

# MICROSAR FlexRay Transceiver Driver NXP TJA1080

## Technical Reference

Version 1.18.01

Status	Released
--------	----------

# 1 Document Information

## 1.1 History

Date	Version	Remarks
2006-08-17	1.00.00	Creation of document
2006-09-11	1.01.00	Minor updates
2007-08-31	1.02.00	Added Limitation L003
2008-02-12	1.03.00	Update for AUTOSAR 3
2008-04-17	1.04.00	ESCAN00025578 Rename TechnicalReference and BSWMD file to MSR ShortName (MSR 3 only!)
2008-07-08	1.05.00	ESCAN00027654 Extensions for Wakeup Interrupt-Mode ESCAN00028206 Configurability of Read Out Wait States ESCAN00028280 Change to STANDBY ends in SLEEP because the EN signal is delayed by the Tja1080 Removed L003
2008-08-25	1.06.00	ESCAN00029309 Transceiver Driver shall support Diagnosis functionality
2008-10-13	1.07.00	Minor modifications and clarifications
2009-08-31	1.08.00	ESCAN00033495 Wrong Memmap sections used
2009-11-18	1.09.00	ESCAN00038804 Support Vendor Id and vendor api infix in file and API names
2010-02-08	1.10.00	ESCAN00040646 Remove dio in naming
2010-06-07	1.11.00	Minor modifications and clarifications
2010-11-11	1.12.00	ESCAN00046814 Add TJA1081 as supported derivative
2011-01-03	1.13.00	Minor modifications and clarifications
2012-11-20	1.14.00	ESCAN00063174 Describe support of user <u>dio</u> and new timer interface
2013-04-30	1.15.00	ESCAN00066893 Clarify Interrupt lock and safe context usage
2013-08-20	1.16.00	ESCAN00069910 Describe ASR4 API extension
2014-10-14	1.16.01	ESCAN00077240 AR3-2679: Description BCD-coded return-value of XXX_GetVersionInfo() in TechRef ESCAN00078930 Missing explanation of API FrTrcv_30_Tja1080_GetVersionInfo
2015-02-06	1.16.02	ESCAN00080891 Rename <u>Techref</u> TechnicalReference_FrTrcv_Tja1080.pdf
2015-09-17	1.17.00	ESCAN00085291 Add NCV7381 as supported derivative
2015-10-26	1.18.00	Rework for SafeBsw
2016-02-10	1.18.01	Updates for SafeBsw

Table 1-1 History of the document

## 1.2 Reference Documents

No.	Title	Version
[1]	AUTOSAR_SWS_FlexRayTransceiver.pdf	1.2.1
[2]	AUTOSAR_SWS_DET.pdf	2.2.1
[3]	AUTOSAR_SWS_DEM.pdf	2.2.0
[4]	AUTOSAR_BasicSoftwareModules.pdf	1.0.0
[5]	TJA1080A.pdf	Rev. 6
[6]	NXP - ApplicationHints_Rev 3_TJA1080.pdf	
[7]	TJA1081B.pdf	Rev. 1
[8]	NCV7381-D.PDF	Rev. 2

Table 1-2 Reference documents

## 1.3 Scope of the Document

This technical reference describes the general use of the FlexRay Transceiver driver basis software. Please refer to your Release Notes to get a detailed description of the platform (Host, CC, Compiler, Transceiver) your Vector FlexRay Bundle has been configured for.



### Please note

We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

## Contents

<b>1</b>	<b>Document Information .....</b>	<b>2</b>
1.1	History .....	2
1.2	Reference Documents .....	3
1.3	Scope of the Document.....	3
<b>2</b>	<b>Introduction.....</b>	<b>10</b>
2.1	Supported Devices.....	10
2.2	Architecture Overview .....	11
<b>3</b>	<b>Functional Description .....</b>	<b>13</b>
3.1	Initialization .....	13
3.1.1	High-Level Initialization .....	13
3.1.2	Low-Level Initialization .....	13
3.2	States .....	13
3.3	Main Function .....	13
3.4	Error Handling.....	13
3.4.1	Development Error Reporting.....	13
3.4.2	Production Code Error Reporting .....	14
3.4.2.1	Transceiver Diagnosis.....	15
<b>4</b>	<b>Integration.....</b>	<b>16</b>
4.1	Scope of Delivery.....	16
4.1.1	Static Files .....	16
4.1.2	Dynamic Files .....	16
4.2	Compiler Abstraction and Memory Mapping.....	16
4.3	Data Consistency.....	17
4.3.1	FRTRCV_30_TJA1080_EXCLUSIVE_AREA_0 .....	17
4.3.2	FRTRCV_30_TJA1080_EXCLUSIVE_AREA_1 .....	17
4.4	The Software Timers.....	17
4.5	Dependencies to Dio component and User Dio functions.....	18
4.6	Using ICU Functionality.....	19
4.7	Adaptation of FrTrcvPhy_30_Tja1080.c .....	19
4.7.1	Dio pin configuration .....	19
4.7.2	FrTrcvPhy_30_Tja1080_Init.....	19
4.7.3	FrTrcvPhy_30_Tja1080_SetMode.....	19
4.7.4	FrTrcvPhy_30_Tja1080_GetMode .....	19
4.7.5	FrTrcvPhy_30_Tja1080_ReadStatusWord .....	19

<b>5</b>	<b>API Description.....</b>	<b>20</b>
5.1	Interfaces Overview .....	20
5.2	Type Definitions .....	20
5.3	Structures .....	21
5.4	Services provided by FlexRay Transceiver Driver .....	22
5.4.1	Administrative Functions .....	22
5.4.1.1	FrTrcv_30_Tja1080_InitMemory: Initialization of Transceiver Driver.....	22
5.4.1.2	FrTrcv_30_Tja1080_Init: Initialization of Transceiver Driver .....	22
5.4.1.3	FrTrcv_30_Tja1080_GetVersionInfo: Read Version Information of the Driver .....	23
5.4.1.4	FrTrcv_30_Tja1080_MainFunction: Main Function of Transceiver Driver.....	23
5.4.2	Service Functions .....	25
5.4.2.1	FrTrcv_30_Tja1080_SetTransceiverMode: Set the Transceiver in the requested mode .....	25
5.4.2.2	FrTrcv_30_Tja1080_GetTransceiverMode: Get the current Transceiver mode .....	26
5.4.2.3	FrTrcv_30_Tja1080_GetTransceiverWUReason: Get the wake up reason.....	26
5.4.2.4	FrTrcv_30_Tja1080_ClearTransceiverWakeup: Clear pending wake up events .....	27
5.4.2.5	FrTrcv_30_Tja1080_DisableTransceiverBranch: Disable an individual branch .....	28
5.4.2.6	FrTrcv_30_Tja1080_EnableTransceiverBranch: Enable an individual branch .....	29
5.4.2.7	FrTrcv_30_Tja1080_GetTransceiverError: Read current Transceiver error.....	29
5.5	Services used by FlexRay Transceiver Driver .....	30
5.6	Callback Functions.....	30
5.6.1	FrTrcv_30_Tja1080_CheckWakeupByTransceiver.....	31
5.7	Callout Functions .....	31
5.7.1	Appl_FrTrcv_30_Tja1080_Wait.....	31
5.7.2	Appl_FrTrcv_30_Tja1080_Dio_WriteChannel .....	32
5.7.3	Appl_FrTrcv_30_Tja1080_Dio_ReadChannel .....	32
5.7.4	Appl_FrTrcv_30_Tja1080_DisableNotification.....	33
5.7.5	Appl_FrTrcv_30_Tja1080_EnableNotification .....	33
5.7.6	Appl_FrTrcv_30_Tja1080_ReportErrorStatusPreFailed .....	34
5.7.7	Appl_FrTrcv_30_Tja1080_ReportErrorStatusPrePassed .....	34
<b>6</b>	<b>AUTOSAR Standard Compliance.....</b>	<b>35</b>
6.1	Deviations .....	35
6.1.1	SetTransceiverMode validation .....	35

