

Value	(0x3AU)

5.3.2.2.45. FRIF_GETGLOBALTIME_SERVICE_ID

Purpose	
Value	(0x0EU)

5.3.2.2.46. FRIF_GETIRQSTATUS_SERVICE_ID

Purpose	
Value	(0x89U)

5.3.2.2.47. FRIF_GETMACROTICKDURATION_SERVICE_ID

Purpose	
Value	(0x31U)

5.3.2.2.48. FRIF_GETMACROTICKSPERCYCLE_SERVICE_ID

Purpose	
Value	(0x11U)

5.3.2.2.49. FRIF_GETNMVECTOR_SERVICE_ID

Purpose	
Value	(0x0FU)

5.3.2.2.50. FRIF_GETNUMOFSTARTUPFRAMES_SERVICE_ID

Purpose	
Value	(0x34U)

5.3.2.2.51. FRIF_GETPOCSTATUS_SERVICE_ID

Purpose	
Value	(0x0DU)

5.3.2.2.52. FRIF_GETRELATIVETIMERIRQSTATUS_SERVICE_ID

Purpose	
Value	(0x96U)

5.3.2.2.53. FRIF_GETSTATE_SERVICE_ID

Purpose	
Value	(0x07U)

5.3.2.2.54. FRIF_GETSYNCFRAMELIST_SERVICE_ID

Purpose	
Value	(0x2AU)

5.3.2.2.55. FRIF_GETSYNCSTATE_SERVICE_ID

Purpose	
Value	(0x99U)

5.3.2.2.56. FRIF_GETTRANSCEIVERERROR_SERVICE_ID

Purpose	
Value	(0x35U)

5.3.2.2.57. FRIF_GETTRANSCEIVERMODE_SERVICE_ID

Purpose	

Value	(0x14U)

5.3.2.2.58. FRIF_GETTRANSCEIVERWUREASON_SERVICE_ID

Purpose	
Value	(0x15U)

5.3.2.2.59. FRIF_GETVERSIONINFO_SERVICE_ID

Purpose	
Value	(0x01U)

5.3.2.2.60. FRIF_GETWAKEUPRXSTATUS_SERVICE_ID

Purpose	
Value	(0x2BU)

5.3.2.2.61. FRIF_HALTCOMMUNICATION_SERVICE_ID

Purpose	
Value	(0x05U)

5.3.2.2.62. FRIF_INDICATE_FRAME

Purpose	
Value	(0x05U)

5.3.2.2.63. FRIF_INIT_SERVICE_ID

Purpose	
Value	(0x02U)

5.3.2.2.64. FRIF_INSTANCE_ID

Purpose	AUTOSAR module identification.
Value	0U

5.3.2.2.65. FRIF_INVALID_ENTRY

Purpose	
Value	(0xFFU)

5.3.2.2.66. FRIF_IRQ_CYCLE_START

Purpose	IRQ source.
Value	(0x01U)
Description	Cycle start interrupt source index.

5.3.2.2.67. FRIF_IRQ_DYNAMICSEGMENT_START

Purpose	IRQ source.
Value	(0x02U)
Description	Dynamic segment start interrupt source index.

5.3.2.2.68. FRIF_IRQ_NMVECTOR_CHANGED

Purpose	IRQ source.
Value	(0x03U)
Description	Change of NM-Vector interrupt source index.

5.3.2.2.69. FRIF_IRQ_STARTUP_COMPLETED

Purpose	IRQ source.
Value	(0x00U)
Description	Startup completed interrupt source index.

5.3.2.2.70. FRIF_JOBLISTEXECUTION_SERVICE_ID

Purpose	
Value	(0x32U)

5.3.2.2.71. FRIF_MAINFUNCTION_SERVICE_ID

Purpose	
Value	(0x27U)

5.3.2.2.72. FRIF_PREPARE_FRAME

Purpose	
Value	(0x07U)

5.3.2.2.73. FRIF_PREPARE_SIMPLE_FRAME

Purpose	
Value	FRIF_PREPARE_FRAME

5.3.2.2.74. FRIF_READCCCONFIG_SERVICE_ID

Purpose	
Value	(0x3BU)

5.3.2.2.75. FRIF_RECEIVE_FRAME

Purpose	
Value	(0x02U)

5.3.2.2.76. FRIF_RECEIVE_INDICATE_FRAME

Purpose	

Value	(0x06U)

5.3.2.2.77. FRIF_RECEIVE_INDICATE_RXFIFO

Purpose	
Value	(0x0DU)

5.3.2.2.78. FRIF_RECEIVE_INDICATE_SIMPLE_FRAME

Purpose	
Value	(0x0CU)

5.3.2.2.79. FRIF_RECONFIGLPDU_SERVICE_ID

Purpose	
Value	(0x00U)

5.3.2.2.80. FRIF_SENDMTS_SERVICE_ID

Purpose	
Value	(0x90U)

5.3.2.2.81. FRIF_SENDWUP_SERVICE_ID

Purpose	
Value	(0x0AU)

5.3.2.2.82. FRIF_SETABSOLUTETIMER_SERVICE_ID

Purpose	
Value	(0x19U)

5.3.2.2.83. FRIF_SETEXTSYNC_SERVICE_ID

Purpose	
Value	(0x9AU)

5.3.2.2.84. FRIF_SETRELATIVETIMER_SERVICE_ID

Purpose	
Value	(0x93U)

5.3.2.2.85. FRIF_SETSTATE_SERVICE_ID

Purpose	
Value	(0x08U)

5.3.2.2.86. FRIF_SETTRANSCEIVERMODE_SERVICE_ID

Purpose	
Value	(0x13U)

5.3.2.2.87. FRIF_SETWAKEUPCHANNEL_SERVICE_ID

Purpose	
Value	(0x09U)

5.3.2.2.88. FRIF_STARTCOMMUNICATION_SERVICE_ID

Purpose	
Value	(0x04U)

5.3.2.2.89. FRIF_STOPMTS_SERVICE_ID

Purpose	

Value	(0x91U)

5.3.2.2.90. FRIF_TRANSMIT_FRAME

Purpose	
Value	(0x00U)

5.3.2.2.91. FRIF_TRANSMIT_FRAME_NONE_MODE

Purpose	
Value	(0x08U)

5.3.2.2.92. FRIF_TRANSMIT_SERVICE_ID

Purpose	
Value	(0x12U)

5.3.2.2.93. FRIF_TRANSMIT_SIMPLE_FRAME

Purpose	
Value	(0x09U)

5.3.2.2.94. FRIF_TRANSMIT_SIMPLE_FRAME_NONE_MODE

Purpose	
Value	(0x0BU)

5.3.2.2.95. FRIF_UNKNOWN_SERVICE_ID

Purpose	
Value	(0xFFU)

5.3.2.3. Objects

5.3.2.3.1. FrIf_ClstRuntimeData

Purpose	
Type	FrIf_ClstRuntimeDataType

5.3.2.3.2. FrIf_FirstASR43OwnerIdx

Purpose	
Type	const uint8

5.3.2.3.3. FrIf_FrFuncPtr

Purpose	
Type	const FrIf_FrFunctionType

5.3.2.3.4. FrIf_LcfgSignature

Purpose	
Type	const uint32

5.3.2.3.5. FrIf_Mem

Purpose	
Type	TS_MaxAlignedType

5.3.2.3.6. FrIf_PduOwnerFuncASR40Ptr

Purpose	
Type	const FrIf_PduOwnerFunctionASR40Type

5.3.2.3.7. FrIf_PduOwnerFuncASR43Ptr

Purpose	
Type	const FrIf_PduOwnerFunctionASR43Type

5.3.2.3.8. FrIf_TrcvFuncPtr

Purpose	
Type	const FrIf_TrcvFunctionType

5.3.2.4. Functions

5.3.2.4.1. FrIf_AbortCommunication

Purpose	Invokes CHI command 'FREEZE'.	
Synopsis	Std_ReturnType **FrIf_AbortCommunication** (uint8 FrIf_CtrlIdx);	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.2. FrIf_AckAbsoluteTimerIRQ

Purpose	Acknowledges the absolute timer IRQ.

Synopsis	`Std_ReturnType` **`FrIf_AckAbsoluteTimerIRQ`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_AbsTimerIdx );`	
Service ID	0x21	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_AbsTimerIdx`	Absolute timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.3. FrIf_AckIRQ

Purpose	Acknowledges an interrupt of the FlexRay controller.	
Synopsis	`Std_ReturnType` **`FrIf_AckIRQ`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_-IRQIdx );`	
Service ID	0x86	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_IRQIdx`	Interrupt source index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.4. FrIf_AckRelativeTimerIRQ

Purpose	Acknowledges the relative timer IRQ.
Synopsis	`Std_ReturnType` **`FrIf_AckRelativeTimerIRQ`** `( uint8 FrIf_CtrlIdx ,` `uint8 FrIf_RelTimerIdx );`
Service ID	0x97
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx` — FlexRay controller index.
	`FrIf_RelTimerIdx` — Relative timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.5. FrIf_AllSlots

Purpose	Invokes CHI command 'ALL_SLOTS'.
Synopsis	`Std_ReturnType` **`FrIf_AllSlots`** `( uint8 FrIf_CtrlIdx );`
Service ID	0x33
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx` — FlexRay controller index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.6. FrIf_AllowColdstart

Purpose	Invokes CHI command 'ALLOW_COLDSTART'.
Synopsis	`Std_ReturnType` **`FrIf_AllowColdstart`** `( uint8 FrIf_CtrlIdx );`
Service ID	0x10
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx` FlexRay controller index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.7. FrIf_CancelAbsoluteTimer

Purpose	Cancels the absolute timer.
Synopsis	`Std_ReturnType` **`FrIf_CancelAbsoluteTimer`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_AbsTimerIdx );`
Service ID	0x1B
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx` FlexRay controller index.
	`FrIf_AbsTimerIdx` Absolute timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.8. FrIf_CancelRelativeTimer

Purpose	Cancels the relative timer.	
Synopsis	Std_ReturnType **FrIf_CancelRelativeTimer** (uint8 FrIf_CtrlIdx , uint8 FrIf_RelTimerIdx);	
Service ID	0x94	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_RelTimerIdx	Relative timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.9. FrIf_CheckMTS

Purpose	Reads the MTS receive status information.	
Synopsis	Std_ReturnType **FrIf_CheckMTS** (uint8 FrIf_CtrlIdx , Fr_Channel-Type FrIf_ChnlIdx , Fr_MTSStatusType * FrIf_MTSStatusPtr);	
Service ID	0x92	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	Channel the MTS status shall read from.
Parameters (out)	FrIf_MTSStatusPtr	Address the MTS status is written to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service.	

	If DET is enabled and Frlf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and Frlf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.10. Frlf_CheckWakeupByTransceiver

Purpose	Service checks for wakeup events by the transceiver.	
Synopsis	void **FrIf_CheckWakeupByTransceiver** (uint8 FrIf_CtrlIdx , Fr_-ChannelType FrIf_ChnlIdx);	
Service ID	0x39	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different value pairs of Frlf_CtrlIdx/Frlf_ChnlIdx	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	FlexRay channel index.
Description	This service translates the Frlf_CtrlIdx and Frlf_ChnlIdx to the configured FlexRay transceiver driver and FlexRay transceiver driver transceiver index and calls the equivalent transceiver driver service. If DET is enabled and Frlf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and Frlf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and Frlf_ChnlIdx contains an invalid value FRIF_E_INV_CHNL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.11. Frlf_ClearTransceiverWakeup

Purpose	Clears the transceiver driver's wakeup information.
Synopsis	Std_ReturnType **FrIf_ClearTransceiverWakeup** (uint8 FrIf_CtrlIdx , Fr_ChannelType FrIf_ChnlIdx);
Service ID	0x18
Sync/Async	Synchronous
Reentrancy	Re-entrant for different value pairs of Frlf_CtrlIdx/Frlf_ChnlIdx only

Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	FlexRay channel index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx and FrIf_ChnlIdx to the configured FlexRay transceiver driver and FlexRay transceiver driver transceiver index and calls the equivalent transceiver driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_ChnlIdx contains an invalid value FRIF_E_INV_CHNL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.12. FrIf_ControllerInit

Purpose	Initializes FlexRay controller.
Synopsis	Std_ReturnType **FrIf_ControllerInit** (uint8 FrIf_CtrlIdx);
Service ID	0x03
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	FrIf_CtrlIdx Index of FlexRay controller to initialize.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.13. FrIf_DisableAbsoluteTimerIRQ

Purpose	Disables the absolute timer IRQ.

Synopsis	`Std_ReturnType` **`FrIf_DisableAbsoluteTimerIRQ`** `( uint8 FrIf_Ctr-lIdx , uint8 FrIf_AbsTimerIdx );`	
Service ID	0x23	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_AbsTimerIdx`	Absolute timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.14. FrIf_DisableIRQ

Purpose	Disables an interrupt of the FlexRay controller.	
Synopsis	`Std_ReturnType` **`FrIf_DisableIRQ`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_IRQIdx );`	
Service ID	0x88	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_IRQIdx`	Interrupt source index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.15. FrIf_DisableLPdu

Purpose	Dynamically disables a LPdu.	
Synopsis	`Std_ReturnType` **`FrIf_DisableLPdu`** `( uint8 FrIf_CtrlIdx , uint16 FrIf_LPduIdx );`	
Service ID	0x28	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_LPduIdx`	LPdu index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.16. FrIf_DisableRelativeTimerIRQ

Purpose	Disables the relative timer IRQ.	
Synopsis	`Std_ReturnType` **`FrIf_DisableRelativeTimerIRQ`** `( uint8 FrIf_Ctr-lIdx , uint8 FrIf_RelTimerIdx );`	
Service ID	0x98	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_RelTimerIdx`	Relative timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned.	

	If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.17. FrIf_DisableTransceiverBranch

Purpose	Wraps the FlexRay Transceiver Driver API function FrTrcv_DisableTransceiver-Branch.	
Synopsis	Std_ReturnType **FrIf_DisableTransceiverBranch** (uint8 FrIf_Ctr-lIdx , Fr_ChannelType FrIf_ChnlIdx , uint8 FrIf_BranchIdx);	
Service ID	0x37	
Sync/Async	Synchronous	
Reentrancy	Re	
Parameters (in)	FrIf_CtrlIdx	(in) FlexRay controller index.
	FrIf_ChnlIdx	(in) FlexRay channel index.
	FrIf_BranchIdx	(in) FlexRay active star branch index.
Parameters (in,out)	FrIf_CtrlIdx	(in) FlexRay controller index.
	FrIf_ChnlIdx	(in) FlexRay channel index.
	FrIf_BranchIdx	(in) FlexRay active star branch index.
Return Value	E_OK	Function serviced successfully.
E_NOT_OK	Function execution failed.	

5.3.2.4.18. FrIf_DispatchComOps

Purpose	Dispatches the communication operations within a communication job.	
Synopsis	Std_ReturnType **FrIf_DispatchComOps** (uint8 FrIf_ClstIdx , uint16 FrIf_JobIdx , uint8 FrIf_Cycle);	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same FlexRay cluster	
Parameters (in)	FrIf_ClstIdx	FlexRay cluster index.
	FrIf_JobIdx	FlexRay job index.
	FrIf_Cycle	FlexRay cycle number.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned.	

If DET is enabled and FrIf_JobIdx contains an invalid value FRIF_E_INV_JOB_IDX is reported to DET and E_NOT_OK returned.

If DET is enabled and FrIf_ClstIdx contains an invalid value FRIF_E_INV_CLST_IDX is reported to DET and E_NOT_OK returned.

If DET is enabled and FrIf_Cycle contains an invalid value FRIF_E_INV_CYCLE is reported to DET and E_NOT_OK returned.

5.3.2.4.19. FrIf_EnableAbsoluteTimerIRQ

Purpose	Enables the absolute timer IRQ.	
Synopsis	`Std_ReturnType` **`FrIf_EnableAbsoluteTimerIRQ`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_AbsTimerIdx );`	
Service ID	0x1D	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_AbsTimerIdx`	Absolute timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.20. FrIf_EnableBusMirroring

Purpose	Enables or disables mirroring for all FlexRay controllers connected to the addressed FlexRay cluster.
Synopsis	`Std_ReturnType` **`FrIf_EnableBusMirroring`** `( uint8 FrIf_ClstIdx , boolean FrIf_MirroringActive );`
Service ID	0x4B
Sync/Async	Synchronous

Reentrancy	Re-entrant	
Parameters (in)	`FrIf_ClstIdx`	(in) Index of the FlexRay cluster to address.
	`FrIf_MirroringActive`	(in) TRUE: Mirror_ReportFlexRayFrame will be called for each frame received or transmitted on the addressed FlexRay CC. FALSE: Mirror_ReportFlexRayFrame will not be called for the addressed FlexRay CC.
Parameters (in,out)	`FrIf_ClstIdx`	(in) Index of the FlexRay cluster to address.
	`FrIf_MirroringActive`	(in) TRUE: Mirror_ReportFlexRayFrame will be called for each frame received or transmitted on the addressed FlexRay CC. FALSE: Mirror_ReportFlexRayFrame will not be called for the addressed FlexRay CC.
Return Value	`E_OK`	Mirroring mode was changed.
`E_NOT_OK`	Wrong FrIf_CtrlIdx, or mirroring is globally disabled (see FrIfBusMirroringSupport).	

5.3.2.4.21. FrIf_EnableIRQ

Purpose	Enables an interrupt of the FlexRay controller.	
Synopsis	`Std_ReturnType` **`FrIf_EnableIRQ`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_IRQIdx );`	
Service ID	0x87	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_IRQIdx`	Interrupt source index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned.	

	If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.22. FrIf_EnableRelativeTimerIRQ

Purpose	Enables the relative timer IRQ.	
Synopsis	Std_ReturnType **FrIf_EnableRelativeTimerIRQ** (uint8 FrIf_CtrlIdx , uint8 FrIf_RelTimerIdx);	
Service ID	0x95	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_RelTimerIdx	Relative timer index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.23. FrIf_EnableTransceiverBranch

Purpose	Wraps the FlexRay Transceiver Driver API function FrTrcv_EnableTransceiverBranch.	
Synopsis	Std_ReturnType **FrIf_EnableTransceiverBranch** (uint8 FrIf_CtrlIdx , Fr_ChannelType FrIf_ChnlIdx , uint8 FrIf_BranchIdx);	
Service ID	0x36	
Sync/Async	Synchronous	
Reentrancy	Re-entrant	
Parameters (in)	FrIf_CtrlIdx	(in) FlexRay controller index.
	FrIf_ChnlIdx	(in) FlexRay channel index.
	FrIf_BranchIdx	(in) FlexRay active star branch index.

