

Document Title	Specification of FlexRay Transceiver Driver
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Version	1.1.0
Document Status	Draft
Part of Release	2.1
Revision	0014

Document Change History			
Date	Version	Changed by	Change Description
31.01.2007	1.1.0	AUTOSAR Administration	<ul style="list-style-type: none"> • Added header file includes: MemMap_<ModuleId>.h and SchM_<ModuleId>.h • Renamed error codes • Support of wake up interrupt sharing (callback only if wake up occurred) • FrTrcv API is only called via FrIf FlexRay Interface, which is transparent to the transceiver driver • Legal disclaimer revised • Release Notes added • “Advice for users” revised • “Revision Information” added
18.05.2006	1.0.0	AUTOSAR Administration	Initial release

Release Notes

Errata and known deficiencies

The wakeup concept is currently neither harmonized nor consistent throughout all wakeup related specification documents and therefore subject to change.

Known and potential problems resulting from known deficiencies

Due to the fact that the wakeup concept is not harmonized, inconsistent assumptions may lead to

- the duplication of functionalities across multiple modules
- proprietary implementation extensions
- difficulties during integration

Changes planned for next release

The harmonized wakeup concept throughout all wakeup related documents will result in

- adapted specification texts in Chapter 7 of the specification documents
- adapted APIs in Chapter 8 of the specification documents
- adapted wakeup sequences in Chapter 9 of the specification documents

Disclaimer

Any use of these specifications requires membership within the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of these specifications.

Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Copyright © 2004-2006 AUTOSAR Development Partnership. All rights reserved.

Advice to users of AUTOSAR Specification Documents:

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

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

Table of Contents

Release Notes	2
Errata and known deficiencies	2
Known and potential problems resulting from known deficiencies	2
Changes planned for next release	2
1 Introduction and functional overview	6
1.1 Goal of FlexRay transceiver driver	7
1.2 Explicitly uncovered FlexRay Transceiver Functionality	7
1.3 System Basis Chip and FlexRay Transceiver Driver	8
2 Acronyms and abbreviations	9
3 Related documentation.....	10
3.1 Input documents.....	10
3.2 Related standards and norms	10
4 Constraints and assumptions	12
4.1 Limitations	12
4.2 Applicability to car domains.....	12
5 Dependencies to other modules.....	13
5.1 File structure	14
5.1.1 Naming convention for transceiver driver implementation	14
5.1.2 Code file structure.....	14
5.1.3 Header file structure	15
6 Requirements traceability	17
6.1 Document: AUTOSAR requirements on Basic Software, general	17
6.2 Document: AUTOSAR_SRS_FlexRay.SRS	21
6.3 Document: AUTOSAR_SWS_ECU_StateManager.pdf.....	26
6.4 Document: AUTOSAR_SWS_ComStackTypes.doc.....	28
6.5 Document: AUTOSAR_SWS_BSW_Scheduler.pdf	28
6.6 Document: AUTOSAR_SWS_MemoryMapping.pdf	28
7 Functional specification	29
7.1 AUTOSAR FlexRay Transceiver Operation Mode Model.....	29
7.2 FlexRay transceiver hardware operation modes	30
7.2.1 Temporary “Go-To-Sleep” Mode.....	30
7.2.2 “Active Star” Mode	31
7.3 Error classification.....	31
7.4 Error detection.....	31
7.5 Error notification	32
7.5.1 BusTrcvErrorType.....	33
7.6 Preconditions for driver initialization	33
7.7 Instance concept	33
8 API specification.....	35
8.1 Imported types.....	35

8.1.1	Standard types.....	35
8.1.2	EcuM types.....	35
8.1.3	ComStack types.....	35
8.2	Type definitions	35
8.2.1	FrTrcv_TrvcModeType	35
8.2.2	FrTrcv_TrvcWUReasonType	36
8.3	Function definitions	36
8.3.1	FrTrcv_TrvcInit.....	36
8.3.2	FrTrcv_SetTransceiverMode	37
8.3.3	FrTrcv_GetTransceiverMode.....	38
8.3.4	FrTrcv_GetTransceiverWUReason.....	39
8.3.5	FrTrcv_GetVersionInfo	40
8.3.6	FrTrcv_DisableTransceiverWakeup.....	40
8.3.7	FrTrcv_EnableTransceiverWakeup	41
8.3.8	FrTrcv_ClearTransceiverWakeup.....	42
8.4	Scheduled functions	43
8.4.1	FrTrcv_MainFunction.....	43
8.5	Call-back notifications	44
8.5.1	FrTrcv_Cbk_WakeupByTransceiver	44
8.6	Expected Interfaces.....	44
8.6.1	Mandatory Interfaces	45
8.6.2	Optional Interfaces.....	45
8.6.3	Configurable interfaces.....	46
9	Sequence diagrams.....	47
10	Configuration specification	49
10.1	How to read this chapter	49
10.1.1	Configuration and configuration parameters.....	49
10.1.2	Variants	49
10.1.3	Containers	50
10.2	Containers and configuration parameters	51
10.2.1	Variants	51
10.2.2	General configuration requirements.....	51
10.2.3	Container FlexRayTransceiverDriverBasic	51
10.2.4	Container FlexRayTransceiverNode.....	53
10.3	Published Information.....	55
11	Changes to Release 1	56
11.1	Deleted SWS Items	56
11.2	Replaced SWS Items	56
11.3	Changed SWS Items.....	56
11.4	Added SWS Items.....	56

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module FlexRay Transceiver Driver which handles the FlexRay transceivers on an ECU.

The FlexRay Transceiver is a hardware device, which mainly transforms the logical I/O signals of the μ C ports to the bus compliant electrical levels, currents and timings.

Within an automotive environment there is currently only one single physical layer specification for FlexRay.

In addition, the transceivers could be able to detect electrical malfunctions like break of cable harness in the lifetime of the ECU, ground offsets (a certain ground shift is tolerated) or bus collisions. Depending on the interface they flag the detected error summarized by a single port pin or very detailed via SPI.

Currently wakeup via bus is mandatory for FlexRay transceivers. Some transceivers also support power supply control. Future markets will probably see a lot of different wakeup/sleep and power supply concepts.

A typical FlexRay transceiver is the TJA1080 for a FlexRay bus with support for fault tolerant communication and wakeup via bus.

Figure 1 depicts the basic structure of the FlexRay stack. One FlexRay Interface accesses several FlexRay Transceivers (Trcv Type X .. Z) using one or several FlexRay Transceiver Driver(s) (FrTrcv Driver Vendor A...C) from different vendors.

A zero based index (FrTrcv_TrvcIdx is) identifies the transceiver within the context of the transceiver driver:

E.g. Node A of FlexRay transceiver type Z is addressed by the index 0, node B by the index 1 in the example in Figure 1.

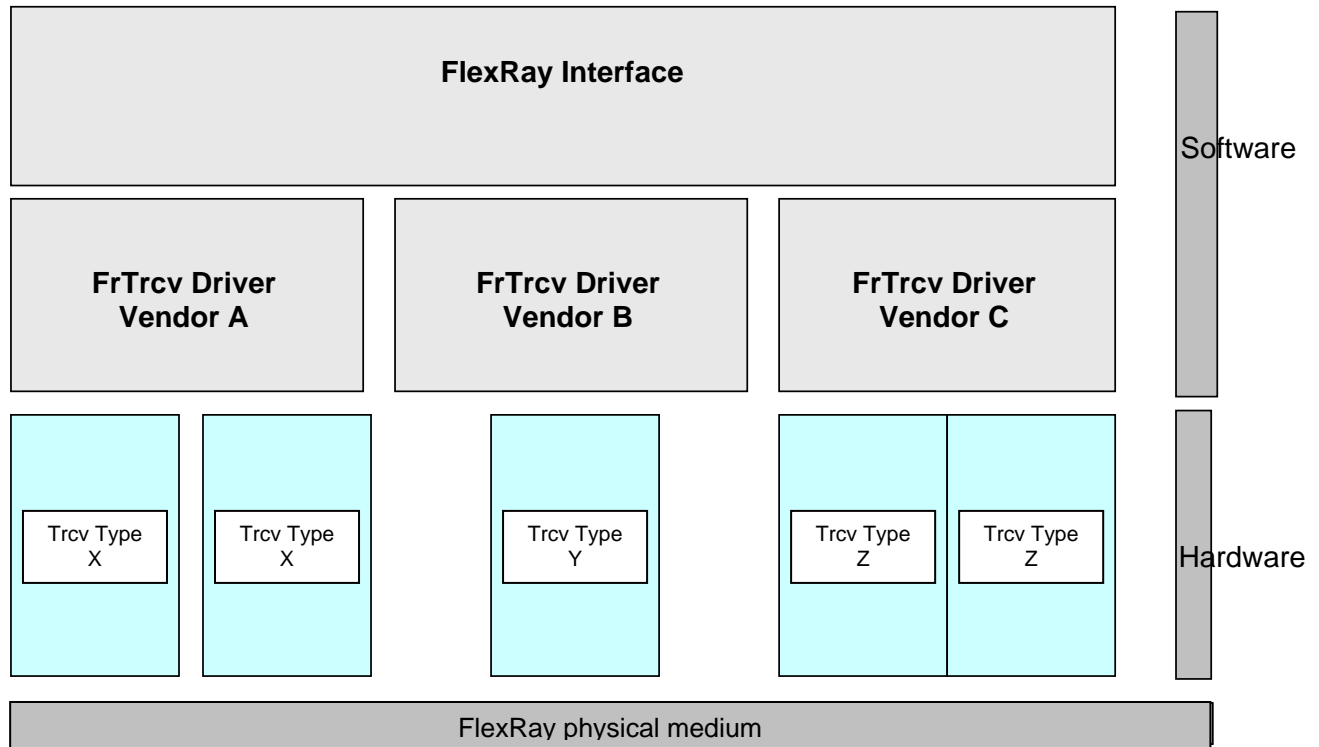


Figure 1: Description of the basic structure of the FlexRay Stack

1.1 Goal of FlexRay transceiver driver

This document specifies interfaces and sequence models which apply to current and future FlexRay transceiver hardware devices.

FrTrcv013: The FlexRay transceiver driver abstracts the usage of FlexRay transceiver hardware chips. It offers a hardware independent interface to the higher layers. It abstracts also from ECU layout by using APIs of MCAL layer to access FlexRay transceiver hardware.

1.2 Explicitly uncovered FlexRay Transceiver Functionality

Some FlexRay Transceivers offer additional functionality like ECU self test or error detection capability for diagnostics.

ECU self test and error detection are not defined within AUTOSAR and requiring such functionality in general will lock out most currently used (and cheap) transceiver devices. Therefore features like “ground shift detection”, “selective wake up”, “slope control” and others are not supported.

1.3 System Basis Chip and FlexRay Transceiver Driver

System basis chips (SBCs) contain beside FlexRay transceiver hardware additional hardware like voltage regulators or watchdogs.

The AUTOSAR concept provides a separate driver for each identified hardware. For AUTOSAR releases 1 and 2 such a driver is missing. Also the application of available drivers which cover functionality inside the SBC is not possible due to shared communication and dependencies of the integrated functions.

Thus AUTOSAR releases 1 and 2 do not support SBCs.

2 Acronyms and abbreviations

Abbreviation / Acronym:	Description:
ComM	Communication Manager, See [15] for details
Dem	Diagnostic Event Manager
Det	Development Error Tracer
Dio	Digital input output, one of the SPAL SW modules
EB	Externally buffered channel. Buffers containing data to transfer are outside the SPI Handler/Driver.
EcuM	ECU State Manager, see [14] for details
IB	Internally buffered channel. Buffers containing data to transfer are inside the SPI Handler/Driver.
ISR	Interrupt service routine
MCAL	Micro controller Abstraction Layer
Port	Port, one of the SPAL SW modules
n/a	Not applicable
SBC	System Basis Chip; a device, which integrates e.g. CAN and/or FlexRay and/or LIN transceiver, watchdog and power control.
SPI Channel	A channel is a software exchange medium for data that are defined with the same criteria: configuration parameters, number of data elements with same size and data pointers (source & destination) or location. See specification of SPI driver for more details.
SPI Job	A job is composed of one or several channels with the same chip select. A job is considered to be atomic and therefore cannot be interrupted. A job has also an assigned priority. See specification of SPI driver for more details.
SPI Sequence	A sequence is a number of consecutive jobs to be transmitted. A sequence depends on a static configuration. See specification of SPI driver for more details.