6.1.2	Re-Initialization .....	35
6.1.3	Error detection during Init .....	35
6.2	Additions/ Extensions .....	35
6.2.1	Transceiver Diagnosis .....	35
6.2.2	User Dio functions .....	35
6.3	Limitations .....	35
6.3.1	L001: Clock Frequency .....	35
<b>7</b>	<b>Glossary and Abbreviations .....</b>	<b>36</b>
7.1	Glossary .....	36
7.2	Abbreviations .....	36
<b>8</b>	<b>Contact .....</b>	<b>37</b>

## Illustrations

Figure 2-1	AUTOSAR architecture.....	11
Figure 2-2	Interfaces to adjacent modules of the FlexRay Transceiver Driver.....	12
Figure 5-1	FlexRay Transceiver Driver API .....	20

## Tables

Table 1-1	History of the document.....	2
Table 1-2	Reference documents.....	3
Table 1-3	Component history.....	9
Table 3-1	Mapping of service IDs to services .....	14
Table 3-2	Errors reported to DET .....	14
Table 3-3	Errors reported to DEM.....	14
Table 3-4	Extended errors reported to DEM .....	15
Table 3-5	Extended errors not detected by the Tja1080 .....	15
Table 4-1	Static files .....	16
Table 4-2	Generated files .....	16
Table 4-3	Compiler abstraction and memory mapping.....	17
Table 5-1	Type definitions.....	21
Table 5-2	FrTrcvChannel .....	21
Table 5-3	Services used by the FlexRay Transceiver Driver .....	30
Table 7-1	Glossary .....	36
Table 7-2	Abbreviations.....	36

## Component History

Component Version	New Features
3.00.00	Support of AUTOSAR 3
3.01.00	The API <MSN>_GetVersionInfo now correctly returns the instanceID
3.02.00	Wakeup is now correctly detected via the ERRN pin
3.05.00	Several bug fixes
3.05.01	ESCAN00033397 Missing extern in call-back declaration ESCAN00029695 AUTOSAR Dummy Statements ESCAN00033495 Wrong MemMap.h sections used ESCAN00034964 Single Channel API change for AUTOSAR SWS compliance
3.05.02	ESCAN00034905 Wrong MemMap.h Stop section defines
3.05.03	ESCAN00034037 Compile warnings by extended version checks
3.05.04	ESCAN00036155 pre-processor checks for building the library
3.05.05	ESCAN00038804 Support Vendor Id and vendor api infix in file and API names
3.05.06	ESCAN00040215 Extended Version Check causes compile error ESCAN00040646 Remove dio in naming
3.05.07	ESCAN00041864 EcuM_SetWakeupEvent call with active interrupt locks ESCAN00040839 Move FrTrcv_TrvcModeType and FrTrcv_TrvcWUReasonType to Fr_GeneralTypes.h
3.05.08	ESCAN00042990 Missing MISRA2004 justifications ESCAN00043289 Interrupts are enabled to early in FrTrcv_Phy
3.05.09	ESCAN00044081 Transceiver might hang in Goto-Sleep Mode
3.06.00	ESCAN00047306 Adapt MainFunction for usage with IdentityManagerConfig
3.06.01	ESCAN00050103 Extern declaration with array size is used ESCAN00035705 Similar DET Usage ESCAN00048980 Support of FrTrcvWakeUpSupport configuration parameter
3.06.02	ESCAN00054673 Compilation error if different FlexRay transceivers are used ESCAN00054897 Wake up is detected while in Normal Mode
3.07.00	ESCAN00054957 Extend implementation with ASR4 API
3.07.01	ESCAN00053417 AR3-2069: Remove non-SchM code for critical section handling
3.07.02	ESCAN00060219 Support ASR4 critical section handling ESCAN00060220 Cfg5: compiler error with missing DRVTRANS_TJA1080_FR_ASR_DLL_VERSION ESCAN00060980 compiler error: GetVersionInfo API writes Instance ID
3.08.00	ESCAN00056613 Delay loop optimized away, replace with timer call outs ESCAN00061053 Support user dio call out ESCAN00060091 DET error caused by early ICU notification. ESCAN00060220 Cfg5: compiler error with missing DRVTRANS_TJA1080_FR_ASR_DLL_VERSION
3.08.01	ESCAN00066931 FrTrcv_30_Tja1080_TrvcInit shall be named FrTrcv_30_Tja1080_Init ESCAN00066941 Variable in section noinit is initialized during InitMemory ESCAN00066940 Variable enclosed in memory mapping sections inside function



Component Version	New Features
	ESCAN00066856 AR4-220: redefine STATIC to static ESCAN00067027 Create seperate Exclusive Areas for SetMode/ReadStatusWord
3.09.00	ESCAN00066938 Implement FrTrcv_30_Tja1080_GetTransceiverError ESCAN00069547 Improve AR compatibility by adding <vendor>_<device> infix
3.09.01	ESCAN00081075 DEM status not triggered continuously when healed
4.00.00	ESCAN00080595 UNINIT DET check of function FrTrcv_GetTransceiverError does not work ESCAN00080737 Diagnosis not working if wake up detection is not enabled ESCAN00081035 DEM status not triggered continuously when healed ESCAN00080313 FEAT-427: SafeBSW Step I ESCAN00081798 FEAT-1275: SafeBSW Step 2 ESCAN00085850 Wake Up reason not set to FRTRCV_WU_INTERNALLY if mode is set to NORMAL ESCAN00085867 Default mode might overwrite wake up during FrTrcv_Init
4.01.00	ESCAN00088026 Implement HSI description

Table 1-3 Component history

## 2 Introduction

This document describes the functionality, API and configuration of the AUTOSAR BSW module FlexRay Transceiver Driver as specified in [1].

<b>Supported AUTOSAR Release*:</b>	3/4	
<b>Supported Configuration Variants:</b>	pre-compile	
<b>Vendor ID:</b>	FlexRay Transceiver Driver_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
<b>Module ID:</b>	FlexRay Transceiver Driver_MODULE_ID	71 decimal (according to ref. [4])

\* For the precise AUTOSAR Release 3.x please see the release specific documentation.

The FlexRay Transceiver Driver provides hardware independent access to control connected Transceivers in a generic way. It offers the functionality to control the mode of operation of connected Transceivers as well as to determine their current state, e.g. if events like wake up or bus errors happened.

