MICROSAR Fr ERay

Technical Reference

Communication Controller E-Ray CANoe

Version 1.38

Authors	Juergen Schaeffer, Sebastian Gärtner, Sebastian Schmar, Mario Kunz, Roland Hocke
Status	Released

1 Document Information

1.1 History

Author	Date	Version	Remarks
Roland Hocke	2013-05-14	1.34	Creation and content copy from MSR3 document
	2013-06-26	1.35	Remove obsolete MTS
Roland Hocke	2014-01-29	1.37	Buffer alignment hints

Table 1-1 History of the document

1.2 Reference Documents

No.	Title	Version
[1]	AUTOSAR_SWS_DevelopmentErrorTracer.pdf	3.2.0
[2]	AUTOSAR_SWS_DiagnosticEventManager.pdf	4.2.0
[3]	TechnicalReference_Asr_Fr.pdf	1.0 or later
[4]	E-Ray_Errata_Sheet_20100215.pdf and later	REL20100215 and later
[5]	TechnicalReference_Asr_SchM.pdf	2.5 or later

Table 1-2 Reference documents

1.3 Scope of the Document

This technical reference describes the specific use of the FlexRay CANoeEmu driver software. It supplements the general FlexRay driver technical reference [3].

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

Illustrations

Tables

2 Component History

Component Version	New Features
3.0	Initial version
3.3	ISR routines in separate file
3.9	Fifo added
3.11	Read and verification of the FlexRay configuration
3.13	Added Fr_GetChannelStatus,Fr_GetClockCorrection, Fr_GetSyncFrameList, Fr_DisableLPdu and Fr_ReconfigLPdu
3.14	Optimization

Table 2-1 Component history

3 Introduction

This document describes the functionality, API and configuration of the AUTOSAR BSW module Fr ERay that is specific for CANoe with CANoeEmu driver. The basic functionality API architecture is already described within the document [3].

4 Functional Description

Please refer to [3].

4.1 Error Handling

4.1.1 Development Error Reporting

Please refer to [3].

The additionally errors of this platform reported to DET are described in the following table:

Error Code		Description
0x70	FR_E_INV_CANOE_CLUSTER_NAME	parameter “CANoe Cluster Name” does not match the name of the remaining bus

Table 4-1 Errors reported to DET

5 Integration

This chapter gives necessary information for the integration of the MICROSAR Fr ERay into an application environment of an ECU. It describes ERay specific issues in addition to [3].

5.1 Scope of Delivery

The delivery of the Fr ERay contains the files which are described in the chapters 5.1.1 and 5.1.2:

5.1.1 Static Files

File Name	Description
Fr_ERay.h	This file replaces the file Fr_<CC>.h listed in the document [3].

Table 5-1 Static files

5.1.2 Dynamic Files

No additional dynamic files. Please refer to [3].

5.2 Compiler Abstraction and Memory Mapping

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

Compiler Abstraction Definitions / Memory Mapping Sections	[Abstraction Name]	FR_CODE	FR_VAR_NOINIT	FR_VAR_NOINIT_FAST	FR_VAR_FRM	FR_CONST	FR_PBCFG	FR_APPL_CODE	FR_CODE_ISR
[Section Name]									
FR_START_SEC_CODE FR_STOP_SEC_CODE		■							
FR_START_SEC_VAR_NOINIT_UNSPECIFIED FR_STOP_SEC_VAR_NOINIT_UNSPECIFIED			■						
FR_START_SEC_VAR_FAST_NOINIT_UNSPECIFIED FR_STOP_SEC_VAR_FAST_NOINIT_UNSPECIFIED				■					

FR_START_SEC_CONST_UNSPECIFIED FR_STOP_SEC_CONST_UNSPECIFIED						■	■		
FR_START_SEC_PBCFG_ROOT FR_ STOP _SEC_PBCFG_ROOT							■		
FR_START_SEC_PBCFG FR_STOP_SEC_PBCFG							■		
FR_START_SEC_CONST_32BIT FR_STOP_SEC_CONST_32BIT						■	■		
FR_START_SEC_CONST_16BIT FR_STOP_SEC_CONST_16BIT						■	■		
FR_START_SEC_CONST_8BIT FR_STOP_SEC_CONST_8BIT						■	■		
FR_APPL_START_SEC_CODE FR_APPL_STOP_SEC_CODE								■	
FR_START_SEC_CODE_ISR FR_STOP_SEC_CODE_ISR									■

