MCAL User Manual for Fr_17_Eray

32-bit TriCore™ AURIX™ TC3xx microcontroller

About this document

Scope and purpose

This User Manual is intended to enable users to integrate the Microcontroller Abstraction Layer (MCAL) software for the TriCore™ AURIX™ family of 32-bit microcontrollers.

This document describes responsibilities of integrator in-charge of integrating MCAL software with the basic software (BSW) stack. This document also provides detailed information on safety, configuration and functions along with examples of usage of significant features.

***Note**: Detailed information about package installation, safety and other generic information that are common across all modules are provided in MCAL User Manual General.*

Intended audience

This document is intended for anyone using the Fr_17_Eray module of the TC3xx MCAL software.

Document conventions

Table 1 Conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, table and figure captions, screen names, windows, dialog boxes, menus, sub-menus
Italics	Denotes variable(s) and reference(s)

`Courier`
 	
`New`
	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets

>	Indicates that a cascading sub-menu opens when you select a menu item
[cover parentID=<alpha numeric value>]	Used for traceability completeness. Reader should ignore these.

Reference documents

This User Manual should be read in conjunction with the following documents:

- AURIX™ TC3xx MCAL User Manual General
- Specification of FlexRay Driver, AUTOSAR_SWS_FlexRay_Driver, AUTOSAR Release 4.2.2
- Specification of FlexRay Driver, AUTOSAR_SWS_FlexRay_Driver, AUTOSAR Release 4.4.0

Table of contents

About this document . . . 1
Table of contents . . . 2
1 Fr_17_Eray driver . . . 7
1.1 User information . . . 7
1.1.1 Description . . . 7
1.1.2 Hardware-software mapping . . . 7
1.1.2.1 SRC: dependent hardware peripheral . . . 8
1.1.2.2 ERAY: primary hardware peripheral . . . 8
1.1.2.3 SCU: dependent hardware peripheral . . . 9
1.1.2.4 PORT: dependent hardware peripheral . . . 9
1.1.3 File structure . . . 9
1.1.3.1 C file structure . . . 9
1.1.3.2 Code generator plugin files . . . 11
1.1.4 Integration hints . . . 12
1.1.4.1 Integration with AUTOSAR stack . . . 13
1.1.4.2 Multicore and Resource Manager . . . 16
1.1.4.3 MCU support . . . 16
1.1.4.4 Port support . . . 16
1.1.4.5 DMA support . . . 20
1.1.4.6 Interrupt connections . . . 20
1.1.4.7 Example usage . . . 21
1.1.5 Key architectural considerations . . . 23
1.1.5.1 Buffer reconfiguration . . . 23
1.1.5.2 BSW scheduler mechanism . . . 24
1.1.5.3 Clock configuration . . . 24
1.1.5.4 Input channel selection . . . 25
1.1.5.5 Additional receive FIFO - related parameters . . . 25
1.1.5.6 Configuration parameter for timeout event . . . 25
1.1.5.7 Get absolute timer status . . . 25
1.1.5.8 Configuration parameter for the `Fr_GetNmVector` API . . . 25
1.1.5.9 User mode support . . . 25
1.1.5.10 Transmission Conflict Detection . . . 26
1.2 Assumptions of Use (AoU) . . . 27
1.3 Reference information . . . 28
1.3.1 Configuration interfaces . . . 28
1.3.1.1 Container: Fr . . . 28
1.3.1.2 Container: FrGeneral . . . 28
1.3.1.2.1 FrCtrlTestCount . . . 29
1.3.1.2.2 FrDevErrorDetect . . . 29

Table of contents

1.3.1.2.3 FrDisableLPduSupport . . . 30
1.3.1.2.4 FrEcucPartitionRef . . . 30
1.3.1.2.5 FrExtendedLPduReporting . . . 31
1.3.1.2.6 FrIndex . . . 31
1.3.1.2.7 FrInitApiMode . . . 32
1.3.1.2.8 FrNmVectorEnable . . . 32
1.3.1.2.9 FrNumCtrlSupported . . . 33
1.3.1.2.10 FrPrepareLPduSupport . . . 33
1.3.1.2.11 FrReconfigLPduSupport . . . 34
1.3.1.2.12 FrRunTimeErrorDetect . . . 34
1.3.1.2.13 FrRxStringentCheck . . . 35
1.3.1.2.14 FrRxStringentLengthCheck . . . 35
1.3.1.2.15 FrTimeoutDurationFactor . . . 36
1.3.1.2.16 FrTxConflictDetection . . . 37
1.3.1.2.17 FrVersionInfoApi . . . 37
1.3.1.3 Container: FrMultipleConfiguration . . . 38
1.3.1.4 Container: FrController . . . 38
1.3.1.4.1 FrCtrlEcucPartitionRef . . . 38
1.3.1.4.2 FrCtrlIdx . . . 39
1.3.1.4.3 FrPAllowHaltDueToClock . . . 39
1.3.1.4.4 FrPAllowPassiveToActive . . . 40
1.3.1.4.5 FrPChannels . . . 40
1.3.1.4.6 FrPClusterDriftDamping . . . 41
1.3.1.4.7 FrPDecodingCorrection . . . 41
1.3.1.4.8 FrPDelayCompensationA . . . 42
1.3.1.4.9 FrPDelayCompensationB . . . 43
1.3.1.4.10 FrPExternalSync . . . 43
1.3.1.4.11 FrPFallBackInternal . . . 44
1.3.1.4.12 FrPKeySlotId . . . 44
1.3.1.4.13 FrPKeySlotOnlyEnabled . . . 45
1.3.1.4.14 FrPKeySlotUsedForStartup . . . 45
1.3.1.4.15 FrPKeySlotUsedForSync . . . 46
1.3.1.4.16 FrPLatestTx . . . 47
1.3.1.4.17 FrPMacroInitialOffsetA . . . 47
1.3.1.4.18 FrPMacroInitialOffsetB . . . 48
1.3.1.4.19 FrPMicroInitialOffsetA . . . 48
1.3.1.4.20 FrPMicroInitialOffsetB . . . 49
1.3.1.4.21 FrPMicroPerCycle . . . 49
1.3.1.4.22 FrPNmVectorEarlyUpdate . . . 50
1.3.1.4.23 FrPOffsetCorrectionOut . . . 50
1.3.1.4.24 FrPOffsetCorrectionStart . . . 51
1.3.1.4.25 FrPPayloadLengthDynMax . . . 52