The Transceiver itself is a hardware device, which mainly transforms the logical 1/0 signals of the FlexRay Controller to the bus compliant electrical levels, currents and timings.

### 2.1 Supported Devices

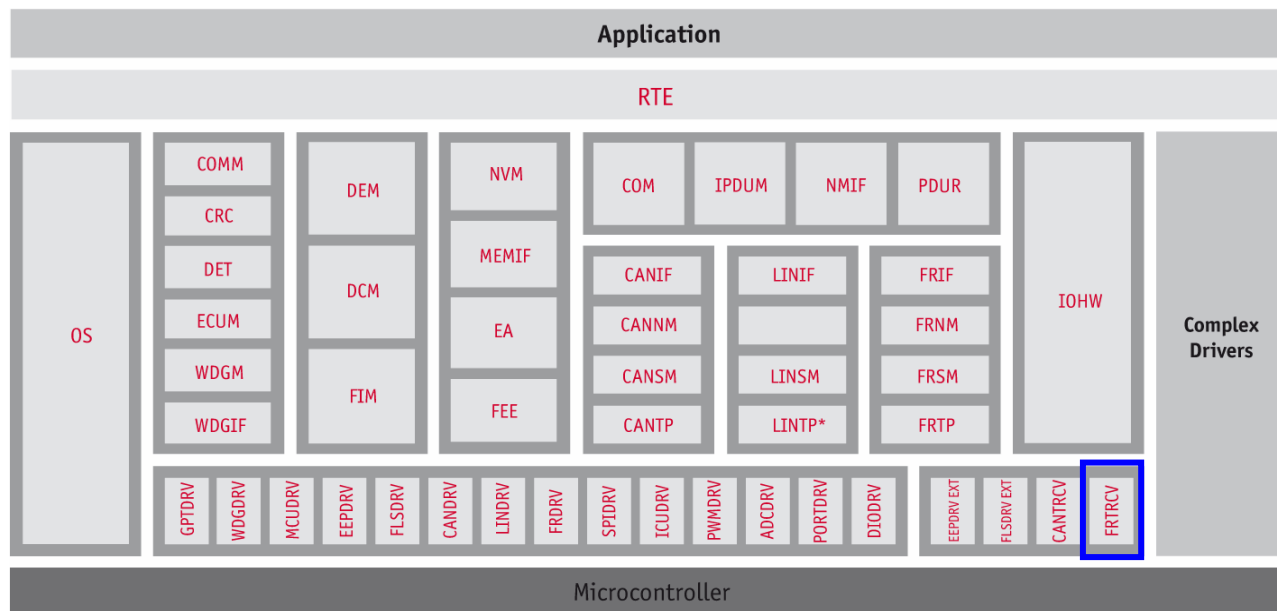
There are devices that are compatible to the Tja1080 regarding the host side interface. The devices, supported by this Transceiver Driver, are:

Device	Device Data Sheet	Version
TJA1080A	TJA1080A.pdf	Rev. 6
TJA1081B	TJA1081B.pdf	Rev. 1
NCV7381	NCV7381-D.PDF	Rev. 2

Different timing for accessing the Status register and setting the mode must be taken into consideration during Integration. Please refer to the device data sheet for details.

## 2.2 Architecture Overview

The following figure shows where the FlexRay Transceiver Driver is located in the AUTOSAR architecture.



Vector MICROSAR Product

Service by Vector

\* Option included in LINIF

Figure 2-1 AUTOSAR architecture

The next figure shows the interfaces to adjacent modules of the FlexRay Transceiver Driver. These interfaces are described in chapter 5.

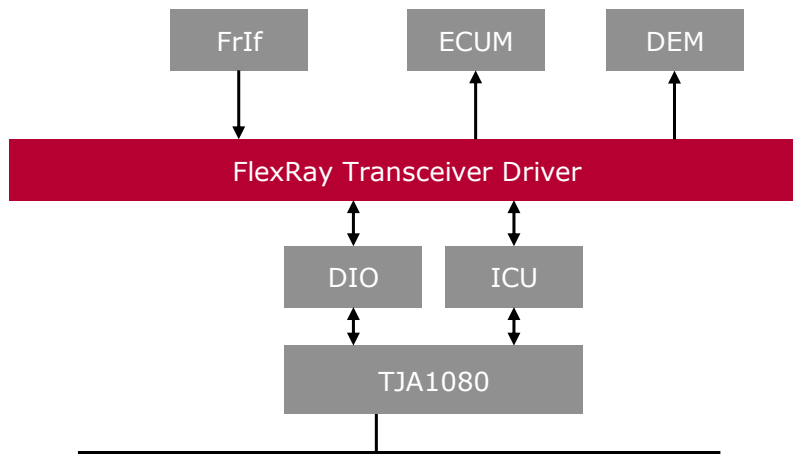


Figure 2-2 Interfaces to adjacent modules of the FlexRay Transceiver Driver

Applications do not access the services of the BSW modules directly. They use the service ports provided by the BSW modules via the RTE.

The FlexRay Transceiver Driver does not have any service ports, therefore no connection to the RTE exists.

## 3 Functional Description

The following chapters highlight some functional aspects of the Transceiver Driver.

### 3.1 Initialization

#### 3.1.1 High-Level Initialization

The Transceiver Driver is initialized by calling the `FrTrcv_30_Tja1080_Init`. The default operation mode of the Transceiver after Init is pre-defined by the GenTool during configuration process.

If a startup code is used that does not initialize the memory please call `FrTrcv_30_Tja1080_InitMemory` first.

#### 3.1.2 Low-Level Initialization

The user is responsible to initialize all I/O-ports used by the Transceiver before any Transceiver Driver function is called. The same holds true for the ICU module, if it is used.

### 3.2 States

After initialization the Transceiver is in a predetermined state which has been configured in the GenTool.

The Transceiver driver differentiates between active and low power modes. Only in low power modes (STANDBY/SLEEP) an optional wake up detection is performed while error diagnosis is optionally performed in active (NORMAL/RECEIVE\_ONLY) as well as low power modes.

### 3.3 Main Function

The Transceiver Driver has one task `FrTrcv_30_Tja1080_MainFunction` which has to be called cyclically. This task is responsible for polling all connected Transceivers and perform action if so required. Please note that this main function will not be present if polling is not used and the call cycle time is configured as 0.

### 3.4 Error Handling

#### 3.4.1 Development Error Reporting

Development errors are reported to DET using the service `Det_ReportError()`, (specified in [2]), if this feature is enabled in the GenTool.

The reported FlexRay Transceiver Driver ID is 71.