Table 5-2 Compiler abstraction and memory mapping

Caution
Please ensure that the define FR_PBCFG get the same value as FR_ CONST in configuration type “Pre-compile Configuration” or “Link-time Configuration”

5.3 Critical Sections

To ensure data consistency and a correct function of the Fr ERay the exclusive area `FR_EXCLUSIVE_AREA_0` has to be provided during the integration.

Considering the timing behavior of your system (e.g. depending on the CPU load of your system, priorities and interruptibility of interrupts and OS tasks and their jitter and delay times) the integrator has to choose and configure a critical section solution in such way that it is ensured that the API functions do not interrupt each other. You can find a set of rules below which describes whether an exclusive area is needed by the Fr ERay or not.

The `FR_EXCLUSIVE_AREA_0` has to be used if at least one of the following rules hold true:

- It is possible that the function `Fr_TransmitTxLPdu` is interrupted by the function `Fr_TransmitTxLPdu` itself. E.g. at execution of method `Fr_TransmitTxLPdu`, that was triggered by upper layer because the attribute `FrIfImmediate` is set, is interrupted by the job list execution of FrIf that contains a call of the method `Fr_TransmitTxLPdu`, or vice versa.
- At enabled feature “Reconfig LPdu Support” it is possible that the functions `Fr_DisableLPdu` or `Fr_ReconfigLPdu` are interrupted by the function `Fr_TransmitTxLPdu` and vice versa. E.g. at execution of the method `Fr_DisableLPdu` or `Fr_ReconfigLPdu` is interrupted by the job list execution of FrIf that contains a call of the method `Fr_TransmitTxLPdu`.

The recommended implementation for the `FR_EXCLUSIVE_AREA_0` of the component Fr ERay depends on the integration context of the job list execution.

The `FR_EXCLUSIVE_AREA_0` shall disable/enable the FlexRay timer interrupt or call `SuspendAllInterrupts()` and `ResumeAllInterrupts()` in case the job list execution is done at interrupt context to ensure data consistency.

The `FR_EXCLUSIVE_AREA_0` shall call `SuspendOSInterrupts()` and `ResumeOSInterrupts()` in case the case the job list execution is done at task context to ensure data consistency. Alternative the undesired task activation can be prevent by implement OS resource lock at the involved tasks.

The Fr ERay supports one of the following two alternatives as implementation for these exclusive areas depending on your questionnaire:

- The BSW Scheduler (refer to [5] for a detailed description)
- the Vector Standard Library (VStdLib)

The VStdLib offers the possibility of mapping the interrupt handling to OS services or to user defined functions. In the first case interrupt handling is done by the OS, in the second case the user has to take care by providing corresponding functions.

5.4 General Integration notes

A workaround for errata #27 in [4] is implemented in module FrSm. If no FrSm is used, the application has to implement a workaround.

5.5 Integration notes for CANoeEmu Driver

5.5.1 CANoe Version

Caution
The service pack for CANoe 7.0 shall be at least SP5. The service pack for CANoe 7.1 shall be at least SP3. The version of the used extended OSCAN library shall be at least Version 5.1.

5.5.2 Calculation of Timeout Loops

Hardware Loop with Cancellation is not supported as the software runs on PC.

5.5.3 Configuration in CANoe

The following settings have to be made in CANoe:

5.5.3.1 Hardware Configuration

You have to assure that the option “Tx Acknowledge” is active in the Hardware Configuration settings.