Table of contents

1.3.1.4.26 FrPRateCorrectionOut 52
1.3.1.4.27 FrPSamplesPerMicrotick 53
1.3.1.4.28 FrPSecondKeySlotId 53
1.3.1.4.29 FrPTwoKeySlotMode 54
1.3.1.4.30 FrPWakeupChannel 54
1.3.1.4.31 FrPWakeupPattern 55
1.3.1.4.32 FrPdAcceptedStartupRange 55
1.3.1.4.33 FrPdListenTimeout 56
1.3.1.4.34 FrPdMicrotick 57
1.3.1.4.35 FrRxInputSelectionA 57
1.3.1.4.36 FrRxInputSelectionB 58
1.3.1.5 Container: FrAbsoluteTimer 58
1.3.1.5.1 FrAbsTimerIdx 58
1.3.1.6 Container: FrControllerDemEventParameterRefs 59
1.3.1.6.1 FR_E_CTRL_TESTRESULT 59
1.3.1.7 Container: FrFifo 60
1.3.1.7.1 FrAdmitWithoutMessageId 60
1.3.1.7.2 FrBaseCycle 61
1.3.1.7.3 FrChannels 61
1.3.1.7.4 FrCycleRepetition 62
1.3.1.7.5 FrFifoDepth 62
1.3.1.7.6 FrFrameIdRejectionFilter 63
1.3.1.7.7 FrFrameIdRejectionFilterMask 63
1.3.1.7.8 FrMsgIdMask 64
1.3.1.7.9 FrMsgIdMatch 64
1.3.1.7.10 FrRejectNullFrames 65
1.3.1.7.11 FrRejectStaticSegment 65
1.3.1.8 Container: FrRange 66
1.3.1.8.1 FrRangeMax 66
1.3.1.8.2 FrRangeMin 66
1.3.1.9 Container: FrClockConfiguration 67
1.3.1.9.1 FrClockDivider 67
1.3.1.10 Container: CommonPublishedInformation 68
1.3.1.10.1 ArMajorVersion 68
1.3.1.10.2 ArMinorVersion 68
1.3.1.10.3 ArPatchVersion 69
1.3.1.10.4 ModuleId 69
1.3.1.10.5 Release 69
1.3.1.10.6 SwMajorVersion 70
1.3.1.10.7 SwMinorVersion 70
1.3.1.10.8 SwPatchVersion 71
1.3.1.10.9 VendorApiInfix 71

Table of contents

1.3.1.10.10 VendorId 72
1.3.2 Functions - Type definitions 72
1.3.2.1 Fr_SlotAssignmentType 72
1.3.2.2 Fr_17_Eray_ConfigType 73
1.3.2.3 Fr_POCStateType 73
1.3.2.4 Fr_SlotModeType 74
1.3.2.5 Fr_ErrorModeType 74
1.3.2.6 Fr_WakeupStatusType 74
1.3.2.7 Fr_StartupStateType 75
1.3.2.8 Fr_POCStatusType 75
1.3.2.9 Fr_TxLPduStatusType 76
1.3.2.10 Fr_RxLPduStatusType 76
1.3.2.11 Fr_ChannelType 77
1.3.2.12 FR_CIDX_[CONFIGPARAM_NAME] 77
1.3.2.13 FR_SLOTMODE_SINGLE 81
1.3.3 Functions - APIs 81
1.3.3.1 Fr_17_Eray_Init 81
1.3.3.2 Fr_17_Eray_ControllerInit 82
1.3.3.3 Fr_17_Eray_StartCommunication 83
1.3.3.4 Fr_17_Eray_AllowColdstart 84
1.3.3.5 Fr_17_Eray_AllSlots 85
1.3.3.6 Fr_17_Eray_HaltCommunication 86
1.3.3.7 Fr_17_Eray_AbortCommunication 86
1.3.3.8 Fr_17_Eray_SendWUP 87
1.3.3.9 Fr_17_Eray_SetWakeupChannel 88
1.3.3.10 Fr_17_Eray_GetPOCStatus 89
1.3.3.11 Fr_17_Eray_TransmitTxLPdu 90
1.3.3.12 Fr_17_Eray_TransmitTxLPdu 91
1.3.3.13 Fr_17_Eray_CancelTxLPdu 92
1.3.3.14 Fr_17_Eray_ReceiveRxLPdu 93
1.3.3.15 Fr_17_Eray_ReceiveRxLPdu 94
1.3.3.16 Fr_17_Eray_CheckTxLPduStatus 95
1.3.3.17 Fr_17_Eray_CheckTxLPduStatus 96
1.3.3.18 Fr_17_Eray_PrepareLPdu 97
1.3.3.19 Fr_17_Eray_ReconfigLPdu 98
1.3.3.20 Fr_17_Eray_DisableLPdu 100
1.3.3.21 Fr_17_Eray_GetGlobalTime 101
1.3.3.22 Fr_17_Eray_GetNmVector 102
1.3.3.23 Fr_17_Eray_GetNumOfStartupFrames 102
1.3.3.24 Fr_17_Eray_GetChannelStatus 103
1.3.3.25 Fr_17_Eray_GetClockCorrection 104
1.3.3.26 Fr_17_Eray_GetSyncFrameList 105

Table of contents

1.3.3.27 Fr_17_Eray_GetWakeupRxStatus 107
1.3.3.28 Fr_17_Eray_SetAbsoluteTimer 108
1.3.3.29 Fr_17_Eray_CancelAbsoluteTimer 109
1.3.3.30 Fr_17_Eray_EnableAbsoluteTimerIRQ 110
1.3.3.31 Fr_17_Eray_AckAbsoluteTimerIRQ 110
1.3.3.32 Fr_17_Eray_DisableAbsoluteTimerIRQ 111
1.3.3.33 Fr_17_Eray_GetAbsoluteTimerIRQStatus 112
1.3.3.34 Fr_17_Eray_GetVersionInfo 113
1.3.3.35 Fr_17_Eray_ReadCCConfig 114
1.3.4 Notifications and Callbacks 115
1.3.5 Scheduled functions 115
1.3.6 Interrupt service routines 115
1.3.7 Callout 115
1.3.8 Errors Handling 115
1.3.9 Deviations and limitations 116
1.3.9.1 Deviations 116
1.3.9.1.1 Software specification deviations 116
1.3.9.1.2 AMDC Violations 117
1.3.9.1.3 VSMD Violations 117
1.3.9.2 Limitations 121

Revision history 122

Disclaimer 124

1 Fr_17_Eray driver

1.1 User information

1.1.1 Description

The FlexRay (FR) driver abstracts the hardware related implementation details of specific FlexRay Communication Controllers (CC). The APIs provide abstract functional operations that are mapped to a sequence of hardware accesses depending on the actual implemented FR driver. Thus, it provides to the FlexRay Interface (FrIf) an access to FlexRay functionality independent of the underlying FlexRay CC hardware. Some of the functionalities provided by the FR driver are to configure the node as a coldstart node or non-coldstart node, initialize the controller through the protocol states so that the node can participate in the cluster, initialize and assign the message buffers to all the LPdus for transmission and reception, send wakeup signal to wakeup the cluster, send sync frames and startup frames when acting as a coldstart node to aid in cluster formation, and so on. The driver is delivered as a Post-Build variant.

1.1.2 Hardware-software mapping

This section describes the system view of the FlexRay driver and peripherals administered by it.