The reported service IDs identify the services which are described in 5.4. The following table presents the service IDs and the related services:

Service ID	Service
0	<code>FrTrcv_30_Tja1080_Init()</code>
1	<code>FrTrcv_30_Tja1080_SetTransceiverMode()</code>
5	<code>FrTrcv_30_Tja1080_GetTransceiverMode()</code>

Service ID	Service
6	FrTrcv_30_Tja1080_GetTransceiverWUReason()
7	FrTrcv_30_Tja1080_GetVersionInfo()
8	FrTrcv_30_Tja1080_GetTransceiverError()
12	FrTrcv_30_Tja1080_ClearTransceiverWakeup()
13	FrTrcv_30_Tja1080_MainFunction()
14	FrTrcv_30_Tja1080_CheckWakeupByTransceiver()
15	FrTrcv_30_Tja1080_DisableTransceiverBranch()
16	FrTrcv_30_Tja1080_EnableTransceiverBranch()

Table 3-1 Mapping of service IDs to services

The errors reported to DET are described in the following table:

Error Code	Description
0x01 BUSTRCV_E_FR_INVALID_TRCVIDX	The Transceiver Driver was called with an invalid transceiver Index.
0x10 BUSTRCV_E_FR_UNINIT	A Transceiver Driver service was called without initializing the module first by calling FrTrcv_Init.
0x15 FRTRCV_30_TJA1080_E_FR_NO_CONTROL_TRCV	The Transceiver Driver has no control over the attached hardware.
0x20 BUSTRCV_E_FR_INVALID_POINTER	A Transceiver Driver service was called with a zero pointer as parameter.

Table 3-2 Errors reported to DET

### 3.4.2 Production Code Error Reporting

Production code related errors are reported to DEM using the service Dem\_ReportErrorStatus() (specified in [3]).

The errors reported to DEM are described in the following table:

Error Code	Description
DEM_EVENT_E_FR_ERRN_TRCV_x	If the diagnosis detects a bus error the DEM is notified with DEM_EVENT_STATUS_PREFAILED

Table 3-3 Errors reported to DEM

### 3.4.2.1 Transceiver Diagnosis

The Transceiver Driver provides extended diagnosis functionality if the 'Transceiver Diagnosis' option in the GenTool is enabled. The following errors can be reported by the API specified in 5.4.2.7:

Error Code	Description
FRTRCV_30_TJA1080_BDE_GLOBAL	If any of the below errors are reported.
FRTRCV_30_TJA1080_BDE_NOISE	The Transceiver issues this error if a branch is clamped active.
FRTRCV_30_TJA1080_BDE_OVER_TEMPERATURE	This error is issued if the junction temperature is above a critical limit.
FRTRCV_30_TJA1080_BDE_TXEN_TIMEOUT	This error is issued if the TXEN pin is continuously clamped and therefore the transmitter is permanently transmitting.
FRTRCV_30_TJA1080_BDE_UNDERVOLTAGE	This error is issued if the either the VBAT, VCC or VIO voltage is below a critical limit.

Table 3-4 Extended errors reported to DEM

The following errors are specified by AUTOSAR but are not detected by the Tja1080 Transceiver and therefore not reported:

Error Code	Description
FRTRCV_30_TJA1080_BDE_SHORT_CIRCUIT_BM_BP	-
FRTRCV_30_TJA1080_BDE_BP_SHORT_CIRCUIT_GND	-
FRTRCV_30_TJA1080_BDE_BP_SHORT_CIRCUIT_VCC	-
FRTRCV_30_TJA1080_BDE_BM_SHORT_CIRCUIT_GND	-
FRTRCV_30_TJA1080_BDE_BM_SHORT_CIRCUIT_VCC	-
FRTRCV_30_TJA1080_BDE_GENERAL_FAULT	-

Table 3-5 Extended errors not detected by the Tja1080

Please take a look in the device data sheet [5] for a detailed explanation of these errors.

Every time such an error appears the DEM notification `FRTRCV_30_TJA1080_E_FR_ERRN_TRCV(n)` will be called with status `DEM_EVENT_STATUS_PREFAILED`. This notification can be used to request the errors described above for further bus diagnosis.

When a error disappears the same DEM notification will be called again with `DEM_EVENT_STATUS_PREPASSED`.

## 4 Integration

This chapter gives necessary information for the integration of the MICROSAR FlexRay Transceiver Driver into an application environment of an ECU.

### 4.1 Scope of Delivery

The delivery of the FlexRay Transceiver Driver contains the files which are described in the chapters 4.1.1 and 4.1.2:

#### 4.1.1 Static Files

The static files are not to be modified




File Name	Description	
FrTrcv_30_Tja1080.c	Source code of Transceiver Driver.	
FrTrcv_30_Tja1080.h	API definitions of the Transceiver Driver.	
FrTrcv_30_Tja1080_Cbk.h	API definitions for call-back functions.	

Table 4-1 Static files

#### 4.1.2 Dynamic Files

The dynamic files can be modified if necessary.





File Name	Description	
_FrTrcvPhy_30_Tja1080.c	Source code of Low-Level Transceiver Driver. This file may be modified according to the used transceiver!	
_FrTrcvPhy_30_Tja1080.h	API definitions of the Low-Level Transceiver Driver. This file may be modified according to the used transceiver!	
FrTrcv_30_Tja1080_Cfg.c	Parameter Configuration source file for Transceiver Driver. Can be modified if the GenTool is not used.	
FrTrcv_30_Tja1080_Cfg.h	Parameter Configuration header file for Transceiver Driver. Can be modified if the GenTool is not used.	

Table 4-2 Generated files

### 4.2 Compiler Abstraction and Memory Mapping

The objects (e.g. variables, functions, constants) are declared by compiler independent definitions – the compiler abstraction definitions. Each compiler abstraction definition is assigned to a memory section.

The following table contains the memory section names and the compiler abstraction definitions defined for the FlexRay Transceiver Driver and illustrates their assignment among each other.



Memory Mapping Sections	Compiler Abstraction Definitions			
	FRTRCV_CONST	FRTRCV_VAR_NOINIT	FRTRCV_VAR_ZERO_INIT	FRTRCV_CODE
FRTRCV_START_SEC_CONST_UNSPECIFIED	■			
FRTRCV_START_SEC_CONST_32BIT	■			
FRTRCV_START_SEC_CONST_16BIT	■			
FRTRCV_START_SEC_CONST_8BIT	■			
FRTRCV_START_SEC_VAR_NOINIT_UNSPECIFIED		■		
FRTRCV_START_SEC_VAR_NOINIT_32BIT		■		
FRTRCV_START_SEC_VAR_ZERO_INIT_UNSPECIFIED			■	
FRTRCV_START_SEC_CODE				■

Table 4-3 Compiler abstraction and memory mapping

### 4.3 Data Consistency

The FlexRay Transceiver Driver calls service functions of upper layers in order to prevent interruption of critical sections (e.g. accessing Transceiver pins).

These service functions have to be provided (normally by the Schedule Manager) and configured accordingly. The following critical areas are used:

#### 4.3.1 FRTRCV\_30\_TJA1080\_EXCLUSIVE\_AREA\_0

This exclusive area is used to lock transceiver functionality against interruption by each other. They may be interrupted by application functionality or time critical CAT1 interrupts.

#### 4.3.2 FRTRCV\_30\_TJA1080\_EXCLUSIVE\_AREA\_1

This exclusive area is used to lock time critical functionality against interruption by any other interrupts. Please use a global interrupt lock for this exclusive area.