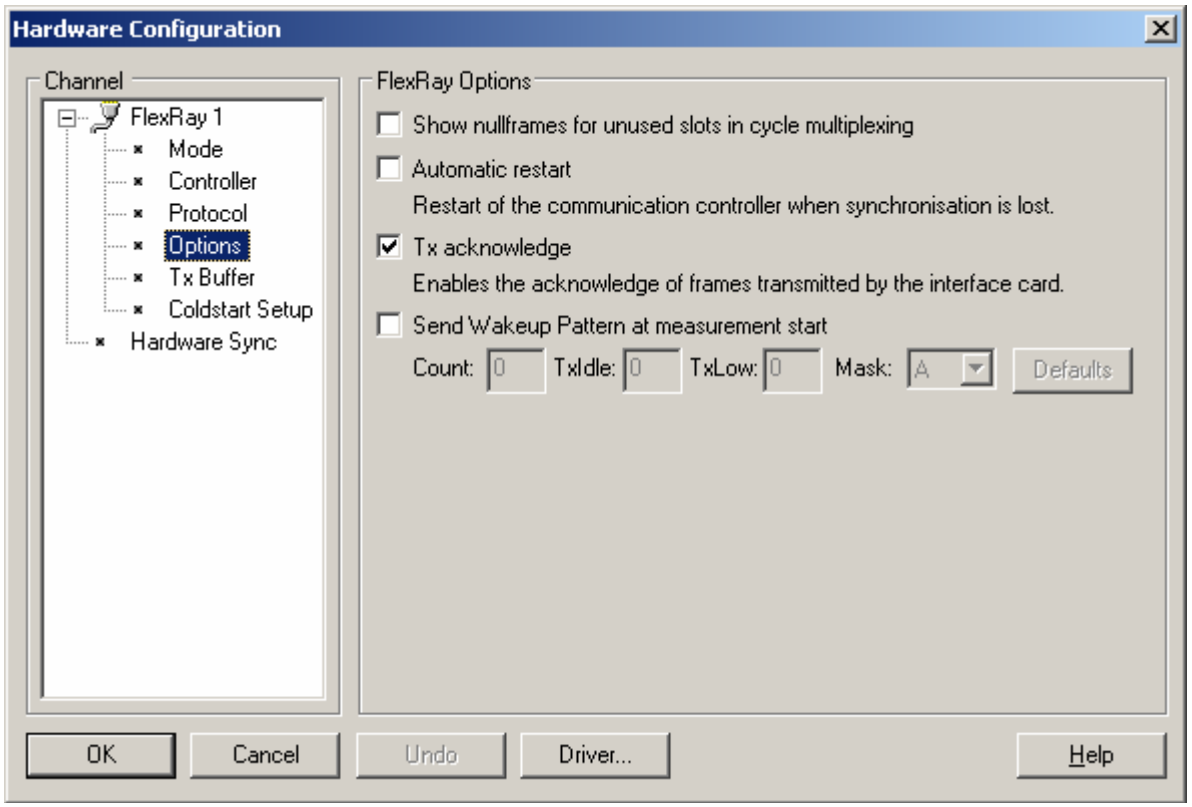

Figure 5-1 Options for using the FlexRay CANoeEmu in CANoe

5.5.3.2 Simulation Setup

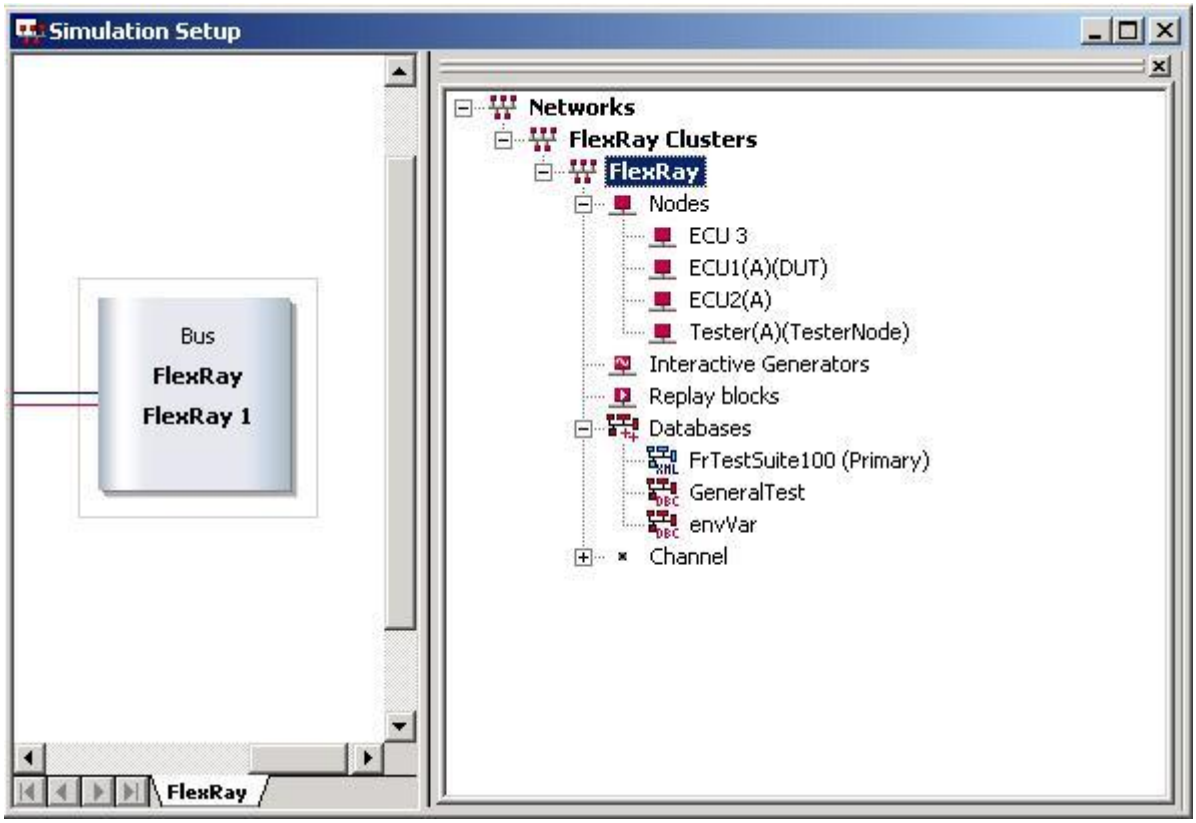

Figure 5-2 Cluster Name at Simulation Setup

The cluster name of at the remaining bus shall be the same as the configuration at “CANoe Cluster Name”. The error `FR_E_INV_CANOE_CLUSTER_NAME` is thrown in case the settings are not analog.

5.5.4 OS Configuration

The CANoeEmu flexRay driver does not use interrupts to detect the cycle start event or the expiration of the FlexRay timer validity event. All the events are detected with polling the current states respectively current time. The lowest sample time is equivalent to the OS TickTime.

	**Caution** Please ensure that the OS TickTime is short enough. For example the OS TickTime for a 5ms FlexRay cycle shall be 250µs.

6 API Description

6.1 Type Definitions

The types defined by the Fr ERay are described in this chapter.

The Fr ERay does not define specific type definitions.

6.2 Services provided by Fr ERay

No additional services are provided but the following hints shall be considered.

6.3 Services provided by Fr ERay and differs from standard