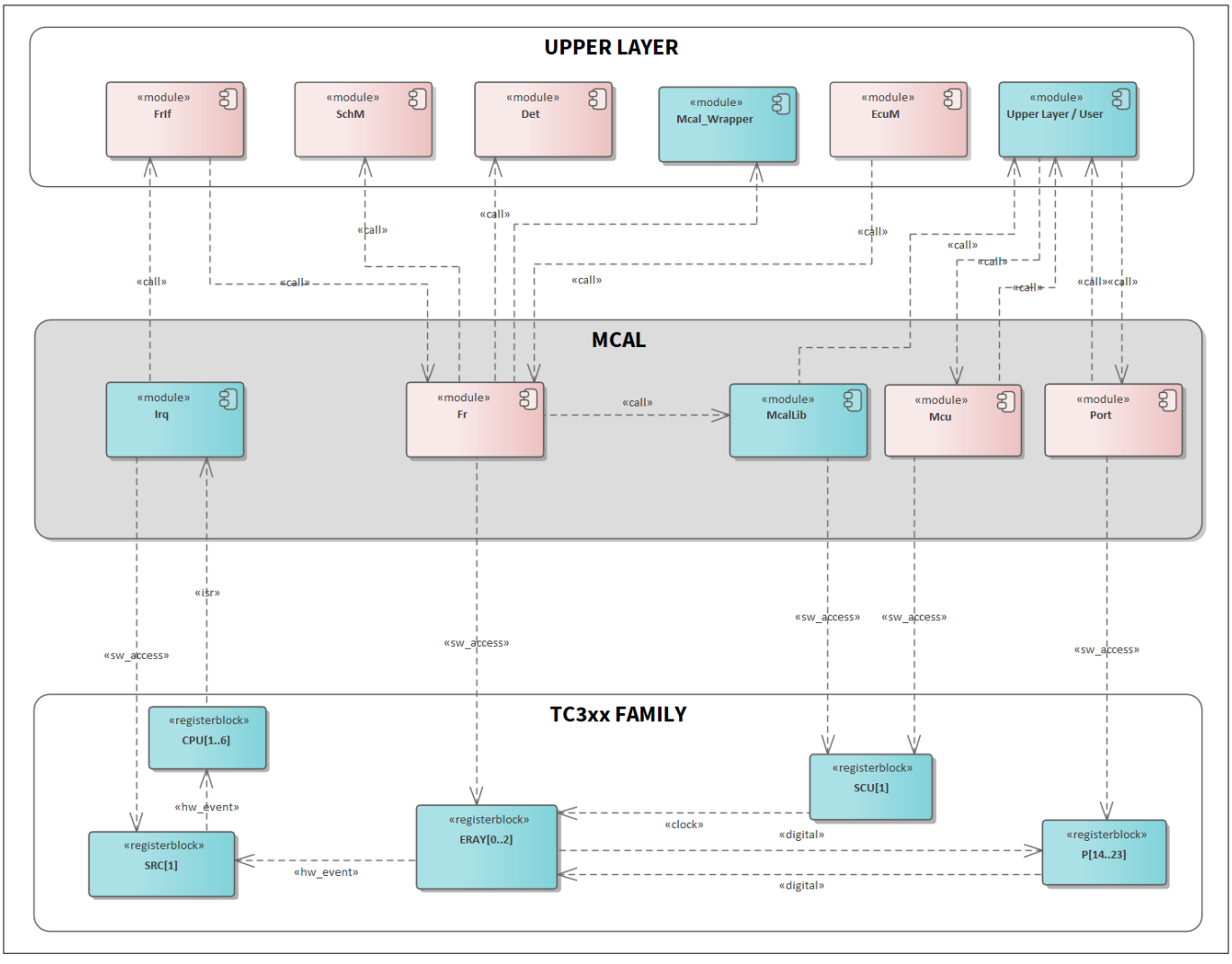

Figure 1 Mapping of hardware-software interfaces

1.1.2.1 SRC: dependent hardware peripheral

Hardware functional features

The FR driver depends on the interrupt router for raising an interrupt to the CPU based on the absolute timer hardware event.

Users of the hardware

The interrupt router is configured either by the IRQ driver or the user software.

Hardware diagnostic features

The SMU alarms configured for interrupt router are not monitored by the FR driver.

Hardware events

The interrupt events raised by the interrupt router are serviced by the CPU. The interrupt handlers are provided by the FrIf module, which must be invoked from the ISR. The FR driver does not provide any interrupt handlers.

1.1.2.2 ERAY: primary hardware peripheral

Hardware functional features

The FR driver uses the ERAY module for communication. The key hardware functional features used by the driver are:

- Baud rate of 10 MBit/s is supported for each channel
- Support of up to 128 Message Buffers based on number of configured frames
- Configuration of message buffers with different payload lengths
- Configuration of the receive FIFO based on FIFO rejection criteria. These criteria also include the rejection for NULL frames and / or static segment frames
- Configuration of each message buffer as receive buffer, transmit buffer or as part of receive FIFO
- Read / Write access to the header and data sections of the message buffers via Input and Output buffer
- Filtering of frames based on slot counter, cycle counter and channel
- Enable / disable and reconfiguration support of the absolute timer
- Node configuration as a sync node and also as a leading / following cold starter
- Communication channel selection - either Channel A / B or both
- Wakeup channel selection - either Channel A/ B to transmit wakeup pattern
- Support for Network Management

The unsupported features of the ERAY are:

- Relative timer
- Stop watch functionality

Users of the hardware

The FR driver exclusively utilizes the ERAY IP.

Hardware diagnostic features

The SMU alarms configured for the ERAY are not monitored by the FR driver.

Hardware events

The FR driver uses the following hardware events from the ERAY IP:

- TX flag upon transmission complete
- RX flag upon reception of data into the message buffer
- Error flags upon occurrence of errors during transmission and reception
- The interrupt service requests are not handled by the FR driver, but they are expected to be handled by the FlexRay Interface(FrIf) module.

1.1.2.3 SCU: dependent hardware peripheral

Hardware functional features

The FR driver depends on the SCU IP for the clock, ENDINIT and reset functionalities. The driver requires the fSPB, fSCLK and fERAY clock signals for functioning. The fCLC_ERAY is configured by the FlexRay driver. The fCLC_ERAY clock is used by the main protocol controller state machine and is derived from fSPB.

Users of the hardware

The SCU IP supplies clock for all the peripherals and the MCU driver is responsible for configuring the clock tree. To avoid conflicts due to simultaneous writes, update to all the ENDINIT protected registers is performed using the MCALLIB APIs.

Hardware diagnostic features

The SMU alarms configured for the SCU IP are not monitored by the FR driver.

Hardware events

Hardware events from the SCU are not used by the FR driver.

1.1.2.4 PORT: dependent hardware peripheral

Hardware functional features

TXDA, TXDB, RXDA, RXDB, TXENA and TXENB signals are routed to the ERAY through the port pads. These pins are configured and enabled through the PORT driver.

Users of the hardware

The port pads are configured by the PORT driver.

Hardware diagnostic features

Not applicable.

Hardware events

Hardware events from port pads are not used by the FR driver.

1.1.3 File structure

1.1.3.1 C file structure

This section provides details of the C files of the FR driver.