3 Related documentation

3.1 Input documents

[1] List of Basic Software Modules

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_BasicSoftwareModules.pdf

[2] Layered Software Architecture

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_LayeredSoftwareArchitecture.pdf

[3] ECU Configuration Specification

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_ECU_Configuration.pdf

[4] General Requirements on Basic Software

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_SRS_General.pdf

[5] FlexRay - EPL-Specification - V2.1

http://www.flexray.com/specification_request.php

FlexRay - EPL-Specification - V2.1.pdf

[6] FlexRay - EPL-Application Notes - V2.1

http://www.flexray.com/specification_request.php

FlexRay - EPL-Application Notes - V2.1.pdf

[8] FlexRay Communications System - EPL - V2.1 - Errata_3.pdf

http://www.flexray.com/specification_request.php

FlexRay Communications System - EPL - V2.1 - Errata_3.pdf

3.2 Related standards and norms

[14] Specification of ECU State Manager

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_SWS_ECU_StateManager.pdf

[15] Specification of Communication Manager

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_SWS_ComManager.pdf

[16] Specification of ICU driver

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_SWS_ICU_Driver.pdf

[17] Specification of DIO Driver

<https://svn.autosar.org/repos/10Releases>

AUTOSAR_SWS_DIO_Driver.pdf

[18] Specification of SPI Handler/Driver
<https://svn.autosar.org/repos/10Releases>
AUTOSAR_SWS_SPI_HandlerDriver.pdf

[19] Requirements on FlexRay
<https://svn.autosar.org/repos/10Releases>
AUTOSAR_SRS_FlexRay.pdf

[20] Specification of Communication Stack Types
<https://svn.autosar.org/repos/10Releases>
AUTOSAR_SWS_ComStackTypes.pdf

[21] Specification of BSW Scheduler
<https://svn.autosar.org/repos/10Releases>
AUTOSAR_SWS_BSW_Scheduler.pdf

[22] Specification of Memory Mapping
<https://svn.autosar.org/repos/10releases/>
AUTOSAR_SWS_MemoryMapping.pdf

4 Constraints and assumptions

4.1 Limitations

The FlexRay Transceiver must provide functionality and an interface, mapped to the operation mode model assumed for the AUTOSAR FlexRay Transceiver Driver. See 7.1 AUTOSAR FlexRay Transceiver Operation Modes.

FrTrcv231:

The used APIs of underlying drivers (SPI and Dio) shall be synchronous. Implementations of underlying drivers which does not support synchronous behaviour cannot be used together with FlexRay Transceiver Driver.

4.2 Applicability to car domains

This driver shall be applicable in all car domains using FlexRay for communication.

5 Dependencies to other modules

<i>Module</i>	<i>Dependencies</i>
ComM	ComM steers FlexRay Transceiver Driver communication modes. Independent steering of each single FlexRay transceiver node via FrIf.
Det	Det gets development error information from FlexRay Transceiver Driver.
Dem	Dem gets production error information from FlexRay Transceiver Driver.
Dio	Dio module is used to access FlexRay transceiver hardware connected via ports.
EcuM	EcuM gets wake up event information from FlexRay Transceiver Driver.
Icu	Icu module performs FlexRay transceiver hardware interrupts and calls appropriate call-back function inside FlexRay Transceiver Driver.
OS	FlexRay Transceiver Driver may use OS services to guarantee data consistency.
SPI	SPI module is used to access FlexRay transceiver hardware connected via SPI.

Please be aware although this documentation of the FlexRay transceiver consumes more of 50 pages of paper, in the end it will still resolve to setting a few bits in RAM and transferring them via SPI or setting a few port pins. This can be VERY small code (e.g. inline functions) in case post build time configuration is not required.

If an upper layer wants to call any node specific FlexRay API, knowledge which FlexRay transceiver driver it has to call for a specific communication node **is not required**. Only a mapping (=knowledge) generated by configuration is required! Here is an example:

Upper layer:

"Set transceivers of cluster C (within a single ECU) to state NORMAL"

FrIf (has cluster knowledge):

Cluster C uses CC Y which is connected to Xcvr Xa (node A) and Xb (node B)

"Set transceivers Xa and Xb to state NORMAL"

FrXcvr (has transceiver driver knowledge, assuming different drivers):

transceiver Xa is the 1st device within driver D1

transceiver Xb is the 3rd device within driver D2

"set Xa to normal via D1(1st device)"

"set Xb to normal via D2(3rd device)"

FlexRay Transceiver Driver FrXcfrD1 (has Xcvr HW knowledge):

NORMAL for 1st device is achieved by setting Dio signal S1 to HIGH and DIO Signal S2 to HIGH

"DIO set S1 and S2 to HIGH"

ECU Abstraction Layer (has ECU layout information):

Signal S1 is mapped to DIO channel C7

Signal S2 is mapped to DIO channel C8

DIO (has port/pin knowledge)

configuration maps C7 to PORTs.PINn and C8 to PORTt.PINm

set S1 to HIGH via PORTs.PINn ((Dio_WriteChannel(S1, Std_High);)

set S2 to HIGH via PORTt.PINm ((Dio_WriteChannel(S2, Std_High);)

5.1 File structure

5.1.1 Naming convention for transceiver driver implementation

FrTrcv059: A FlexRay Transceiver Driver implementation may support different FlexRay transceiver hardware. Thus BSW00347 is applied for the naming in a way that no FlexRay transceiver hardware specific naming extensions are used.

5.1.2 Code file structure

FrTrcv021: Naming convention is applied to all files.

FrTrcv058: Configuration is contained in *_Cfg.c

FrTrcv009: Module consists of listed files:

File name	Requirements	Description
FrTrcv.c	FrTrcv033:	The implementation general c file. It does not contain interrupt routines.
FrTrcv_Cfg.c	FrTrcv080: FrTrcv057: FrTrcv079: FrTrcv117:	Pre compile time configuration code file. It is generated by the configuration tool.

5.1.3 Header file structure

The header file structure shall include the following FlexRay-specific header files:

File name	Requirements	Description
FrTrcv.h	FrTrcv022: FrTrcv113:	General header file of the FlexRay Transceiver Driver. It contains only information relevant for other BSW modules (API). Differences in API depending on configuration are encapsulated.
FrTrcv_Cbk.h	FrTrcv070:	FrTrcv_Cbk.h contains call-back functions implemented in FrTrcv.c and called by other modules.
FrTrcv_Cfg.h	FrTrcv110:	Pre compile time configuration parameter file. It's generated by the configuration tool.
Fr_GeneralTypes.h		contains declarations shared by all AUTOSAR FlexRay BSW modules
SchM_FrTrcv.h	FrTrcv266:	contains schedule manager declarations used by the FlexRay Transceiver Driver specified by [21]
MemMap.h	FrTrcv267:	apply the memory mapping abstraction mechanisms specified by [22]

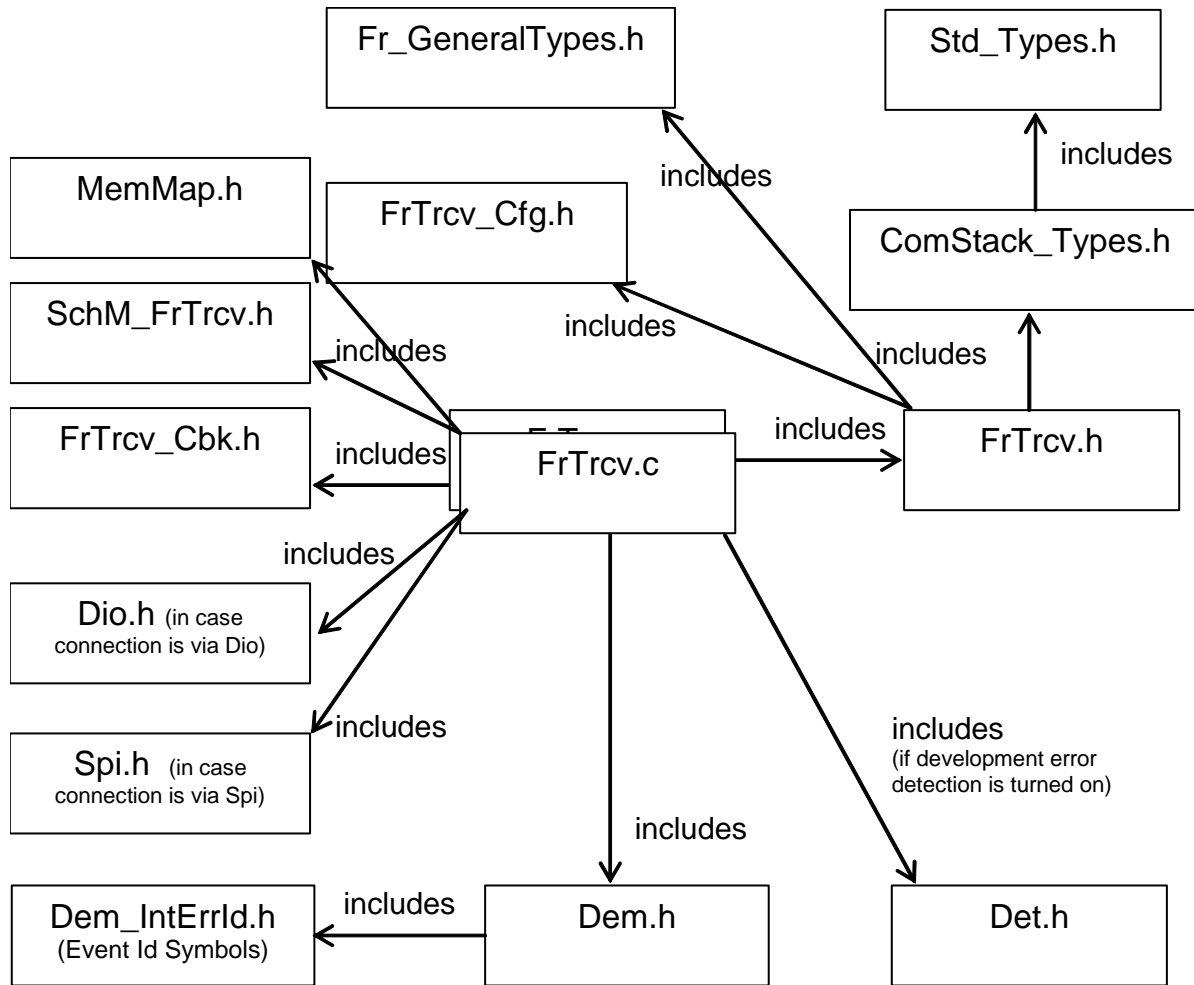


Figure 2 FlexRay Transceiver Driver Header File Structure

FrTrcv060: Std_Types.h includes platform specific header files and compiler specific header files. It defines standard data types and values for standard defines. This file is indirectly included via ComStack_Types.h.

FrTrcv068: Name of compiler specific header file is Compiler.h. All mappings of not standardized keywords of compiler specific scope shall be placed and organized in this compiler specific type and keyword header. This file is indirectly included via ComStack_Types.h.

FrTrcv062: Name of platform specific header file is Platform_Types.h. All integer type definitions of target and compiler specific scope shall be placed and organized in this single type header. This file is indirectly included via ComStack_Types.h.

Fr_GeneralTypes.h contains definitions shared by the FlexRay Transceiver Driver with the FlexRay driver and the FlexRay interface.

FrTrcv266: BSW scheduler information is included via SchM_FrTrcv.h.

FrTrcv267: BSW memory mapping information is included via MemMap.h.