The Fr ERay API consists of services, which are realized by function calls. General services are described in [3]. The following services differ from standard behavior.

6.3.1 Fr_EnableAbsoluteTimerIRQ

There is no interrupt functionality used with CANoe.

6.3.2 Fr_DisableAbsoluteTimerIRQ

Please refer to 6.3.1

6.4 Services used by Fr ERay

General services are described in [3].

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

Component	API
osCAN Library	CANoeAPI_GetCurrentSimTime
osCAN Library	CANoeAPI_MapChannel
osCAN Library	CANoeAPI_RegisterFlexRayMessage
osCAN Library	CANoeAPI_ResetFlexRayCC
osCAN Library	CANoeAPI_ResetFlexRayCCAndSendWUP
osCAN Library	CANoeAPI_SendFlexRayMessage
osCAN Library	CANoeAPI_SetFlexRayCycleHandler
osCAN Library	CANoeAPI_SetFlexRayMessageHandler
osCAN Library	CANoeAPI_SetFlexRayMode
osCAN Library	CANoeAPI_SetFlexRaySlotHandler
osCAN Library	CANoeAPI_SetFlexRayStatusHandler
osCAN Library	CANoeAPI_SetFlexRayTimerHandler

osCAN Library	CANoeAPI_SetFlexRayPOCStateHandler
osCAN Library	CANoeAPI_SetRequestedFlexRaySlotIDs
osCAN Library	CANoeAPI_GetSimulationMode

Table 6-1 Services used by Fr ERay

Depending on configuration services used by Fr ERay are listed in next table. For details about prototype and functionality refer to the documentation of the providing component.