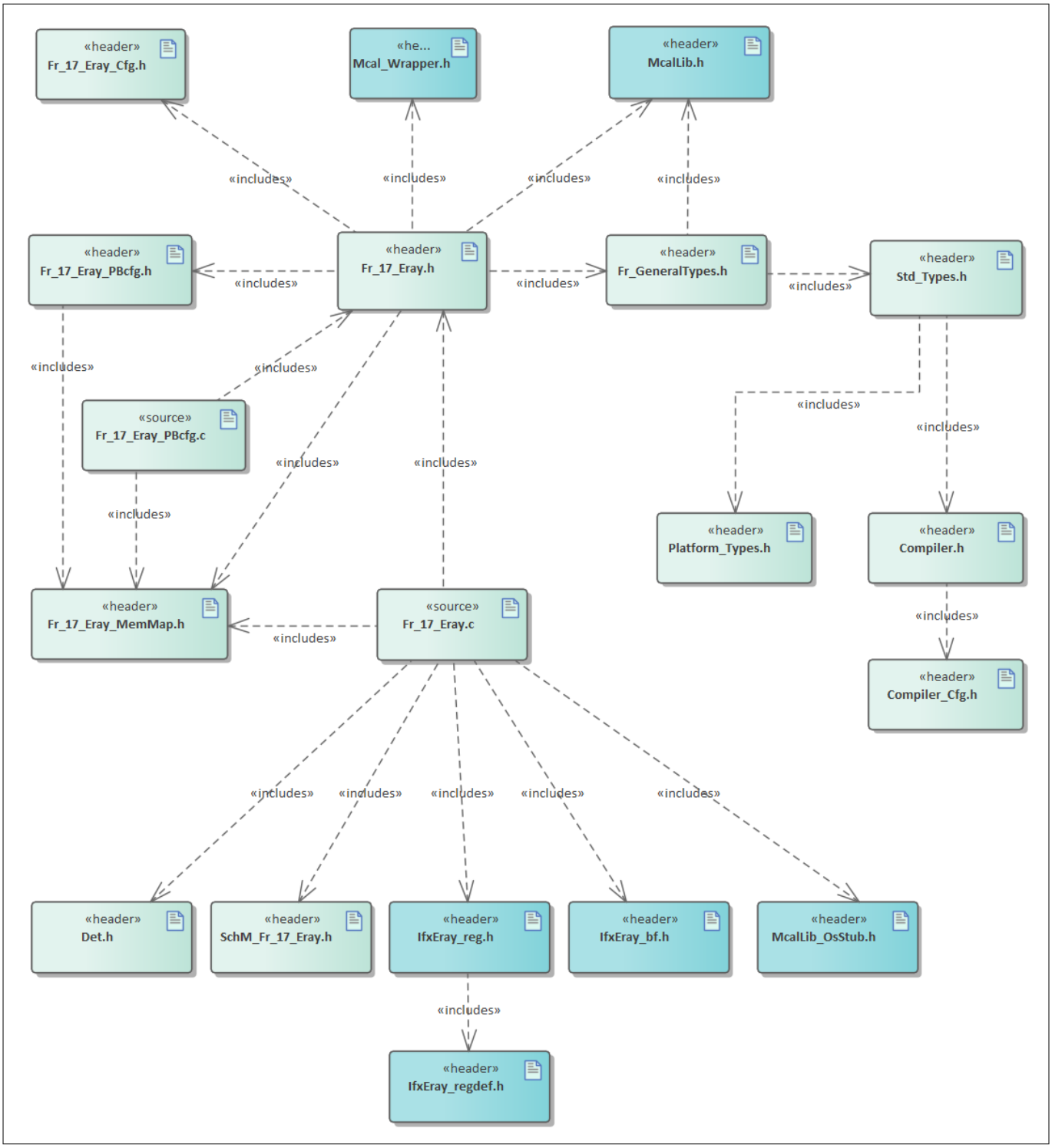

Figure 2 Fr_C_File_Structure-1.png

Table 2 C file structure

File name	Description
`Compiler.h`	Provides abstraction from compiler-specific keywords
`Compiler_Cfg.h`	Configuration header file for compiler abstraction

(table continues...)

Table 2 (continued) C file structure

File name	Description
Det.h	Provides the exported interfaces of Development Error Tracer
Fr_17_Eray.c	Contains the functionality of the FR driver
Fr_17_Eray.h	Contains macros, type definitions and function prototypes of the FR driver
Fr_17_Eray_Cfg.h	Contains driver Pre-compile configuration parameters
Fr_17_Eray_MemMap.h	Mapping of code and data (variables, constant variables) to specific memory sections
Fr_17_Eray_PBcfg.c	Contains driver post-build configuration parameters
Fr_17_Eray_PBcfg.h	File (generated) containing declaration of the post-build configuration data structures
Fr_GeneralTypes.h	Contains all types and constants that are shared among the AUTOSAR FlexRay modules Fr, FrIf and FrTrcv. Defines the macros that can be passed into API function Fr_ReadCCConfig as parameter Fr_ConfigParamIdx.
IfxEray_bf.h	SFR header file for ERAY
IfxEray_reg.h	SFR header file for ERAY
IfxEray_regdef.h	SFR header file for ERAY
McalLib.h	Static header file defining prototypes of data structure and APIs exported by the MCALLIB.
McalLib_OsStub.h	McalLib_OsStub.h provides macros to support user mode of Tricore. This shall be included by other drivers to call OS APIs.
Mcal_Wrapper.h	Provides the exported interfaces for Production Error and Runtime Development Errors. Implemented by default to include functions of Dem.h and Det.h files. This file can be modified by the user but function prototype is not user modifiable.
Platform_Types.h	Platform-specific type declaration file as defined by AUTOSAR
SchM_Fr_17_Eray.h	Contains data consistency mechanisms
Std_Types.h	Standard type declaration file as defined by AUTOSAR. It is independent of compiler or platform.

1.1.3.2 Code generator plugin files

This section provides details of the code generator plugin files of the FR driver.