6 Requirements traceability

6.1 Document: AUTOSAR requirements on Basic Software, general

Requirement	Description	Satisfied by
[BSW00003]	Version identification	FrTrcv001
[BSW00004]	Version check	not applicable to this SWS, general implementation requirement
[BSW00005]	No hard coded horizontal interfaces within MCAL	not applicable, FlexRay transceiver driver is part of ECU abstraction layer. It is not part of uC abstraction layer.
[BSW00006]	Platform independency	FrTrcv267
[BSW00007]	HIS MISRA C	not applicable to this SWS, general implementation requirement
[BSW00009]	Module User Documentation	not applicable, General documentation requirement
[BSW00010]	Memory resource documentation	not applicable, General documentation requirement
[BSW00101]	Initialization interface	FrTrcv008
[BSW00158]	Separation of configuration from implementation	FrTrcv009
[BSW00159]	Tool-based configuration	FrTrcv010
[BSW00160]	Human-readable configuration data	FrTrcv011
[BSW00161]	Microcontroller abstraction	not applicable, FlexRay transceiver driver is part of ECU abstraction layer. It is not part of uC abstraction layer.
[BSW00162]	ECU layout abstraction	FrTrcv013
[BSW00164]	Implementation of interrupt service routines	not applicable, FlexRay transceiver driver does not implement interrupt service routines
[BSW00167]	Static configuration checking	FrTrcv016
[BSW00168]	Diagnostic Interface of SW components	not applicable, FlexRay transceiver driver does not implement tester routines
[BSW00170]	Data for reconfiguration of AUTOSAR SW-Components	FrTrcv018
[BSW00171]	Configurability of optional functionality	FrTrcv019
[BSW00172]	Compatibility and documentation of scheduling strategy	FrTrcv020
[BSW00300]	Module naming convention	FrTrcv021
[BSW00301]	Limit imported information	FrTrcv022
[BSW00302]	Limit exported information	FrTrcv023
[BSW00304]	AUTOSAR integer data types	not applicable to this SWS, general implementation requirement
[BSW00305]	Self-defined data types naming convention	FrTrcv025
[BSW00306]	Avoid direct use of compiler and platform specific keywords	not applicable to this SWS, general implementation requirement
[BSW00307]	Global variables naming convention	not applicable to this SWS, general implementation requirement

[BSW00308]	Definition of global data	not applicable to this SWS, general implementation requirement
[BSW00309]	Global data with read-only constraint	not applicable to this SWS, general implementation requirement
[BSW00310]	API naming convention	FrTrcv030
[BSW00312]	Shared code shall be reentrant	not applicable to this SWS, general implementation requirement
[BSW00314]	Separation of interrupt frames and service routines	FrTrcv033
[BSW00318]	Format of module version numbers	FrTrcv034
[BSW00321]	Enumeration of module version numbers	not applicable to this SWS, general implementation requirement
[BSW00323]	API parameter checking	FrTrcv037
[BSW00325]	Runtime of interrupt service routines	not applicable, FlexRay transceiver driver does not implement interrupt service routines
[BSW00326]	Transition from ISRs to OS tasks	not applicable, no such transitions are performed
[BSW00327]	Error values naming convention	FrTrcv041
[BSW00328]	Avoid duplication of code	not applicable to this SWS, general implementation requirement
[BSW00329]	Avoidance of generic interfaces	FrTrcv043
[BSW00330]	Usage of macros / inline functions instead of functions	not applicable to this SWS, general implementation requirement
[BSW00331]	Separation of error and status values	FrTrcv045
[BSW00333]	Documentation of call-back function context	not applicable, General documentation requirement
[BSW00334]	Provision of XML file	FrTrcv011 not applicable to this SWS, general implementation requirement
[BSW00335]	Status values naming convention	FrTrcv048
[BSW00336]	Shutdown interface	not applicable, FlexRay transceiver driver does not have a need for such an interface
[BSW00337]	Classification of errors	FrTrcv050
[BSW00338]	Detection and Reporting of development errors	FrTrcv051
[BSW00339]	Reporting of production relevant error status	FrTrcv052
[BSW00341]	Microcontroller compatibility documentation	not applicable, General documentation requirement
[BSW00342]	Usage of source code and object code	not applicable to this SWS, general implementation requirement
[BSW00343]	Specification and configuration of time	FrTrcv055
[BSW00344]	Reference to link-time configuration	not applicable, FlexRay transceiver driver supports only pre compile time configuration
[BSW00345]	Pre-compile-time configuration	FrTrcv057
[BSW00346]	Basic set of module files	FrTrcv058
[BSW00347]	Naming separation of different instances of BSW drivers	FrTrcv059
[BSW00348]	Standard type header	FrTrcv060

[BSW00350]	Development error detection keyword	FrTrcv061
[BSW00353]	Platform specific type header	FrTrcv062
[BSW00355]	Do not redefine AUTOSAR integer data types	not applicable to this SWS, general implementation requirement
[BSW00357]	Standard API return type	FrTrcv064
[BSW00358]	Return type of init() functions	FrTrcv065
[BSW00359]	Return type of call-back functions	FrTrcv066
[BSW00360]	Parameters of call-back functions	FrTrcv067
[BSW00361]	Compiler specific language extension header	FrTrcv068
[BSW00369]	Do not return development error codes via API	FrTrcv069
[BSW00370]	Separation of call-back interface from API	FrTrcv070
[BSW00371]	Do not pass function pointers via API	FrTrcv071
[BSW00373]	Main processing function naming convention	FrTrcv072
[BSW00374]	Module vendor identification	FrTrcv073
[BSW00375]	Notification of wake-up reason	FrTrcv074
[BSW00376]	Return type and parameters of main processing functions	FrTrcv075
[BSW00377]	Module specific API return types	FrTrcv076
[BSW00378]	AUTOSAR boolean type	not applicable to this SWS, general implementation requirement
[BSW00379]	Module identification	FrTrcv078
[BSW00380]	Separate C-Files for configuration parameters	FrTrcv079
[BSW00381]	Separate configuration header file for pre-compile time parameters	FrTrcv080
[BSW00382]	Not-used configuration elements need to be listed [rejected]	not applicable, General documentation requirement
[BSW00383]	List dependencies of configuration files	not applicable, General documentation requirement
[BSW00384]	List dependencies to other modules	not applicable, General documentation requirement
[BSW00385]	List possible error notifications	FrTrcv084
[BSW00386]	Configuration for detecting an error	FrTrcv085
[BSW00387]	Specify the configuration class of call-back function	FrTrcv086
[BSW00388]	Introduce containers	FrTrcv087
[BSW00389]	Containers shall have names	FrTrcv088
[BSW00390]	Parameter content shall be unique within the module	FrTrcv089
[BSW00391]	Parameter shall have unique names	FrTrcv090
[BSW00392]	Parameters shall have a type	FrTrcv091
[BSW00393]	Parameters shall have a range	FrTrcv092
[BSW00394]	Specify the scope of the	FrTrcv093

	parameters	
[BSW00395]	List the required parameters (per parameter)	FrTrcv094
[BSW00396]	Configuration classes	FrTrcv095
[BSW00397]	Pre-compile-time parameters	FrTrcv096
[BSW00398]	Link-time parameters	not applicable, FlexRay transceiver driver supports only pre compile time configuration
[BSW00399]	Loadable Post-build time parameters	not applicable, FlexRay transceiver driver supports only pre compile time configuration
[BSW00400]	Selectable Post-build time parameters	not applicable, FlexRay transceiver driver supports only pre compile time configuration
[BSW00401]	Documentation of multiple instances of configuration parameters	not applicable, General documentation requirement
[BSW00402]	Published information	FrTrcv101
[BSW00404]	Reference to post build time configuration	not applicable, FlexRay transceiver driver supports only pre compile time configuration
[BSW00405]	Reference to multiple configuration sets	not applicable, FlexRay transceiver driver supports only pre compile time configuration
[BSW00406]	Check module initialization	FrTrcv104
[BSW00407]	Function to read out published parameters	FrTrcv105
[BSW00408]	Configuration parameter naming convention	FrTrcv106
[BSW00409]	Header files for production code error IDs	FrTrcv107
[BSW00410]	Compiler switches shall have defined values	not applicable to this SWS, general implementation requirement
[BSW00411]	Get version info keyword	FrTrcv109
[BSW00412]	Separate H-File for configuration parameters	FrTrcv110
[BSW00413]	Accessing instances of BSW modules	FrTrcv111
[BSW00414]	Parameter of init function	FrTrcv112
[BSW00415]	User dependent include files	FrTrcv113
[BSW00416]	Sequence of Initialization	not applicable, this is out of FlexRay transceiver driver's scope
[BSW00417]	Reporting of Error Events by Non-Basic Software	not applicable, requirement concerns application components only
[BSW00419]	Separate C-Files for pre-compile time configuration parameters	FrTrcv117
[BSW00420]	Production relevant error event rate detection	not applicable, it's an Dem requirement
[BSW00421]	Reporting of production relevant error events	FrTrcv119
[BSW00422]	Debouncing of production relevant error status	not applicable, it's an Dem requirement
[BSW00423]	Usage of SW-C template to	not applicable to this SWS, general

	describe BSW modules with AUTOSAR Interfaces	implementation requirement
[BSW00424]	BSW main processing function task allocation	FrTrcv122
[BSW00425]	Trigger conditions for schedulable objects	FrTrcv123
[BSW00426]	Exclusive areas in BSW modules	not applicable (no exclusive areas specified for this module)
[BSW00427]	ISR description for BSW modules	not applicable, no such areas or function in FlexRay transceiver driver
[BSW00428]	Execution order dependencies of main processing functions	FrTrcv126
[BSW00429]	Restricted BSW OS functionality access	not applicable to this SWS, general implementation requirement
[BSW00431]	The BSW Scheduler module implements task bodies	not applicable, requirement concerns BSW scheduler module
[BSW00432]	Modules should have separate main processing functions for read/receive and write/transmit data path	not applicable, FlexRay transceiver driver does not propagate data
[BSW00433]	Calling of main processing functions	not applicable, requirement concerns BSW scheduler module
[BSW00434]	The Schedule Module shall provide an API for exclusive areas	not applicable, requirement concerns BSW scheduler module
[BSW00435]	Header File Structure for the Basic Software Scheduler	FrTrcv266
[BSW00436]	Module Header File Structure for the Memory Mapping	FrTrcv267

6.2 Document: AUTOSAR_SRS_FlexRay.SRS

Requirement	Description	Satisfied by
[BSW05000]	Support of Synchronous SW Modules	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05001]	Support of Asynchronous SW Modules	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05002]	FlexRay Modules as Only Necessarily Synchronous SW Modules	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05003]	Support of Slot/Cycle Multiplexing	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05004]	PDU-Based Data API	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05005]	Support of Hardware FIFO Mechanism	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05006]	Abstraction of FlexRay-Specific Features	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05007]	Number of FlexRay CCs per	not applicable, this is out of FlexRay

	Interface	transceiver driver's scope
[BSW05009]	Local Memory Space Usage	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05010]	Unique PDU-ID	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05011]	Initialize Low-Level Parameters	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05012]	Initialize FlexRay CC Transmit/Receive Buffers	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05013]	Initialize Local Memory Space	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05015]	Start-up of a FlexRay CC	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05016]	Abortion of a FlexRay CC Communication	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05018]	Sending of a Wake-Up Pattern	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05019]	Get FlexRay Global Time	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05022]	Get FlexRay CC POC Status	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05023]	Get FlexRay CC Sync State	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05024]	Abstraction from CC Buffer Configuration	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05025]	Consistent Access to All Local Memory	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05027]	Transmit PDU	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05031]	Initialization of a FlexRay CC	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05033]	Tick Conversion	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05034]	Configuration Modifiable by a Flashing Process	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05035]	MTS Sending	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05038]	Get MTS Reception Status	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05039]	Set FlexRay Transceiver Operation Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05040]	Get FlexRay Transceiver Error State	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05041]	Consistent Access to a PDU's Local Memory	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05042]	Switch Configuration in Normal Active Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05044]	Set Absolute Timer	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05045]	Set Relative Timer	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05046]	Enable Absolute Alarms	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05047]	Disable Absolute Alarms	not applicable, this is out of FlexRay

		transceiver driver's scope
[BSW05048]	Acknowledge Absolute Alarms	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05049]	Enable Relative Alarms	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05050]	Disable Relative Alarms	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05051]	Acknowledge Relative Alarms	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05052]	Get Cycle Length in Macroticks	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05053]	Cluster External Clock Synchronization	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05055]	Avoid Timer Interrupts during Shutdown	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05056]	Configuration of the FlexRay Interface at System Configuration Time	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05058]	Configuration of FlexRay Driver at System Configuration Time	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05059]	Transmit/Receive Buffer Configuration	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05060]	Scheduling of Copy Operation into/from FlexRay CC	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05063]	Halt of a FlexRay CC Communication	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05064]	Abstraction of FlexRay CC-specific Implementation	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05065]	Number of FlexRay CCs per Driver	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05066]	L-SDU-Based API	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05067]	Set FlexRay Cluster Offline Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05068]	Set FlexRay Cluster Online Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05069]	Get FlexRay Cluster Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05072]	FlexRay Time Services Access if CC is Out of Sync	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05073]	Usage of ISO 15765-2 and ISO 15765-4 Specifications	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05074]	FlexRay Transport Layer Interfaces	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05075]	Independence of the Network Configuration	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05076]	Multiple Logical FlexRay Transport Layer Channels	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05077]	Unique Identifier of N-SDU	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05078]	Initialization of a FlexRay Cluster	not applicable, cluster abstraction is handled by the FlexRay interface FrIf
[BSW05079]	Transport Connection Properties	not applicable, this is out of FlexRay transceiver driver's scope