Parameters (in,out)	FrIf_CtrlIdx	(in) FlexRay controller index.
	FrIf_ChnlIdx	(in) FlexRay channel index.
	FrIf_BranchIdx	(in) FlexRay active star branch index.
Return Value	E_OK	Function serviced successfully.
E_NOT_OK	Function execution failed.	

5.3.2.4.24. FrIf_GetAbsoluteTimerIRQStatus

Purpose	Reads the absolute timer IRQ status.
Synopsis	Std_ReturnType **FrIf_GetAbsoluteTimerIRQStatus** (uint8 FrIf_CtrlIdx , uint8 FrIf_AbsTimerIdx , boolean * FrIf_IRQStatusPtr);
Service ID	0x1F
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_AbsTimerIdx	Absolute timer index.
Parameters (out)	FrIf_IRQStatusPtr	Address the timer IRQ status is stored to
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.25. FrIf_GetChannelStatus

Purpose	Returns the FlexRay aggregated channel status.
Synopsis	Std_ReturnType **FrIf_GetChannelStatus** (uint8 FrIf_CtrlIdx , uint16 * FrIf_ChannelAStatusPtr , uint16 * FrIf_ChannelBStatusPtr);
Service ID	0x26
Sync/Async	Synchronous

Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Parameters (out)	`FrIf_ChannelStatusPtr`	Address to write the aggregated channel status to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.26. FrIf_GetClockCorrection

Purpose	Returns the FlexRay clock correction terms.	
Synopsis	`Std_ReturnType` **`FrIf_GetClockCorrection`** `( uint8 FrIf_CtrlIdx , sint16 * FrIf_RateCorrectionPtr , sint32 * FrIf_OffsetCorrectionPtr );`	
Service ID	0x29	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same FlexRay controller	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Parameters (out)	`FrIf_RateCorrectionPtr`	Address to write the rate correction value to.
	`FrIf_OffsetCorrectionPtr`	Address to write the offset correction value to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service reads the FlexRay clock correction terms and writes them into *FrIf_Rate-CorrectionPtr and *FrIf_RateCorrectionPtr. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.27. FrIf_GetControllerErrorStatus

Purpose	Returns a FlexRay controller error status.
Synopsis	`Std_ReturnType` **`FrIf_GetControllerErrorStatus`** `( uint8 FrIf_CtrlIdx , uint16 * FrIf_ControllerErrorStatusPtr );`
Service ID	0x83
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Parameters (out)	`FrIf_ControllerErrorStatusPtr`	Address to write the error status to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.28. FrIf_GetCycleLength

Purpose	This API returns the configured time of the configuration parameter "GdCycle" in nanoseconds for the FlexRay controller with index FrIf_CtrlIdx.	
Synopsis	`uint32` **`FrIf_GetCycleLength`** `( uint8 FrIf_CtrlIdx );`	
Service ID	0x3A	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_TxPduId only	
Parameters (in)	`FrIf_CtrlIdx`	(in) Index of the FlexRay CC to address.
Parameters (in,out)	`FrIf_CtrlIdx`	(in) Index of the FlexRay CC to address.
Return Value		
`Time`	in unit of nanoseconds	

5.3.2.4.29. FrIf_GetGlobalTime

Purpose	Reads the FlexRay clusters global time.

Synopsis	`Std_ReturnType` **`FrIf_GetGlobalTime`** `( uint8 FrIf_CtrlIdx , uint8 * FrIf_CyclePtr , uint16 * FrIf_MacroTickPtr );`	
Service ID	0x0E	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Parameters (out)	`FrIf_CyclePtr`	Address to write the current cycle counter value to.
	`FrIf_MacroTickPtr`	Address to write the current macrotick counter value to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.30. FrIf_GetIRQStatus

Purpose	Returns the interrupt status of the FlexRay controller.	
Synopsis	`Std_ReturnType` **`FrIf_GetIRQStatus`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_IRQIdx , boolean * FrIf_IRQStatusPtr );`	
Service ID	0x89	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_IRQIdx`	Interrupt source index.
Parameters (out)	`FrIf_IRQStatusPtr`	Address to write the IRQ interrupt status.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned.	

	If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.31. FrIf_GetMacrotickDuration

Purpose	Get macrotick duration.	
Synopsis	uint16 **FrIf_GetMacrotickDuration** (uint8 FrIf_CtrlIdx);	
Service ID	0x31	
Sync/Async	Synchronous	
Reentrancy	Re-entrant	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
Return Value	Macrotick length in units of nanoseconds	
Description	This service returns the configured macrotick time of the cluster that the controller requested is part of. The time returned is in units of nanoseconds. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_INITIALIZED is reported to DET and 0 returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and 0 returned.	

5.3.2.4.32. FrIf_GetMacroticksPerCycle

Purpose	Retunrs macroticks per communication cycle.	
Synopsis	uint16 **FrIf_GetMacroticksPerCycle** (uint8 FrIf_CtrlIdx);	
Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same FlexRay controller	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
Return Value	Duration of a communication cycle in units of macroticks.	
Description	This service returns the number of macroticks a single FlexRay communication cycle of the FlexRay cluster the requested controller is joining consists of. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_INITIALIZED is reported to DET and E_NOT_OK returned.	

	If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.33. FrIf_GetNmVector

Purpose	Reads the FlexRay NM-Vector of the last FlexRay cycle.	
Synopsis	Std_ReturnType **FrIf_GetNmVector** (uint8 FrIf_CtrlIdx , uint8 * FrIf_NmVectorPtr);	
Service ID	0x0F	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same FlexRay controller	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
Parameters (out)	FrIf_NmVectorPtr	Address to write the Nm-Vector to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the driver service Fr_GetNmVector(). If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_NmVectorPtr is NULL_PTR, FRIF_E_INV_POINTER is reported to DET and E_NOT_OK returned.	

5.3.2.4.34. FrIf_GetNumOfStartupFrames

Purpose	Wraps the FlexRay Driver API function Fr_GetNumOfStartupFrames and gets the current number of startup frames seen on the cluster.	
Synopsis	Std_ReturnType **FrIf_GetNumOfStartupFrames** (uint8 FrIf_CtrlIdx , uint8 * FrIf_NumOfStartupFramesPtr);	
Service ID	0x34	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	FrIf_CtrlIdx	(in) FlexRay controller index.

	FrIf_NumOfStartupFramesPtr	(out) FlexRay active star branch index.
Parameters (in,out)	FrIf_CtrlIdx	(in) FlexRay controller index.
	FrIf_NumOfStartupFramesPtr	(out) FlexRay active star branch index.
Return Value	E_OK	Function serviced successfully.
E_NOT_OK	Function execution failed.	

5.3.2.4.35. FrIf_GetPOCStatus

Purpose	Reads FlexRay CC POC-status.
Synopsis	Std_ReturnType **FrIf_GetPOCStatus** (uint8 FrIf_CtrlIdx , Fr_-POCStatusType * FrIf_POCStatusPtr);
Service ID	0x0D
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
Parameters (out)	FrIf_POCStatusPtr	Address the POC-status is written to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.36. FrIf_GetRelativeTimerIRQStatus

Purpose	Reads the relative timer IRQ status.
Synopsis	Std_ReturnType **FrIf_GetRelativeTimerIRQStatus** (uint8 FrIf_Ctr-lIdx , uint8 FrIf_RelTimerIdx , boolean * FrIf_IRQStatusPtr);
Service ID	0x96
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.

	FrIf_RelTimerIdx	Relative timer index.
Parameters (out)	FrIf_IRQStatusPtr	Address the timer IRQ status is stored to
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.37. FrIf_GetState

Purpose	Reads the cluster's state.
Synopsis	Std_ReturnType **FrIf_GetState** (uint8 FrIf_ClstIdx , FrIf_State-Type * FrIf_StatePtr);
Service ID	0x07
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only

Parameters (in)	FrIf_ClstIdx	FlexRay cluster index.
Parameters (out)	FrIf_StatePtr	Address the current state is written to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service returns the current state of the FrIf state machine of a cluster. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_ClstIdx contains an invalid value FRIF_E_INV_CLST_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_StatePtr is NULL_PTR, FRIF_E_INV_POINTER is reported to DET and E_NOT_OK returned.	

5.3.2.4.38. FrIf_GetSyncFrameList

Purpose	Returns a list of sync frames.

Synopsis	`Std_ReturnType` **`FrIf_GetSyncFrameList`** `( uint8 FrIf_CtrlIdx`
	`, uint8 FrIf_ListSize , uint16 * FrIf_ChannelAEvenListPtr ,`
	`uint16 * FrIf_ChannelBEvenListPtr , uint16 * FrIf_ChannelAOd-`
	`dListPtr , uint16 * FrIf_ChannelBOddListPtr );`	

Service ID	0x2A	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same FlexRay controller	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_ListSize`	Size of list passed to the output pointers.
Parameters (out)	`FrIf_ChannelAEvenListPtr`	Address to write the list of even sync frames of channel A.
	`FrIf_ChannelBEvenListPtr`	Address to write the list of even sync frames of channel B.
	`FrIf_ChannelAOddListPtr`	Address to write the list of odd sync frames of channel A.
	`FrIf_ChannelBOddListPtr`	Address to write the list of odd sync frames of channel B.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service writes a list of sync frames observed in the last even/odd communication cycle couple into *FrIf_ChannelAEvenListPtr, *FrIf_ChannelBEvenListPtr, *FrIf_ChannelAOddListPtr and *FrIf_ChannelBOddListPtr If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.39. FrIf_GetSyncState

Purpose	Reads the FlexRay CC synchronization state.
Synopsis	`Std_ReturnType` **`FrIf_GetSyncState`** `( uint8 FrIf_CtrlIdx , Fr_-` `SyncStateType * FrIf_SyncStatePtr );`
Service ID	0x99
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only

Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
Parameters (out)	FrIf_SyncStatePtr	Address the synchronization state is written to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.40. FrIf_GetTransceiverError

Purpose	Returns the transceivers error status.	
Synopsis	`Std_ReturnType FrIf_GetTransceiverError ( uint8 FrIf_CtrlIdx , Fr_ChannelType FrIf_ChnlIdx , uint8 FrIf_BranchIdx , uint32 * FrIf_BusErrorState );`	
Service ID	0x35	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same transceiver	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	FlexRay channel index.
	FrIf_BranchIdx	FlexRay branch index (active star).
Parameters (out)	FrIf_BusErrorState	Address to write the transceiver error status to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service returns the FlexRay transceiver's error status. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_ChnlIdx contains an invalid value FRIF_E_INV_CHNL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.41. FrIf_GetTransceiverMode

Purpose	Receive the transceiver's current mode.
Synopsis	Std_ReturnType **FrIf_GetTransceiverMode** (uint8 FrIf_CtrlIdx , Fr_ChannelType FrIf_ChnlIdx , FrTrcv_TrcvModeType * FrIf_Trcv-ModePtr);
Service ID	0x14
Sync/Async	Synchronous
Reentrancy	Re-entrant

Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	FlexRay channel index.
Parameters (out)	FrIf_TrcvModePtr	Address to write the transceiver mode to.

Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx and FrIf_ChnlIdx to the configured FlexRay transceiver driver and FlexRay transceiver driver transceiver index and calls the equivalent transceiver driver service.

If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned.

If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

If DET is enabled and FrIf_ChnlIdx contains an invalid value FRIF_E_INV_CHNL_IDX is reported to DET and E_NOT_OK returned. |

5.3.2.4.42. FrIf_GetTransceiverWUReason

Purpose	Returns the transceivers wakeup reason.
Synopsis	Std_ReturnType **FrIf_GetTransceiverWUReason** (uint8 FrIf_Ctr-lIdx , Fr_ChannelType FrIf_ChnlIdx , FrTrcv_TrcvWUReasonType * FrIf_TrcvWUReasonPtr);
Service ID	0x15
Sync/Async	Synchronous
Reentrancy	Re-entrant for different value pairs of FrIf_CtrlIdx/FrIf_ChnlIdx only

Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	FlexRay channel index.
Parameters (out)	FrIf_TrcvWUReasonPtr	Address to write the transceiver WU Reason to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx and FrIf_ChnlIdx to the configured FlexRay transceiver driver and FlexRay transceiver driver transceiver index and calls the equivalent transceiver driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_ChnlIdx contains an invalid value FRIF_E_INV_CHNL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.43. FrIf_GetVersionInfo

Purpose	Get version information of the FlexRay Interface.
Synopsis	void **FrIf_GetVersionInfo** (Std_VersionInfoType * FrIf_VersionInfoPtr);
Service ID	0x01
Sync/Async	Synchronous
Reentrancy	Re-entrant
Parameters (out)	FrIf_VersionInfoPtr	Pointer where to store the version information of this module.
Description	This service returns the version information of this module. The version information includes:	

This service returns the version information of this module. The version information includes:

▶ Module Id

▶ Vendor Id

▶ Vendor specific version numbers

If DET is enabled and VersionInfoPtr is NULL_PTR, FRIF_E_INV_POINTER is reported to DET.

5.3.2.4.44. FrIf_GetWakeupRxStatus

Purpose	Reads the wakeup rx status of the FlexRay controller.	
Synopsis	`Std_ReturnType` **`FrIf_GetWakeupRxStatus`** `( uint8 FrIf_CtrlIdx , uint8 * FrIf_WakeupRxStatusPtr );`	
Service ID	0x2B	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Parameters (out)	`FrIf_WakeupRxStatusPtr`	Address to write the wakeup rx status to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.45. FrIf_HaltCommunication

Purpose	Invokes CHI command 'HALT'.	
Synopsis	`Std_ReturnType` **`FrIf_HaltCommunication`** `( uint8 FrIf_CtrlIdx );`	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.46. FrIf_Init

Purpose	Initializes the module.
Synopsis	void **FrIf_Init** (const FrIf_ConfigType * FrIf_ConfigPtr);
Service ID	0x02
Sync/Async	Synchronous
Reentrancy	Non re-entrant
Parameters (in)	FrIf_ConfigPtr	(in) Address of module post-build-time configuration
Parameters (in,out)	FrIf_ConfigPtr	(in) Address of module post-build-time configuration
Description	This service initializes the module and registers the post-build-time configuration passed as argument for usage by other service of this module. If FrIf_ConfigPtr is NULL_PTR, FRIF_E_INV_POINTER is reported to DET.

5.3.2.4.47. FrIf_IsValidConfig

Purpose	Validate configuration.
Synopsis	Std_ReturnType **FrIf_IsValidConfig** (const void * voidConfigPtr);
Sync/Async	Synchronous
Reentrancy	Reentrant
Return Value	E_OK if the given module configurations is valid otherwise E_NOT_OK.
Description	Checks if the post build configuration fits to the link time configuration part. {}

5.3.2.4.48. FrIf_JobListExec

Purpose	Cluster independent FrIf_JobListExec_# implementation.
Synopsis	void **FrIf_JobListExec** (uint8 FrIf_ClstIdx);

Service ID	0x32	
Sync/Async	Synchronous	
Reentrancy	Non Re-entrant	
Parameters (in)	`FrIf_ClstIdx`	FlexRay cluster index.
Description	This service executes the joblist and the contained communication operations. If this service detects that it is not properly executed (in time), it calls DEM and reports FRIF_E_JLE_SYNC with status DEM_EVENT_STATUS_FAILED. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET.	

5.3.2.4.49. FrIf_MainFunction

Purpose	Cluster independent FrIf_MainFunction_# implementation.	
Synopsis	`void FrIf_MainFunction ( uint8 FrIf_ClstIdx );`	
Service ID	0x27	
Sync/Async	Synchronous	
Reentrancy	Non Re-entrant	
Production Errors	▸ FRIF_E_ACS_CH_A: thrown, if any of the error bits in ACS of channel A is set ▸ FRIF_E_ACS_CH_B: thrown, if any of the error bits in ACS of channel B is set ▸ FRIF_E_NIT_CH_A: thrown, if any of the error bits in NIT of channel A is set ▸ FRIF_E_NIT_CH_B: thrown, if any of the error bits in NIT of channel B is set ▸ FRIF_E_SW_CH_A: thrown, if any of the error bits in SW of channel A is set ▸ FRIF_E_SW_CH_B: thrown, if any of the error bits in SW of channel B is set ▸ FRIF_E_JLE_SYNC: thrown, if Job List Execution lost synchronization to the FlexRay Global Time	
Parameters (in)	`FrIf_ClstIdx`	FlexRay cluster index.
Description	This service performs the joblist (re-)synchronization continously monitors the correct execution of the joblist. If this service detects that FrIf_JobListeExec is not properly executed, it calls DEM and reports FRIF_E_JLE_SYNC with status DEM_EVENT_STATUS_FAILED. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET.	

5.3.2.4.50. FrIf_ReadCCConfig

Purpose	Wraps the FlexRay Driver API function Fr_ReadCCConfig().	
Synopsis	`Std_ReturnType` **`FrIf_ReadCCConfig`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_ConfigParamIdx , uint32 * FrIf_ConfigParamValuePtr );`	
Service ID	0x3B	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	(in) FlexRay controller index.
	`FrIf_ConfigParamIdx`	(in) FlexRay configuration parameter to read.
	`FrIf_ConfigParamValuePtr`	(out) Pointer to the localtion where output value will be stored.
Parameters (in,out)	`FrIf_CtrlIdx`	(in) FlexRay controller index.
	`FrIf_ConfigParamIdx`	(in) FlexRay configuration parameter to read.
	`FrIf_ConfigParamValuePtr`	(out) Pointer to the localtion where output value will be stored.
Return Value	`E_OK`	Function serviced successfully.
`E_NOT_OK`	Function execution failed.	

5.3.2.4.51. FrIf_ReconfigLPdu

Purpose	Dynamically reconfigures a LPdu.	
Synopsis	`Std_ReturnType` **`FrIf_ReconfigLPdu`** `( uint8 FrIf_CtrlIdx , uint16 FrIf_LPduIdx , uint16 FrIf_FrameId , Fr_ChannelType FrIf_Chn-lIdx , uint8 FrIf_CycleRepetition , uint8 FrIf_CycleOffset , uint8 FrIf_PayloadLength , uint16 FrIf_HeaderCRC );`	
Service ID	0x00	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_LPduIdx`	LPdu index.