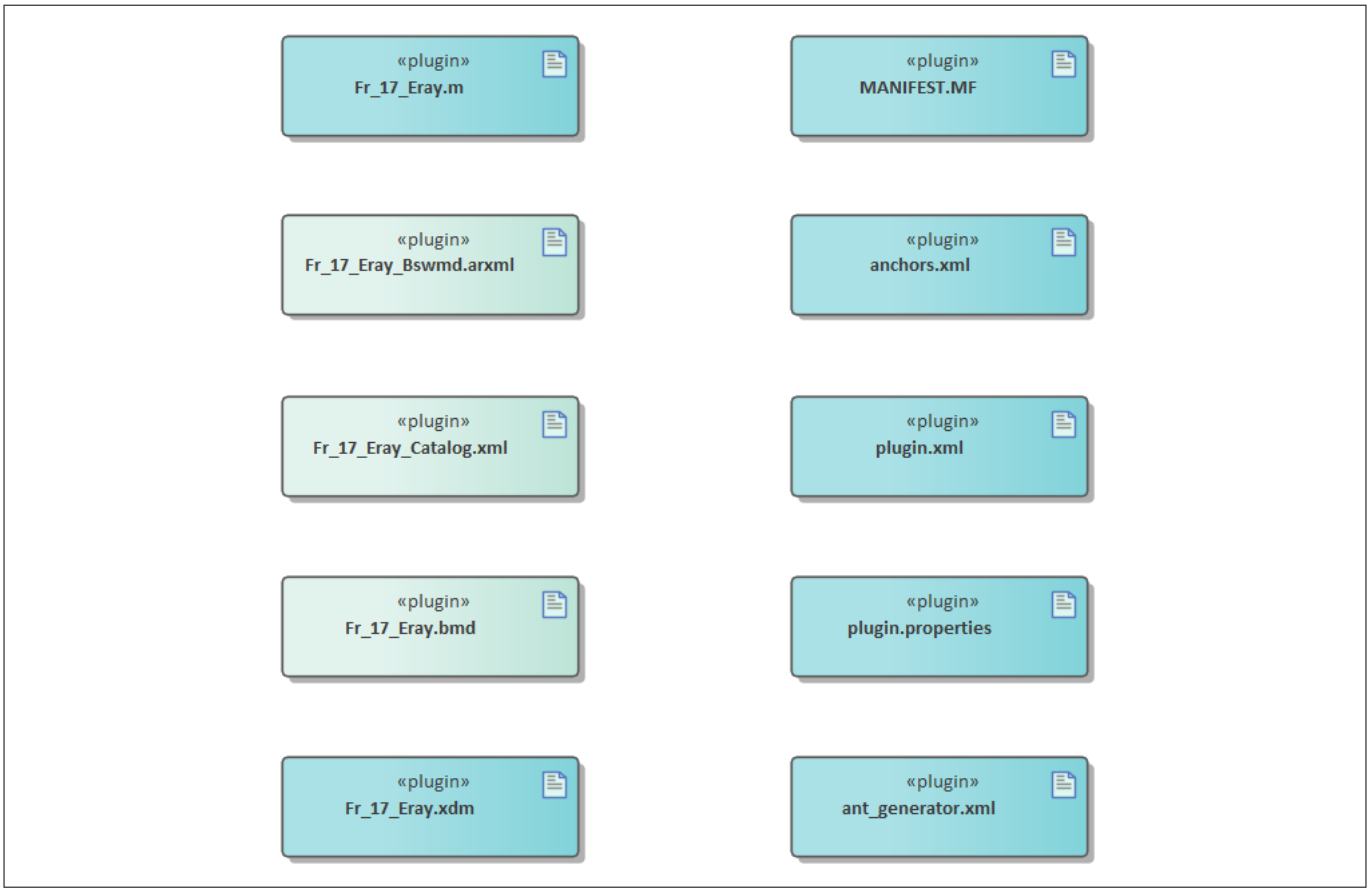

Figure 3 **Fr_Code_Generator_Plugin_Files-1.png**

Table 3 **Code generator plugin files**

File name	Description
`Fr_17_Eray.bmd`	AUTOSAR format XML data model schema file (for each device)
`Fr_17_Eray.m`	Code template macro file for the FR driver
`Fr_17_Eray.xdm`	Tresos format XML data model schema file
`Fr_17_Eray_Bswmd.arxml`	AUTOSAR format module description file
`Fr_17_Eray_Catalog.xml`	AUTOSAR format catalog file
`MANIFEST.MF`	Tresos plugin support file containing the meta-data for FR driver
`anchors.xml`	Tresos anchors support file for the FR driver
`ant_generator.xml`	Tresos support file to generate and rename multiple post-build configuration when using variation point feature
`plugin.properties`	Tresos plugin support file for the FR driver
`plugin.xml`	Tresos plugin support file for the FR driver

1.1.4 Integration hints

This section lists the key points that an integrator or user of the FlexRay driver must consider.

1.1.4.1 Integration with AUTOSAR stack

This section lists the modules, which are not part of the MCAL, but are required to integrate the FlexRay driver.

- **EcuM**

  The ECU Manager module is a part of the AUTOSAR stack that manages common aspects of ECU. Specifically, in the context of the MCAL, EcuM is used for initialization and deinitialization of the software drivers. The EcuM module provided in the MCAL package is a stub code and needs to be replaced with a complete EcuM module during the integration phase.

  Note: The FR driver does not provide any de-initialization API.

- **Memory mapping**

  Memory mapping is a concept from AUTOSAR that allows relocation of text, variables, constants and configuration data to user specific memory regions. To achieve this, all the relocatable elements of the driver are encapsulated in different memory-section macros. These macros are defined in the file `Fr_17_Eray_MemMap.h`.

  The `Fr_17_Eray_MemMap.h` file is provided in the MCAL package as a stub code. The integrator must place appropriate compiler pragmas within the memory-section macros. The pragmas ensure that the elements

are re-located to the correct memory region. A sample implementation listing the memory-section macros are shown as follows.

```
/**** GLOBAL RAM DATA ****/
#if defined FR_17_ERAY_START_SEC_VAR_CLEARED_QM_LOCAL_32
/*****User pragmas here for LMU*****/
#undef FR_17_ERAY_START_SEC_VAR_CLEARED_QM_LOCAL_32
#undef MEMMAP_ERROR

#elif defined FR_17_ERAY_STOP_SEC_VAR_CLEARED_QM_LOCAL_32
/*****User pragmas here for LMU*****/
#undef FR_17_ERAY_STOP_SEC_VAR_CLEARED_QM_LOCAL_32
#undef MEMMAP_ERROR

/**** CONFIG DATA -- PF[x] ****/
#elif defined FR_17_ERAY_START_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
/*****User pragmas here for PF[x]*****/
#undef FR_17_ERAY_START_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
#undef MEMMAP_ERROR

#elif defined FR_17_ERAY_STOP_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
/*****User pragmas here for PF[x]*****/
#undef FR_17_ERAY_STOP_SEC_CONFIG_DATA_QM_LOCAL_UNSPECIFIED
#undef MEMMAP_ERROR

/**** CODE -- PF[x] ****/
#elif defined FR_17_ERAY_START_SEC_CODE_QM_LOCAL
/*****User pragmas here for PF[x]*****/
#undef FR_17_ERAY_START_SEC_CODE_QM_LOCAL
#undef MEMMAP_ERROR
#elif defined FR_17_ERAY_STOP_SEC_CODE_QM_LOCAL
/*****User pragmas here for PF[x]*****/
#undef FR_17_ERAY_STOP_SEC_CODE_QM_LOCAL
#undef MEMMAP_ERROR

#endif

#if defined MEMMAP_ERROR
#error "Fr_17_Eray_MemMap.h, wrong pragma command"
#endif
```

- **DET**

  The DET module is a part of the AUTOSAR stack that handles all the development errors reported by the BSW modules. The FR driver reports all the development errors to the DET module through the `Det_ReportError()` API. The user of the FR driver must process all the errors reported to the DET module through the `Det_ReportError()` API.

  The `Det.h` and `Det.c` files are provided in the MCAL package as a stub code and needs to be replaced with a complete DET module during the integration phase.

- **Mcal_Wrapper**

  This Driver performs reporting of the Production and Runtime errors. The Handling of the reported errors shall be done by the user. The `Mcal_Wrapper_Det_ReportRuntimeError()` API,

`Mcal_Wrapper_Dem_ReportErrorStatus()` API and `Mcal_Wrapper_Dem_SetEventStatus()` API are provided in the `Mcal_Wrapper.h` and `Mcal_Wrapper.c` files as a stub code and can be updated by the integrator to handle the reported errors. The files `Mcal_Wrapper.h` and `Mcal_Wrapper.c` are user modifiable but the function prototype is not user modifiable and by default the Mcal_Wrapper function shall call AUTOSAR DEM and DET modules.