[BSW05082]	Acknowledgement without Retry	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05083]	Acknowledgement with Retry	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05084]	PDU Length	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05085]	Segmented 1:n Connections without Flow Control	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05088]	FlexRay Transport Layer Initialization	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05089]	FlexRay Transport Layer Availability	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05090]	Support of Optional ISO 15765-2 Service	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05093]	Transmit Cancellation	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05095]	Bandwidth Control	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05096]	Assignment of Drivers to Controllers	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05097]	Number of FlexRay Drivers per FlexRay Interface	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05101]	Start-up of a FlexRay Cluster	not applicable, cluster abstraction is handled by the FlexRay interface FrIf
[BSW05102]	Halt of a FlexRay Cluster Communication	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05104]	Default Separation Time	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05106]	Buffer Reconfiguration in Normal Active Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05107]	MTS Sending	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05109]	Start-up of a FlexRay CC	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05111]	Get MTS Reception Status	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05113]	Abortion of a FlexRay Cluster Communication	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05114]	Abortion of FlexRay CC Communication	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05115]	Halt of FlexRay CC Communication	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05116]	Initialization of FlexRay CC	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05117]	Sending of Wake-Up Pattern	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05120]	Get FlexRay CC POC Status	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05121]	Get FlexRay CC Sync State	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05123]	Configuration Modifiable by a Flashing Process	not applicable, this is out of FlexRay transceiver driver's scope FrTrcv219
[BSW05124]	Global Transport Layer Properties	not applicable, this is out of FlexRay transceiver driver's scope

[BSW05125]	Interrupt Handling	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05126]	PDU Update/Valid Information	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05129]	Mismatch of Service Call and Connection Properties	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05130]	Transmit Request Queuing	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05131]	Configuration Data for FlexRay Transceiver	FrTrcv225
[BSW05132]	Support for More than One FlexRay Transceiver	FrTrcv226
[BSW05133]	Configuration of Bus Operation Mode after Initialization for Each FlexRay Transceiver	FrTrcv227
[BSW05134]	Initialization Sequence for FlexRay Transceiver Driver	FrTrcv228
[BSW05136]	Configuration "Notification for Wake-up by Bus"	FrTrcv229
[BSW05137]	Initialize the FlexRay Transceiver Driver	FrTrcv230
[BSW05138]	FlexRay Transceiver Driver API shall be Synchronous	FrTrcv231
[BSW05144]	Get FlexRay Transceiver Wake-up Reason	FrTrcv232
[BSW05147]	Notification for Wake-up by Bus	FrTrcv233
[BSW05148]	Support for Wake-up During Sleep Transition	FrTrcv234
[BSW05149]	Support API to Enable/Disable and Clear Wake-up Events	FrTrcv235
[BSW05150]	Safe System Shutdown for FlexRay Transceiver Driver	FrTrcv236
[BSW05151]	FlexRay Transceiver Driver Must Check Transceiver Control	FrTrcv237
[BSW05152]	Transceiver-specific Timing Requirements	FrTrcv238
[BSW05153]	Get FlexRay Controller Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05154]	Set FlexRay Controller Offline Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05155]	Set FlexRay Controller Online Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05156]	Controller External Clock Synchronization	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05157]	Get FlexRay Transceiver Operation Mode	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05158]	Get FlexRay Transceiver Wake-up Reason	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05159]	Enable FlexRay Transceiver Wake-up Indication	FrTrcv245
[BSW05160]	Disable FlexRay Transceiver Wake-up Indication	FrTrcv246
[BSW05161]	Clear FlexRay Transceiver	FrTrcv247

	Wake-up Events	
[BSW05162]	Set Cluster-wide FlexRay Transceiver Operation Mode	not applicable, cluster abstraction is handled by the FlexRay interface FrIf
[BSW05163]	Cluster-wide Enable FlexRay Transceiver Wake-up Indication	not applicable, cluster abstraction is handled by the FlexRay interface FrIf
[BSW05164]	Cluster-wide Disable FlexRay Transceiver Wake-up Indication	not applicable, cluster abstraction is handled by the FlexRay interface FrIf
[BSW05165]	Cluster-wide Clear FlexRay Transceiver Wake-up Events	not applicable, cluster abstraction is handled by the FlexRay interface FrIf
[BSW05166]	Set FlexRay Transceiver Operation Mode	FrTrcv252
[BSW05167]	Get FlexRay Transceiver Operation Mode	FrTrcv253
[BSW05168]	Indicate FlexRay Transceiver Error State	FrTrcv254
[BSW05169]	Avoid Timer Interrupts during Start-up	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05170]	Receive PDU	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05171]	Provide PDU Transmit Confirmation	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05172]	Get NMVector	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05173]	Get NMVector	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05174]	Interrupt Handling	not applicable, this is out of FlexRay transceiver driver's scope
[BSW05175]	Provide Error Information	not applicable, this is out of FlexRay transceiver driver's scope

6.3 Document: AUTOSAR_SWS_ECU_StateManager.pdf

Requirement	Description	Satisfied by
[EcuM2330]	Wakeup sources have to be handled and encapsulated by drivers. The implementation must follow the protocols and requirements presented in this document to ensure a seamless integration into AUTOSAR BSW.	FrTrcv074
EcuM2482:	To support the wakeup and validation protocol, the driver has to fulfill the following requirements:	FrTrcv074
EcuM2483:	The driver has to notify ECU State Manager by invoking the EcuM_SetWakeupEvent service once when a wakeup event is detected. The same service should also be invoked during initialization of the driver if a pending wakeup event is detected during the initialization. Preferably, the invocation is done from a callout or function stub of the caller, to decouple driver modules and ECU State Manager.	FrTrcv262
EcuM2486:	The driver shall provide an explicit service to put the wakeup source to sleep. This service shall put the wakeup source into a energy saving and inert	FrTrcv236

	operation mode and re-arm the wakeup notification mechanism.	
EcuM2485:	If the wakeup source is capable of generating faulty events then the driver or the software stack consuming the driver or another appropriate BSW module shall either provide a validation callout for the wakeup event under validation or directly call the wakeup validation service of the ECU State Manager. If validation is not necessary, then this requirement is not applicable for the according wakeup source.	not applicable, FlexRay transceiver driver has no such needs
EcuM2560:	Some drivers may need re-initialization when the ECU is woken up. This is especially true for drivers with wakeup sources. For re-initialization, a restart block is defined. The restart block is part of the WAKEUP state.	not applicable, FlexRay transceiver driver has no such needs
EcuM2561:	The restart list will typically only contain a subset of drivers. But drivers shall appear in the same order as in the combined list of init block I and init block II (see 10.3 Configurable Parameters, ECUM_DRIVER_RESTART_LIST).	not applicable, this is out of FlexRay transceiver driver's scope
EcuM2562:	Drivers which serve wakeup sources must be re-initialized in the restart block. The driver restart shall re-arm the trigger mechanism of the 'wakeup detected' call-back (see 7.7.4.1 WAKEUP I).	FrTrcv263
EcuM2563:	If hardware is put into a sleep mode during SHUTDOWN then this hardware must be restarted by its driver.	not applicable, this is out of FlexRay transceiver driver's scope
EcuM2745:	The restart list will be invoked in state WAKEUP I (see 7.1.5 WAKEUP State).	not applicable, this is out of FlexRay transceiver driver's scope
EcuM2627:	During the initialization, the driver must detect if the CAN transceiver switched on the power supply because of a passive wakeup. The driver shall notify this. In this case, the system designer is responsible to clear all previous wakeup sources and to set the CAN transceiver as the wakeup source by using the EcuM_SetWakeupEvent. This requirement only applies for systems where the CAN transceiver controls the power supply to implement the SLEEP state.	not applicable, this is out of FlexRay transceiver driver's scope
EcuM2569:	Pending events are validated with a call to EcuM_ValidateWakeupEvent. This call must be placed in the driver or the consuming stack on top of the driver (e.g. the handler). The best place to put this depends on hardware and software design. See also 7.8.5 Requirements for Drivers with Wakeup Sources.	not applicable, this is out of FlexRay transceiver driver's scope

6.4 Document: AUTOSAR_SWS_ComStackTypes.doc

Requirement	Description	Satisfied by
COMTYPE021:	General Codes	FrTrcv264
COMTYPE022:	The Communication System dependent Return codes shall be named as follows: <BUS>TRCV_E_FR_<Error Code Name>. Error Code Name: self explaining name of error return code.	FrTrcv265

6.5 Document: AUTOSAR_SWS_BSW_Scheduler.pdf

Requirement	Description	Satisfied by
INTEGR092:	Each BSW module implementation <ModulePrefix>.c shall include its respective header file SchM_<ModulePrefix>.h.	FrTrcv266

6.6 Document: AUTOSAR_SWS_MemoryMapping.pdf

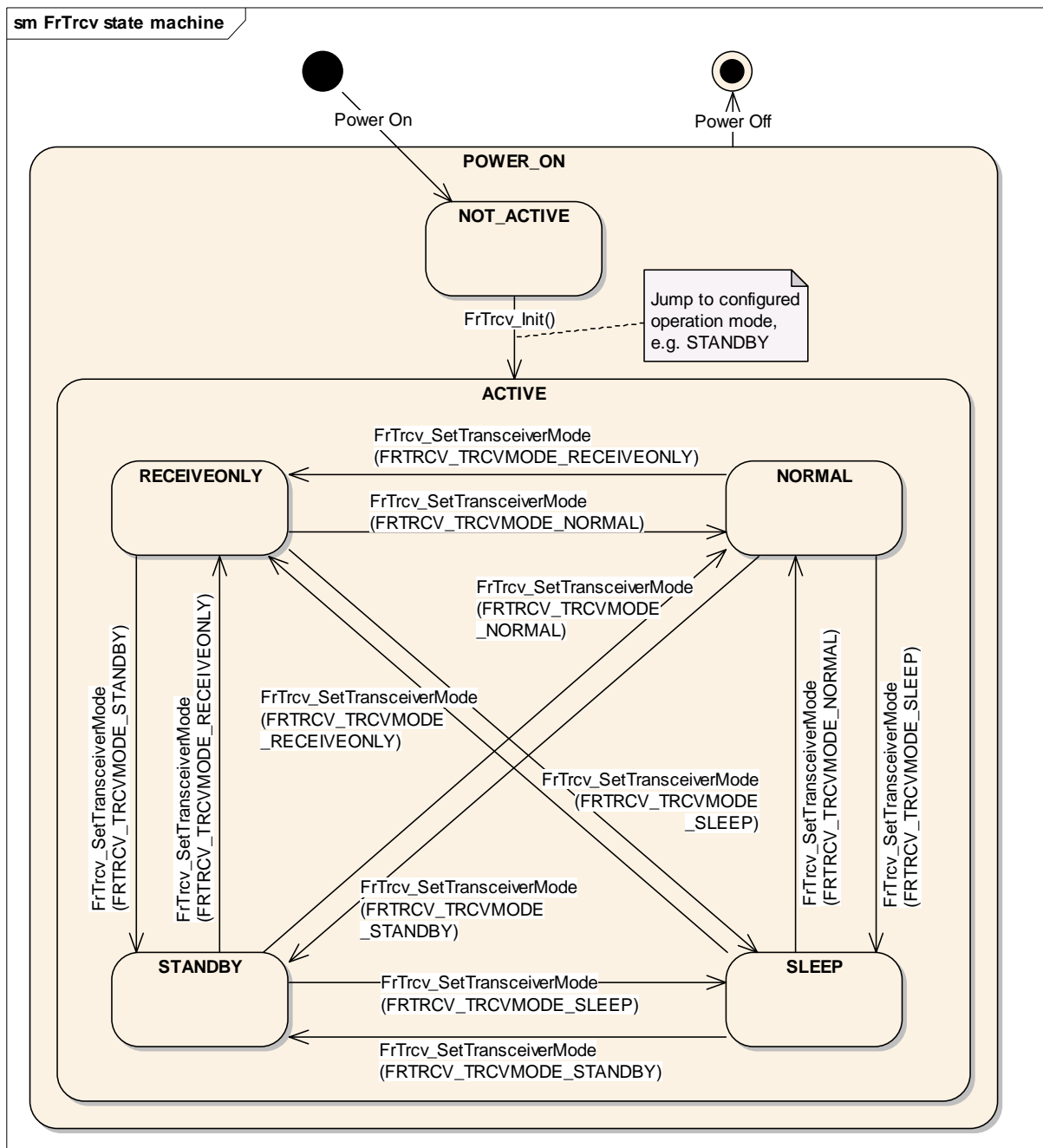
Requirement	Description	Satisfied by
MEMMAP003:	Each AUTOSAR software module shall wrap declaration and definition of code, variables and constants using the following mechanism: <ol style="list-style-type: none"> 1. Definition of start symbol for module memory section 2. Inclusion of MemMap.h 3. Declaration/definition of code, variables or constants belonging to the specified section 4. Definition of stop symbol for module memory section 5. Inclusion of MemMap.h The inclusion of MemMap.h within the code is a MISRA violation. As neither executable code nor symbols are included (only pragmas) this violation is an approved exception without side effects.	FrTrcv267

7 Functional specification

7.1 AUTOSAR FlexRay Transceiver Operation Mode Model

The FlexRay Transceiver operation modes are described in the state diagram below. The main idea behind this diagram is to support a lot of up to now available FlexRay Transceivers in a common model view. Depending on the transceiver device, the model may have one or two states more than necessary for a given device but this will clearly decouple the ComM and EcuM from the used hardware.