### 4.4 The Software Timers

In order to access the Transceiver a certain timing is required by the Transceiver Driver software. To generate this timing a software timer callback function named `Appl_FrTrcv_30_Tja1080_Wait` is used. This function has to be implemented by the user to generate the correct timing. The following implementation is an example and has to be completed to generate the correct timing:

**Example**

```
#include "FrTrcv_30_Tja1080.h"
#include "FrTrcv_30_Tja1080_Cbk.h"
#include "FrTrcvPhy_30_Tja1080.h"

FUNC(void, FRTRCV_CODE) Appl_FrTrcv_30_Tja1080_Wait(uint8 TimerIndex)
{
    switch(TimerIndex)
    {
        case kFrTrcv_30_Tja1080_delay_tEN_half:
            /* Insert code to delay fo tEN/2 which is 2us..10us on TJA1080
            and TJA1081 or 0,5us..2,5us on TJA1081B
            Please refer to the device datasheet for correct numbers */
            break;
        case kFrTrcv_30_Tja1080_delay_tDet:
            /* Insert code to delay for tDet(EN) which is at least 80us on
            TJA1080 and TJA1081 or at least 20us on TJA1081B
            Please refer to the device datasheet for correct numbers */
            break;
    }
}
```

**Hint!**

**Try to make the software timer as fast as allowed by the manufacturer device data sheet of the Transceiver as reading the Transceiver is done with active interrupt locks and takes quite some time!**

To verify correct Timing, measurement of the delay with an Oscilloscope is recommended. Please take also into consideration to map this function to the same context as the Transceiver Driver. Otherwise a call of this function might cause an context switch which would increase the delay considerably.

#### 4.5 Dependencies to Dio component and User Dio functions

The FlexRay Transceiver Driver performs hardware access by calling service functions of the lower layer component Dio Driver.

- > Function Dio\_WriteChannel is used to set the logical level of the channel pins to which the Transceiver hardware is connected.
- > Function Dio\_ReadChannel is used to get the logical level of the channel pins to which the Transceiver hardware is connected.
- > The Dio Driver has to provide the pin assignment for the Transceiver hardware pins EN, STBN and ERRN. These pins are referred by the FlexRay Transceiver Driver by using the symbolic names which must be specified in the FrTrcvPhy\_30\_Tja1080.c file.

In case the AUTOSAR DIO component is too slow to guarantee the required timing it is possible to use custom DIO functions instead. For this the functions `Appl_FrTrcv_30_Tja1080_Dio_WriteChannel` and `Appl_FrTrcv_30_Tja1080_Dio_ReadChannel` are called if this feature is enabled in the GenTool. The same channel names are used for the AUTOSAR DIO component. That means that the port pins still need to be configured in the AUTOSAR DIO component.

## 4.6 Using ICU Functionality

The Transceiver Driver supports a feature to enable/disable Transceiver interrupts. For this purpose two application call-backs `Appl_FrTrcv_30_Tja1080_DisableNotification` and `Appl_FrTrcv_30_Tja1080_EnableNotification` are called which can be used by the user to either call the respective ICU services or disable the port interrupts directly. Whenever the Transceiver is set to a low power mode the interrupts are enabled. Whenever the Transceiver is set to a normal operation mode the interrupts are disabled.

## 4.7 Adaptation of `FrTrcvPhy_30_Tja1080.c`

Depending on the number of used Transceiver or how your Transceiver is connected, adaptation of the file `FrTrcvPhy_30_Tja1080.c` might be required. The following subchapters highlight some details.

### 4.7.1 Dio pin configuration

For access of Dio controlled pins the correct name must be known. This name can be configured in the structure `FrTrcvChannel`. For each Transceiver there must be one entry defining the names for the I/O pins used to access the Transceiver.

### 4.7.2 `FrTrcvPhy_30_Tja1080_Init`

This method can be used to initialize any necessary low level functions.

### 4.7.3 `FrTrcvPhy_30_Tja1080_SetMode`

This method is used to set the respective Transceiver into the requested mode. Configuring the Transceiver for a different operation mode is done by setting the I/O ports to a certain state, reflecting the requested operation mode. The given method is made for the TJA1080/81.

### 4.7.4 `FrTrcvPhy_30_Tja1080_GetMode`

This method is used to read back the current mode of the Transceiver. The given method is made for the TJA1080/81.

### 4.7.5 `FrTrcvPhy_30_Tja1080_ReadStatusWord`

This method is used to read the status word of the Transceiver. This method has to return the gathered information in a standardised status word which is defined in the `FrTrcv_StatusBitField` type in the file `FrTrcvPhy_30_Tja1080.h`. If you use a different Transceiver, please make sure to return the status bits in this same format. Otherwise it is possible to move the values in the `FrTrcv_StatusBitField` type to the location required.

## 5 API Description

### 5.1 Interfaces Overview

The AUTOSAR Transceiver Driver provides the following services:

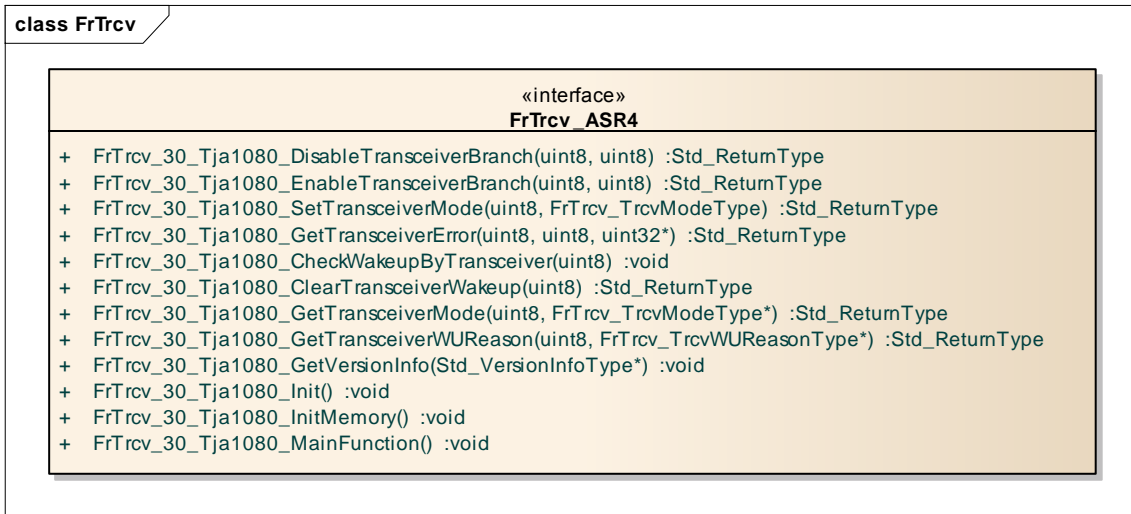


Figure 5-1 FlexRay Transceiver Driver API

### 5.2 Type Definitions

Type Name	C-Type	Description	Value Range
FrTrcv_TrcvModeType	uint8	Defines all possible Transceiver modes	FRTRCV_TRCVMODE_UNKNOWN Temporary state before initialization FRTRCV_TRCVMODE_NORMAL Normal operation mode FRTRCV_TRCVMODE_STANDBY Standby operation mode FRTRCV_TRCVMODE_SLEEP Sleep operation mode FRTRCV_TRCVMODE_RECEIVEONLY Receive only operation mode
FrTrcv_TrcvWUReasonType	uint8	The reason for the last recent wakeup	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_GotoNormalMode API call.