	`FrIf_FrameId`	FlexRay frame ID.
	`FrIf_ChnlIdx`	FlexRay Channel.
	`FrIf_CycleRepetition`	Cycle Repetition pattern.
	`FrIf_CycleOffset`	Cycle Offset part of the cycle filter mechanism FrIf_LPdu shall be configured to.
	`FrIf_PayloadLength`	Payload length in units of bytes.
	`FrIf_HeaderCRC`	FlexRay Header CRC.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.52. FrIf_SendMTS

Purpose	Initiates transmission of a MTS symbol.	
Synopsis	`Std_ReturnType `**`FrIf_SendMTS`**` ( uint8 FrIf_CtrlIdx , Fr_Channel-Type FrIf_ChnlIdx );`	
Service ID	0x90	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_ChnlIdx`	Channel the MTS should be transmitted on.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.53. FrIf_SendWUP

Purpose	Invokes CHI command 'WAKEUP'.
Synopsis	`Std_ReturnType` **`FrIf_SendWUP`** `( uint8 FrIf_CtrlIdx );`
Service ID	0x0A
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.54. FrIf_SetAbsoluteTimer

Purpose	Sets an absolute timer.	
Synopsis	`Std_ReturnType` **`FrIf_SetAbsoluteTimer`** `( uint8 FrIf_CtrlIdx , uint8 FrIf_AbsTimerIdx , uint8 FrIf_Cycle , uint16 FrIf_Offset );`	
Service ID	0x19	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_AbsTimerIdx`	Absolute timer index.
	`FrIf_Cycle`	Communication Cycle the alarm should elapse.
	`FrIf_Offset`	Macrotick offset the alarm should elapse.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service.	

If DET is enabled and Frlf_Init() was not called before this service FRIF_E_NOT_-
INITIALIZED is reported to DET and E_NOT_OK returned.

If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX
is reported to DET and E_NOT_OK returned.

5.3.2.4.55. FrIf_SetExtSync

Purpose	Initiates external clock synchronization.	
Synopsis	Std_ReturnType **FrIf_SetExtSync** (uint8 FrIf_CtrlIdx , Fr_Off-setCorrectionType FrIf_Offset , Fr_RateCorrectionType FrIf_Rate);	
Service ID	0x9A	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_Offset	Offset correction application mode.
	FrIf_Rate	Rate correction application mode.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and Frlf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.56. FrIf_SetRelativeTimer

Purpose	Sets a relative timer.
Synopsis	Std_ReturnType **FrIf_SetRelativeTimer** (uint8 FrIf_CtrlIdx , uint8 FrIf_RelTimerIdx , uint16 FrIf_Offset);
Service ID	0x93
Sync/Async	Synchronous

Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only	
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
	`FrIf_RelTimerIdx`	Relative timer index.
	`FrIf_Offset`	Macrotick offset the alarm should elapse.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.57. FrIf_SetState

Purpose	Sets the cluster state.	
Synopsis	`Std_ReturnType` **`FrIf_SetState`** `( uint8 FrIf_ClstIdx , FrIf_State-` `TransitionType FrIf_StateTransition );`	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant for same FlexRay controller	
Parameters (in)	`FrIf_StateTransition`	Requested transition.
	`FrIf_ClstIdx`	FlexRay cluster index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service sets the FrIf state (FRIF_STATE_ONLINE, FRIF_STATE_OFFLINE) of the respective FrIf cluster. In FRIF_STATE_ONLINE the joblist executes all communication operations. In FRIF_STATE_OFFLINE the joblist executes only the communication operation PREPARE_LPDU. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIF_ClstIdx contains an invalid value FRIF_E_INV_CLST_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.58. FrIf_SetTransceiverMode

Purpose	Sets the transceiver's mode.
Synopsis	Std_ReturnType **FrIf_SetTransceiverMode** (uint8 FrIf_CtrlIdx , Fr_ChannelType FrIf_ChnlIdx , FrTrcv_TrcvModeType FrIf_TrcvMode);
Service ID	0x13
Sync/Async	Synchronous
Reentrancy	Re-entrant for different value pairs of FrIf_CtrlIdx/FrIf_ChnlIdx only
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	FlexRay channel index.
	FrIf_TrcvMode	Transceiver Mode to set.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx and FrIf_ChnlIdx to the configured FlexRay transceiver driver and FlexRay transceiver driver transceiver index and calls the equivalent transceiver driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_ChnlIdx contains an invalid value FRIF_E_INV_CHNL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.59. FrIf_SetWakeupChannel

Purpose	Selects a channel for wakeup pattern transmission.
Synopsis	Std_ReturnType **FrIf_SetWakeupChannel** (uint8 FrIf_CtrlIdx , Fr_ChannelType FrIf_ChnlIdx);
Service ID	0x09
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	Channel the wakeup pattern should be transmitted on.

Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.

5.3.2.4.60. FrIf_StartCommunication

Purpose	Invokes CHI command 'RUN'.
Synopsis	`Std_ReturnType` **`FrIf_StartCommunication`** `( uint8 FrIf_CtrlIdx );`
Service ID	0x04
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only
Parameters (in)	`FrIf_CtrlIdx`	FlexRay controller index.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.61. FrIf_StopMTS

Purpose	Stops the transmission of a MTS symbol.
Synopsis	`Std_ReturnType` **`FrIf_StopMTS`** `( uint8 FrIf_CtrlIdx , Fr_Channel-Type FrIf_ChnlIdx );`
Service ID	0x91
Sync/Async	Synchronous
Reentrancy	Re-entrant for different values of FrIf_CtrlIdx only

Parameters (in)	FrIf_CtrlIdx	FlexRay controller index.
	FrIf_ChnlIdx	Channel the MTS transmission shall be stopped on.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service translates the FrIf_CtrlIdx to the configured FlexRay driver and FlexRay driver controller index and calls the equivalent driver service. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_CtrlIdx contains an invalid value FRIF_E_INV_CTRL_IDX is reported to DET and E_NOT_OK returned.	

5.3.2.4.62. FrIf_Transmit

Purpose	Transmits a Pdu.	
Synopsis	`Std_ReturnType` **`FrIf_Transmit`** `( PduIdType FrIf_TxPduId , const PduInfoType * FrIf_PduInfoPtr );`	
Service ID	0x12	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different values of FrIf_TxPduId only	
Parameters (in)	FrIf_TxPduId	Id of Pdu to be transmitted.
	FrIf_PduInfoPtr	Description of Pdu content to be transmitted.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	Tjis service allows upper layers to request the transmission of of Pdus via the FlexRay communication system. In case of an immediate transmission (single Pdu per Frame, no Update-Bit), the Pdu payload data is immediately passed to the FlexRay drivers transmit service. In case of decoupled transmission this service just remembers the transmission request and carries out the transmission within th econtext of the joblist execution. If DET is enabled and FrIf_Init() was not called before this service FRIF_E_NOT_-INITIALIZED is reported to DET and E_NOT_OK returned. If DET is enabled and FrIf_TxPduId contains an invalid value FRIF_E_INV_TXPDUID is reported to DET and E_NOT_OK returned.	

If DET is enabled and the Pdu to transmit is an immediate Tx-Pdu and FrIf_PduInfoPtr or FrIf_PduInfoPtr->SduDataPtr is NULL_PTR, FRIF_E_INV_POINTER is reported to DET and E_NOT_OK returned.

If DET is enabled and the Pdu to transmit is dynamic length Pdu and FrIf_PduInfoPtr is NULL_PTR, FRIF_E_INV_POINTER is reported to DET and E_NOT_OK returned.

5.3.3. Integration notes

5.3.3.1. Exclusive areas

This section describes the exclusive areas used by the `FrIf` module.

5.3.3.1.1. SCHM_FRIF_TX_ADMINISTRATION

Protected data structures	This exclusive area protects the Tx counter and flags which are accessed by potentially concurring functions FrIf_Transmit() and FrIf_JobListExec().
Recommended locking mechanism	The locking mechanism for this exclusive area can be disabled if at least one of the following conditions is true: ▶ FrIf_Transmit() does not interrupt FrIf_JobListExec() (and vice versa) ▶ no PDU is configured for decoupled transmission mode If the conditions listed above do not apply, the exclusive area shall be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to the section `Mapping exclusive areas in the basic software modules` in the `Integration notes` section for details.

5.3.3.1.2. SCHM_FRIF_JOBLIST_EXECUTION

Protected data structures	This exclusive area protects the joblist timeout monitoring counter which is accessed by potentially concurring functions FrIf_MainFunction() and FrIf_JobListExec().

Recommended locking mechanism	The locking mechanism for this exclusive area can be disabled if: ► FrIf_MainFunction() does not interrupt FrIf_JobListExec() (and vice versa) If the conditions listed above do not apply, the exclusive area shall be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to the section `Mapping exclusive areas in the basic software modules` in the `Integration notes` section for details.

5.3.3.2. Production errors

FRIF_E_ACS_CH_A	► FrIf_MainFunction
FRIF_E_ACS_CH_B	► FrIf_MainFunction
FRIF_E_JLE_SYNC	► FrIf_MainFunction
FRIF_E_NIT_CH_A	► FrIf_MainFunction
FRIF_E_NIT_CH_B	► FrIf_MainFunction
FRIF_E_SW_CH_A	► FrIf_MainFunction
FRIF_E_SW_CH_B	► FrIf_MainFunction

5.3.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
CODE
CONST_32
CONST_8
VAR_CLEARED_UNSPECIFIED

| VAR_FAST_INIT_UNSPECIFIED |
| VAR_INIT_8 |
| CONFIG_DATA_UNSPECIFIED |
| CONST_UNSPECIFIED |

5.3.3.4. Integration requirements

WARNING	Integration requirements list is not exhaustive
	The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the FrIf module.

5.4. FrNm

5.4.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
FrNmChannelConfig	1..1	This container contains all configuration parameters of FlexRay NM configured from the channel perspective.
FrNmGeneral	1..1	
FrNmGlobalConfig	1..1	This container contains all global configuration parameters for the FrNm module.
FrNmDefensiveProgramming	1..1	**Label:** Defensive Programming Options Parameters for defensive programming
PublishedInformation	1..1	**Label:** EB Published Information

Containers included		
		Additional published parameters not covered by Common-PublishedInformation container.

Parameters included	
Parameter name	**Multiplicity**
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT	
Label	Configuration Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantPostBuild	
Range	VariantPostBuild	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
Label	AUTOSAR Major Version

Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	4
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArMinorVersion
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	5

Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	16
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	8
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	32
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID

Description	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

5.4.1.2. FrNmChannelConfig

Containers included		
Container name	**Multiplicity**	**Description**
FrNmChannel	1..n	This container contains the configuration parameters for a FlexRay NM Channel.

5.4.1.3. FrNmChannel

Containers included		
Container name	**Multiplicity**	**Description**
FrNmChannelIdentifiers	1..1	This container contains module instance specific identifiers related to the respective FlexRay Channel.
FrNmChannelTiming	1..1	This container contains module instance specific timing related to the respective FlexRay Channel.

Parameters included	
Parameter name	**Multiplicity**

Parameters included	
FrNmAllNmMessagesKeepAwake	0..1

Parameter Name	FrNmAllNmMessagesKeepAwake	
Description	Specifies if FrNm drops irrelevant NM messages.	
Multiplicity	0..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.4.1.4. FrNmChannelIdentifiers

Containers included		
Container name	Multiplicity	Description
FrNmRxPdu	1..n	This container describes the FlexRay NM RX PDUs. Dependency on parameter(s): ► The PDU length configured in ECU configuration must be equal when seperate PDUs are used for NM-Vote and NM-Data.
FrNmTxPdu	0..4	This container describes the FlexRay NM TX PDUs. Dependency on parameter(s): ► The PDU length configured in ECU configuration must be less than or equal to the PDU length configured for `FrNmRxPdu`.
FrNmUserDataTxPdu	0..1	This optional container is used to configure the UserNm PDU. This container is only available if FrNmComUserDataSupport is enabled.
FrNmUserDataRxPdu	0..1	This optional container is used to configure the UserNm PDU. This container is only available if FrNmComUserDataSupport is enabled.

Parameters included	
Parameter name	Multiplicity

Parameters included	
FrNmActiveWakeupBitEnabled	1..1
FrNmCarWakeUpBitPosition	1..1
FrNmCarWakeUpBytePosition	1..1
FrNmCarWakeUpFilterEnabled	1..1
FrNmCarWakeUpFilterNodeId	1..1
FrNmCarWakeUpRxEnabled	1..1
FrNmControlBitVectorActive	1..1
FrNmNodeDetectionEnabled	1..1
FrNmNodeId	1..1
FrNmPduScheduleVariant	1..1
FrNmPnEnabled	1..1
FrNmPnEraCalcEnabled	1..1
FrNmRepeatMessageBitActive	1..1
FrNmSourceNodeIdentifierEnabled	1..1
FrNmSynchronizationPointEnabled	1..1
FrNmPnEraRxNSduRef	0..1
FrNmChannelHandle	1..1
FrNmComMNetworkHandleRef	1..1

Parameter Name	FrNmActiveWakeupBitEnabled	
Label	Active Wakeup Bit Enable	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCarWakeUpBitPosition
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=7

	>=0

Configuration class	**PreCompile:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCarWakeUpBytePosition	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCarWakeUpFilterEnabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCarWakeUpFilterNodeId	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCarWakeUpRxEnabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmControlBitVectorActive	
Description	This parameter is used to activate or deactivate the control bit vector support for a Fr Nm Channel.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmNodeDetectionEnabled	
Description	Pre-processor switch for enabling node detection support. Dependency on parameter(s): ▶ This parameters has effect only if the parameter `FrNmPassiveModeEnabled` is set as `false`.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmNodeId	
Description	NM node identifier configured for the respective FlexRay Channel. It is used for identifying the respective NM node in the NM-cluster. It must be unique for each NM node within one NM cluster.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPduScheduleVariant	
Description	Selects the Schedule Variant.	

	▶ FRNM_PDU_SCHEDULE_VARIANT_1: NM-Vote and NM-Data in static segment (one PDU)	
	▶ FRNM_PDU_SCHEDULE_VARIANT_2: NM-Vote and NM-Data in dynamic segment (one PDU)	
	▶ FRNM_PDU_SCHEDULE_VARIANT_3: NM-Vote and NM-Data in static segment (separate PDU)	
	▶ FRNM_PDU_SCHEDULE_VARIANT_4: NM-Vote in static segment and NM-Data in dynamic segment	
	▶ FRNM_PDU_SCHEDULE_VARIANT_5: NM-Vote in dynamic segment and NM-Data in static segment	
	▶ FRNM_PDU_SCHEDULE_VARIANT_6: NM-Vote and NM-Data in dynamic segment (separate PDU)	
	▶ FRNM_PDU_SCHEDULE_VARIANT_7: Combined NM-Vote and CBV in static segment and NM-Data in dynamic segment	
	Note: Schedule Variants FRNM_PDU_SCHEDULE_VARIANT_3 and FRNM_-PDU_SCHEDULE_VARIANT_5 are not supported.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	FRNM_PDU_SCHEDULE_VARIANT_1	
	FRNM_PDU_SCHEDULE_VARIANT_2	
	FRNM_PDU_SCHEDULE_VARIANT_3	
	FRNM_PDU_SCHEDULE_VARIANT_4	
	FRNM_PDU_SCHEDULE_VARIANT_5	
	FRNM_PDU_SCHEDULE_VARIANT_6	
	FRNM_PDU_SCHEDULE_VARIANT_7	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	ECUC_AUTOSAR	

Parameter Name	**FrNmPnEnabled**	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPnEraCalcEnabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmRepeatMessageBitActive	
Description	This parameter is used to activate or deactivate the repeat message bit support for a FrNm Channel.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmSourceNodeIdentifierEnabled	
Description	Pre-processor switch for enabling SourceNodeIdentifier support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmSynchronizationPointEnabled	
Description	This parameter defines if this channel shall provide the synchronization point indication to the NM Interface.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPnEraRxNSduRef

Multiplicity	0..1	
Type	REFERENCE	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmChannelHandle**	
Description	Channel identifier configured for the respective instance of the NM. The FrNmChannelHandle shall be encoded in the FrNmRxPduId parameter which is passed to FrNm_RxIndication() function called by the FrIf.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmComMNetworkHandleRef**	
Description	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.5. FrNmRxPdu

Parameters included	
Parameter name	**Multiplicity**
FrNmRxPduContainsData	1..1
FrNmRxPduContainsVote	1..1
FrNmRxPduId	1..1
FrNmRxPduRef	1..1

Parameter Name	**FrNmRxPduContainsData**
Description	This parameted defines if the PDU contains NM Data.
Multiplicity	1..1

Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmRxPduContainsVote**	
Description	This parameted defines if the PDU contains NM Vote information.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmRxPduId**	
Description	PDU identifier configured for the respective FlexRay Channel. It is used for referring to the FlexRay Interface receive function. It must be consistent with the value configured in the FlexRay Interface. This ID is used for the combined reception of NM Vote and NM Data or for the reception of the NM Vote if NM Data is received in a separate PDU.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmRxPduRef**	
Description	The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the FrIf module to derive the PDU Id.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

5.4.1.6. FrNmTxPdu

Parameters included	
Parameter name	**Multiplicity**
FrNmTxConfirmationPduId	1..1
FrNmTxPduContainsData	1..1
FrNmTxPduContainsVote	1..1
FrNmTxPduRef	1..1

Parameter Name	FrNmTxConfirmationPduId	
Description	Handle Id to be used by the Lower Layer to confirm the transmission of the FrNmTxPdu to the LowerLayer.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmTxPduContainsData	
Description	This parameter defines if the PDU contains NM Data.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmTxPduContainsVote
Description	This parameted defines if the PDU contains NM Vote information.
Multiplicity	1..1
Type	BOOLEAN

Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmTxPduRef**	
Description	The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference is used to derive the PDU Id that is defined by the FrIf module.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.7. FrNmUserDataTxPdu

Parameters included	
Parameter name	**Multiplicity**
FrNmTxUserDataPduId	1..1
FrNmTxUserDataPduRef	1..1

Parameter Name	**FrNmTxUserDataPduId**	
Description	This parameter defines the Handle ID of the NM User Data I-PDU.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmTxUserDataPduRef**	
Description	Reference to the NM User Data I-PDU in the global PDU collection.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.8. FrNmUserDataRxPdu

Parameters included	
Parameter name	**Multiplicity**
FrNmRxUserDataPduId	1..1
FrNmRxUserDataPduRef	1..1

Parameter Name	FrNmRxUserDataPduId	
Description	This parameter defines the Handle ID of the NM User Data I-PDU.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrNmRxUserDataPduRef	
Description	Reference to the NM User Data I-PDU in the global PDU collection.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.4.1.9. FrNmChannelTiming

Parameters included	
Parameter name	**Multiplicity**
FrNmDataCycle	1..1
FrNmMainFunctionPeriod	1..1
FrNmMsgTimeoutTime	1..1
FrNmReadySleepCnt	1..1
FrNmRemoteSleepIndTime	1..1
FrNmRepeatMessageTime	1..1
FrNmRepetitionCycle	1..1
FrNmSyncLossTimer	1..1

Parameters included	
FrNmVoteInhibitionEnabled	1..1
FrNmVotingCycle	1..1

Parameter Name	FrNmDataCycle	
Description	Number of FlexRay Schedule Cycles needed to transmit the NM Data of all ECUs on the FlexRay bus.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	FRNM_CYCLE_VALUE_1	
	FRNM_CYCLE_VALUE_16	
	FRNM_CYCLE_VALUE_2	
	FRNM_CYCLE_VALUE_32	
	FRNM_CYCLE_VALUE_4	
	FRNM_CYCLE_VALUE_64	
	FRNM_CYCLE_VALUE_8	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmMainFunctionPeriod	
Description	This parameter defines the processing cycle of the main function of FrNm module.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.005	
Range	<=0.020	
	>=0.001	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmMsgTimeoutTime
Description	Timeout of a NM-message. It determines in seconds how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.

Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmReadySleepCnt**	
Description	Numbers of repetitions in the ready sleep state before NM switches to bus sleep mode. On a value of "1", the NM-State Machine will leave the Ready Sleep State after one NM Repetition Cycle with no "keep awake" votes. Value of parameter according to AUTOSAR:	

Value of parameter according to AUTOSAR:

► According to AUTOSAR 4.2.2, if bus communication is released the FlexRay NM module shall perform the transition into the Bus-Sleep Mode at the end of the FrNmReadySleepCnt + 1 repetition cycle without any positive NM vote.

► According to AUTOSAR 4.0.3, if bus communication is released the FlexRay NM module shall perform the transition into the Bus-Sleep Mode at the end of the FrNmReadySleepCnt repetition cycle without any positive NM vote.

Default implementation is according to AUTOSAR 4.2.2

To met AUTOSAR 4.0.3 behaviour, set required FrNmReadySleepCnt -1

Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmRemoteSleepIndTime**	
Description	Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep. The value "0" denotes that no Remote Sleep Indication functionality is configured.	

Dependency on parameter(s):

▶ If parameter `FrNmRemoteSleepIndicationEnabled` is set to false, then the Remote Sleep Indication Time must be set to zero.

▶ The Remote Sleep Indication Time must be a multiple of the Main Function Period.

▶ The Remote Sleep Indication Time must not be less than one Repetition Cycle time.

▶ The Remote Sleep Indication Time should be greater than one Repetition Cycle time to avoid the possibility of `Nm_RemoteSleepIndication()` and `Nm_RemoteSleepCancellation()` being called within the same Repetition Cycle.

Multiplicity	1..1
Type	FLOAT
Default value	0.0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrNmRepeatMessageTime**
Description	Timeout for Repeat Message State. Defines the time in seconds how long the NM shall stay in the Repeat Message State. The value "0" denotes that no Repeat Message State is configured, which means that Repeat Message State is transient and implies that it is left immediately after entry and consequently no startup stability is guaranteed and no node detection procedure is possible.
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrNmRepetitionCycle**
Description	Number of Flexray Schedule Cycles used to repeat the transmission of the Nm vote of all ECUs on the Flexray Bus.
Multiplicity	1..1
Type	ENUMERATION
Range	FRNM_CYCLE_VALUE_1
	FRNM_CYCLE_VALUE_2
	FRNM_CYCLE_VALUE_4

	FRNM_CYCLE_VALUE_8
	FRNM_CYCLE_VALUE_16
	FRNM_CYCLE_VALUE_32
	FRNM_CYCLE_VALUE_64

Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmSyncLossTimer**	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Range	<=65535.0	
	>=0.0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmVoteInhibitionEnabled**	
Description	Pre-processor switch for enabling the inhibition of vote changes from the next-to-last repetition cycle to the last repetition cycle before the Ready Sleep Counter expires.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmVotingCycle**	
Description	Number of FlexRay Schedule Cycles needed to transmit the Nm vote of all ECUs on the FlexRay Bus.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	FRNM_CYCLE_VALUE_1	
	FRNM_CYCLE_VALUE_2	
	FRNM_CYCLE_VALUE_4	

	FRNM_CYCLE_VALUE_8	
	FRNM_CYCLE_VALUE_16	
	FRNM_CYCLE_VALUE_32	
	FRNM_CYCLE_VALUE_64	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.10. FrNmGeneral

Parameters included	
Parameter name	**Multiplicity**
FrNmMultiCoreSupport	1..1
FrNmPnSupported	1..1
FrNmRelocatablePbcfgEnable	1..1
FrNmMaxPn	0..1

Parameter Name	**FrNmMultiCoreSupport**	
Label	FrNm multicore support	
Description	Enables MultiCoreSupport.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmPnSupported**
Label	Support for Partial Network Cluster (PNC)
Description	Enables or disables support of partial networking. ▶ `False`: Partial Networking is disabled ▶ `True`: Partial Networking is enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrNmRelocatablePbcfgEnable	
Label	FrNmRelocatablePbcfgEnable	
Description	Enables/disables support for relocatable postbuild configuration. ► True: Postbuild configuration relocatable in memory. ► False: Postbuild configuration not relocatable in memory.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrNmMaxPn	
Label	FrNmMaxPn	
Description	The maximum number of Partial Networking Clusters that can be configured.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.4.1.11. FrNmGlobalConfig

Containers included		
Container name	**Multiplicity**	**Description**
FrNmGlobalConstants	1..1	This container contains module constants related to the FlexRay NM functionality.
FrNmGlobalFeatures	1..1	This container contains module features related to the FlexRay NM functionality.
FrNmGlobalProperties	1..1	This container contains module properties related to the FlexRay NM functionality.

5.4.1.12. FrNmGlobalConstants

Parameters included	
Parameter name	**Multiplicity**
FrNmNumberOfClusters	1..1

Parameter Name	FrNmNumberOfClusters	
Description	This is the maximum number of supported FrNm clusters the Node may be attached to. The actual number of connected clusters (i.e. number of containers in FrNmChannelIdentifiers or FrNmChannelTimings) must be lower or equal to this constant.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.13. FrNmGlobalFeatures

Containers included		
Container name	**Multiplicity**	**Description**
FrNmPnInfo	0..1	

Parameters included	
Parameter name	**Multiplicity**
FrNmBusSynchronizationEnabled	1..1
FrNmComUserDataSupport	1..1
FrNmControlBitVectorEnabled	1..1
FrNmCoordinatorSyncSupport	1..1
FrNmCycleCounterEmulation	1..1
FrNmDualChannelPduEnable	1..1
FrNmHwVoteEnable	1..1
FrNmPostBuildRamSize	1..1
FrNmPassiveModeEnabled	1..1

Parameters included	
FrNmPduRxIndicationEnabled	1..1
FrNmPnEiraCalcEnabled	1..1
FrNmPnResetTime	1..1
FrNmRemoteSleepIndicationEnabled	1..1
FrNmRepeatMessageBitEnabled	1..1
FrNmVoteBitValue	0..1
FrNmStateChangeIndicationEnabled	1..1
FrNmSynchErrExtended	0..1
FrNmUserDataEnabled	1..1
FrNmVotingNextToLastRepetitionCycleDisable	1..1
FrNmPnEiraRxNSduRef	1..1

Parameter Name	FrNmBusSynchronizationEnabled	
Description	Pre-processor switch for enabling the bus synchronisation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmComUserDataSupport	
Description	Enable/disable the user data support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmControlBitVectorEnabled
Description	Pre-processor switch for enabling control bit vector support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCoordinatorSyncSupport	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmCycleCounterEmulation	
Description	Pre-processor switch for enabling the cycle counter emulation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmDualChannelPduEnable	
Description	Pre-processor switch for enabling the support of dual channel transmission and reception of NM messages.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmHwVoteEnable	
Description	Pre-processor switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	FrNmPostBuildRamSize	
Label	FrNmPostBuildRamSize	
Description	Number of bytes for TX and RX buffers Value should be set as: the sum of the largest RxPdu length on each channel multiplied with 2 (in case passive mode is disabled). Size should be big enough to hold eventual changes of PDU lenghts at postbuild time	
Multiplicity	1..1	
Type	INTEGER	
Default value	96	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrNmPassiveModeEnabled	
Description	Pre-processor switch for enabling Passive Mode Configuration support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPduRxIndicationEnabled	
Description	Pre-processor switch for enabling PDU reception indication.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPnEiraCalcEnabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPnResetTime	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.001	
Range	<=65.535	
	>=0.0010	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmRemoteSleepIndicationEnabled	
Description	Pre-processor switch for enabling remote sleep indication. Dependency on parameter(s): ▶ This parameters has effect only if the parameter `FrNmPassiveModeEnabled` is set as `false`.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmRepeatMessageBitEnabled	
Description	Pre-processor switch for enabling the repeat message bit support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmVoteBitValue	
Description	Specifies the value of the vote bit in case FrNmPduScheduleVariant is configured with FRNM_PDU_SCHEDULE_VARIANT_2 or FRNM_PDU_SCHED-	

	ULE_VARIANT_6. In case any other schedule variant is selected, this value is ignored.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=1	
	>=0	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmStateChangeIndicationEnabled**	
Description	Pre-processor switch for enabling state change indication.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmSynchErrExtended**	
Description	Enables or disables AUTOSAR 4.2.1 error handling ► false : BusOff handling is done as described in AUTOSAR 4.0.2 ► true : BusOff handling is done as described in AUTOSAR 4.2.1	
Multiplicity	0..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmUserDataEnabled**	
Description	Pre-processor switch for enabling user data support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmVotingNextToLastRepetitionCycleDisable	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmPnEiraRxNSduRef	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.14. FrNmPnInfo

Containers included		
Container name	Multiplicity	Description
FrNmPnFilterMaskByte	0..7	

Parameters included	
Parameter name	Multiplicity
FrNmPnInfoLength	1..1
FrNmPnInfoOffset	1..1

Parameter Name	FrNmPnInfoLength	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=7	
	>=1	

Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmPnInfoOffset**	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=31	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.15. FrNmPnFilterMaskByte

Parameters included	
Parameter name	**Multiplicity**
[FrNmPnFilterMaskByteIndex](#)	1..1
[FrNmPnFilterMaskByteValue](#)	1..1

Parameter Name	**FrNmPnFilterMaskByteIndex**	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrNmPnFilterMaskByteValue**	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.16. FrNmGlobalProperties

Parameters included	
Parameter name	Multiplicity
FrNmDevErrorDetect	1..1
FrNmMainAcrossFrCycle	1..1
FrNmVersionInfoApi	1..1

Parameter Name	FrNmDevErrorDetect	
Description	Pre-processor switch for enabling development error detection.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmMainAcrossFrCycle	
Description	Parameter describing if the execution of FrNm_Main function crosses the FlexRay cycle boundary or not.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrNmVersionInfoApi
Description	Pre-processor switch for enabling version info API support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false

Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.4.1.17. FrNmDefensiveProgramming

Parameters included	
Parameter name	**Multiplicity**
FrNmDefProgEnabled	1..1
FrNmPrecondAssertEnabled	1..1
FrNmPostcondAssertEnabled	1..1
FrNmStaticAssertEnabled	1..1
FrNmUnreachAssertEnabled	1..1
FrNmInvariantAssertEnabled	1..1

Parameter Name	**FrNmDefProgEnabled**	
Label	Enable Defensive Programming	
Description	Enables or disables the defensive programming feature for the module FrNm. Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows: 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmPrecondAssertEnabled**
Label	Enable Precondition Assertions
Description	Enables handling of precondition assertion checks reported from the module FrNm. Dependency on parameter(s):

▶ Enable Development Error Detection (`FrNmDevErrorDetect`): must be enabled ▶ Enable Defensive Programming (`FrNmDefProgEnabled`): must be enabled		
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmPostcondAssertEnabled**	
Label	Enable Postcondition Assertions	
Description	Enables handling of postcondition assertion checks reported from the module FrNm. Dependency on parameter(s): ▶ Enable Development Error Detection (`FrNmDevErrorDetect`): must be enabled ▶ Enable Defensive Programming (`FrNmDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmStaticAssertEnabled**
Label	Enable Static Assertions
Description	Enables handling of static assertion checks reported from the module FrNm. Dependency on parameter(s): ▶ Enable Development Error Detection (`FrNmDevErrorDetect`): must be enabled ▶ Enable Defensive Programming (`FrNmDefProgEnabled`): must be enabled
Multiplicity	1..1

Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmUnreachAssertEnabled**	
Label	Enable Unreachable Code Assertions	
Description	Enables handling of unreachable code assertion checks reported from the module FrNm. Dependency on parameter(s): ► Enable Development Error Detection (`FrNmDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrNmDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrNmInvariantAssertEnabled**	
Label	Enable Invariant Assertions	
Description	Enables handling of invariant assertion checks reported from functions of the module FrNm. Dependency on parameter(s): ► Enable Development Error Detection (`FrNmDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrNmDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	Elektrobit Automotive GmbH

5.4.1.18. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport	
Label	PbcfgM support	
Description	Specifies whether or not the FrNm can use the PbcfgM module for post-build support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

5.4.2. Application programming interface (API)

5.4.2.1. Macro constants

5.4.2.1.1. FRNM_AR_RELEASE_MAJOR_VERSION

Purpose	AUTOSAR release major version.
Value	4U

5.4.2.1.2. FRNM_AR_RELEASE_MINOR_VERSION

Purpose	AUTOSAR release minor version.

Value	0U

5.4.2.1.3. FRNM_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.
Value	3U

5.4.2.1.4. FRNM_E_BUSSLEEPMODE

Purpose	Error code for SchM Nm_BusSleepMode.
Value	248U

5.4.2.1.5. FRNM_E_CARWAKEUPINDICATION

Purpose	Error code for SchM Nm_CarWakeUpIndication.
Value	240U

5.4.2.1.6. FRNM_E_COORDREADYTOSLEEPINDICATION

Purpose	Error code for SchM Nm_CoordReadyToSLeepIndication.
Value	245U

5.4.2.1.7. FRNM_E_INIT_FAILED

Purpose	Error code reported to DET in case FrNm_Init fails.
Value	0x06U

5.4.2.1.8. FRNM_E_INVALID_CHANNEL

Purpose	Error Code for Invalid channel.

Value	0x02U

5.4.2.1.9. FRNM_E_INVALID_FUNCTION_ARG

Purpose	
Value	0x05U

5.4.2.1.10. FRNM_E_INVALID_POINTER

Purpose	Error code for Invalid pointers.
Value	0x03U

5.4.2.1.11. FRNM_E_NETWORKMODE

Purpose	Error code for SchM Nm_NetworkMode.
Value	249U

5.4.2.1.12. FRNM_E_NETWORKSTARTINDICATION

Purpose	Error code for SchM Nm_NetworkStartIndication.
Value	250U

5.4.2.1.13. FRNM_E_PDURXINDICATION

Purpose	Error code for SchM Nm_PduRxIndication.
Value	243U

5.4.2.1.14. FRNM_E_PDU_ID_INVALID

Purpose	Error code for PDU ID as input parameter.

Value	0x04U

5.4.2.1.15. FRNM_E_REMOTESLEEPCANCELLATION

Purpose	Error code for SchM Nm_RemoteSleepCancellation.
Value	246U

5.4.2.1.16. FRNM_E_REMOTESLEEPINDICATION

Purpose	Error code for SchM Nm_RemoteSleepIndication.
Value	247U

5.4.2.1.17. FRNM_E_STATECHANGENOTIFICATION

Purpose	Error code for SchM Nm_StateChangeNotification.
Value	242U

5.4.2.1.18. FRNM_E_SYNCHRONIZATIONPOINT

Purpose	Error code for SchM Nm_SynchronizationPoint.
Value	244U

5.4.2.1.19. FRNM_E_TXTIMEOUTEXCEPTION

Purpose	Error code for SchM Nm_TxTimeoutException.
Value	241U

5.4.2.1.20. FRNM_E_UNINIT

Purpose	Initialization status before module initilaization.
Value	0x01U

5.4.2.1.21. FRNM_INSTANCE_ID

Purpose	Instance Id of FrNm.
Value	0U

5.4.2.1.22. FRNM_MODULE_ID

Purpose	AUTOSAR module identification.
Value	32U

5.4.2.1.23. FRNM_PDU_BYTE_0

Purpose	AUTOSAR API service ID.
Value	0U
Description	Definition of FRNM_PDU_BYTE_0.

5.4.2.1.24. FRNM_PDU_BYTE_1

Purpose	AUTOSAR API service ID.
Value	1U
Description	Definition of FRNM_PDU_BYTE_1.

5.4.2.1.25. FRNM_PDU_OFF

Purpose	AUTOSAR API service ID.
Value	3U
Description	Definition of FRNM_PDU_OFF.

5.4.2.1.26. FRNM_SERVID_CHECKREMOTESLEEPINDICATION

Purpose	AUTOSAR API service ID.

Value	0x0DU
Description	Definition of FRNM_SERVID_CHECKREMOTESLEEPINDICATION.

5.4.2.1.27. FRNM_SERVID_DISABLECOMMUNICATION

Purpose	AUTOSAR API service ID.
Value	0x0CU
Description	Definition of FRNM_SERVID_DISABLECOMMUNICATION.

5.4.2.1.28. FRNM_SERVID_ENABLECOMMUNICATION

Purpose	AUTOSAR API service ID.
Value	0x0DU
Description	Definition of FRNM_SERVID_ENABLECOMMUNICATION.

5.4.2.1.29. FRNM_SERVID_GETLOCALNODEIDENTIFIER

Purpose	AUTOSAR API service ID.
Value	0x0bU
Description	Definition of FRNM_SERVID_GETLOCALNODEIDENTIFIER.

5.4.2.1.30. FRNM_SERVID_GETNODEIDENTIFIER

Purpose	AUTOSAR API service ID.
Value	0x0aU
Description	Definition of FRNM_SERVID_GETNODEIDENTIFIER.

5.4.2.1.31. FRNM_SERVID_GETPDUDATA

Purpose	AUTOSAR API service ID.
Value	0x08U

Description	Definition of FRNM_SERVID_GETPDUDATA.

5.4.2.1.32. FRNM_SERVID_GETSTATE

Purpose	AUTOSAR API service ID.
Value	0x0EU
Description	Definition of FRNM_SERVID_GETSTATE.

5.4.2.1.33. FRNM_SERVID_GETUSERDATA

Purpose	AUTOSAR API service ID.
Value	0x07U
Description	Definition of FRNM_SERVID_GETUSERDATA.

5.4.2.1.34. FRNM_SERVID_GETVERSIONINFO

Purpose	AUTOSAR API service ID.
Value	0x0FU
Description	Definition of FRNM_SERVID_GETVERSIONINFO.