FrTrcv227:



State	Description
POWER_ON	ECU is fully powered.
NOT_ACTIVE	State of FlexRay transceiver hardware depends on ECU hardware and on SPAL driver configuration. FlexRay Transceiver Driver is not initialized and therefore not active.
ACTIVE	The function FrTrcv_TrcevInit() was called. This moves FlexRay Transceiver Driver to the active state selected by configuration.
NORMAL	Full bus communication is possible depending on ComM state. If FlexRay transceiver hardware controls ECU power supply, ECU is fully powered. The FlexRay Transceiver Driver detects no further wake up information.
STANDBY	No communication is possible. ECU is still powered if FlexRay transceiver hardware controls ECU power supply. A wake up by bus or by a local wake up event is possible.
SLEEP	No communication is possible. ECU may be unpowered depending on responsibility to handle power supply. A wake up by bus or by a local wake up event is possible.
RECEIVEONLY	Similar to NORMAL, but only reception is possible.

If a FlexRay transceiver driver covers more than one FlexRay node, all nodes are either in state NOT_ACTIVE or in state ACTIVE. In state ACTIVE each node may be in a different sub state.

If a state is optional and NOT supported by the transceiver and ECU hardware (e.g. SLEEP or RECEIVEONLY), the transceiver driver substitutes an equivalent state (i.e. STANDBY instead of SLEEP; and NORMAL instead of RECEIVEONLY) and returns the real state by the FrTrcv_GetTransceiverMode() function.

7.2 FlexRay transceiver hardware operation modes

The FlexRay transceiver hardware may support more mode transitions than shown in the state diagram above. The dependencies and the recommended implementation are explained in this chapter.

7.2.1 Temporary “Go-To-Sleep” Mode

The often “Go-To-Sleep” called mode is a temporary mode when switching from Normal to Sleep. The driver shall encapsulate such a temporary mode within one of the “official” state transitions. The higher layers will always use the allowed call sequences when switching to standby and sleep mode.

I.e. a FlexRay bus transceiver driver for the TJA1080 will implement the transition from Standby to Sleep via a temporary transition in “Go-To-Sleep” and then into

Sleep. The transition from Normal to Standby is not affected and will be performed directly.

7.2.2 “Active Star” Mode

If a transceiver supports active star mode, the driver must not incorrectly assume it is in node mode. This mode is transparent to the transceiver driver.

7.3 Error classification

FrTrcv107: Values for production code Event Ids are assigned externally by the configuration of the Dem. They are published in the file Dem_IntErrId.h and included via Dem.h.

FrTrcv050: Development error values are of type uint8.

FrTrcv041: The naming of errors has to be compliant to BSW000327.

FrTrcv085:

Type or error	Relevance	Related error code	Value [hex]
API service called with wrong parameter	Development	FRTRCV_E_FR_INVALID_TRCVIDX FlexRay transceiver index out of range	0x01
API Service used without initialization	Development	FRTRCV_E_FR_UNINIT	0x10
API service called in wrong transceiver operation mode	Development	FRTRCV_E_FR_TRCV_NOT_STANDBY	0x11
		FRTRCV_E_FR_TRCV_NOT_NORMAL	0x12
		FRTRCV_E_FR_TRCV_NOT_SLEEP	0x13
		FRTRCV_E_FR_TRCV_NOT_RECEIVEONLY	0x14
No error, all ok	Production	FRTRCV_E_FR_TRCV_NONE	*
No/incorrect communication to transceiver.	Production	FRTRCV_E_FR_NO_TRCV_CONTROL	*

* Assignment is done in a header file of module Dem.

7.4 Error detection

FrTrcv061: The detection of development errors is configurable (*ON/ OFF*) at pre-compile time. The switch `FRTRCV_DEV_ERROR_DETECT` shall activate or deactivate the detection of all development errors.

[FrTrcv048]: FrTrcv037: If the *FRTRCV_DEV_ERROR_DETECT* switch is enabled API parameter checking is enabled. The detailed description of the detected errors can be found in chapter 7.3

[FrTrcv058]: FrTrcv119: The detection of production code errors cannot be switched off.

FrTrcv237: The FlexRay Transceiver Driver shall check the control communication to the transceiver and the reaction of the transceiver for correctness; in case of faults, a production error (*FRTRCV_E_FR_NO_TRCV_CONTROL*) is raised via Dem.

Depending on the supported transceiver device, the driver shall check the correctness of the executed control communication and the operation mode of the transceiver in order to detect defective or faulty transceiver hardware and/or corrupted SPI communication.

The check shall only be applied to errors within the transceiver or the transceiver control communication (ports or SPI), i.e. errors caused by malfunction of the μ C, SW or a defect transceiver device.

"Errors" caused by the "outer world" (e.g. disturbed communication nodes or ground offsets) are not in the scope of this API.

7.5 Error notification

FrTrcv051: Detected development errors will be reported to the error hook of the Development Error Tracer (DET) if the pre-processor switch *FRTRCV_DEV_ERROR_DETECT* is set (see chapter 10).

FrTrcv052: Production errors shall be reported to Diagnostic Event Manager.

FrTrcv254: Error cases are reported to the Dem.

In addition, the state of the FlexRay transceiver is periodically reported to Dem by the *FrTrcv_MainFunction* to ease healing of errors.

FrTrcv265: The Communication System dependent Return codes shall be named as follows: *<BUS>TRCV_E_FR_<Error Code Name>*. Error Code Name: self explaining name of error return code.

7.5.1 BusTrcvErrorType

Type:	uint8	
Range:	0x5B – 0x78	Error notification: Error notification codes specific for the communication system FlexRay..
	> 0x78	Currently values in this range are invalid. In future it might be possible that further return codes are specified for other communication systems.
	COMTYPE020: Variables of this type shall be used to return the bus status evaluated by a transceiver.	--
	--	--
Description:	v	

7.6 Preconditions for driver initialization

The FlexRay bus transceiver driver has to use drivers for SPI, Dio and or Icu to control the FlexRay bus transceiver hardware. Thus these drivers must be available and ready to operate before the FlexRay bus transceiver driver is initialized.

FrTrcv228:The FlexRay bus transceiver driver may have timing requirements for the initialization sequence and the access to the transceiver device, which must be fulfilled by these used underlying drivers.

FrTrcv238:The timing requirements might be that

1. The call of the FlexRay bus transceiver driver initialization has to be performed very early after power up to be able to read all necessary information out of the transceiver device in time for all other users within the ECU.
2. The runtime of the used underlying services is very short and synchronous to enable the driver to keep its own timing requirements limited by the hardware device used ([FrTrcv231](#)).
3. The runtime of the driver may be enlarged in order to support setup and hold times of the hardware devices in low power states, e.g. sleep.

7.7 Instance concept

FrTrcv111:An ECU may contain multiple FlexRay transceivers. These transceivers can be of different types. Each transceiver type is handled by a dedicated FlexRay Transceiver Driver.

For your convenience, assume that any API call is not executed directly but is resolved by configuration to a zero based index into a function pointer table (per driver).

This issue is already resolved for Flexray Interface FrIf and the FlexRay communication controller.

FrTrcv226: Multiple FlexRay transceivers of the same type are handled by a single FlexRay transceiver driver; there is no need for multiple instances of this single FlexRay transceiver driver.

FrTrcv supports exactly one transceiver per CC and channel (i.e., it is not permitted that two CCs of one ECU share one FlexRay transceiver)!

8 API specification

8.1 Imported types

8.1.1 Standard types

For AUTOSAR standard data types the header file Std_Types.h is included. Used type is Std_VersionInfoType.

8.1.2 EcuM types

EcuM type EcuM_WakeupSourceType is the only EcuM type used.

8.1.3 ComStack types

The communication stack types header file defines communication types based on the platform types [PltfTypes] (Platform_Types.h) and Compiler (Compiler.h) header file [CompTypes]. To prevent multiple includes of header files, the communication stack header file includes the standard types header file [StdTypes] which already includes both other files. Used type is BusTrcvErrorType.

8.2 Type definitions

FrTrcv030: FrTrcv045: FrTrcv069: FrTrcv071:

The following two type definitions shall be kept in a file named Fr_GeneralTypes.h and be protected by a FR_GENERAL_TYPES define in order:

- to be shared between different FlexRay Transceiver Drivers
- to be included into the FrIf

If different FlexRay Transceiver Drivers are used, only one instance of this file has to be included in the source tree.

8.2.1 FrTrcv_TrcvModeType

FrTrcv252: FrTrcv048:

Type:	enumeration	
	FRTRCV_TRCVMODE_NORMAL	Transceiver is in state NORMAL
	FRTRCV_TRCVMODE_STANDBY	Transceiver is in state STANDBY
	FRTRCV_TRCVMODE_SLEEP	Transceiver is in state SLEEP
	FRTRCV_TRCVMODE_RECEIVEONLY	Transceiver is in state RECEIVEONLY
Description:	See also 8.3.3 for details.	

8.2.2 FrTrcv_TrcvWUReasonType

FrTrcv074:

Type:	enumeration	
	FRTRCV_WU_NOT_SUPPORTED	The transceiver does not support any information for the wake up reason.
	FRTRCV_WU_BY_BUS	The transceiver has detected that the bus has caused the wake up of the ECU.
	FRTRCV_WU_INTERNALLY	The transceiver has detected that the bus has woken up by the ECU via FrTrcv_SetTransceiverMode() API call
	FRTRCV_WU_RESET	The transceiver has detected that the "wake up" is due to an ECU reset.
	FRTRCV_WU_POWER_ON	The transceiver has detected that the "wake up" is due to an ECU reset after power on.
Description:	This type shall be used to specify the wake up reason detected by the FR transceiver in detail. See 8.3.4, too.	

8.3 Function definitions

FrTrcv043: FrTrcv089: FrTrcv090: FrTrcv091: FrTrcv092: FrTrcv093: FrTrcv094: FrTrcv104:

8.3.1 FrTrcv_TrcvInit

Service name:	FrTrcv_TrcvInit	
Syntax:	void FrTrcv_TrcvInit (uint8 FrTrcv_TrcvIdx)	
Service ID [hex]:	0x0	
Behavior:	Synchronous	
Reentrancy:	non reentrant	
Parameters (in):	FrTrcv_TrcvIdx	This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.
Parameters (out):	none	-
Return value:	none	FrTrcv065:
Description:	FrTrcv008: FrTrcv230: FrTrcv228 FrTrcv112: FrTrcv263: After this call the FlexRay transceiver driver and hardware are in the state FRTRCV_INIT_STATE defined by configuration, i.e. either in state FRTRCV_TRCVMODE_NORMAL, FRTRCV_TRCVMODE_STANDBY, FRTRCV_TRCVMODE_RECEIVEONLY or FRTRCV_TRCVMODE_SLEEP. In the time span between power up and call of FrTrcv_TrcvInit FlexRay transceiver hardware maybe in a different state. This depends on hardware and SPAL driver configuration. The transceiver driver has to check within the init routine whether there has been	

	<p>a wake up due to transceiver activity and if this is the case report this to the EcuM via EcuM_SetWakeupEvent(event). Hint: Every FlexRay conformant transceiver is able to detect wakeup! The driver has to notify ECU State Manager by invoking the EcuM_SetWakeupEvent service once when a wakeup event is detected. The same service should also be invoked during initialization of the driver if a pending wakeup event is detected during the initialization. This is done indirectly via FrTrcv_Cbk_WakeupByTransceiver. Drivers which serve wakeup sources must be re-initialized in the restart block. The driver restart shall re-arm the trigger mechanism of the 'wakeup detected' call-back. Actually, the first code to be executed may be an ISR, e.g. a wakeup ISR. However, this is specific to hardware and/or driver implementation</p> <p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range</p> <p><u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access</p>
Caveats:	<p>The initialization sequence after reset (e.g. power up) is a critical phase for the FlexRay transceiver driver. The driver will use SPAL functionality (SPI or Dio) to access the transceiver hardware. Therefore all necessary BSW drivers must be initialized and usable before.</p> <p>Enable the detection of a power on reset due to transceiver activity. If this detection is not done the ECU will wake up and directly go to sleep again.</p>
Configuration:	--