Type Name	C-Type	Description	Value Range
			FRTRCV_WU_RESET The Transceiver has detected that the "wake up" is due to an ECU reset.
			FRTRCV_WU_BY_PIN The transceiver has detected a wake-up event at one of the transceiver's pins (not at the FlexRay bus).
FrTrcv_StatusWordType	uint16	The status word read from the transceiver	See FrTrcvPhy_30_Tja1080.h for further details.

Table 5-1 Type definitions

## 5.3 Structures

### FrTrcvChannel

Contains the Dio pin description of the Transceiver. The Dio name must be configured in the AUTOSAR DIO component accordingly.

Struct Element Name	C-Type	Description
TrcvPinEN	Dio_ChannelType	Dio name of the respective pin
TrcvPinSTBN	Dio_ChannelType	Dio name of the respective pin
TrcvPinERRN	Dio_ChannelType	Dio name of the respective pin

Table 5-2 FrTrcvChannel

## 5.4 Services provided by FlexRay Transceiver Driver

The FlexRay Transceiver Driver API consists of services, which are realized by function calls.

### 5.4.1 Administrative Functions

#### 5.4.1.1 FrTrcv\_30\_Tja1080\_InitMemory: Initialization of Transceiver Driver

FrTrcv\_30\_Tja1080\_InitMemory

Prototype	
<code>void FrTrcv_30_Tja1080_InitMemory( void );</code>	
Parameters [in/out/both]	
Void	-
Return code	
Void	-
Service ID	
Service ID	-
Functional Description	
Initialization of the Transceiver Driver memory in case no start-up code is used that zeroes out the memory.	
Preconditions	
None.	
Postconditions	
The Transceiver Driver memory is initialized.	
Particularities and Limitations	
<ul style="list-style-type: none"><li>&gt; Call context: task level</li><li>&gt; Not re-entrant</li><li>&gt; Synchronous</li></ul>	

#### 5.4.1.2 FrTrcv\_30\_Tja1080\_Init: Initialization of Transceiver Driver

FrTrcv\_30\_Tja1080\_Init

Prototype	
<code>void FrTrcv_30_Tja1080_Init( void );</code>	
Parameters [in/out/both]	
void	-
Return code	
void	-
Service ID	
Service ID	0

Functional Description
Initialization of the Transceiver Driver module as well as the physical Transceiver itself.
Preconditions
The I/O ports, used to access the Transceiver, have to be initialized!
Postconditions
The Transceiver will be initialized to the configured operation state.
Particularities and Limitations
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>

### 5.4.1.3 FrTrcv\_30\_Tja1080\_GetVersionInfo: Read Version Information of the Driver

FrTrcv\_30\_Tja1080\_GetVersionInfo

Prototype	
<pre>void FrTrcv_30_Tja1080_GetVersionInfo( P2VAR(Std_VersionInfoType, AUTOMATIC, FRTRCV_APPL_DATA) versioninfo );</pre>	
Parameters [in/out/both]	
Versioninfo [out]	Pointer to the location where the Version information shall be stored.
Return code	
void	-
Service ID	
Service ID	7
Functional Description	
FrTrcv_30_Tja1080_GetVersionInfo() returns version information, vendor ID and AUTOSAR module ID of the component. The versions are BCD-coded.	
Preconditions	
FRTRCV_30_TJA1080_VERSION_INFO_API is STD_ON	
Postconditions	
None.	
Particularities and Limitations	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

### 5.4.1.4 FrTrcv\_30\_Tja1080\_MainFunction: Main Function of Transceiver Driver

FrTrcv\_30\_Tja1080\_MainFunction

Prototype	
<code>void FrTrcv_30_Tja1080_MainFunction( void );</code>	
Parameters [in/out/both]	
-	
Return code	
void	-
Service ID	
Service ID	13
Functional Description	
Main function of the Transceiver Driver for one instance. This service polls the respective Transceiver for any wake up events or bus errors. This functionality is optional. In case a wake up is detected and notifications are allowed the ECU Manager is notified via <code>EcuM_SetWakeupEvent</code> .	
Preconditions	
The Transceiver Driver module must be initialized.	
Postconditions	
If enabled a callback in case of a wake-up event is triggered.	
Particularities and Limitations	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	



## 5.4.2 Service Functions

### 5.4.2.1 FrTrcv\_30\_Tja1080\_SetTransceiverMode: Set the Transceiver in the requested mode

FrTrcv\_30\_Tja1080\_SetTransceiverMode

#### Prototype

```
Std_ReturnType FrTrcv_30_Tja1080_SetTransceiverMode
(
    const uint8 FrTrcv_TrcvIdx,
    FrTrcv_TrcvModeType FrTrcv_TrcvMode
);
```

#### Parameters [in/out/both]

FrTrcv_TrcvIdx [in]	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 [in]	<p>This parameter describes the mode the Transceiver shall be set in. It can have one of the following values:</p> <ul style="list-style-type: none"> <li>■ FRTRCV_TRCVMODE_NORMAL</li> <li>■ FRTRCV_TRCVMODE_STANDBY</li> <li>■ FRTRCV_TRCVMODE_SLEEP</li> <li>■ FRTRCV_TRCVMODE_RECEIVEONLY</li> </ul>

#### Return code

Std_ReturnType	The service returns E_NOT_OK if the Transceiver could not be set to the requested mode, otherwise E_OK is returned.
----------------	---------------------------------------------------------------------------------------------------------------------

#### Service ID

Service ID	1
------------	---

#### Functional Description

This service sets the Transceiver in the requested mode.

#### Preconditions

The Transceiver Driver module must be initialized.

#### Postconditions

None.

#### Particularities and Limitations

- > Call context: task level
- > Not re-entrant
- > Synchronous

### 5.4.2.2 FrTrcv\_30\_Tja1080\_GetTransceiverMode: Get the current Transceiver mode

#### FrTrcv\_30\_Tja1080\_GetTransceiverMode

##### Prototype

```
Std_ReturnType FrTrcv_30_Tja1080_GetTransceiverMode
(
    const uint8 FrTrcv_TrcvIdx,
    FrTrcv_TrcvModeType *FrTrcv_TrcvModePtr
);
```

##### Parameters [in/out/both]

FrTrcv_TrcvIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.
FrTrcv_TrcvModePtr [out]	This parameter describes the current Transceiver mode. It can have one of the following values: <ul style="list-style-type: none"><li>■ FRTRCV_TRCVMODE_NORMAL</li><li>■ FRTRCV_TRCVMODE_STANDBY</li><li>■ FRTRCV_TRCVMODE_SLEEP</li><li>■ FRTRCV_TRCVMODE_RECEIVEONLY</li></ul>

##### Return code

Std_ReturnType	The service returns E_NOT_OK if the Transceiver status could not be determined, otherwise E_OK is returned.
----------------	-------------------------------------------------------------------------------------------------------------

##### Service ID

Service ID	5
------------	---

##### Functional Description

This service determines the current Transceiver mode.

##### Preconditions

The Transceiver Driver module must be initialized.

##### Postconditions

None.