Configuration/Precondition	Component	API
Application callback at timer 0 interrupt and BSM FrIf is used	FrIf	FrIf_JobListExec_0

Table 6-2 Configuration dependent services used by Fr ERay

6.5 Callback Functions

With CANoe Emu Fr defines a callout function in PostBuild configuration variant. The declaration of the callout function is provided by the Fr BSW module. It is the integrator's task to provide the corresponding function definitions. The definitions of the callouts can be adjusted to the system's needs. The callout function is only needed for postbuild configuration. It is described in the following table:

6.5.1.1 ApplFr_ GetFrConfigPtr

Prototype	
`FUNC(void, FR_APPL_CODE) ApplFr_GetFrConfigPtr(P2CONST(Fr_ConfigType, AUTOMATIC, FR_PBCFG) *Fr_ConfigPtr);`	
Parameter	
`ConfigPtr`	Pointer to used Fr_Config
Return code	
-	
Functional Description	
This method should return the pointer to the FlexRay driver configuration struct e.g. "Fr_Config". ’	
Particularities and Limitations	
■ Particularities, limitations, post-conditions, pre-conditions: Only needed in Postbuild configuration variant.	
Expected Caller Context	
■ It is called at start of measurement from CANoe before any OS or stack is initialized.	

Table 6-3 ApplFr_ GetFrConfigPtr

6.6 Configurable Interfaces

6.6.1 Notifications

There is no Fr ERay specific notification function available.

6.6.2 Callout Functions

There is no Fr ERay specific callout function available.

7 Configuration

In the Fr ERay the attributes can be configured with the following methods:

> Configuration in DaVinci Configurator 5; for a detailed description see 7.2.

7.1 Hardware Fifo

This platform does not support Hardware Fifo support. Therefore it cannot be configured.

7.2 Configuration with DaVinci Configurator 5

The Fr ERay is configured with the help of the configuration tool DaVinci Configurator 5. For general configuration information please refer to [3].

8 AUTOSAR Standard Compliance

8.1 Deviations

For general deviations please refer to [3] and [4].

8.2 Additions/ Extensions

Please refer to [3].

8.3 Limitations

Please refer to [3].

The following features are currently not supported by the driver:

- Returning of the sync frame list with the API `Fr_GetSyncFrameList`.
- Reconfiguration of the LPdu with the API `Fr_ReconfigLPdu` and `Fr_DisableLpdu`.
- Returning of the channel status with the API `Fr_GetChannelStatus`
- Returning of the offset and rate correction values with the API `Fr_GetClockCorrection`.
- Buffer reconfiguration. Because the count of message buffer is not limited.
- Network Management Vector
- Self diagnostic extension
- CC read back support
- Rx Stringent Checks do not work at Canoeemu.

8.3.1 Behavior of the FlexRay bus POC states

The FlexRay driver simulates the POC states for the upper layer software of the FlexRay-Stack. The POC states shall be reached after calls `Fr_StartCommunication`, `Fr_HaltCommunication`, etc.. The simulation of the POC states is not equivalent to the “real” state of the FlexRay bus.

9 Glossary and Abbreviations

9.1 Glossary

Term	Description
EAD	Embedded Architecture Designer; generation tool for MICROSAR components
Cfg 5	DaVinci Configurator 5. Configuration tool for Microsar 4

Table 9-1 Glossary

9.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CC	FlexRay Communication Controller
E-Ray	Specific implementation of a FlexRay Communication Controller
DBA	Direct Buffer Access
DEM	Diagnostic Event Manager
DET	Development Error Tracer
EAD	Embedded Architecture Designer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
PPort	Provide Port
RPort	Require Port
RTE	Runtime Environment
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 9-2 Abbreviations

10 Contact

Visit our website for more information on

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

www.vector-informatik.com