8.3.2 FrTrcv_SetTransceiverMode

Service name:	FrTrcv_SetTransceiverMode	
Syntax:	BusTrcvErrorType FrTrcv_SetTransceiverMode (uint8 FrTrcv_TrcvIdx, FrTrcv_TrcvModeType FrTrcv_TrcvMode)	
Service ID [hex]:	0x1	
Behavior:	Synchronous	
Reentrancy:	non reentrant	
Parameters (in):	FrTrcv_TrcvIdx	This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.
	FrTrcv_TrcvMode	Selects the state the transceiver will transit to (transitions to optional states may fail)
Parameters (out):	none	
Return value:	BUSTRCV_E_OK	BUSTRCV_E_OK will be returned if the transceiver state has been changed to the requested mode. FrTrcv264:
	BUSTRCV_E_ERROR	BUSTRCV_E_ERROR will be returned if the transceiver state change has failed or the parameter is out of the allowed range. The previous state has not been changed.
Description:	FrTrcv252 FrTrcv064: The internal state of the FlexRay transceiver driver identified by FrTrcv_TrcvIdx is	

	<p>switched to the state indicated by FrTrcv_TrcvMode (one of FRTRCV_TRCVMODE_NORMAL, FRTRCV_TRCVMODE_STANDBY, FRTRCV_TRCVMODE_SLEEP, FRTRCV_TRCVMODE_RECEIVEONLY)</p> <p>In case an illegal transition is requested, it will fail (BUSTRCV_E_ERROR) and the original state is maintained.</p> <p>If a state is optional and NOT supported by the transceiver and ECU hardware (e.g. FrTRCV_TRCVMODE_SLEEP or FrTRCV_TRCVMODE_RECEIVEONLY), the transceiver driver substitutes an equivalent state (i.e. FrTRCV_TRCVMODE_STANDBY instead of FrTRCV_TRCVMODE_SLEEP; and FrTRCV_TRCVMODE_NORMAL instead of FrTRCV_TRCVMODE_RECEIVEONLY).</p> <p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range FRTRCV_E_FR_TRCV_NOT_STANDBY: Transition to FRTRCV_TRCVMODE_STANDBY failed FRTRCV_E_FR_TRCV_NOT_NORMAL: Transition to FRTRCV_TRCVMODE_NORMAL failed FRTRCV_E_FR_TRCV_NOT_SLEEP: Transition to FRTRCV_TRCVMODE_SLEEP failed FRTRCV_E_FR_TRCV_NOT_RECEIVEONLY: Transition to FRTRCV_TRCVMODE_RECEIVEONLY failed FRTRCV_E_FR_UNINIT: not yet initialized</p> <p><u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access</p>
Caveats:	<p>ComM is responsible not to call any of these functions as long as a previous call is not finished.</p> <p>The application controller (host) has to ensure that the BD enters BD_Normal or BD_Receiveonly mode, before the CC enters one of its states where the CC starts to listen to the channel.</p> <p>In case the host commands BD_Normal and the BD does not enter BD_Normal, an error is indicated at the BD host interface. In this case the host shall force the CC to step back to a non listening state. The reason for this is that as long as the BD is not in BD_Normal or BD_Receiveonly mode, no information about the status of the channel is available via the signals RxD and RxEN.</p>
Configuration:	The number of supported transceivers is set up in the configuration phase.

8.3.3 FrTrcv_GetTransceiverMode

Service name:	FrTrcv_GetTransceiverMode	
Syntax:	BusTrcvErrorType FrTrcv_GetTransceiverMode (uint8 FrTrcv_Trcvidx, FrTrcv_TrcvModeType * FrTrcv_TrcvModePtr)	
Service ID [hex]:	0x5	
Behavior:	Synchronous	
Reentrancy:	Re-entrant	
Parameters (in):	FrTrcv_Trcvidx	This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.

Parameters (out):	FrTrcv_TrvcModePtr	FrTrcv076: Pointer to structure of current transceiver state; the FlexRay transceiver driver will write the transceiver state information there.
Return value:	BUSTRCV_E_OK	BUSTRCV_E_OK will be returned if the transceiver state has been provided
	BUSTRCV_E_ERROR	BUSTRCV_E_ERROR will be returned if the transceiver state is illegal or the parameter is out of range. Output parameters remain unchanged.
Description:	<p>FrTrcv253: The return value reflects always the actual state of the transceiver. Output data is valid until the next call of FrTrcv_GetTransceiverMode() OR until the next call of FrTrcv_TrvcInit()</p> <p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range FRTRCV_E_FR_UNINIT: not yet initialized</p> <p><u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access</p>	
Caveats:	See FrTrcv_TrvcInit for the provided state after the FlexRay transceiver driver initialization till the first operation mode change request. If a transceiver supports active star mode, do NOT incorrectly assume it is in node mode.	
Configuration:	The number of supported nodes and the transceiver type for each node is statically set in the configuration phase.	

8.3.4 FrTrcv_GetTransceiverWUReason

Service name:	FrTrcv_GetTransceiverWUReason	
Syntax:	<pre>BusTrcvErrorType FrTrcv_GetTransceiverWUReason (uint8 FrTrcv_TrvcIdx, FrTrcv_TrvcWUReasonType * FrTrcv_TrvcWUReasonPtr)</pre>	
Service ID [hex]:	0x6	
Behavior:	Synchronous	
Reentrancy:	reentrant	
Parameters (in):	FrTrcv_TrvcIdx	This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.
Parameters (out):	FrTrcv_TrvcWUReasonPtr	Pointer to structure of least recent wakeup source, the FlexRay transceiver driver will write the transceiver state information there
Return value:	BUSTRCV_E_OK	BUSTRCV_E_OK will be returned if the transceiver wake up source has been provided
	BUSTRCV_E_ERROR	BUSTRCV_E_ERROR will be returned if the transceiver state is illegal or the parameter is out of range. Output parameters remain unchanged.
Description:	<p>FrTrcv232: This API returns the reason for the wake up that the FlexRay transceiver has detected. The ability to detect and differentiate the possible wake up reasons depends strongly on the FlexRay transceiver hardware. Output data is valid until the next call of FrTrcv_GetTransceiverWUReason () OR until the next call of FrTrcv_TrvcInit()</p>	

	<p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range FRTRCV_E_FR_UNINIT: not yet initialized</p> <p><u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access</p>
Caveats:	<p>Please be aware, that if more than one bus is available each bus may report a different wake up reason. E.g. if an ECU has FlexRay, a wake up by FlexRay may occur and the incoming data may cause an internal wake up for another FlexRay bus.</p> <p>The FlexRay bus transceiver driver has a “per bus” view and does not vote the more important reason or sequence internally. The same may be true if e.g. one transceiver controls the power supply and the other is just powered or un-powered. Then one may be able to return “FRTRCV_WU_POWER_ON” whereas the other may state e.g. “FRTRCV_WU_RESET”.</p> <p>It is up to the EcuM and the ComM, to decide what shall happen with that wake up information.</p>
Configuration:	The number of supported nodes and the transceiver type for each node is statically set in the configuration phase.

8.3.5 FrTrcv_GetVersionInfo

Service name:	FrTrcv_GetVersionInfo
Syntax:	<pre>void FrTrcv_GetVersionInfo (Std_VersionInfoType *versioninfo)</pre>
Service ID [hex]:	0x7
Behavior:	Synchronous
Reentrancy:	reentrant
Parameters (in):	-
Parameters (out):	versioninfo Pointer to structure with version information
Return value:	- -
Description:	FrTrcv001: FrTrcv105: FrTrcv109: This API returns the version information of this module- <u>Development errors:</u> none
Caveats:	This function returns the version of the FrTrcv transceiver <u>driver</u> , <u>NOT</u> the version of the FlexRay transceiver <u>hardware</u> .
Configuration:	Depending on configuration parameter FRTRCV_GET_VERSION_INFO this function shall be present in compiled code or not. See chapter 10.2.3 for more details.

8.3.6 FrTrcv_DisableTransceiverWakeup

Service name:	FrTrcv_DisableTransceiverWakeup
Syntax:	<pre>BusTrcvErrorType FrTrcv_DisableTransceiverWakeup (uint8 FrTrcv_TrcvIdx</pre>

)
Service ID [hex]:	0xA
Behavior:	Synchronous
Reentrancy:	non-reentrant
Parameters (in):	FrTrcv_TrcvIdx This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.
Parameters (out):	none -
Return value:	BUSTRCV_E_OK BUSTRCV_E_OK will be returned if the transceiver wake up has been disabled
	BUSTRCV_E_ERRO R BUSTRCV_E_ERROR will be returned if the transceiver state is illegal or the parameter is out of range. Wake up state remains unchanged.
Description:	FrTrcv246: FrTrcv235: This API disables the notification for wake up events on the addressed bus. After the call of the API, no wake up notification to higher layers is executed anymore. If necessary by the transceiver device and the underlying communication, the driver has to detect the wake up event nevertheless and stores it internally to raise the event when the wake up notification is enabled again. <u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range FRTRCV_E_FR_UNINIT: not yet initialized <u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access
Caveats:	It is very important not to lose wake up events during the disabled period. In order to keep the transceiver driver simple, this API should refer to all kinds of wake up. Further differentiation of wakeup sources requires knowledge available only to higher software layers and is out of scope of the transceiver driver.
Configuration:	The number of supported nodes is statically set in the configuration phase.

8.3.7 FrTrcv_EnableTransceiverWakeup

Service name:	FrTrcv_EnableTransceiverWakeup
Syntax:	BusTrcvErrorType FrTrcv_EnableTransceiverWakeup (uint8 FrTrcv_TrcvIdx)
Service ID [hex]:	0xB
Behavior:	Synchronous
Reentrancy:	non-reentrant
Parameters (in):	FrTrcv_TrcvIdx This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.
Parameters (out):	none -
Return value:	BUSTRCV_E_OK BUSTRCV_E_OK will be returned if the transceiver wake up has been enabled
	BUSTRCV_E_ERRO R BUSTRCV_E_ERROR will be returned if the transceiver state is illegal or the parameter is out of range. Wake up state remains unchanged.

Description:	<p>FrTrcv245:</p> <p>This API enables the notification for wake up events on the addressed bus. After the call of the API the wake up notification to higher layers is performed when a wake up event is detected.</p> <p>If the driver has a stored wake up event pending, the notification is executed within the API call or immediately after (depending on the implementation).</p> <p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range FRTRCV_E_FR_UNINIT: not yet initialized</p> <p><u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access</p>
Caveats:	<p>The implementation may be e.g. enabling the interrupt source for the wake up. If the interrupt is level triggered a pending interrupt is automatically stored and raised after enabling the notification again.</p> <p>It is very important not to loose wake up events during the disabled period. In order to keep the transceiver driver simple, this API should refer to all kinds of wake up. Further differentiation of wakeup sources requires knowledge available only to higher software layers and is out of scope of the transceiver driver.</p>
Configuration:	<p>The number of supported nodes is statically set in the configuration phase.</p>

8.3.8 FrTrcv_ClearTransceiverWakeup

Service name:	FrTrcv_ClearTransceiverWakeup	
Syntax:	BusTrcvErrorType FrTrcv_ClearTransceiverWakeup (uint8 FrTrcv_TrcvIdx)	
Service ID [hex]:	0xC	
Behavior:	Synchronous	
Reentrancy:	non-reentrant	
Parameters (in):	FrTrcv_TrcvIdx	This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied.
Parameters (out):	none	-
Return value:	BUSTRCV_E_OK	BUSTRCV_E_OK will be returned if the transceiver wakeup source has been cleared
	BUSTRCV_E_ERROR	BUSTRCV_E_ERROR will be returned if the transceiver could not clear its wakeup state or the parameter is out of range. Wake up state remains unchanged.
Description:	<p>FrTrcv247:</p> <p>This API clears a pending wake up event. The API is intended to be used when the wake up notification is disabled to clear all pending wake up events under control of the higher layer.</p> <p>The expectation to the implementation is that all "old" and currently no more valid wake up events are cleared. If a "real" wake up is pending, the clearance shall be performed, too, but the "real" wake up will cause an immediate occurrence of a new wake up event.</p> <p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range</p>	

	FRTRCV_E_FR_UNINIT: not yet initialized <u>Production errors:</u> FRTRCV_E_FR_NO_TRCV_CONTROL: incorrect transceiver access
Caveats:	The implementation may be e.g. clearing a pending, level triggered wake up interrupt. If the interrupt is “old”, e.g. from the last communication cycle, the interrupt is cleared and ready for the next trigger. If communication is starting just in that moment, the level triggered interrupt fires immediately again and causes a new wake up after enabling the notification again. In order to keep the transceiver driver simple, this API should refer to all kinds of wake up. Further differentiation of wakeup sources requires knowledge available only to higher software layers and is out of scope of the transceiver driver.
Configuration:	The number of supported nodes is statically set in the configuration phase.