The user of the FR driver shall process all the Production errors (fail/pass) and Runtime errors reported to the Mcal_Wrapper module. The interface used for reporting Production error in AUTOSAR version 4.2.2 is `Mcal_Wrapper_Dem_ReportErrorStatus()` and for AUTOSAR version 4.4.0 is `Mcal_Wrapper_Dem_SetEventStatus()`, for reporting Runtime error is `Mcal_Wrapper_Det_ReportRuntimeError() API`. The `Mcal_Wrapper.h` and `Mcal_Wrapper.c` files are provided in the MCAL package as a stub code and can be replaced with user specific Production and Runtime error handling module/s during the integration phase.

- **SchM**

  The SchM module is a part of the RTE that manages the BSW Scheduler.

  In FR driver, in order to enter the READY state from CONFIG state, it is required to execute an unlock sequence before writing to the `SUCC1.CMD` field in the SUC Configuration Register 1. The write operation to `SUCC1.CMD` field has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (`LCK.CLK`). If this write sequence is pre-empted by other read or write accesses, the Communication Controller remains in CONFIG state and the sequence has to be repeated. The FR driver implements this write sequence within a critical section using the exclusive area defined in `SchM_Fr_17_Eray.c` in order to prevent the pre-emption of the sequence. The identified SchM section for FR driver is:

  - ConfigLockKey

  The `SchM_Fr_17_Eray.h` and `SchM_Fr_17_Eray.c` files are provided in the MCAL package as an example code and needs to updated by the integrator. The user must implement the SchM functions defined by the FR driver as **suspend / resume** of interrupts for the CPU on which the API is invoked. A sample implementation of the SchM functions is shown as follows:

```
/**** Sample implementation of SchM_Fr_17_Eray.c ****/
#include "Os.h"
#include "SchM_Fr_17_Eray.h"

/* Start of Critical Section */
/* Suspend CPU core interrupt */
void SchM_Enter_Fr_17_Eray_ConfigLockKey(void)
{
 SuspendAllInterrupts();
}

/* End of Critical Section */
/* Resume CPU core interrupt */
void SchM_Exit_Fr_17_Eray_ConfigLockKey(void)
{
 ResumeAllInterrupts();
}
```

- **Safety error**

  The FR driver does not report any safety errors.

- **Notifications and callbacks**

The FR driver does not provide any callbacks or notifications.

- **Operating system**

  The OS or the application must ensure correct type of service and interrupt priority is configured in the SR register. Enabling and disabling of interrupts must also be managed by the OS or application.

  OS files provided by the MCAL package are only an example code and must be updated by the integrator with the actual OS files for the desired function.

1.1.4.2 Multicore and Resource Manager

The FlexRay driver does not support execution on multiple cores simultaneously.

1.1.4.3 MCU support

The FlexRay driver is dependent on MCU driver for clock configuration. The initialization of FR driver must be started only after completing MCU initialization. The following must be considered while configuring the MCU driver in EB tresos:

- The ERAY peripheral clock (f_{ERAY}) must always be enabled and should be configured to 80 MHz.

  ERAY clock settings:

  - To enable the ERAY peripheral clock, the MCU parameter `McuErayClkEnable` must be enabled. This MCU parameter is found in EB Tresos tool in the following path:

  `/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuClockReferencePointConfig/McuPllDistributionSettingConfig/McuErayClkEnable`.

  - To configure the ERAY peripheral clock, the MCU parameter `McuErayFrequency` must be set to 80 MHz. This MCU parameter is found in EB Tresos tool in the following path:

  `/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuClockReferencePointConfig/McuPllDistributionSettingConfig/McuErayFrequency`.

1.1.4.4 Port support

The PORT driver configures the port pins of the entire microcontroller. The user must configure port pins used by the FR driver through the PORT configuration and initialize the port pins prior to invoking the FR initialization. The configuration of the PORT driver should be done based on the hardware connectivity between the microcontroller and the FlexRay transceiver chip.

Port configuration

- The port pin connected to the RXD pin must be selected using the `FrRxInputSelectionA` and `FrRxInputSelectionB` parameters within the `FrController` container. *Note: For FlexRay controller 1, the values `FR_RXSEL2` and `FR_RXSEL3` should not be selected as there are no port lines connected to the corresponding interface signals.*
- *The port pins connected to the TXD and TXEN pins needs to be configured in the PORT driver.*

Table 4 Connectivity of I/O signals for FR controller 0 Channel A - TC39x, TC38x, TC357, TC37x, TC337, TC365, TC367 and TC3E7 devices

Interface signals	*Port lines*	*I/O direction*	*Configuration parameter provided in FR driver*
RXD0A0	*P14.8*	*In*	*`FrRxInputSelectionA` - value `FR_RXSEL0`*

(table continues...)

Table 4 *(continued) Connectivity of I/O signals for FR controller 0 Channel A - TC39x, TC38x, TC357, TC37x, TC337, TC365, TC367 and TC3E7 devices*

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0A1	P11.9	In	`FrRxInputSelectionA` - value `FR_RXSEL1`
RXD0A2	P02.1	In	`FrRxInputSelectionA` - value `FR_RXSEL2`
RXD0A3	P14.1	In	`FrRxInputSelectionA` - value `FR_RXSEL3`
TXD0A	P02.0, P11.3, P14.10, P14.0	Out	Not applicable
TXEN0A	P02.4, P11.6, P14.9	Out	Not applicable

Table 5 *Connectivity of I/O signals for FR controller 0 Channel B - TC39x, TC38x, TC357, TC37x, TC337, TC365, TC367 and TC3E7 devices*

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0B0	P14.7	In	`FrRxInputSelectionB` - value `FR_RXSEL0`
RXD0B1	P11.10	In	`FrRxInputSelectionB` - value `FR_RXSEL1`
RXD0B2	P02.3	In	`FrRxInputSelectionB` - value `FR_RXSEL2`
RXD0B3	P14.1	In	`FrRxInputSelectionB` - value `FR_RXSEL3`
TXD0B	P02.2, P14.0, P14.5, P11.12	Out	Not applicable
TXEN0B	P02.5, P14.6, P14.9, P11.11, P11.6	Out	Not applicable

Table 6 *Connectivity of I/O signals for FR controller 1 Channel A - TC399 and TC389 devices*

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD1A0	P14.8	In	`FrRxInputSelectionA` - value `FR_RXSEL0`
RXD1A1	P01.1	In	`FrRxInputSelectionA` - value `FR_RXSEL1`
RXD1A2	No Connection	--	--
RXD1A3	No Connection	--	--
TXD1A	P01.12, P14.10	Out	Not applicable
TXEN1A	P01.14, P14.9	Out	Not applicable

Table 7 Connectivity of I/O signals for FR controller 1 Channel B - TC399 and TC389 devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD1B0	P14.7	In	`FrRxInputSelectionB` - value `FR_RXSEL0`
RXD1B1	P01.8	In	`FrRxInputSelectionB` - value `FR_RXSEL1`
RXD1B2	No Connection	--	--
RXD1B3	No Connection	--	--
TXD1B	P01.13, P14.5	Out	Not Applicable
TXEN1B	P02.15, P14.6	Out	Not Applicable