5.4.2.1.35. FRNM_SERVID_INIT

Purpose	AUTOSAR API service ID.
Value	0x00U
Description	Definition of FRNM_SERVID_INIT.

5.4.2.1.36. FRNM_SERVID_MAINFUNCTION_X

Purpose	AUTOSAR API service ID.
Value	0xF0U
Description	Definition of FRNM_SERVID_MAINFUNCTION_X.

5.4.2.1.37. FRNM_SERVID_NETWORKGWERAREQUEST

Purpose	AUTOSAR API service ID.
Value	0xFEU
Description	Definition of FRNM_SERVID_NETWORKGWERAREQUEST

5.4.2.1.38. FRNM_SERVID_NETWORKRELEASE

Purpose	AUTOSAR API service ID.
Value	0x03U
Description	Definition of FRNM_SERVID_NETWORKRELEASE.

5.4.2.1.39. FRNM_SERVID_NETWORKREQUEST

Purpose	AUTOSAR API service ID.
Value	0x02U
Description	Definition of FRNM_SERVID_NETWORKREQUEST.

5.4.2.1.40. FRNM_SERVID_PASSIVESTARTUP

Purpose	AUTOSAR API service ID.
Value	0x01U
Description	Definition of FRNM_SERVID_PASSIVESTARTUP

5.4.2.1.41. FRNM_SERVID_REPEATMESSAGEREQUEST

Purpose	AUTOSAR API service ID.
Value	0x09U
Description	Definition of FRNM_SERVID_REPEATMESSAGEREQUEST.

5.4.2.1.42. FRNM_SERVID_REQUESTBUSSYNCHRONIZATION

Purpose	AUTOSAR API service ID.

Value	0xC0U
Description	Definition of FRNM_SERVID_REQUESTBUSSYNCHRONIZATION.

5.4.2.1.43. FRNM_SERVID_RXINDICATION

Purpose	AUTOSAR API service ID.
Value	0xE1U
Description	Definition of FRNM_SERVID_RXINDICATION.

5.4.2.1.44. FRNM_SERVID_SETSLEEPREADYBIT

Purpose	AUTOSAR API service ID.
Value	0x12U
Description	Definition of FRNM_SERVID_SETSLEEPREADYBIT.

5.4.2.1.45. FRNM_SERVID_SETUSERDATA

Purpose	AUTOSAR API service ID.
Value	0x06U
Description	Definition of FRNM_SERVID_SETUSERDATA.

5.4.2.1.46. FRNM_SERVID_STARTUPERROR

Purpose	AUTOSAR API service ID.
Value	0x10U
Description	Definition of FRNM_SERVID_STARTUPERROR.

5.4.2.1.47. FRNM_SERVID_TRANSMIT

Purpose	AUTOSAR API service ID.
Value	0x00U

Description	Definition of FRNM_SERVID_TRANSMIT.

5.4.2.1.48. FRNM_SERVID_TRIGGERTRANSMIT

Purpose	AUTOSAR API service ID.
Value	0xE4U
Description	Definition of FRNM_SERVID_TRIGGERTRANSMIT.

5.4.2.1.49. FRNM_SERVID_TXCONFIRMATION

Purpose	AUTOSAR API service ID.
Value	0xE0U
Description	Definition of FRNM_SERVID_TXCONFIRMATION.

5.4.2.1.50. FRNM_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	5U

5.4.2.1.51. FRNM_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	16U

5.4.2.1.52. FRNM_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	8U

5.4.2.1.53. FRNM_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.

Value	1U

5.4.2.2. Functions

5.4.2.2.1. FrNm_CheckRemoteSleepIndication

Purpose	This function checks if remote sleep indication has taken place or not.	
Synopsis	`Std_ReturnType` **`FrNm_CheckRemoteSleepIndication`** `( NetworkHandle-Type NetworkHandle , boolean * nmRemoteSleepIndPtr );`	
Service ID	0x0d	
Sync/Async	Synchronous	
Reentrancy	Reentrant(but not for the same Nm-channel)	
Parameters (in)	`NetworkHandle`	Identification of the NM-Cluster.
Parameters (out)	`nmRemoteSleepIndPtr`	Pointer to the location where the check result of remote sleep indication shall be copied.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Checking of remote sleep indication bits has failed
Description	This FlexRay NM function shall provide the information about current status of Remote Sleep Indication.	

5.4.2.2.2. FrNm_DisableCommunication

Purpose	Disable the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.
Synopsis	`Std_ReturnType` **`FrNm_DisableCommunication`** `( NetworkHandleType nmChannelHandle );`
Service ID	0x0c
Sync/Async	Asynchronous
Reentrancy	Reentrant (but not for the same NM-channel)

Parameters (in)	nmChannelHandle	Identification of the NM-channel.
Return Value	Std_ReturnType	
	E_OK	No error
	E_NOT_OK	Disabling of NM PDU transmission ability has failed

5.4.2.2.3. FrNm_EnableCommunication

Purpose	Enable the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.	
Synopsis	Std_ReturnType **FrNm_EnableCommunication** (NetworkHandleType nm-ChannelHandle);	
Service ID	0x05	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel.
Return Value	Std_ReturnType	
	E_OK	No error
	E_NOT_OK	Enabling of NM PDU transmission ability has failed

5.4.2.2.4. FrNm_GetLocalNodeIdentifier

Purpose	This function gets the node identifier configured for the local node.	
Synopsis	Std_ReturnType **FrNm_GetLocalNodeIdentifier** (NetworkHandleType NetworkHandle , uint8 * nmNodeIdPtr);	
Service ID	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkHandle	Identification of the NM-channel.
Parameters (out)	nmNodeIdPtr	Pointer the location where the node identifier of the local node shall be copied.
Return Value	Std_ReturnType	
	E_OK	No error

	E_NOT_OK	Getting of the node identifier of the local node has failed
Description	If node detection is enabled, then this function shall provide the node identifier configured for the local host node.	

5.4.2.2.5. FrNm_GetNodeIdentifier

Purpose	This function gets the node identifier from the last successfully received NM-message.	
Synopsis	Std_ReturnType **FrNm_GetNodeIdentifier** (NetworkHandleType NetworkHandle , uint8 * nmNodeIdPtr);	
Service ID	0x0a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkHandle	Identification of the NM-channel.
Parameters (out)	nmNodeIdPtr	Pointer to the location where the node identifier from the last successfully received NM-message shall be copied.
Return Value	Std_ReturnType	
	E_OK	No error
	E_NOT_OK	Getting of the node identifier out of the last received NM-message has failed
Description	If the node detection feature is enabled, then this function shall provide the node identifier from the most recently received NM-message.	

5.4.2.2.6. FrNm_GetPduData

Purpose	This function Gets PDU data.	
Synopsis	Std_ReturnType **FrNm_GetPduData** (NetworkHandleType NetworkHandle , uint8 * nmPduData);	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkHandle	Identification of the NM-channel.

Parameters (out)	`nmPduData`	Pointer where NM PDU shall be copied to.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Getting of NM PDU data has failed
Description	This function shall get the whole NM PDU data out of the most recently received NM message.	

5.4.2.2.7. FrNm_GetState

Purpose	This function returns the state and the mode of the network management.	
Synopsis	`Std_ReturnType` **`FrNm_GetState`** `( NetworkHandleType NetworkHandle , Nm_StateType * nmStatePtr , Nm_ModeType * nmModePtr );`	
Service ID	0x0e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-Cluster.
Parameters (out)	`nmStatePtr`	Pointer to the location where the state of the network management shall be copied.
	`nmModePtr`	Pointer to the location where the mode of the network management shall be copied.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Getting of NM state has failed
Description	This function shall provide consistent information about the current state and the current mode of the NM state machine. Note: Consistency between the provided values and the current values of the state and mode should be ensured.	

5.4.2.2.8. FrNm_GetUserData

Purpose	This function gets user data from the last successfully received NM message.
Synopsis	`Std_ReturnType` **`FrNm_GetUserData`** `( NetworkHandleType NetworkHandle , uint8 * nmUserDataPtr );`

Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.
Parameters (out)	`nmUserDataPtr`	Pointer to the location where the user data from the last successfully received NM message shall be copied.
Return Value	Std_ReturnType	
	`E_OK`	No error.
	`E_NOT_OK`	Getting of user data has failed
Description	If user data handling is enabled for the FrNm module, then this function shall provide the user data from the last received NM-Data PDU.	

5.4.2.2.9. FrNm_GetVersionInfo

Purpose	Returns the version information.	
Synopsis	void **FrNm_GetVersionInfo** (Std_VersionInfoType * NmVerInfoPtr);	
Service ID	0x0f	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	`NmVerInfoPtr`	Pointer to the location where the version information of this module shall be copied.
Description	This function shall return the version information of this module. The version information includes: ► Module Id ► Vendor Id ► Vendor specific version numbers(BSW00407).	

5.4.2.2.10. FrNm_Init

Purpose	Initialization of FrNm module.

Synopsis	void **FrNm_Init** (const FrNm_ConfigType *const nmConfigPtr);	
Service ID	0x00	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	nmConfigPtr	Pointer to selected configuration set.
Description	This function shall initialize the FrNm module.	

5.4.2.2.11. FrNm_IsValidConfig

Purpose	Validate configuration.	
Synopsis	Std_ReturnType **FrNm_IsValidConfig** (const void * voidConfigPtr);	
Service ID	0x60	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Return Value	E_OK if the given module configurations is valid otherwise E_NOT_OK.	
Description	Checks if the post build configuration fits to the link time configuration part.	

5.4.2.2.12. FrNm_NetworkGwEraRequest

Purpose	This function requests the network because the ECU needs to communicate on the bus. Network state shall be changed to "requested".	
Synopsis	Std_ReturnType **FrNm_NetworkGwEraRequest** (NetworkHandleType NetworkHandle);	
Service ID	0xFE	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	NetworkHandle	Identification of the NM-channel.
Return Value	Std_ReturnType	
	E_OK	No error.
	E_NOT_OK	Requesting of bus communication has failed.

5.4.2.2.13. FrNm_NetworkRelease

Purpose	This function releases the network because the ECU doesn't have to communicate on the bus. Network state shall be changed to "released".	
Synopsis	`Std_ReturnType` **`FrNm_NetworkRelease`** `( NetworkHandleType Net-workHandle );`	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.
Return Value	Std_ReturnType	
	`E_OK`	No error.
	`E_NOT_OK`	Releasing of bus communication has failed

5.4.2.2.14. FrNm_NetworkRequest

Purpose	This function requests the network because the ECU needs to communicate on the bus. Network state shall be changed to "requested".	
Synopsis	`Std_ReturnType` **`FrNm_NetworkRequest`** `( NetworkHandleType Net-workHandle );`	
Service ID	0x02	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.
Return Value	Std_ReturnType	
	`E_OK`	No error.
	`E_NOT_OK`	Requesting of bus communication has failed.

5.4.2.2.15. FrNm_PassiveStartUp

Purpose	Initiates the Passive Startup of the FlexRay NM.

Synopsis	`Std_ReturnType` **`FrNm_PassiveStartUp`** `( NetworkHandleType Net-workHandle );`	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Start of network management has failed.
Description	This function shall initiate the Passive Startup of the FlexRay NM.	

5.4.2.2.16. FrNm_RepeatMessageRequest

Purpose	This function causes a Repeat Message Request to be transmitted next on the bus.	
Synopsis	`Std_ReturnType` **`FrNm_RepeatMessageRequest`** `( NetworkHandleType NetworkHandle );`	
Service ID	0x09	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Repeat Message Request has failed
Description	If the node detection feature is enabled, then this function shall request node detection on the FlexRay Bus NM nodes.	

5.4.2.2.17. FrNm_RequestBusSynchronization

Purpose	This function has no functionality - the service is provided only to be compatible to future extensions and to be compatible to the FR-NM interface.	
Synopsis	`Std_ReturnType` **`FrNm_RequestBusSynchronization`** `( NetworkHandle-Type NetworkHandle );`	

Service ID	0xc0	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`NetworkHandle`	Identification of the NM-Cluster.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Function failed

5.4.2.2.18. FrNm_RxIndication

Purpose	Indication of a received I-PDU from a lower layer communication module.	
Synopsis	void **FrNm_RxIndication** (PduIdType RxPduId , PduInfoType * PduInfoPtr);	
Service ID	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds but not for same	
Parameters (in)	`RxPduId`	ID of the received I-PDU.
	`PduInfoPtr`	Contains the length of the received I-PDU and a pointer to a buffer containing the I-PDU.
Description	This function shall copy the received FlexRay NM PDU and store it locally associated with the received FlexRay NM PDU ID. This function might be called by the FrNm module's environment in an interrupt context.	

5.4.2.2.19. FrNm_SetSleepReadyBit

Purpose	Set the NM Coordinator Sleep Ready bit in the Control Bit Vector.
Synopsis	Std_ReturnType **FrNm_SetSleepReadyBit** (NetworkHandleType nmChannelHandle , boolean nmSleepReadyBit);
Service ID	0x12
Sync/Async	Synchronous
Reentrancy	Non-Reentrant

Parameters (in)	`nmChannelHandle`	Identification of the NM-channel.
	`nmSleepReadyBit`	Value written to ReadySleep Bit in CBV
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Writing of remote sleep indication bit has failed.

5.4.2.2.20. FrNm_SetUserData

Purpose	This function sets user data for NM-Data transmitted next on the bus.
Synopsis	`Std_ReturnType` **`FrNm_SetUserData`** `( NetworkHandleType NetworkHandle , const uint8 * nmUserDataPtr );`
Service ID	0x06
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.
	`nmUserDataPtr`	User data for the next transmitted NM message.
Return Value	Std_ReturnType	
	`E_OK`	No error
	`E_NOT_OK`	Setting of user data has failed
Description	If user data handling is enabled for the FrNm module, then this function shall set the user data.

5.4.2.2.21. FrNm_StartupError

Purpose	This function is called by the FrSM when synchronization of the FlexRay cluster could not be achieved.
Synopsis	`void` **`FrNm_StartupError`** `( NetworkHandleType NetworkHandle );`
Service ID	0x10
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	`NetworkHandle`	Identification of the NM-channel.

5.4.2.2.22. FrNm_Transmit

Purpose	This is an empty function returning E_NOT_OK at any time. This requirement is relevant to avoid linker errors as PduR expects this API to be provided.	
Synopsis	`Std_ReturnType` **`FrNm_Transmit`** `( PduIdType FrNmTxPduId , const PduInfoType * PduInfoPtr );`	
Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	`FrNmTxPduId`	L-PDU handle of FlexRay L-PDU to be transmitted. This handle specifies the corresponding FlexRay L-PDU ID and implicitly the FlexRay Driver instance as well as the corresponding FlexRay controller device.
	`PduInfoPtr`	Pointer to a structure with FlexRay L-PDU related data: DLC and pointer to FlexRay L-SDU buffer.
Return Value	Std_ReturnType	
	`E_OK`	No Transmit request has been accepted.
	`E_NOT_OK`	Transmit request has not been accepted (FrNm is not in RM or NO).

5.4.2.2.23. FrNm_TriggerTransmit

Purpose	The lower layer communication module requests the buffer of the SDU for transmission from the upper layer module.	
Synopsis	`Std_ReturnType` **`FrNm_TriggerTransmit`** `( PduIdType TxPduId , PduInfoType * PduInfoPtr );`	
Service ID	0x41	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds but not for same	
Parameters (in)	`TxPduId`	ID of the SDU that is requested to be transmitted.
	`PduInfoPtr`	Contains a pointer to a buffer to where the SDU shall be copied to. On return,

		the service will indicate the length of the copied SDU data in SduLength.
Return Value	Std_returnType	
	`E_OK`	SDU has been copied and SduLength indicates the number of copied bytes
	`E_NOT_OK`	No SDU has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data
Description	This function shall copy the triggered FlexRay NM PDU with respect to the triggered FlexRay NM PDU ID. This function might be called by the FrNm module's environment in an interrupt context.	

5.4.2.2.24. FrNm_TxConfirmation

Purpose	The lower layer communication module confirms the transmission of an I-PDU.
Synopsis	void **FrNm_TxConfirmation** (PduIdType TxPduId);
Service ID	0x40
Sync/Async	Synchronous
Reentrancy	Reentrant for different PduIds but not for same
Parameters (in)	`TxPduId`	ID of the I-PDU that has been transmitted.

5.4.3. Integration notes

5.4.3.1. Exclusive areas

This section describes the exclusive areas used by the `FrNm` module.

5.4.3.1.1. SCHM_FRNM_EXCLUSIVE_AREA_0

Protected data structures	All shared data that shall be protected from mutual access.
Recommended locking mechanism	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the EB

	tresos AutoCore Generic documentation. Refer to the section `Mapping exclusive areas in the basic software modules` in the `Integration notes` section for details.

`FrNm` uses exclusive areas for protecting the global data against concurrent read/write access:

▶ The status of `FrNm` channels - the consistency of this global variable must be assured as it can be read/written by the `FrNm` state machine and/or following user interfaces:

    ▶ `FrNm_NetworkRequest()`

    ▶ `FrNm_NetworkRelease()`

    ▶ `FrNm_PassiveStartUp()`

    ▶ `FrNm_RepeatMessageRequest()`

▶ The partial networking bits - the consistency of this global data must be assured as it can be read/written by the `FrNm` state machine and/or `RxIndication`.

▶ The NM PDU data - the consistency of the PDU data must be assured as it can be read/written by the `RxIndication` and/or following interfaces:

    ▶ `FrNm_GetUserData()`

    ▶ `FrNm_GetPduData()`

    ▶ `FrNm_SetUserData()`

    ▶ `FrNm_GetNodeIdentifier()`

    ▶ `FrNm_TriggerTransmit()`

    ▶ `FrNm_RepeatMessageRequest()`

▶ The syncronization loss data - the consistency of this global data must be assured as it can be read/written by the `FrNm` state machine and/or `RxIndication` and/or following interface:

    ▶ `FrNm_StartupError()`

5.4.3.2. Production errors

Production errors are not reported by the `FrNm` module.

5.4.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
CODE
CONFIG_DATA_UNSPECIFIED
CONFIG_DATA_8
VAR_INIT_BOOLEAN
VAR_INIT_8
CONST_8
CONST_32
CONST_UNSPECIFIED
VAR_CLEARED_8
VAR_CLEARED_UNSPECIFIED

5.4.3.4. Integration requirements

WARNING	Integration requirements list is not exhaustive
⚠	The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

5.4.3.4.1. lim.FrNm.EB_INTREQ_FrNm_0001

Description	Length of FrNmRxPduId and FrNmTxPduId. The length of the FrNmRxPduId configured in the ECU configuration must be equal when separate PDUs are used for NM-Vote and NM-Data and length of the FrNmTxPduId must not be greater than length of the FrNmRxPduId.
Rationale	This limitation allows a more efficient implementation.