8.4 Scheduled functions

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

8.4.1 FrTrcv_MainFunction

Service name:	FrTrcv_MainFunction
Service ID [hex]:	0xD
Description:	FrTrcv020: FrTrcv072: FrTrcv075: FrTrcv126: The FlexRay bus transceiver driver may have cyclic jobs like polling for wake up events (if configured). This cyclic called function has to scan all busses in STANDBY and SLEEP for wake up events and has to perform these events by calling appropriate call-back functions. FrTrcv122: According [BSW00424] BSW module main processing functions are only allowed to be allocated to basic tasks (see extended and basic task according to AUTOSAR OS classification). <u>Development errors:</u> FRTRCV_E_FR_UNINIT: Report this development error if any of the supported transceivers is not initialised. <u>Production errors:</u> FRTRCV_E_FR_TRCV_NONE no transceiver error, all transceivers are ok.
Timing:	fixed cyclic
Pre condition:	-
Configuration:	Configuration parameter “FRTRCV_MAIN_FUNCTION_CYCLE_TIME” specify cycle time. For a cycle time of 0 this function need not be present in compiled code. See chapter 10.2.3 for more details.

8.5 Call-back notifications

This is a list of functions provided for lower layer modules. The function prototypes of the call-back functions shall be provided in the file FrTrcv_Cbk.h

8.5.1 FrTrcv_Cbk_WakeupByTransceiver

Service name:	FrTrcv_Cbk_WakeupByTransceiver	
Syntax:	<pre>void FrTrcv_Cbk_WakeupByTransceiver (uint8 FrTrcv_TrcvIdx)</pre>	
Service ID [hex]:	0xE	
Behavior:	Synchronous	
Reentrancy:	reentrant	
Parameters (in):	FrTrcv_TrcvIdx	This zero based index identifies the transceiver within the context of the transceiver driver to which the API call has to be applied (FrTrcv067:).
Parameters (out):	none	
Return value:	-	FrTrcv066:
Description:	<p>FrTrcv233: FrTrcv086: FrTrcv262:</p> <p>The underlying SPAL Icu driver calls this API via the FlexRay Interface (Frlf) in case a wake up interrupt is detected or periodically by a polling process. The API calls the appropriate function of EcuM (EcuM_SetWakeupEvent) with the parameter value ECUM_WKSOURCE_FRTRCV_FR of EcuM_WakeupSourceType <u>only</u> in case a valid wakeup originated from the transceiver identified by FrTrcv_TrcvIdx.</p> <p>Thus shared interrupts are easily de-multiplexed: Drivers which did not trigger the interrupt just return doing nothing.</p> <p><u>Development errors:</u> FRTRCV_E_FR_INVALID_TRCVIDX: FlexRay transceiver index out of range FRTRCV_E_FR_UNINIT: not yet initialized</p>	
Caveats:	<p>Wake up by bus is always asynchronous to the transition to sleep and standby. In worst case wake up occurs during transition to sleep. In such a case the driver shall create a wake up by bus notification immediately after the API calls to enter standby or sleep has finished. The EcuM must be able to handle the wake up event immediately after requesting the standby or sleep mode.</p> <p>Drivers which serve wakeup sources must be re-initialized in the restart block. The driver restart shall re-arm the trigger mechanism of the 'wakeup detected' call-back.</p>	
Configuration:	<p>If no wake up by bus is used this function need not be present in compiled code. See configuration parameters FRTRCV_WAKEUP_BY_NODE_USED in chapter 8.6.2 for more details.</p>	
Call Context	<p>The call context of this API is expected to be within the ISR handler of the underlying driver. This has to be documented (BSW00333). It may also be called by a polling process in sleep mode or by FrTrcv_MainFunction().</p>	

8.6 Expected Interfaces

In this chapter all interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

This chapter defines all interfaces which are required to fulfill the core functionality of the module.

<i>Req Id</i>	<i>API</i>	<i>Mod.</i>	<i>Description</i>	<i>Configuration</i>
FrTrcv052	Dem_ReportErrorStatus	Dem	Production errors notification.	-

8.6.2 Optional Interfaces

FrTrcv019: This chapter defines all interfaces which are required to fulfil an optional functionality of the module.

<i>API function</i>	<i>Module</i>	<i>Description</i>	<i>Configuration</i>
EcuM_SetWake up Event	EcuM	FrTrcv084: Wake up by bus notification. Called in operation modes sleep and standby. Not called if call to FrTrcv_SetTransceiverMode() has caused wake up.	Container FlexRayTransceiverNodes, parameters FRTRCV_WAKEUP_BY_NODE_USED and FRTRCV_WAKEUP_POLLING
Det_ReportError	Det	FrTrcv051 Development error notification.	Container FlexRayTransceiverDriverBasic, parameter FRTRCV_DEVELOPMENT_ERROR.
Spi_WriteIB	Spi ¹	Write data to an IB of a SPI channel.	Container FlexRayTransceiverSPIChannel gives reference for appropriate SPI container with all necessary information.
Spi_ReadIB		Read to from an IB of a SPI channel.	
Spi_SetupEB		Configure EB for a SPI channel	
Spi_GetStatus		Returns current state of SPI driver	No configuration necessary.
Spi_SyncTransmit		Starts synchronous transmission on a SPI connection	Sequence name given by container FlexRayTransceiver SPISequences.
Dio_ReadChannel	Dio ²	Read access to one dio channel.	Appropriate identifier names given by instances of container FlexRayTransceiverDIOAccess.
Dio_WriteChannel		Write access to one dio channel.	
Dio_ReadPort		Read access to one dio port.	
Dio_WritePort		Write access to one dio port.	
Dio_ReadChannelGroup		Read access to one dio channel group.	
Dio_WriteChannelGroup		Write access to one dio channel group.	

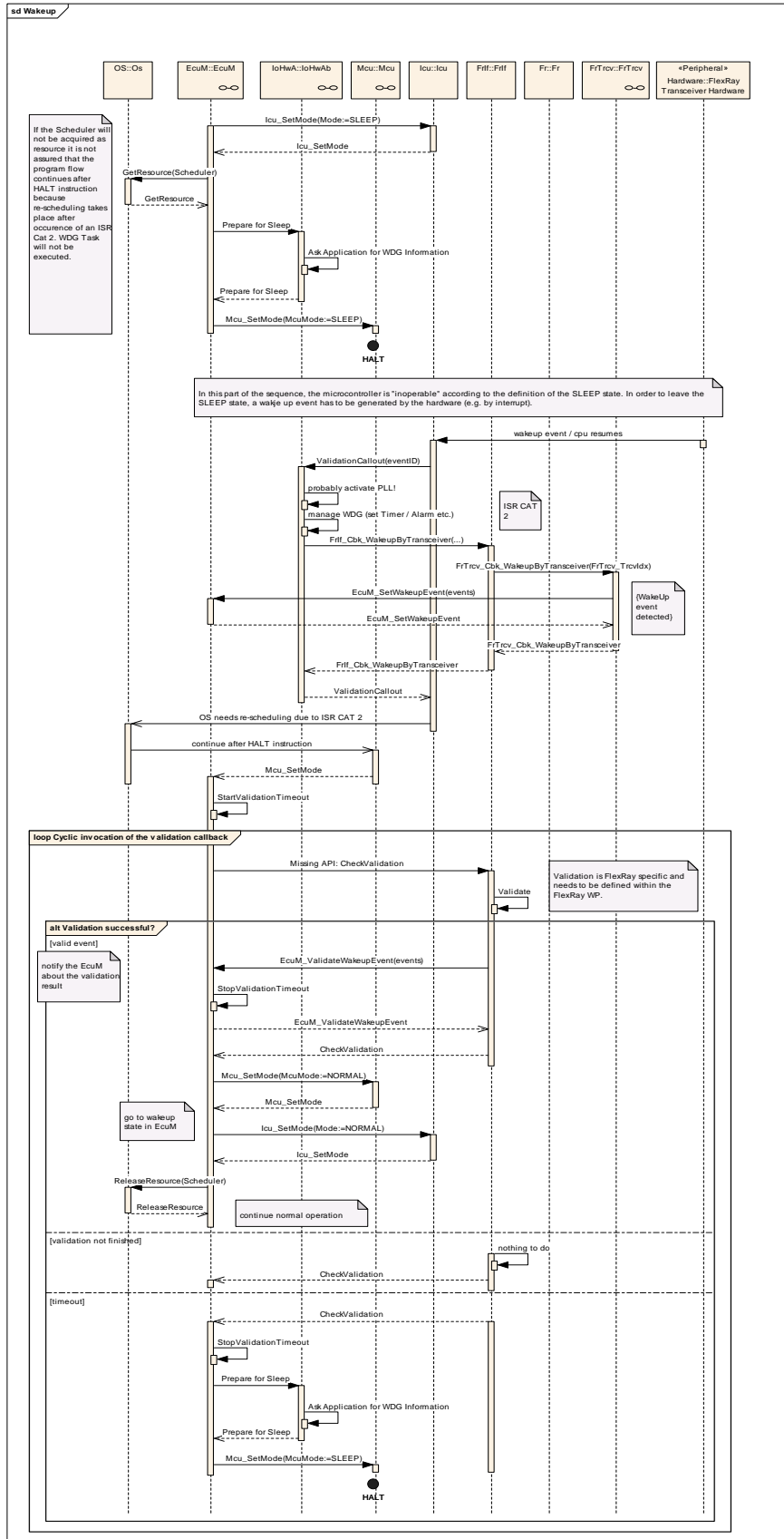
1. Interfaces of SPI module are used if there are instances of the container FlexRayTransceiverSPISequences.
2. Interfaces of DIO module are used if there are instances of the container FlexRayTransceiverDioAccess.
- 3.

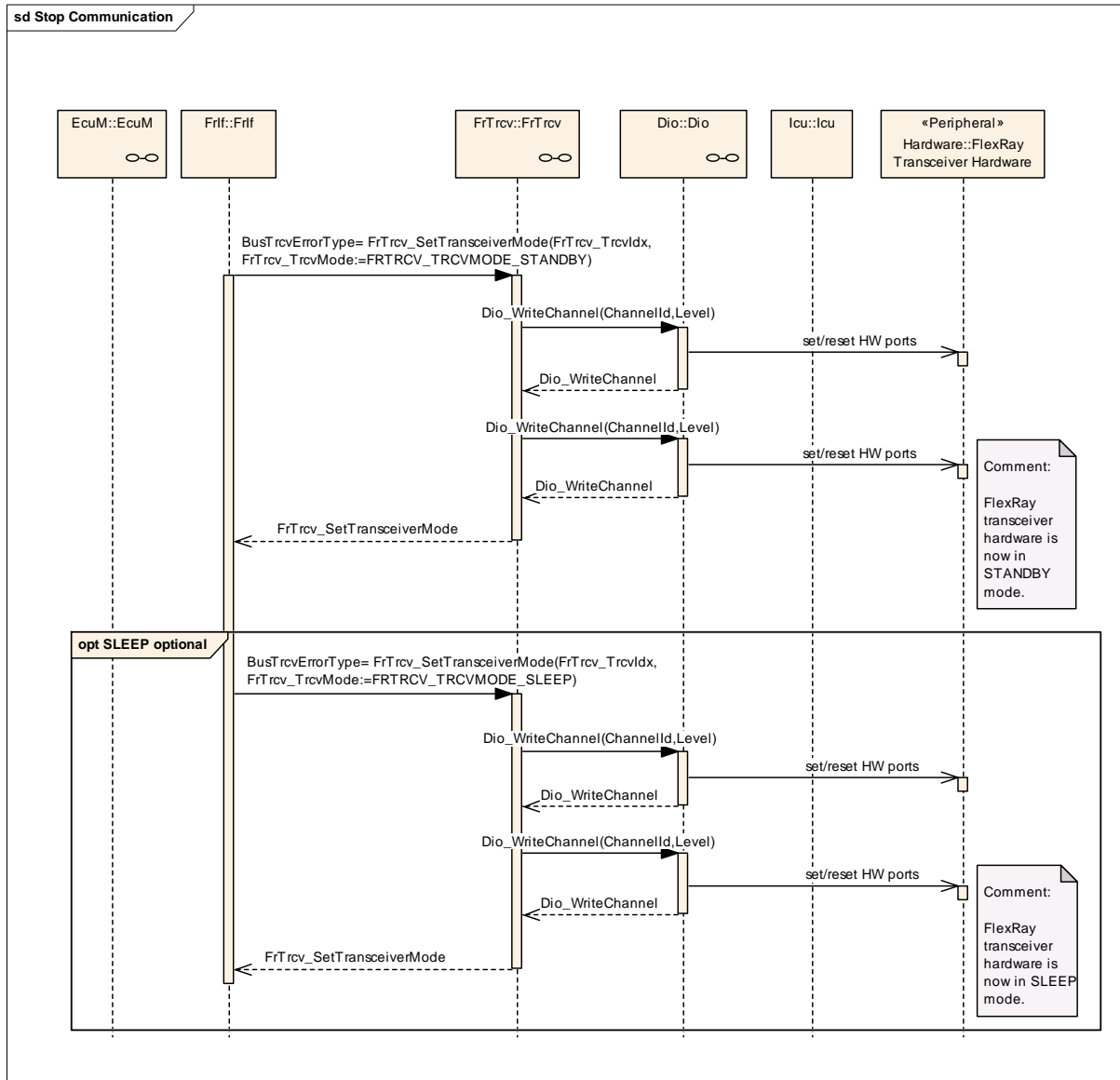
ATTENTION: Either SPI or DIO must be supported depending on FlexRay Transceiver hardware.