Table 8 Connectivity of I/O signals for FR controller 1 Channel A - TC397, TC397 ADAS, TC387 and TC3E7 devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD1A0	P14.8	In	`FrRxInputSelectionA` - value `FR_RXSEL0`
RXD1A1	No Connection	--	--
RXD1A2	No Connection	--	--
RXD1A3	No Connection	--	--
TXD1A	P14.10	Out	Not applicable
TXEN1A	P14.9	Out	Not applicable

Table 9 Connectivity of I/O signals for FR controller 1 Channel B - TC397, TC397 ADAS, TC387 and TC3E7 devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD1B0	P14.7	In	`FrRxInputSelectionB` - value `FR_RXSEL0`
RXD1B1	No Connection	--	--
RXD1B2	No Connection	--	--
RXD1B3	No Connection	--	--
TXD1B	P14.5	Out	Not Applicable
TXEN1B	P14.6	Out	Not Applicable

Table 10 Connectivity of I/O signals for FR controller 0 Channel A - TC356, TC332, TC333, TC334, TC336, TC364_LQFP and TC364_TQFP devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver

(table continues...)

Table 10 (continued) Connectivity of I/O signals for FR controller 0 Channel A - TC356, TC332, TC333, TC334, TC336, TC364_LQFP and TC364_TQFP devices

RXD0A0	No Connection	--	--
RXD0A1	P11.9	In	`FrRxInputSelectionA` - value `FR_RXSEL1`
RXD0A2	P02.1	In	`FrRxInputSelectionA` - value `FR_RXSEL2`
RXD0A3	P14.1	In	`FrRxInputSelectionA` - value `FR_RXSEL3`
TXD0A	P02.0, P11.3, P14.0	Out	Not Applicable
TXEN0A	P02.4, P11.6	Out	Not Applicable

Table 11 Connectivity of I/O signals for FR controller 0 Channel B - TC356, TC334, TC364_LQFP, TC364_TQFP and TC366 devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0B0	No Connection	--	--
RXD0B1	P11.10	In	`FrRxInputSelectionB` - value `FR_RXSEL1`
RXD0B2	P02.3	In	`FrRxInputSelectionB` - value `FR_RXSEL2`
RXD0B3	P14.1	In	`FrRxInputSelectionB` - value `FR_RXSEL3`
TXD0B	P02.2, P11.12, P14.0, P14.5	Out	Not Applicable
TXEN0B	P02.5, P11.6, P11.11,P14.6	Out	Not Applicable

Table 12 Connectivity of I/O signals for FR controller 0 Channel A - TC366 device

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0A0	P14.8	--	--
RXD0A1	P11.9	In	`FrRxInputSelectionA` - value `FR_RXSEL1`
RXD0A2	P02.1	In	`FrRxInputSelectionA` - value `FR_RXSEL2`
RXD0A3	P14.1	In	`FrRxInputSelectionA` - value `FR_RXSEL3`
TXD0A	P02.0, P11.3, P14.0, P14.10	Out	Not Applicable
TXEN0A	P02.4, P11.6	Out	Not Applicable

Table 13 Connectivity of I/O signals for FR controller 0 Channel B - TC332 and TC333 devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0B0	No Connection	--	--
RXD0B1	P11.10	In	`FrRxInputSelectionB` - value `FR_RXSEL1`
RXD0B2	P02.3	In	`FrRxInputSelectionB` - value `FR_RXSEL2`
RXD0B3	P14.1	In	`FrRxInputSelectionB` - value `FR_RXSEL3`
TXD0B	P02.2, P11.12, P14.0, P14.5	Out	Not Applicable
TXEN0B	P02.5, P11.6, P11.11	Out	Not Applicable

1.1.4.5 DMA support

The FlexRay driver does not use any services provided by the DMA driver.

1.1.4.6 Interrupt connections

The FlexRay driver does not use any interrupt source.

Note: The FR driver depends on the interrupt router (IR) if the absolute timer interrupt support is required. The absolute timer interrupt is assigned to the ERAY Service Request 0 line by the FR driver. The interrupt router (IR) needs to be configured to support the absolute timer interrupt. However, the FR driver does not provide any interrupt handler; the interrupt service routine `FrIf_JobListExec_<ClstIdx>` is part of the FrIf module.

1.1.4.7 Example usage

Examples of API usage and other useful information are as follows:

- **Initialization of the FR driver**

  Pre-condition: The initialization of the MCU and PORT modules are successfully done.

  Step 1: Invoke the `Fr_17_Eray_Init( )` API by passing configuration structure pointer as input parameter.

  For example: `Fr_17_Eray_Init(&Fr_17_Eray_Config);`

  Step 2: If the absolute timer interrupt is to be supported, then the corresponding settings needs to be done for Interrupt Router module and then initialization needs to be done.

  Step 3: Invoke the `Fr_17_Eray_ControllerInit()` API to initialize the Communication Controller.

  For example: `Fr_17_Eray_ControllerInit(0);`

  Step 4: Invoke the `Fr_17_Eray_GetPOCStatus()` API till the POC state is READY.

  For example: `while (Fr_17_Eray_GetPOCStatus(0) != FR_POCSTATE_READY);`

- **Synchronization of FR controller**

  FR controller as a cold start node

  Pre-condition: FlexRay channels are connected to the cluster and the FR Controller is in POCState - POC:ready.

  Step 1: Invoke the `Fr_17_Eray_AllowColdstart()` API to make the controller perform the cold start activity.

  Step 2: Invoke the `Fr_17_Eray_StartCommunication()` API to start communication.

  For example: `Fr_17_Eray_StartCommunication(0);`

  Step 3: Invoke the `Fr_17_Eray_GetPOCStatus()` till the POC state is NORMAL ACTIVE.

  For example: `while (Fr_17_Eray_GetPOCStatus(0) != FR_POCSTATE_NORMAL_ACTIVE);`

  Now the FR driver is synchronized with the cluster and starts the communication. Data can be transmitted by invoking `Fr_17_Eray_TransmitTxLPdu()` API.

  FR controller as a non-cold start node

  Pre-condition: FlexRay channels are connected to the running cluster and the FR Controller is in POCState - POC:ready.

  Step 1: Invoke the `Fr_17_Eray_StartCommunication()` API to start communication.

  For example: `Fr_17_Eray_StartCommunication(0);`

  Step 2: Invoke the `Fr_17_Eray_GetPOCStatus()` API till the POC state is NORMAL ACTIVE.

  For example: `while (Fr_17_Eray_GetPOCStatus(0) != FR_POCSTATE_NORMAL_ACTIVE);`

  Now the FR driver is synchronized with the cluster and starts the communication. Data can be transmitted by invoking `Fr_17_Eray_TransmitTxLPdu()` API.

- **Normal communication of FR driver**

  The FlexRay frames are to be transmitted and received in synchronous to the FlexRay global time. To achieve this, the absolute timer interrupt of the ERAY can be used. The frames which are to be transmitted on a slot n shall be updated using the `Fr_17_Eray_TransmitTxLPdu()` API before the start of the slot n and the frames which are to be received on a slot n shall be read using the `Fr_17_Eray_ReceiveRxLPdu()` API after the slot n.