##### Particularities and Limitations

- > Call context: task level
- > Not re-entrant
- > Synchronous

### 5.4.2.3 FrTrcv\_30\_Tja1080\_GetTransceiverWUReason: Get the wake up reason

#### FrTrcv\_30\_Tja1080\_GetTransceiverWUReason

Prototype	
<pre>Std_ReturnType FrTrcv_30_Tja1080_GetTransceiverWUReason (     const uint8 FrTrcv_TrcvIdx,     FrTrcv_TrcvWUReasonType *FrTrcv_TrcvWUReasonPtr );</pre>	
Parameters [in/out/both]	
FrTrcv_TrcvIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.
FrTrcv_TrcvWUReasonPtr [out]	This parameter contains the wake up reason of the last wake-up event. It can have one of the following values: <ul style="list-style-type: none"> <li>■ FRTRCV_WU_POWER_ON</li> <li>■ FRTRCV_WU_BY_BUS</li> <li>■ FRTRCV_WU_INTERNALLY</li> </ul>
Return code	
Std_ReturnType	The service returns E_NOT_OK if the wake up reason could not be determined, otherwise E_OK is returned.
Service ID	
Service ID	6
Functional Description	
This service determines the wake up reason of the last wake up event. It can be used after an EcuM_SetWakeupEvent callback to determine if the wake up event happened locally or was triggered by the bus.	
Preconditions	
The Transceiver Driver module must be initialized.	
Postconditions	
None.	
Particularities and Limitations	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

#### 5.4.2.4 FrTrcv\_30\_Tja1080\_ClearTransceiverWakeup: Clear pending wake up events

##### FrTrcv\_30\_Tja1080\_ClearTransceiverWakeup

Prototype	
<pre>Std_ReturnType FrTrcv_30_Tja1080_ClearTransceiverWakeup( const uint8 FrTrcv_TrcvIdx );</pre>	
Parameters [in/out/both]	
FrTrcv_TrcvIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.

Return code	
Std_ReturnType	The service returns E_NOT_OK if wake up events could not be cleared, otherwise E_OK is returned.
Service ID	
Service ID	12
Functional Description	
This service clears pending wake up events. Furthermore the wake up reason is reset to FRTRCV_WU_RESET.	
Preconditions	
The Transceiver Driver module must be initialized.	
Postconditions	
None.	
Particularities and Limitations	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

#### 5.4.2.5 FrTrcv\_30\_Tja1080\_DisableTransceiverBranch: Disable an individual branch

##### FrTrcv\_30\_Tja1080\_DisableTransceiverBranch

Prototype	
Std_ReturnType FrTrcv_30_Tja1080_DisableTransceiverBranch( const uint8 FrTrcv_TrcvIdx, uint8 FrTrcv_BranchIdx );	
Parameters [in/out/both]	
FrTrcv_TrcvIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.
FrTrcv_BranchIdx [in]	This zero based index identifies the Transceiver branch. On Transceivers without branches this API call will be ignored.
Return code	
Std_ReturnType	The service returns E_NOT_OK.
Service ID	
Service ID	15
Functional Description	
This service disables individual branches. As the Tja1080/81 does not support branches this call is ignored and will always return E_NOT_OK.	
Preconditions	
The Transceiver Driver module must be initialized.	
Postconditions	
None.	

### Particularities and Limitations

- > Call context: task level
- > Not re-entrant
- > Synchronous

#### 5.4.2.6 FrTrcv\_30\_Tja1080\_EnableTransceiverBranch: Enable an individual branch

##### FrTrcv\_30\_Tja1080\_EnableTransceiverBranch

### Prototype

```
Std_ReturnType FrTrcv_30_Tja1080_EnableTransceiverBranch( const uint8
FrTrcv_TrcvIdx, uint8 FrTrcv_BranchIdx );
```

### Parameters [in/out/both]

FrTrcv_TrcvIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.
FrTrcv_BranchIdx [in]	This zero based index identifies the Transceiver branch. On Transceivers without branches this API call will be ignored.

### Return code

Std_ReturnType	The service returns E_OK.
----------------	---------------------------

### Service ID

Service ID	16
------------	----

### Functional Description

This service enables individual branches. As the Tja1080/81 does not support branches this call is ignored and will always return E\_OK.

### Preconditions

The Transceiver Driver module must be initialized.

### Postconditions

None.

### Particularities and Limitations

- > Call context: task level
- > Not re-entrant
- > Synchronous

#### 5.4.2.7 FrTrcv\_30\_Tja1080\_GetTransceiverError: Read current Transceiver error

##### FrTrcv\_30\_Tja1080\_GetTransceiverError

### Prototype

```
Std_ReturnType FrTrcv_30_Tja1080_GetTransceiverError ( const uint8
FrTrcv_TrcvIdx, uint8 FrTrcv_BranchIdx, uint32* FrTrcv_BusErrorState );
```

Parameters [in/out/both]	
FrTrcv_TrcvIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.
FrTrcv_BranchIdx [in]	This zero based index identifies the Transceiver branch. On Transceivers without branches this parameter will be ignored.
FrTrcv_BusErrorState [out]	Pointer to variable where the error status word will be stored.
Return code	
Std_ReturnType	The service returns E_OK if the function call was successful, otherwise E_NOT_OK.
Service ID	
Service ID	8
Functional Description	
This service can be used to return detailed error information.	
Preconditions	
The Transceiver Driver module must be initialized.	
Postconditions	
None.	
Particularities and Limitations	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

## 5.5 Services used by FlexRay Transceiver Driver

In the following table services provided by other components, which are used by the FlexRay Transceiver Driver are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
DET (optional)	Det_ReportError
DEM	Dem_SetEventStatus
ECU Manager	EcuM_SetWakeupEvent
Dio	Dio_WriteChannel Dio_ReadChannel

Table 5-3 Services used by the FlexRay Transceiver Driver

## 5.6 Callback Functions

This chapter describes the callback functions that are implemented by the FlexRay Transceiver Driver and can be invoked by other modules. The prototypes of the callback

functions are provided in the header file `FrTrcv_30_Tja1080_Cbk.h` by the FlexRay Transceiver Driver.

### 5.6.1 FrTrcv\_30\_Tja1080\_CheckWakeupByTransceiver

Prototype	
<pre>void FrTrcv_30_Tja1080_CheckWakeupByTransceiver( const uint8 FrTrcv_TrvcIdx );</pre>	
Parameter	
FrTrcv_TrvcIdx [in]	This zero based index identifies the Transceiver within the context of the Transceiver driver to which the API call has to be applied.
Return code	
void	-
Functional Description	
Callback to trigger wake up detection in case of an interrupt or non-periodically. If the component is not initialized this service will not generate a DET call. In this case the function will simply return.	
Particularities and Limitations	
<ul style="list-style-type: none"> <li>■ Particularities, limitations, post-conditions, pre-conditions</li> </ul>	
Expected Caller Context	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

## 5.7 Callout Functions

This chapter describes the user callout functions that are called by the FlexRay Transceiver Driver and must be implemented by the user.