8.6.3 Configurable interfaces

There are no configurable interfaces for FlexRay transceiver driver.

9 Sequence diagrams





ATTENTION: These sequence charts are application examples only. They focus on interaction between the FlexRay transceiver driver (FrTrcv), FlexRay Interface (FrIf) and BSW modules ComM, EcuM, Icu and Dio. For details see [14] and [15]. Depending on FlexRay transceiver hardware one or more calls to Dio_WriteChannels may be necessary.

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals.

Chapter 10.2 specifies the structure (containers) and the parameters of the module FrTrcv.

Chapter 10.3 specifies published information of the module FrTrcv.

10.1 How to read this chapter

In addition to this section, it is highly recommended to read the documents:

- AUTOSAR Layered Software Architecture
- AUTOSAR ECU Configuration Specification
This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

10.1.1 Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an implementation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/hardware) in use during system and/or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term “configuration class” (of a parameter) shall be used in order to refer to a specific configuration point in time.

10.1.2 Variants

FrTrcv095: [FrTrcv219](#)

Variants describe sets of configuration parameters. E.g., variant 1: only pre-compile time configuration parameters; variant 2: mix of pre-compile- and post build time-configuration parameters. In one variant a parameter can only be of one configuration class.

Each Variant must have a unique name which could be referenced to in later chapters.

10.1.3 Containers

FrTrcv087: Containers structure the set of configuration parameters. This means:

- *all* configuration parameters are kept in containers.
- (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

Configuration parameters shall be clustered into a container whenever

- the configuration parameters logically belong together
(e.g. general parameters which are valid for the entire module NVRAM manager)
- the configuration parameters need to be instantiated
(e.g. parameters of the memory block specification of the NVRAM manager – those parameters must be instantiated for each memory block)

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters are described in the preceding chapters.

10.2.1 Variants

Variant 1: Only pre compile time parameters.

Variant 2: Mix of pre compile- and link time parameters (unused in this driver).

Variant 3: Mix of pre compile-, link time and post build time parameters (unused in this driver).

FrTrcv096: Pre compile time configuration shall be supported.

Link time parameters are not used within the FlexRay Transceiver Driver.

Post build time configuration changes by flashing are not used within the FlexRay Transceiver Driver.

10.2.2 General configuration requirements

All following configuration requirements has to be fulfilled by configuration tool. Configuration information is part of files FrTrcv.h and FrTrcv_Cfg.c.

Requirement	Description
FrTrcv010:	A configuration tool is used to generate the configuration data and code if any.
FrTrcv011:	[BSW00160] Human-readable configuration data
FrTrcv018:	[BSW00170] Data for reconfiguration of AUTOSAR SW-Components
FrTrcv023:	[BSW00302] Limit exported information
FrTrcv225:	[BSW05131] Configuration Data for FlexRay Transceiver
	[BSW05132] Support for More than One FlexRay Transceiver
FrTrcv016:	The configuration tool has to check the validity of the provided input data and the usability in the project context.
FrTrcv080	Provide configuration dependency information.

FrTrcv088:

10.2.3 Container FlexRayTransceiverDriverBasic

SWS Item	FrTrcv055: FrTrcv106: FrTrcv123:
Container Name	FlexRayTransceiverDriverBasic
Description	Container gives FlexRay transceiver driver basic information.
Configuration Parameters	

Name	FRTRCV_DEV_ERROR_DETECT		
Description	Switches development error detection and notification on and off. If switched on, "#define FRTRCV_DEV_ERROR_DETECT ON" shall be generated. If switched off, "#define FRTRCV_DEV_ERROR_DETECT OFF" shall be generated. Define shall be part of file FrTrcv_Cfg.h.		
Type or Unit	BooleanParamDef		
Range	TRUE	Is used.	
	FALSE (default)	Is not used.	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Module		
Dependency	-		

Name	FRTRCV_MAIN_FUNCTION_CYCLE_TIME		
Description	Cyclic call time for function FrTrcvMainFunction. A call time of 0ms indicates no calls for this function. In this case function need not be present in compiled code.		
	FloatParamDef		
Range	0 – 0,1	Cycle time in s.	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Module		
Dependency	FRTRCV_WAKEUP_POLLING		

Name	FRTRCV_GET_VERSION_INFO		
Description	Switches version information API on and off. If switched off, function need not be present in compiled code.		
Type or Unit	BooleanParamDef		
Range	TRUE	Is used.	
	FALSE (default)	Is not used.	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Module		
Dependency	--		

Name	FRTRCV_RECEIVEONLY_SUPPORTED		
Description	Information if the optional transceiver state RECEIVEONLY is supported by the driver and hardware. See also chapter 7.1		
Type or Unit	BooleanParamDef		
Range	TRUE	RECEIVEONLY state is supported	
	FALSE (default)	RECEIVEONLY state is not supported	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Module		
Dependency	--		

Name	FRTRCV_SLEEP_SUPPORTED		
Description	Information if the optional transceiver state SLEEP is supported by the driver and hardware. See also chapter 7.1		
Type or Unit	BooleanParamDef		
Range	TRUE	SLEEP state is supported	
	FALSE (default)	SLEEP state is not supported	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Module		
Dependency	--		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
FlexRayTransceiverNode	1...n	ECU

10.2.4 Container FlexRayTransceiverNode

SWS Item	FrTrcv091
Container Name	FlexRayTransceiverNode
Description	Container gives FlexRay transceiver driver information about a single FlexRay transceiver node. Any FlexRay transceiver driver has such FlexRay transceiver nodes.
Configuration Parameters	

Name	FRTRCV_NODE_ID		
Description	Unique node id. It is used by ComM by Icu and internally.		
Type or Unit	IntegerParamDef		
Range	0 ... (n-1)		
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Instance		
Dependency	--		

Name	FRTRCV_INIT_STATE		
Description	State of FlexRay transceiver after power on.		
Type or Unit	EnumParamDef FrTrcv_TrcvModeType		
Range	FRTRCV_TRCVMODE_NORMAL (default)	In normal mode	
	FRTRCV_TRCVMODE_STANDBY	In standby mode	
	FRTRCV_TRCVMODE_SLEEP	In sleep mode	
	FRTRCV_TRCVMODE_RECEIVEONLY	In read only mode	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Instance		
Dependency	--		

Name	FRTRCV_WAKEUP_BY_NODE_USED		
Description	Is wake up by node supported? If FlexRay transceiver hardware does not support wake up by node value is always FALSE. If FlexRay transceiver hardware supports wake up by node value is TRUE or FALSE depending		

	whether it is used or not.		
Type or Unit	BooleanParamDef		
Range	TRUE	Is used.	
	FALSE (default)	Is not used.	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Instance		
Dependency	<p>This container: FrTrcv019 FrTrcv229: FRTRCV_WAKEUP_POLLING</p> <p>If FRTRCV_WAKEUP_BY_NODE_USED is FALSE or if FRTRCV_WAKEUP_POLLING is TRUE function FrTrcv_MainFunction need not be present in compiled code.</p> <p>If FRTRCV_WAKEUP_POLLING is TRUE for all containers which states FRTRCV_WAKEUP_BY_NODE_USED is TRUE function FrTrcv_Cbk_WakeupByTransceiver need not be present in compiled code.</p>		

Name	FRTRCV_WAKEUP_POLLING		
Description	Information of this entity is only relevant if parameter FrTrcvWakeupByTransceiverUsed is TRUE. Information whether wake up events will be performed in main function (TRUE) or by a call-back function called by an interrupt service (FALSE).		
Type or Unit	BooleanParamDef		
Range	TRUE	Use main function.	
	FALSE (default)	Use appropriate ISR routines if.	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Instance		
Dependency	<p>This container: FRTRCV_WAKEUP_BY_NODE_USED</p> <p>Container FlexRayTransceiverDriverBasic: FRTRCV_MAIN_FUNCTION_CYCLE_TIME</p> <p>If FRTRCV_WAKEUP_BY_NODE_USED is FALSE or if FRTRCV_WAKEUP_POLLING is TRUE function FrTrcv_MainFunction need not be present in compiled code.</p> <p>If FRTRCV_WAKEUP_POLLING is TRUE for all containers which states FRTRCV_WAKEUP_BY_NODE_USED is TRUE function FrTrcv_Cbk_WakeupByTransceiver need not be present in compiled code.</p>		

Name	FRTRCV_CONTROLS_POWER_SUPPLY		
Description	Is ECU power supply controlled by this transceiver?		
Type or Unit	BooleanParamDef		
Range	TRUE	Controlled by transceiver.	
	FALSE (default)	Not controlled by transceiver.	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Instance		
Dependency	--		

Name	FRTRCV_MAX_BAUDRATE		
Description	Maximum baud rate for transceiver hardware type. Only used for validation purposes. Value shall be configured by configuration tool based on		

	FRTRCV_HARDWARE_NAME and internal information about ability of this hardware type.		
Type or Unit	IntegerParamDef		
Range	0-10000000	Baud rate in Baud	
Configuration Class	Pre-compile	x	Variant 1
	Link time	--	--
	Post Build	--	--
Scope	Instance		
Dependency	FRTRCV_HARDWARE_NAME		

10.3 Published Information

FrTrcv101: Published information contains data defined by the implementer of the SW module that does not change when the module is adapted (i.e. configured) to the actual HW/SW environment. It thus contains version and manufacturer information.

SWS Item	FrTrcv034: FrTrcv073: FrTrcv078:	
Information elements		
Information element name	Type / Range	Information element description
FrTrcv_VENDOR_ID	#define/uint16	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
FrTrcv_MODULE_ID	#define/uint8	Module ID of this module from Module List
FrTrcv_AR_MAJOR_VERSION	#define/uint8	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
FrTrcv_AR_MINOR_VERSION	#define/uint8	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
FrTrcv_AR_PATCH_VERSION	#define/uint8	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
FrTrcv_SW_MAJOR_VERSION	#define/uint8	Major version number of the vendor specific implementation of the module. The numbering is vendor specific.
FrTrcv_SW_MINOR_VERSION	#define/uint8	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
FrTrcv_SW_PATCH_VERSION	#define/uint8	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

11 Changes to Release 1

FlexRay transceiver driver was not part of AUTOSAR release 1. Thus this chapter is not applicable for AUTOSAR release 2.

11.1 Deleted SWS Items

<i>SWS Item</i>	<i>Rationale</i>
--	--
--	--

11.2 Replaced SWS Items

<i>SWS Item of Release 1</i>	<i>replaced by SWS Item</i>	<i>Rationale</i>
--	--	--
--	--	--

11.3 Changed SWS Items

<i>SWS Item</i>	<i>Rationale</i>
--	--
--	--

11.4 Added SWS Items

<i>SWS Item</i>	<i>Rationale</i>
--	--
--	--