5.4.3.4.2. lim.FrNm.EB_INTREQ_FrNm_0002

Description	Requirement FRNM307 allows a reinitialization of the module. However, implementation does not support concurrent access to FrNm_Init() and FrNm_Mainfunction(). User has to ensure that FrNm_Init() shall not be called during the execution of FrNm_-Mainfunction() and vice versa.

Rationale	This limitation allows a more efficient implementation.

5.4.3.4.3. FrNm.EB_INTREQ_FrNm_0003

Description	The data to be transmitted as part of the NM PDU is always updated in the FrNm main processing function. The FlexRay job list shall be configured in such a way that the PDU data is always updated after the execution of the FrNm_MainFunction and before the beginning of the static segment in order to ensure that the latest data is always visible also on the bus.
Rationale	The FrNm PDU, including the vote bit, is always updated in the FrNm_MainFunction. As long as all the state transitions in the Network Mode are aligned with the repetition cycle, the update of the vote bit is also aligned with the repetition cycle. In order to have the data on the bus aligned with the changes of the NM PDU from FrNm, the configuration of the FlexRay job list shall ensure that the PDU data is updated after the execution of the FrNM main processing function and before the start of the static segment (especially for the case when the vote is sent in the static segment).

5.5. FrSM

5.5.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
FrSMDefensiveProgramming	1..1	**Label:** Defensive Programming Options Parameters for defensive programming
FrSMConfig	1..1	This container comprises the cluster specific configuration of the FlexRay State Manager.
ReportToDem	1..1	**Label:** Production Error Handling Production error handling

Containers included		
FrSMGeneral	1..1	**Label:** General Configuration This container contains the general configuration parameters of the FlexRay State Manager.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

Parameters included	
Parameter name	**Multiplicity**
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	**IMPLEMENTATION_CONFIG_VARIANT**
Label	Configuration Variant
Description	Configuration variant. Only post-build configuration is supported.
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.5.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	**ArMajorVersion**

Label	AUTOSAR Major Version
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ArMinorVersion
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL

Default value	5
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMinorVersion**
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwPatchVersion**
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	19
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ModuleId**
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	142
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**VendorId**

Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

5.5.1.2. FrSMDefensiveProgramming

Parameters included	
Parameter name	**Multiplicity**
FrSMDefProgEnabled	1..1
FrSMPrecondAssertEnabled	1..1
FrSMPostcondAssertEnabled	1..1
FrSMStaticAssertEnabled	1..1
FrSMUnreachAssertEnabled	1..1
FrSMInvariantAssertEnabled	1..1

Parameter Name	FrSMDefProgEnabled
Label	Enable Defensive Programming
Description	Enables or disables the defensive programming feature for the module FrSM. Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows:

	1. Enable development error detection
	2. Enable defensive programming
	3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMPrecondAssertEnabled**
Label	Enable Precondition Assertions
Description	Enables handling of precondition assertion checks reported from the module FrSM. Dependency on parameter(s): ► Enable Development Error Detection (`FrSMDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrSMDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMPostcondAssertEnabled**
Label	Enable Postcondition Assertions
Description	Enables handling of postcondition assertion checks reported from the module FrSM. Dependency on parameter(s): ► Enable Development Error Detection (`FrSMDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrSMDefProgEnabled`): must be enabled
Multiplicity	1..1

Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrSMStaticAssertEnabled**	
Label	Enable Static Assertions	
Description	Enables handling of static assertion checks reported from the module FrSM. Dependency on parameter(s): ► Enable Development Error Detection (`FrSMDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrSMDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrSMUnreachAssertEnabled**	
Label	Enable Unreachable Code Assertions	
Description	Enables handling of unreachable code assertion checks reported from the module FrSM. Dependency on parameter(s): ► Enable Development Error Detection (`FrSMDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrSMDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrSMInvariantAssertEnabled	
Label	Enable Invariant Assertions	
Description	Enables handling of invariant assertion checks reported from functions of the module FrSM. Dependency on parameter(s): ▶ Enable Development Error Detection (`FrSMDevErrorDetect`): must be enabled ▶ Enable Defensive Programming (`FrSMDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.5.1.3. FrSMConfig

Containers included		
Container name	Multiplicity	Description
FrSMCluster	1..n	This container specifies a FlexRay cluster and all related data.

5.5.1.4. FrSMCluster

Containers included		
Container name	Multiplicity	Description
FrSMClusterDemEventParameterRefs	1..1	**Label:** Container For DemEventParameter References Container for the references to `DemEventParameter` elements which shall be invoked using the API `Dem_ReportErrorStatus()` in case the corresponding error occurs. The EventId is taken from the referenced `DemEventParameter`'s `DemEventId` value. The standard-

Containers included		
		ized errors are provided in this container and may be extended by vendor specific error references.

Parameters included	
Parameter name	**Multiplicity**
FrSMCheckWakeupReason	1..1
FrSMDelayStartupWithoutWakeup	1..1
FrSMDurationT1	1..1
FrSMDurationT2	1..1
FrSMDurationT3	1..1
FrSMIsColdstartEcu	1..1
FrSMIsWakeupEcu	1..1
FrSMMainFunctionCycleTime	1..1
FrSMMinNumberOfColdstarter	0..1
FrSMNumWakeupPatterns	1..1
FrSMStartupRepetitions	0..1
FrSMStartupRepetitionsWithWakeup	0..1
FrSMComMNetworkHandleRef	1..1
FrSMFrIfClusterRef	1..1

Parameter Name	**FrSMCheckWakeupReason**	
Label	Enable Wakeup Reason Check	
Description	Enables wakeup reason check in order to skip the wakeup in case of wakeup by bus. ► `true`: the FrSM will check the wakeup reason in order to skip the wakeup in case of wakeup by bus. ► `false`: the FrSM will always try to perform a wakeup.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrSMDelayStartupWithoutWakeup**

Label	Enable Delay Of Startup Without Wakeup	
Description	This parameter enables the start of timer t1 instead of the immediate call of `FrIf_AllowColdstart()` in case of a startup without wakeup.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrSMDurationT1	
Label	Timer t1 Duration [s]	
Description	The duration of timer t1 in seconds. The timer models the delay of clearing the coldstart inhibit mode (i.e. the delay until calling `FrIf_AllowColdstart()`). If the parameter is set to 0, there is no delay. Range: ► 0 ► `FrSMMainFunctionCycleTime .. FrSMMainFunctionCycleTime * 65535`	
Multiplicity	1..1	
Type	FLOAT	
Default value	0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrSMDurationT2	
Label	Timer t2 Duration [s]	
Description	The duration of timer t2 in seconds. The timer models the time difference after which the FrSM will repeat the startup of the FlexRay cluster. If the parameter is set to 0, the FrSM will not repeat the startup. Range: ► 0 (interpreted as infinite) ► `FrSMMainFunctionCycleTime .. FrSMMainFunctionCycleTime * 65535`	
Multiplicity	1..1	

Type	FLOAT	
Default value	0.5	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrSMDurationT3	
Label	Timer t3 Duration [s]	
Description	The duration of timer t3 in seconds. The timer supervises the transition to communication mode `COMM_FULL_COMMUNICATION`. If the parameter is set to 0, the transition is not supervised, i.e. the duration of timer t3 is infinite. Range: ► 0 (interpreted as infinite) ► `FrSMMainFunctionCycleTime .. FrSMMainFunctionCycleTime * 65535`	
Multiplicity	1..1	
Type	FLOAT	
Default value	2	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrSMIsColdstartEcu	
Label	Is Coldstart ECU	
Description	► `true`: The ECU is a coldstart node for this FlexRay cluster. ► `false`: The ECU is no coldstart node for this FlexRay cluster.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrSMIsWakeupEcu	
Label	Is Wakeup ECU	
Description	► `true`: FrSM shall perform a wakeup for this cluster.	

	► `false`: FrSM shall never perform a wakeup for this FlexRay cluster.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrSMMainFunctionCycleTime**	
Label	Main Function Cycle Time [s]	
Description	This parameter defines the cycle time of the periodic calling of the FrSM main function in seconds. Range: 0.00025 ..	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.005	
Range	<=1	
	>=0.00025	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrSMMinNumberOfColdstarter**	
Label	Minimum Number of Cold Starters	
Description	This parameter defines the number of coldstarter that should not be underrun. If this parameter is not configured the mainfunction shall not check the number of startup frames. Range: 0..255 **This feature is currently not supported.**	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	**FrSMNumWakeupPatterns**	
Label	Maximum Number Of Wakeup Patterns	
Description	Maximum number of Wakeup Patterns the node may send before going to `FRSM_STARTUP`. Range: 1 .. 255	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrSMStartupRepetitions**	
Label	Startup Repetitions	
Description	The number of times an ECU may repeat the startup procedure for a FlexRay cluster. If this parameter is disabled, the number of repetitions is unlimited. Range: ► Parameter enabled : 0 .. 65534 ► Parameter disabled : Infinite Dependency on parameter: ► `FrSMStartupRepetitionsWithWakeup` must be smaller than or equal to this parameter.	
Multiplicity	0..1	
Type	INTEGER	
Default value	100	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrSMStartupRepetitionsWithWakeup**

Label	Startup Repetitions With Wakeup
Description	The number of times an ECU may repeat the startup procedure including a wakeup for a FlexRay cluster.
	If this parameter is disabled, the number of repetitions is unlimited.
	Range:
	► Parameter enabled : 0 .. 65534
	► Parameter disabled : Infinite
Multiplicity	0..1
Type	INTEGER
Default value	10
Range	<=65534
	>=0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrSMComMNetworkHandleRef**
Label	ComM Network Handle Reference
Description	Reference to the unique handle to identify one certain FlexRay network.
	Note: The reference corresponds to one of the network handles of the ComM configuration.
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrSMFrIfClusterRef**
Label	FrIf Cluster Configuration Reference
Description	References the cluster configuration in the FlexRay Interface configuration.
	Note: The assigned controllers and transceivers are defined in the FrIf configuration and can be accessed via this reference.
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.5. FrSMClusterDemEventParameterRefs

Parameters included	
Parameter name	**Multiplicity**
FRSM_E_CLUSTER_STARTUP	0..1
FRSM_E_CLUSTER_SYNC_LOSS	0..1

Parameter Name	FRSM_E_CLUSTER_STARTUP
Label	Reference to FRSM_E_CLUSTER_STARTUP
Description	Reference to the `DemEventParameter` that shall be issued when the error `FRSM_E_CLUSTER_STARTUP` occurs. Dependency on parameter(s): ► `FrSMClusterStartupReportToDem`: Select DEM to enable the reporting of `FRSM_E_CLUSTER_STARTUP`. Further notes: ► Activation: This error is reported if the start-up is not performed within time T3. ► Healing: This error is healed as soon as a start-up is successfully performed. ► Trigger debounce: None. The error is reported on first occurrence. ► Rate of diagnostic checks: Checked on every Flexray start-up.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FRSM_E_CLUSTER_SYNC_LOSS
Label	Reference to FRSM_E_CLUSTER_SYNC_LOSS
Description	Reference to the `DemEventParameter` that shall be issued when the error `FRSM_E_CLUSTER_SYNC_LOSS` occurs.

	Dependency on parameter(s): ► `FrSMClusterSyncLossReportToDem`: Select DEM to enable the reporting of `FRSM_E_CLUSTER_SYNC_LOSS`. Further notes: ► Activation: This error is reported if the FlexRay controller moves from FlexRay POC state normal-active to any other FlexRay POC state without user request. ► Healing: This error is healed as soon as a start-up is successfully performed and FlexRay POC state normal-active is reached. ► Trigger debounce: None. The error is reported on first occurrence. ► Rate of diagnostic checks: Checked on every FrSM_Mainfunction() call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

5.5.1.6. ReportToDem

Parameters included	
Parameter name	**Multiplicity**
FrSMClusterStartupReportToDem	1..1
FrSMClusterStartupReportToDemDetErrorId	1..1
FrSMClusterSyncLossReportToDem	1..1
FrSMClusterSyncLossReportToDemDetErrorId	1..1

Parameter Name	FrSMClusterStartupReportToDem
Label	FlexRay Startup Error Handling
Description	Selects the handling of the production error: *FlexRay startup could not reach the state normal active within the configured time* ► `DEM`: All errors are reported to the Diagnostics Event Manager (Dem). ► `DET`: All errors are reported to the Development Error Tracer (Det) if enabled. ► `DISABLE`: Production errors are not reported at all.

	Optimization Effect: ▶ **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM
Range	DEM DET DISABLE
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMClusterStartupReportToDemDetErrorId**
Label	FlexRay Startup Det Error ID
Description	If a production error is reported towards the Det, this parameter defines the error id which is reported towards the Det. The Det instance id is the ComM channel ID (parameter ComMChannelId of the ComM channel referenced by parameter FrSMComMNetworkHandleRef).
Multiplicity	1..1
Type	INTEGER
Default value	129
Range	<=255
Configuration class	**PreCompile:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMClusterSyncLossReportToDem**
Label	FlexRay Sync Loss Error Handling
Description	Selects the handling of the production error: *The FlexRay cluster has lost its synchronization* ▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem). ▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled.

	▶ DISABLE: Production errors are not reported at all. **Optimization Effect:** ▶ **ROM reduction (code):** Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ▶ **Execution time reduction (code):** Setting this parameter to a value of DISABLE reduces the execution time of the module code.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DEM	
Range	DEM	
	DET	
	DISABLE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrSMClusterSyncLossReportToDemDetErrorId**	
Label	FlexRay Sync Loss Det Error ID	
Description	If a production error is reported towards the Det, this parameter defines the error id which is reported towards the Det. The Det instance id is the ComM channel ID (parameter ComMChannelId of the ComM channel referenced by parameter FrSMComMNetworkHandleRef).	
Multiplicity	1..1	
Type	INTEGER	
Default value	130	
Range	<=255	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.5.1.7. FrSMGeneral

Parameters included	
Parameter name	**Multiplicity**

Parameters included	
FrSMDevErrorDetect	1..1
FrSMSyncLossErrorIndicationName	0..1
FrSMVersionInfoApi	1..1
FrSMFrTrcvControlEnable	1..1
FrSMComMIndicationEnable	1..1
FrSMSingleClstOptEnable	1..1
FrSMReportToBswMEnable	1..1
FrSMSetEcuPassiveEnable	1..1
FrSMFrNmStartupErrorEnable	1..1
FrSMKeySlotOnlyModeEnable	1..1
FrSMSyncLossErrorIndicationHeaderName	0..1
FrSMMultiCoreSupportEnable	1..1

Parameter Name	FrSMDevErrorDetect	
Label	Enable Development Error Detection	
Description	Enables and disables the development error detection and notification mechanism. ► `true`: Development error detection and development error reporting is enabled. ► `false`: Development error detection and development error reporting is disabled. **Optimization Effect:** ► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code. ► **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrSMSyncLossErrorIndicationName

Label	SyncLoss Error Indication function
Description	Name of <Cdd>_SyncLossErrorIndication function that shall be called on loss of synchronization. If this parameter is omitted no indication shall take place.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrSMVersionInfoApi
Label	Enable Version Info API
Description	Enables and disables the version info API. ▶ `true`: `FrSM_GetVersionInfo()` is available. ▶ `false`: `FrSM_GetVersionInfo()` is not available. **Optimization Effect:** ▶ **ROM reduction (config):** Disabling this parameter reduces the ROM consumption of the module configuration.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrSMFrTrcvControlEnable
Label	Enable Control Of FlexRay Transceiver
Description	Allows the FlexRay state manager to control the FlexRay transceiver module according to the specification. ▶ `true`: The FlexRay State Manager controls the FlexRay transceiver module according to the specification. ▶ `false`: The FlexRay State Manager does not control the FlexRay transceiver module. **Optimization Effect:** ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.

Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMComMIndicationEnable**
Label	Enable Call Of ComM_ComModeIndication()
Description	Switches the call of the API service `ComM_ComModeIndication()` on or off.
	► `true`: API service `ComM_ComModeIndication()` is called according to the specification.
	► `false`: API service `ComM_ComModeIndication()` is not called.
	Optimization Effect:
	► **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMSingleClstOptEnable**
Label	Enable Single Cluster Optimization
Description	Optimization for a configuration consisting of a single FlexRay cluster.
	► `true`: Enables optimization but limits the configuration to a single FlexRay cluster.
	► `false`: Disables optimization but allows to configure more than one FlexRay cluster.
	Optimization Effect:
	► **ROM reduction (config):** Enabling this parameter reduces the ROM consumption of the module configuration.
	► **ROM reduction (code):** Enabling this parameter reduces the ROM consumption of the module code.

	► **Execution time reduction (code):** Enabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMReportToBswMEnable**
Label	Enable Report To BswM
Description	Switches the report of transitions of the FrSM state machine to the Basic Software Mode Manager via `BswM_FrSM_CurrentState()` on or off.
	► `true`: State machine transitions are reported to Basic Software Mode Manager as specified in the AUTOSAR SWS.
	► `false`: State machine transitions are not reported to Basic Software Mode Manager.
	Optimization Effect:
	► **Execution time reduction (code):** Disabling this parameter decreases the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrSMSetEcuPassiveEnable**
Label	Provide API FrSM_SetEcuPassive()
Description	Switches the API service `FrSM_SetEcuPassive()` on or off.
	► `true`: `FrSM_SetEcuPassive()` is available.
	► `false`: `FrSM_SetEcuPassive()` is not available.
	Optimization Effect:
	► **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.

Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrSMFrNmStartupErrorEnable**	
Label	Enable Call Of FrNm_StartupError	
Description	Switches the call of API service `FrNm_StartupError()` on or off. ▶ `true`: `FrNm_StartupError()` is called according to the specification. ▶ `false`: `FrNm_StartupError()` is not called. **Optimization Effect:** ▶ **Execution time reduction (code):** Disabling this parameter reduces the execution time of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	**FrSMKeySlotOnlyModeEnable**	
Label	Enable Single Slot Mode Support	
Description	Enables/disables support for single slot (key slot) mode. ▶ `true`: single slot mode is supported and API service `FrSM_AllSlots()` is provided. ▶ `false`: single slot mode is not supported and API service `FrSM_AllSlots()` is not provided. **Optimization Effect:** ▶ **ROM reduction (code):** Disabling this parameter reduces the ROM consumption of the module code.	
Multiplicity	1..1	
Type	BOOLEAN	

Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrSMSyncLossErrorIndicationHeaderName	
Label	SyncLoss Error Indication function header	
Description	Name of header file which contains the declaration of the <Cdd>_SyncLossErrorIndication function. Must be in format: AnyValidFileName.h	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	**PreCompile:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrSMMultiCoreSupportEnable	
Label	Enable multicore support	
Description	Enables/disables support for multicore mode. ▶ `true`: Multi core mode is supported and API service `Schm_Call()` is used. ▶ `false`: Multi core mode is not supported and API service `ComM_BusSM_-ModeIndication()` is used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.5.1.8. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support

Description	Specifies whether or not the FrSM can use the PbcfgM module for post-build support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

5.5.2. Application programming interface (API)

5.5.2.1. Type definitions

5.5.2.1.1. FrSM_ClstRuntimeDataType

Purpose		
Type	struct	
Members	`uint16 T1Counter`	internal counter representing timer T1
	`uint16 T2Counter`	internal counter representing timer T2
	`uint16 T3Counter`	internal counter representing timer t3
	`uint16 StartupCounter`	(saturating) startup counter
	`uint8 WakeupCounter`	wakeup counter
	`uint8 State`	state of internal state machine
	`uint8 RequestedState`	Requested state of internal state machine. Either FRSM_STATE_READY (corresponds to reqComMode = NoCom) or FRSM_STATE_ONLINE (corresponds to reqComMode = FullCom)
	`uint8 WakeupType`	WS state machine variable wakeupType. Values are of type FrSM_WakeupType-Type
	`uint8 preemptionData`	contains information of state machine lock and preemption state Bit 0: statemachine

		is currently processed (mutex) Bit 1: requestComMode_NoCommunication has been suspended Bit 2: mainFunction has been suspended
	`boolean WakeupTransmitted`	SWS state machine variable wakeupTransmitted.
	`boolean PartialWakeupOnChannelA`	Partial wakeup was detected on channel A. Only valid if WakeupType == FRSM_DUAL_CHANNEL_ECHO_WAKEUP

5.5.2.1.2. FrSM_WakeupTypeType

Purpose	Values of SWS state machine variable wakeupType.	
Type	enum	
Constants	FRSM_SINGLE_CHANNEL_WAKEUP	
	FRSM_DUAL_CHANNEL_WAKEUP	
	FRSM_DUAL_CHANNEL_WAKEUP_FORWARD	
	FRSM_DUAL_CHANNEL_ECHO_WAKEUP	
	FRSM_NO_WAKEUP	

5.5.2.2. Macro constants

5.5.2.2.1. FRSM_E_INV_HANDLE

Purpose	DET error code.
Value	2U
Description	Invalid network handle parameter.

5.5.2.2.2. FRSM_E_INV_MODE

Purpose	DET error code.

Value	4U
Description	Invalid communication mode requested.

5.5.2.2.3. FRSM_E_MODEINDICATION

Purpose	DET error code.
Value	250U
Description	SchM_Call_ComM_BusSM_Modeindication failed.

5.5.2.2.4. FRSM_E_NULL_PTR

Purpose	DET error code.
Value	1U
Description	Invalid pointer in parameter list.

5.5.2.2.5. FRSM_E_UNINIT

Purpose	DET error code.
Value	3U
Description	FrSM module was not initialized.

5.5.2.2.6. FRSM_INVALID_DEM_EVENTID

Purpose	0 is not a valid Dem_EventIdType value
Value	0U

5.5.2.2.7. FRSM_MODULE_ID

Purpose	Module information.
Value	142U
Description	Module ID for module FrSM.

5.5.2.2.8. FRSM_REPETITIONS_INFINITE_VALUE

Purpose	If the FrSM_ClstCfgType members StartupRepetitions or StartupRepetitionsWith-Wakeup have this value, they shall be treated as corresponding to positive infinity.
Value	0xffffU

5.5.2.2.9. FRSM_VENDOR_ID

Purpose	Module information.
Value	1U
Description	Vendor ID (EB) for the module FrSM.

5.5.2.3. Objects

5.5.2.3.1. FrSM_ClstRuntimeData

Purpose	
Type	FrSM_ClstRuntimeDataType

5.5.2.4. Functions

5.5.2.4.1. FrSM_AllSlots

Purpose	This API function can be used to leave the KeySlotOnlyMode.	
Synopsis	Std_ReturnType **FrSM_AllSlots** (NetworkHandleType NetworkHandle);	
Service ID	5	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different FlexRay clusters	
Parameters (in)	NetworkHandle	This parameter identifies the FlexRay cluster for which a communication mode is requested.

Return Value	Std_ReturnType	
	E_OK	Request accepted
	E_NOT_OK	Request not accepted
Description	This function calls FrIf_AllSlots for the controller of the FlexRay cluster.	

5.5.2.4.2. FrSM_GetCurrentComMode

Purpose	Service to receive the current communication mode.	
Synopsis	Std_ReturnType **FrSM_GetCurrentComMode** (NetworkHandleType NetworkHandle , ComM_ModeType * ComM_ModePtr);	
Service ID	3	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different FlexRay clusters	
Parameters (in)	NetworkHandle	Handle of communication network
Parameters (out)	ComM_ModePtr	Pointer to the memory location where the current communication mode shall be stored
Return Value	Std_ReturnType	
	E_OK:	Request accepted
	E_NOT_OK:	Request was not accepted as the FrSM has not been initialized using FrSM_Init.
Description	This service receives the current communication mode from the FrSM.

Only COMM_FULL_COMMUNICATION or COMM_NO_COMMUNICATION will be received. | |

5.5.2.4.3. FrSM_GetVersionInfo

Purpose	Get version information of the FlexRay State Manager.
Synopsis	void **FrSM_GetVersionInfo** (Std_VersionInfoType * versioninfo);
Service ID	4
Sync/Async	Synchronous
Reentrancy	Reentrant

Parameters (out)	versioninfo	Pointer where to store the version information of this module.
Description	This service returns the version information of this module. The version information includes: ► Module Id ► Vendor Id ► Vendor specific version numbers	

5.5.2.4.4. FrSM_Init

Purpose	Initialization service for module FrSM.	
Synopsis	void **FrSM_Init** (const FrSM_ConfigType * FrSM_ConfigPtr);	
Service ID	1	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	FrSM_ConfigPtr	Pointer to a selected configuration structure
Description	This service initializes the FlexRay State Manager local variables and the state machine for each cluster (state FRSM_READY). It initially stores the post-build-time configuration passed as argument to enable subsequent service calls to access the configuration.	

5.5.2.4.5. FrSM_MainFunction

Purpose	FrSM Main Function.
Synopsis	void **FrSM_MainFunction** (const uint8 FrSM_ClstIdx);
Service ID	0x80
Sync/Async	Synchronous
Reentrancy	Non Re-entrant
Production Errors	► FRSM_E_CLUSTER_STARTUP: thrown, if Fr startup could not reach the state normal active within the configured time. ► FRSM_E_CLUSTER_SYNC_LOSS: thrown, if Fr cluster has lost its synchronization.

Parameters (in)	FrSM_ClstIdx	Index of FrSMCluster container according to document order
Description	Main function that drives the statemachine. Executed per cluster.	

5.5.2.4.6. FrSM_RequestComMode

Purpose	Communication mode change request service.	
Synopsis	Std_ReturnType **FrSM_RequestComMode** (NetworkHandleType NetworkHandle , ComM_ModeType ComM_Mode);	
Service ID	2	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different FlexRay clusters	
Parameters (in)	NetworkHandle	This parameter identifies the FlexRay cluster for which a communication mode is requested.
	ComM_Mode	This parameter holds the requested communication mode.
Return Value	Std_ReturnType	
	E_OK	Request accepted
	E_NOT_OK	Request not accepted
Description	This service requests a change of the communication mode from the FrSM. This service doesn't necessarily change into the requested communication mode synchronously but might store the request internally and process it via further FrSM_MainFunction_#() invocations. Only COMM_FULL_COMMUNICATION and COMM_NO_COMMUNICATION may be requested. If COMM_SILENT_COMMUNICATION is requested, the service will return E_NOT_OK.	

5.5.2.4.7. FrSM_SetEcuPassive

Purpose	Service to set all FlexRay clusters of the ECU to receive only mode.
Synopsis	Std_ReturnType **FrSM_SetEcuPassive** (boolean FrSM_Passive);

Service ID	6	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	`FrSM_Passive`	If true set to passive mode. If false set to active mode.
Return Value	Std_ReturnType	
	`E_OK`	Request accepted
	`E_NOT_OK`	Request not accepted

5.5.3. Integration notes

5.5.3.1. Exclusive areas

This section describes the exclusive areas used by the `FrSM` module.

5.5.3.1.1. SCHM_FRSM_EXCLUSIVE_AREA_0

Protected data structures	All shared data that shall be protected from mutual access.
Recommended locking mechanism	The locking mechanism for this exclusive area can be disabled if at least one of the following conditions is true: ► FrSM_RequestComMode() does not interrupt FrSM_-MainFunction() (and vice versa) If the condition listed above does not apply, the exclusive area shall be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to the section `Mapping exclusive areas in the basic software modules` in the `Integration notes` section for details.

5.5.3.2. Production errors

FRSM_E_CLUSTER_STARTUP	► FrSM_MainFunction
FRSM_E_CLUSTER_SYNC_LOSS	► FrSM_MainFunction

5.5.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
VAR_INIT_8
VAR_CLEARED_8
VAR_CLEARED_UNSPECIFIED
CONST_UNSPECIFIED
CODE
CONFIG_DATA_UNSPECIFIED

5.5.3.4. Integration requirements

WARNING	Integration requirements list is not exhaustive
⚠️	The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

5.5.3.4.1. lim.FrSM.EB_INTREQ_FrSM_0001

Description	FrSM_SetEcuPassive() and FrSM_MainFunction_<Cluster Id>() must not execute concurrently. The API functions FrSM_SetEcuPassive() and FrSM_MainFunction_<Cluster Id>() must not execute concurrently.
Rationale	If FrSM_SetEcuPassive() and FrSM_MainFunction_<Cluster Id>() execute concurrently, they might change the transceiver state simultaneously. This limitation avoids the overhead associated with consistency actions for the simultaneous transceiver state change.

5.6. FrTp

5.6.1. Configuration parameters

Containers included		
Container name	**Multiplicity**	**Description**
CommonPublishedInformation	1..1	**Label:** Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
FrTpDefensiveProgramming	1..1	**Label:** Defensive Programming Options Parameters for defensive programming
FrTpGeneral	1..1	This container contains the general configuration parameters of the FlexRay Transport Protocol module.
FrTpMultipleConfig	1..1	This container holds one or several multiple configuration sets.
PublishedInformation	1..1	**Label:** EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

Parameters included	
Parameter name	**Multiplicity**
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.6.1.1. CommonPublishedInformation

Parameters included	
Parameter name	**Multiplicity**
ArMajorVersion	1..1
ArMinorVersion	1..1

Parameters included	
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
Label	AUTOSAR Major Version
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	4
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ArMinorVersion
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL

Default value	0
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMajorVersion**
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	4
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwMinorVersion**
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	4
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**SwPatchVersion**
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	27
Configuration class	**PublishedInformation:**
Origin	Elektrobit Automotive GmbH

Parameter Name	**ModuleId**
Label	Numeric Module ID

Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	36
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH

5.6.1.2. FrTpDefensiveProgramming

Parameters included	
Parameter name	**Multiplicity**
FrTpDefProgEnabled	1..1
FrTpPrecondAssertEnabled	1..1
FrTpPostcondAssertEnabled	1..1
FrTpStaticAssertEnabled	1..1

Parameters included	
FrTpUnreachAssertEnabled	1..1
FrTpInvariantAssertEnabled	1..1

Parameter Name	FrTpDefProgEnabled	
Label	Enable Defensive Programming	
Description	Enables or disables the defensive programming feature for the module FrTp. Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows: 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpPrecondAssertEnabled	
Label	Enable Precondition Assertions	
Description	Enables handling of precondition assertion checks reported from the module FrTp. Dependency on parameter(s): ► Enable Development Error Detection (`FrTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrTpDefProgEnabled`): must be enabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpPostcondAssertEnabled

Label	Enable Postcondition Assertions
Description	Enables handling of postcondition assertion checks reported from the module FrTp. Dependency on parameter(s): ► Enable Development Error Detection (`FrTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrTpStaticAssertEnabled**
Label	Enable Static Assertions
Description	Enables handling of static assertion checks reported from the module FrTp. Dependency on parameter(s): ► Enable Development Error Detection (`FrTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrTpUnreachAssertEnabled**
Label	Enable Unreachable Code Assertions
Description	Enables handling of unreachable code assertion checks reported from the module FrTp. Dependency on parameter(s):

	► Enable Development Error Detection (`FrTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	**FrTpInvariantAssertEnabled**
Label	Enable Invariant Assertions
Description	Enables handling of invariant assertion checks reported from functions of the module FrTp. Dependency on parameter(s): ► Enable Development Error Detection (`FrTpDevErrorDetect`): must be enabled ► Enable Defensive Programming (`FrTpDefProgEnabled`): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.6.1.3. FrTpGeneral

Containers included		
Container name	**Multiplicity**	**Description**
VendorSpecific	1..1	

Parameters included	
Parameter name	**Multiplicity**
FrTpAckRt	1..1

Parameters included	
FrTpChanNum	1..1
FrTpChangeParamApi	1..1
FrTpDevErrorDetect	1..1
FrTpFullDuplexEnable	1..1
FrTpMainFuncCycle	1..1
FrTpTransmitCancellation	1..1
FrTpUnknownMsgLength	1..1
FrTpVersionInfoApi	1..1
FrTpTxPduNum	1..1

Parameter Name	FrTpAckRt	
Description	*The functionality related to this parameter is not supported by the current implementation.* Preprocessor switch for enabling the Acknowledgement and retry mechanisms. ► `True`: Acknowledge and Retry is enabled ► `False`: Acknowledge and Retry is disabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpChanNum	
Description	Preprocessor switch for defining the number of concurrent channels the module supports. Up to 32 channels shall be definable here.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=32	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpChangeParamApi	
Description	Preprocessor switch for enabling the API to change FrTp communication parameters. ► `True`: ChangeParameter API is enabled ► `False`: ChangeParameter API is disabled **This feature is currently not supported.**	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpDevErrorDetect	
Description	Preprocessor switch for enabling development error detection. ► `True`: Development Error Detection is enabled ► `False`: Development Error Detection is disabled	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpFullDuplexEnable	
Description	Preprocessor switch for enabling full duplex mechanisms for all channels. ► `True`: Full duplex is enabled ► `False`: Fullduplex is disabled (Half duplex is enabled)	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMainFuncCycle

Description	This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.
Multiplicity	1..1
Type	FLOAT
Default value	0.005
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrTpTransmitCancellation**
Description	Preprocessor switch for enabling Transmit Cancellation. ► `True`: Transmit Cancellation is enabled ► `False`: Transmit Cancellation is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrTpUnknownMsgLength**
Description	*The functionality related to this parameter is not supported by the current implementation.* Preprocessor switch to support data transfer with unknown message length. ► `True`: Transmission with unknown message length is enabled ► `False`: Transmission with unknown message length is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrTpVersionInfoApi**
Description	Preprocessor switch for enabling the Version info API. ► `True`: Version Info API is enabled

	► False: Version Info API is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	**FrTpTxPduNum**	
Description	Preprocessor switch for defining the maximum number of TxPdus that can be configured.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.6.1.4. VendorSpecific

Containers included		
Container name	**Multiplicity**	**Description**
FrTpConnectionLimitConfig	0..n	Maps a maximum number of active connections and a buffer size to a remote address.

Parameters included	
Parameter name	**Multiplicity**
FrTpMainfunctionsPerCommunicationCycle	1..1
FrTpRelocatablePbcfgEnable	1..1
FrTpCopyToLocalBuffer	1..1
FrTpLocalBufferSize	1..1
FrTpLimitNumberOfConnections	1..1
FrTpConnectionBufferSizeUnlimited	1..1

Parameter Name	FrTpMainfunctionsPerCommunicationCycle	
Description	This config parameter contains the number of FrTp_MainFunction() invocations within a single FlexRay communications cycle. Note: Since there is only a single FrTp_MainFunction(), FrTp works only for multiple FlexRay clusters if they all use the same FlexRay communication cycle length.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpRelocatablePbcfgEnable	
Description	Enables/disables support for relocatable postbuild configuration. ► True: Postbuild configuration relocatable in memory. ► False: Postbuild configuration not relocatable in memory.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpCopyToLocalBuffer	
Description	Enables/disables support for copy of STF to the local buffer, in case PduR doesn't provide enough buffer for STF payload. ► True: Enable copy of STF to local buffer. ► False: Disable copy of STF to local buffer.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	Elektrobit Automotive GmbH

Parameter Name	FrTpLocalBufferSize	
Description	Specifies the size of the local buffer used for STF copy.	
Multiplicity	1..1	
Type	INTEGER	
Default value	64	
Range	<=246	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpLimitNumberOfConnections	
Description	Enables/disables support for buffering data transfer requests in cases where all channels are allocated. ► True: Enable buffer ► False: Disable buffer	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpConnectionBufferSizeUnlimited	
Description	Specifies the maximum number of buffered data transfer requests for unlimited connections (i.e. connections whose remote address is not listed in FrTpConnectionLimitConfig).	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	Elektrobit Automotive GmbH

5.6.1.5. FrTpConnectionLimitConfig

Parameters included	
Parameter name	**Multiplicity**
FrTpRa	1..1
FrTpConnectionLimit	1..1
FrTpConnectionBufferSize	1..1

Parameter Name	FrTpRa	
Description	Specifies the remote address.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpConnectionLimit	
Description	Specifies the maximum number of active connections.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=32	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	FrTpConnectionBufferSize
Description	Specifies the maximum number of buffered data transfer requests for this re-mote address.
Multiplicity	1..1
Type	INTEGER
Range	<=255
	>=0

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.6.1.6. FrTpMultipleConfig

Containers included		
Container name	**Multiplicity**	**Description**
FrTpConnection	1..n	This container contains the connection specific parameters to transfer N-PDUs via FlexRay TP.
FrTpConnectionControl	1..n	This container contains the configuration parameters to control a FlexRay TP connection.
FrTpRxPduPool	0..n	This container contains all Pdus that are assigned to that Pdu Pool.
FrTpTxPduPool	0..n	This container contains all Pdus that are assigned to that Pdu Pool.

5.6.1.7. FrTpConnection

Containers included		
Container name	**Multiplicity**	**Description**
FrTpRxSdu	0..1	This parameter defines the Rx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR.
FrTpTxSdu	0..1	This parameter defines the Tx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR.