### 5.7.1 Appl\_FrTrcv\_30\_Tja1080\_Wait

Prototype	
<pre>void Appl_FrTrcv_30_Tja1080_Wait( uint8 TimerIndex );</pre>	
Parameter	
TimerIndex [in]	This zero based index identifies the Timer which shall be used. kFrTrcv_30_Tja1080_delay_tEN_half: Delay time of tEN/2 selected kFrTrcv_30_Tja1080_delay_tDet: Delay time of tDet(EN) selected
Return code	
void	-
Functional Description	
Mandatory call out to delay a certain time in order to provide the Transceiver Driver with a deterministic timing.	

### Particularities and Limitations

- Particularities, limitations, post-conditions, pre-conditions

### Expected Caller Context

- > Call context: task level
- > Not re-entrant
- > Synchronous

## 5.7.2 Appl\_FrTrcv\_30\_Tja1080\_Dio\_WriteChannel

### Prototype

```
void Appl_FrTrcv_30_Tja1080_Dio_WriteChannel( Dio_ChannelType channel,
Dio_LevelType value );
```

### Parameter

channel [in]	This channel identifier as configured in the AUTOSAR DIO component.
value [in]	Either STD_HIGH or STD_LOW

### Return code

void	-
------	---

### Functional Description

Optional call out to set the selected DIO pin to high or low.

### Particularities and Limitations

- Particularities, limitations, post-conditions, pre-conditions

### Expected Caller Context

- > Call context: task level
- > Not re-entrant
- > Synchronous

## 5.7.3 Appl\_FrTrcv\_30\_Tja1080\_Dio\_ReadChannel

### Prototype

```
Dio_LevelType Appl_FrTrcv_30_Tja1080_Dio_ReadChannel( Dio_ChannelType
channel );
```

### Parameter

channel [in]	This channel identifier as configured in the AUTOSAR DIO component.
--------------	---------------------------------------------------------------------

### Return code

Dio_LevelType	Either STD_HIGH or STD_LOW depending on dio pin level.
---------------	--------------------------------------------------------

### Functional Description

Optional call out to get the status of the selected DIO pin.

### Particularities and Limitations

- Particularities, limitations, post-conditions, pre-conditions



#### Expected Caller Context

- > Call context: task level
- > Not re-entrant
- > Synchronous

### 5.7.4 Appl\_FrTrcv\_30\_Tja1080\_DisableNotification

#### Prototype

```
void Appl_FrTrcv_30_Tja1080_DisableNotification( uint8 Icu_Channel );
```

#### Parameter

Icu_Channel [in]	This channel identifies the respective ICU channel.
------------------	-----------------------------------------------------

#### Return code

void	-
------	---

#### Functional Description

Optional call out to disable Transceiver interrupts.

#### Particularities and Limitations

- Particularities, limitations, post-conditions, pre-conditions

#### Expected Caller Context

- > Call context: task level
- > Not re-entrant
- > Synchronous

### 5.7.5 Appl\_FrTrcv\_30\_Tja1080\_EnableNotification

#### Prototype

```
void Appl_FrTrcv_30_Tja1080_EnableNotification( uint8 Icu_Channel );
```

#### Parameter

Icu_Channel [in]	This channel identifies the respective ICU channel.
------------------	-----------------------------------------------------

#### Return code

void	-
------	---

#### Functional Description

Optional call out to enable Transceiver interrupts.

#### Particularities and Limitations

- Particularities, limitations, post-conditions, pre-conditions

#### Expected Caller Context

- > Call context: task level
- > Not re-entrant
- > Synchronous

### 5.7.6 Appl\_FrTrcv\_30\_Tja1080\_ReportErrorStatusPreFailed

Prototype	
<pre>void Appl_FrTrcv_30_Tja1080_ReportErrorStatusPreFailed( Dem_EventIdType eventId );</pre>	
Parameter	
eventId [in]	Id of the configured DEM error.
Return code	
void	-
Functional Description	
Optional call out to notify the DEM about a detected Transceiver error.	
Particularities and Limitations	
■ Particularities, limitations, post-conditions, pre-conditions	
Expected Caller Context	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

### 5.7.7 Appl\_FrTrcv\_30\_Tja1080\_ReportErrorStatusPrePassed

Prototype	
<pre>void Appl_FrTrcv_30_Tja1080_ReportErrorStatusPrePassed( Dem_EventIdType eventId );</pre>	
Parameter	
eventId [in]	Id of the configured DEM error.
Return code	
void	-
Functional Description	
Optional call out to notify the DEM about a healed Transceiver error (i.e. no error was detected).	
Particularities and Limitations	
■ Particularities, limitations, post-conditions, pre-conditions	
Expected Caller Context	
<ul style="list-style-type: none"> <li>&gt; Call context: task level</li> <li>&gt; Not re-entrant</li> <li>&gt; Synchronous</li> </ul>	

## 6 AUTOSAR Standard Compliance

### 6.1 Deviations

#### 6.1.1 SetTransceiverMode validation

According to AUTOSAR [FRTRCV276] the API `FrTrcv_SetTransceiverMode` shall verify if the requested mode is successfully set. This is currently not supported as this can not be verified with DIO transceivers, i.e. there is no feedback from the Transceiver device.

#### 6.1.2 Re-Initialization

According to AUTOSAR [FRTRCV437] the initialization process shall be restarted if access to the transceiver fails. This is currently not supported as access to DIO can not fail.

#### 6.1.3 Error detection during Init

According to AUTOSAR [FRTRCV390] it shall be configurable whether the `Init` function detects an error. This is not supported. It is only possible to enable this error detection capability globally when `Diagnosis` is enabled.

### 6.2 Additions/ Extensions

#### 6.2.1 Transceiver Diagnosis

The Transceiver diagnosis feature is according to AUTOSAR 4 and optional. If enabled errors provided by the Transceiver diagnosis feature will be reported to the DEM. The detailed error reason can be requested by calling `FrTrcv_GetTransceiverError`.

#### 6.2.2 User Dio functions

When the AUTOSAR Dio component is too slow for the software generated timing, a user call out can be enabled instead.

### 6.3 Limitations

#### 6.3.1 L001: Clock Frequency

The current version of the FlexRay Transceiver Driver is only capable of working at a certain clock frequency. This has to be taken into account, if for example the PLL is not running after start up. In this scenario, the cyclic `FrTrcv_30_Tja1080_MainFunction` of the Transceiver Driver can only be used if the CPU is running with nominal clock frequency. The reason for this limitation is the timing requirements for accessing the Transceiver which are defined by software.

## 7 Glossary and Abbreviations

### 7.1 Glossary

Term	Description
EAD	Embedded Architecture Designer; generation tool for MICROSAR components
GENy	Generation tool for CANbedded and MICROSAR components

Table 7-1 Glossary

### 7.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DIO	Digital Input Output
EAD	Embedded Architecture Designer
ECU	Electronic Control Unit
FrTrcv	FlexRay Transceiver Driver
HIS	Hersteller Initiative Software
ICU	Input Capture Unit
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
Platform	Hardware including Host and Communication Controller (might also be integrated in Host) on which the communication stack is implemented.
RTE	Runtime Environment
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 7-2 Abbreviations

## 8 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

**[www.vector-informatik.com](http://www.vector-informatik.com)**