Parameters included	
Parameter name	**Multiplicity**
FrTpBandwidthLimitation	1..1
FrTpLa	1..1
FrTpMultipleReceiverCon	1..1
FrTpRa	1..1
FrTpConCtrlRef	1..1

Parameters included	
FrTpRxPduPoolRef	1..1
FrTpTxPduPoolRef	1..1

Parameter Name	FrTpBandwidthLimitation	
Description	This parameter indicates whether the connection requires a bandwidth limitation or not. If FrTpBandwidthLimitation=True the sender shall send a Start-Frame always on the first PDU of a PDU-Pool.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpLa	
Description	This parameter defines the Local Address for the respective connection. ► When the local instance is the sender, this is the Source Address within the TP frame. ► When the local instance is the receiver, this is the Target Address within the TP frame.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMultipleReceiverCon
Description	This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection. If data segmentation is required this parameter is used to check whether segmentation is possible or not. If the connection is 1:n segmentation is not possible and an error will occur.
Multiplicity	1..1
Type	BOOLEAN

Default value	false	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpRa**	
Description	This parameter defines the Remote Address for the respective connection. ► When the local instance is the sender, this is the Target Address within the TP frame. ► When the local instance is the receiver, this is the Source Address within the TP frame.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpConCtrlRef**	
Description	FrTpConnectionControlReference: This parameter defines a reference to a connection control container.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpRxPduPoolRef**	
Description	This parameter defines a reference to a RxPduPool.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpTxPduPoolRef**	
Description	This parameter defines a reference to a TxPduPool.	
Multiplicity	1..1	
Type	REFERENCE	

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.8. FrTpRxSdu

Parameters included	
Parameter name	**Multiplicity**
FrTpRxSduId	1..1
FrTpRxSduRef	1..1

Parameter Name	**FrTpRxSduId**	
Description	This unique identifier is used for change parameter request or receive cancellation from PduR to FrTp. ImplementationType: `PduIdType` **This parameter is currently not used**	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpRxSduRef**	
Description	Reference to a PDU in the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.9. FrTpTxSdu

Parameters included	
Parameter name	**Multiplicity**

Parameters included	
FrTpTxSduId	1..1
FrTpTxSduRef	1..1

Parameter Name	FrTpTxSduId	
Description	This is a unique identifier for a to be transmitted message from the PduR to the FrTp. ImplementationType: `PduIdType`	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpTxSduRef	
Description	Reference to a PDU in the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.10. FrTpConnectionControl

Parameters included	
Parameter name	**Multiplicity**
FrTpAckType	1..1
FrTpMaxAr	1..1
FrTpMaxAs	1..1
FrTpMaxBufferSize	1..1
FrTpMaxFCWait	1..1
FrTpMaxFrIf	1..1
FrTpMaxNbrOfNPduPerCycle	1..1
FrTpMaxRn	1..1
FrTpSCexp	1..1

Parameters included	
FrTpTimeBr	1..1
FrTpTimeBuffer	1..1
FrTpTimeFrIf	1..1
FrTpTimeoutAr	1..1
FrTpTimeoutAs	1..1
FrTpTimeoutBs	1..1
FrTpTimeoutCr	1..1
FrTpMaxBufReq	1..1

Parameter Name	FrTpAckType	
Description	This parameter defines the type of acknowledgement which is used for the specific channel.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	FRTP_NO	
Range	FRTP_ACK_WITH_RT	
	FRTP_NO	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMaxAr	
Description	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs. **This parameter is currently not used.**	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMaxAs

Description	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs. **This parameter is currently not used.**	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpMaxBufferSize**	
Description	Limits the maximal buffer size the FrTp can choose in order to limit the amount of Tx buffer that will be requested at the sender side in a segmented transfer. **This parameter is currently not used.**	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535	
	>=1	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	**FrTpMaxFCWait**	
Description	This parameter defines the maximum number of FlowControl N-PDUs with FlowState "WAIT".	
Multiplicity	1..1	
Type	INTEGER	
Default value	8	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMaxFrIf	
Description	This parameter defines the maximum number of trying to send a frame when the FrIf returns an error.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMaxNbrOfNPduPerCycle	
Description	This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=31	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMaxRn	
Description	This parameter defines the maximum number of retries (if retry is configured).	
Multiplicity	1..1	
Type	INTEGER	
Default value	4	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpSCexp
Description	This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It represents the exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrTpTimeBr
Description	This parameter defines the time in seconds the FrTp requires to transmit a corresponding FlowControl Frame. According to ISO 10681-2 this parameter is a performance requirement.
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Range	<=0.255 >=0
Configuration class	**VariantPostBuild:** VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	FrTpTimeBuffer
Description	This parameter defines the time in seconds of waiting for the next try to get a Tx or Rx buffer. **This parameter is currently not used.**
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Range	<=65.535 >=0
Configuration class	**VariantPostBuild:** VariantPostBuild

Origin	AUTOSAR_ECUC

Parameter Name	FrTpTimeFrIf	
Description	This parameter defines the time in seconds of waiting for the next try (if retry is activated) to send via FrIf_Transmit. **This parameter is currently not used.**	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Range	<=0.255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpTimeoutAr	
Description	This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.02	
Range	<=65.535	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpTimeoutAs
Description	This parameter specifies the timeout in seconds the FrIf shall confirm a transmitted Pdu to the FrTp.
Multiplicity	1..1
Type	FLOAT
Default value	0.02
Range	<=65.535
	>=0

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpTimeoutBs	
Description	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.5	
Range	<=65.535	
	>=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpTimeoutCr	
Description	This parameter defines the timeout value in seconds a receiver is waiting for a CF or a LF.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.5	
Range	<=65.535	
	>=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpMaxBufReq	
Description	This parameter is used to limit the number of retries for PduR_FrTpCopyTxData when no timer is active.	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	<=255	
	>=0	

Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.6.1.11. FrTpRxPduPool

Containers included		
Container name	**Multiplicity**	**Description**
FrTpRxPdu	0..n	Container to hold the PDU parameters. ImplementationType: `PduInfoType`

5.6.1.12. FrTpRxPdu

Parameters included	
Parameter name	**Multiplicity**
FrTpRxPduId	1..1
FrTpRxPduRef	1..1

Parameter Name	FrTpRxPduId	
Description	This is a unique identifier for a received message which is forwarded from the FrIf to the FrTp. ImplementationType: `PduIdType`	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpRxPduRef	
Description	Reference to a PDU in the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild

Origin	AUTOSAR_ECUC

5.6.1.13. FrTpTxPduPool

Containers included		
Container name	**Multiplicity**	**Description**
FrTpTxPdu	1..n	Container to hold the PDU parameters. ImplementationType: `PduInfoType`

5.6.1.14. FrTpTxPdu

Parameters included	
Parameter name	**Multiplicity**
FrTpTxConfirmationPduId	1..1
FrTpTxPduRef	1..1

Parameter Name	FrTpTxConfirmationPduId	
Description	Handle Id to be used by the FrIf to confirm the transmission of the FrTpTxPdu to the FrIf module.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	
	>=0	
Configuration class	**VariantPostBuild:**	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	FrTpTxPduRef	
Description	Reference to a PDU in the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	**VariantPostBuild:**	VariantPostBuild

Origin	AUTOSAR_ECUC

5.6.1.15. PublishedInformation

Parameters included	
Parameter name	**Multiplicity**
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport	
Label	PbcfgM support	
Description	Specifies whether or not the FrTp can use the PbcfgM module for post-build support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	**PublishedInformation:**	
Origin	Elektrobit Automotive GmbH	

5.6.2. Application programming interface (API)

5.6.2.1. Type definitions

5.6.2.1.1. FrTp_CancelReasonType

Purpose		
Type	enum	
Constants	FRTP_CNLDO	Cancel Transfer because data are outdated
	FRTP_CNLNB	Cancel Transfer because no further buffer can be provided
	FRTP_CNLOR	Cancel Transfer because of another reason

5.6.2.1.2. FrTp_ParameterValueType

Purpose	
Type	`uint8`

5.6.2.2. Macro constants

5.6.2.2.1. FRTP_CANCELRECEIVE_SERVICE_ID

Purpose	
Value	0x08U

5.6.2.2.2. FRTP_CANCELTRANSMIT_SERVICE_ID

Purpose	
Value	0x03U

5.6.2.2.3. FRTP_CHANGEPARAMETER_SERVICE_ID

Purpose	
Value	0x04U

5.6.2.2.4. FRTP_E_INVALID_PARAMETER

Purpose	
Value	0x04U

5.6.2.2.5. FRTP_E_INVALID_PDU_SDU_ID

Purpose	
Value	0x03U

5.6.2.2.6. FRTP_E_NO_CHANNEL

Purpose	
Value	0x07U

5.6.2.2.7. FRTP_E_NULL_PTR

Purpose	
Value	0x02U

5.6.2.2.8. FRTP_E_SEG_ERROR

Purpose	
Value	0x05U

5.6.2.2.9. FRTP_E_UMSG_LENGTH_ERROR

Purpose	
Value	0x06U

5.6.2.2.10. FRTP_E_UNINIT

Purpose	
Value	0x01U

5.6.2.2.11. FRTP_GETVERSIONINFO_SERVICE_ID

Purpose	
Value	0x27U

5.6.2.2.12. FRTP_INIT_SERVICE_ID

Purpose	

Value	0x00U

5.6.2.2.13. FRTP_ISVALIDCONFIG_SERVICE_ID

Purpose	
Value	0x60U

5.6.2.2.14. FRTP_MAINFUNCTION_SERVICE_ID

Purpose	
Value	0x10U

5.6.2.2.15. FRTP_RXINDICATION_SERVICE_ID

Purpose	
Value	0x42U

5.6.2.2.16. FRTP_SID_UNKNOWN_API

Purpose	
Value	0xffU

5.6.2.2.17. FRTP_TRANSMIT_SERVICE_ID

Purpose	
Value	0x02U

5.6.2.2.18. FRTP_TRIGGERTRANSMIT_SERVICE_ID

Purpose	
Value	0x41U

5.6.2.2.19. FRTP_TXCONFIRMATION_SERVICE_ID

Purpose	
Value	0x40U

5.6.2.2.20. NTFRSLT_E_ABORT

Purpose	
Value	0x10U

5.6.2.2.21. NTFRSLT_E_FR_ML_MISMATCH

Purpose	
Value	0x5BU

5.6.2.2.22. NTFRSLT_E_FR_TX_ON

Purpose	
Value	0x5DU

5.6.2.2.23. NTFRSLT_E_FR_WRONG_BP

Purpose	
Value	0x5CU

5.6.2.3. Functions

5.6.2.3.1. FrTp_CancelReceive

Purpose	This service primitive is used to cancel the transfer of pending Fr N-SDUs. The connection is identified by FrTpRxSduId.

Synopsis	`Std_ReturnType` **`FrTp_CancelReceive`** `( PduIdType FrTpRxSduId );`	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	`FrTpRxPduId`	Contains the FlexRay TP instance unique identifier of the Fr N-SDU to be cancelled.
Return Value	E_OK Cancellation request of the transfer (sending or receiving) of the specified Fr N-SDU is accepted. E_NOT_OK Cancellation request of the transfer of the specified Fr N-SDU is rejected.	

5.6.2.3.2. FrTp_CancelTransmit

Purpose	This service primitive is used to cancel the transfer of pending Fr N-SDUs. The connection is identified by FrTpTxSduId.	
Synopsis	`Std_ReturnType` **`FrTp_CancelTransmit`** `( PduIdType FrTpTxPduId );`	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	`FrTpTxPduId`	Contains the FlexRay TP instance unique identifier of the Fr N-SDU to be cancelled.
Return Value	E_OK Cancellation request of the transfer (sending or receiving) of the specified Fr N-SDU is accepted. E_NOT_OK Cancellation request of the transfer of the specified Fr N-SDU is rejected.	

5.6.2.3.3. FrTp_ChangeParameter

Purpose	This service primitive is used to request the change of the value of the FRTP_STMIN parameter.

Synopsis	`Std_ReturnType` **`FrTp_ChangeParameter`** `( PduIdType FrTpTxPduId ,`
	`TPParameterType parameter , FrTp_ParameterValueType FrTpParame-`
	`terValue );`

Service ID	0x04
Sync/Async	Asynchronous
Reentrancy	Reentrant

Parameters (in)	FrTpTxPduId	Gives the ID of the connection (message) for whose channel the change shall be done.
	parameter	The selected parameter that the request shall change.
	FrTpParameterValue	contains the new value of Bandwidth Control (BC).
Return Value	E_OK request is accepted	
	E_NOT_OK request is not accepted	

5.6.2.3.4. FrTp_GetVersionInfo

Purpose	This service returns the version information of this module. The version information includes: Module Id, Vendor Id, Instance ID and Vendor specific version numbers.
Synopsis	void **FrTp_GetVersionInfo** (Std_VersionInfoType * versioninfo);
Service ID	0x27
Sync/Async	Synchronous
Reentrancy	Non reentrant
Parameters (out)	versioninfo

Parameters (out)	versioninfo	Pointer to where to store the version information of this module.

5.6.2.3.5. FrTp_Init

Purpose	This service initializes all global variables of a FlexRay Transport Layer instance and set it in the idle state.
Synopsis	void **FrTp_Init** (const FrTp_ConfigType * PBCfgPtr);
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non reentrant
Parameters (in)	PBCfgPtr

Parameters (in)	PBCfgPtr	Addres of the post-build configuration

5.6.2.3.6. FrTp_IsValidConfig

Purpose	Validate configuration.

Synopsis	`Std_ReturnType FrTp_IsValidConfig ( const void * voidConfigPtr );`
Service ID	0x60
Sync/Async	Synchronous
Reentrancy	Reentrant
Return Value	E_OK if the given module configurations is valid otherwise E_NOT_OK.
Description	Checks if the post build configuration fits to the link time configuration part.

5.6.2.3.7. FrTp_MainFunction

Purpose	The main function for scheduling the TP (Entry point for scheduling).
Synopsis	`void FrTp_MainFunction ( void );`
Service ID	0x10

5.6.2.3.8. FrTp_RxIndication

Purpose	The FlexRay Interface calls this primitive after the reception of an Fr N-PDU.	
Synopsis	`void FrTp_RxIndication ( PduIdType RxPduId , PduInfoType * PduInfoPtr );`	
Service ID	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	`RxPduId`	This parameter contains the identifier of the received Fr N-PDU.
	`PduInfoPtr`	Pointer Pdu info structure.

5.6.2.3.9. FrTp_Transmit

Purpose	Request the transfer of data.
Synopsis	`Std_ReturnType FrTp_Transmit ( PduIdType FrTpTxPduId , const PduInfoType * PduInfoPtr );`
Service ID	0x02
Sync/Async	Asynchronous

Reentrancy	Reentrant	
Parameters (in)	FrTpTxPduId	Contains the FlexRay TP instance unique identifier of the Fr N-SDU to be transmitted.
	PduInfoPtr	A pointer to a structure with Fr N-SDU related data: data length and pointer to an Fr N-SDU buffer.
Return Value	E_OK The request has been accepted E_NOT_OK The request has not been accepted, e. g. due to a still ongoing transmission in the corresponding channel.	

5.6.2.3.10. FrTp_TriggerTransmit

Purpose	This function is called by the FlexRay Interface for sending out a FlexRay frame.	
Synopsis	Std_ReturnType **FrTp_TriggerTransmit** (PduIdType FrTxConfirmationPduId , PduInfoType * PduInfoPtr);	
Service ID	0x41	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	FrTxConfirmationPduId	Confirmation ID of FlexRay L-PDU that is requested to be transmitted.
Parameters (out)	PduInfoPtr	Pointer Pdu info structure.
Return Value	Std_ReturnType	
	E_OK	SDU has been copied and SduLength indicates the number of copied bytes.
	E_NOT_OK	No SDU has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.

5.6.2.3.11. FrTp_TxConfirmation

Purpose	This function is called by the FlexRay Interface after the TP-related Pdu has been transmitted over the network.
Synopsis	void **FrTp_TxConfirmation** (PduIdType FrTxConfirmationPduId);
Service ID	0x40

Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	`FrTxConfirmationPduId`	This parameter contains the confirmation identifier of the transmitted Fr N-PDU.

5.6.3. Integration notes

5.6.3.1. Exclusive areas

This section describes the exclusive areas used by the `FrTp` module.

5.6.3.1.1. SCHM_FRTP_EXCLUSIVE_AREA_0

Protected data structures	All shared data that shall be protected from mutual access.
Recommended locking mechanism	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the `EB tresos AutoCore Generic documentation`. Refer to the section `Mapping exclusive areas in the basic software modules` in the `Integration notes` section for details.

5.6.3.2. Production errors

Production errors are not reported by the `FrTp` module.

5.6.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the `Integration notes` section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section

CODE
CONST_32
CONFIG_DATA_UNSPECIFIED
VAR_CLEARED_UNSPECIFIED
VAR_CLEARED_8
VAR_FAST_INIT_UNSPECIFIED
VAR_INIT_8

5.6.3.4. Integration requirements

WARNING **Integration requirements list is not exhaustive**

⚠️ The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

5.6.3.4.1. lim.FrTp.EB_INTREQ_FrTp_0001

Description	The integrator must assure that the following functions do not interrupt each other: - FrTp_TriggerTransmit - FrTp_TxConfirmation - FrTp_RxIndication - FrTp_MainFunction - FrTp_Init
Rationale	This limitation reduces code size and execution time.

5.6.3.4.2. lim.FrTp.EB_INTREQ_FrTp_0002

Description	It must be assured that the following functions are not preempted by any other function: - FrTp_TriggerTransmit - FrTp_TxConfirmation - FrTp_RxIndication

Rationale	This limitation reduces code size and execution time.

5.6.3.4.3. lim.FrTp.EB_INTREQ_FrTp_0003

Description	FrTp_Init must not preempt any other function.
Rationale	This limitation reduces code size and execution time.

Appendix A. ACG8 FlexRay Stack licenses

A.1. License information

This appendix provides general information about the licenses of third party software used in the external code generators for `Fr` and `FrIf`. The external code generators use the Xerces-C library to parse XML files. Xerces-C is licensed under the Apache License, see Section A.1.1, "Apache License". For parsing the command line parameters, Mark K. Kim's getopt library is used, see Section A.1.2, "Getopt License". MinGW provides the runtime libraries for the code generators, see Section A.1.3, "MinGW License".

A.1.1. Apache License

Copyright 2018 Elektrobit Automotive GmbH

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

A.1.2. Getopt License

Copyright (c)2002-2003 Mark K. Kim

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

▶ Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

▶ Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

▶ Neither the original author of this software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.1.3. MinGW License

MinGW Runtime with gcc is used to build xerces and the external code generators.

Copyright (c) 2012 MinGW.org project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice, this permission notice and the below disclaimer shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Further information can be found under http://mingw.org/license