  Step 1: For the slot n, determine the cycle and macrotick value.

  Step 2: Based on the communication operation to be performed, set the absolute timer by invoking the `Fr_17_Eray_SetAbsoluteTimer()` API.

  Step 3: Enable the absolute timer interrupt by invoking the `Fr_17_Eray_EnableAbsoluteTimerIRQ` API.

  Step 4: Invoke the appropriate API corresponding to the communication operation within the absolute timer ISR handler.

Step 5: Reconfigure the absolute timer for the next communication operation.

- **Buffer reconfiguration**

  The ERay message RAM can be used to configure maximum of 128 message buffers and 8192 bytes of Header and data. In order to accommodate more frames, FR driver implements the buffer reconfiguration mechanism with the support of the `Fr_17_Eray_PrepareLPdu()` API. The configuration parameter `FrPrepareLPduSupport` needs to be configured as true and the Communication Action - `PREPARE_LPDU` has to be configured in FrIf configuration for the frames within the static segment which then take part of buffer reconfiguration.

  Dynamic LPdus are assigned to FIFO and LPdus with FrIfReconfigurable set to TRUE will not participate in buffer reconfiguration as these LPdus require dedicated message buffers. Also if a keyslot frame is present then a dedicated message buffer is assigned to it.

  Invoke `Fr_17_Eray_PrepareLPdu()` API before calling `Fr_17_Eray_TransmitTxLPdu()` API and `Fr_17_Eray_ReceiveRxLPdu()` API, except for dynamic LPdus, LPdus which are assigned to FIFO and LPdus with FrIfReconfigurable set to 'true'.

  Note: Invoking the `Fr_17_Eray_PrepareLPdu()` API does not have any impact on the LPdus which do not participate in buffer reconfiguration.

- **De-initialization of FR driver**

  The FR driver does not implement any de-initialization API.

- **Configuration parameters of the FrIf module**

  The values/ ranges of the cluster related configuration parameters that are part of the FrIf module shall be configured in compliance to **FlexRay Communication Systems Protocol Specification, Version 2.1 Revision A**.

- **Receive FIFO operation - FIFO overrun**

  The detection of the receive FIFO overrun condition is not performed by the FR driver. When a read operation is requested for an LPdu, which is part of the FIFO, the oldest available received message is provided. When an overflow condition occurs, the received new message overwrites the oldest available message in the FIFO resulting in message loss. Hence, the user has to ensure that the read operation for the FIFO LPdus is performed at the right intervals such that there is no occurrence of message loss.

- **Absolute timer mode**

  The AUTOSAR FlexRay specification does not mention about the mode configuration of the absolute timer whether should it be configured in either continuous mode or one-shot mode. However, the `Fr_17_Eray_SetAbsoluteTimer()` API configures the absolute timer in continuous mode. This implementation with continuous timer mode is done in order to avoid the software jitter.

- **Handling of FlexRay frames received in dynamic segment**

  The FlexRay frame received in a dynamic segment slot should be read by the user/ application before the same dynamic minislot number of the next communication cycle. The failure to read this frame within the stipulated time will result in loss of such frame as the Message Buffer Status (MBS) of the assigned message buffer gets updated with the slot status information of the latest slot.

  Consider the example of the FlexRay communication on channel A; there are 12 static slots and a new frame is received in slot 13 which is a dynamic slot. Upon reception of the frame, the message buffer status (bit-field VFRA of register MBS) of the assigned message buffer indicates that a valid frame is received on channel A. Then in this case, the user/ application has to ensure that this received frame is read within the minislot 13 of the next communication cycle. The failure to do so results in the Message Buffer Status (MBS) getting updated with the slot information of the latest minislot 13, which would now indicate that no valid frame was received on channel A.

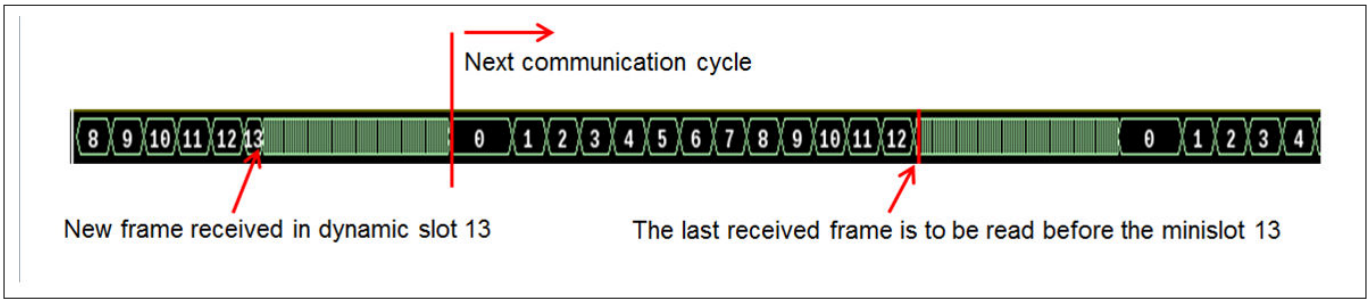

Figure 4 Frame received in dynamic segment

- **Transmission conflict detection feature**

  A configuration parameter `FrTxConflictDetection` is added to enable/ disable the detection of the transmission conflict and this feature is configurable both in AUTOSAR v4.2.2 and AUTOSAR v4.4.0. The `Fr_CheckTxLPduStatus()` API provides the transmit status of the LPdu, i.e whether the LPdu has been transmitted or not. In addition to this, the `Fr_CheckTxLPduStatus()` API can also detect the occurrence of transmission conflict when the `FrTxConflictDetection` parameter is set to true. This is an additional feature which is not a part of AUTOSAR v4.2.2, hence this feature is made configurable (enable/ disable) and by default is kept disabled. However, this feature is part of the AUTOSAR FR SWS from v4.3.0 onwards. The enabling/ disabling of the transmit conflict detection feature is a pre-compile configuration. When a transmission conflict condition is detected, the `Fr_CheckTxLPduStatus()` API returns the transmit status as `FR_TRANSMITTED_CONFLICT`.

  Note that, the enumeration value `FR_TRANSMITTED_CONFLICT` is part of the enumeration type `Fr_TxLPduStatusType`, which is present in the `Fr_GeneralTypes.h` file. So, in case the transmission conflict detection feature is enabled, then the `Fr_GeneralTypes.h` file needs to contain this enumeration value `FR_TRANSMITTED_CONFLICT`. A failure of having this value will result in a compilation error.

  In case the transmission conflict detection feature is disabled, then the enumeration type `Fr_TxLPduStatusType` as in AUTOSAR v4.2.2 can be used, which does not contain the value `FR_TRANSMITTED_CONFLICT`. But in case the enumeration type `Fr_TxLPduStatusType` with value `FR_TRANSMITTED_CONFLICT` is used, then the complete FlexRay Stack must use the enumeration `FR_NOT_TRANSMITTED` as it is and not its value directly since its value differs between the two different definitions of `Fr_TxLPduStatusType`.

1.1.5 Key architectural considerations

1.1.5.1 Buffer reconfiguration

The E-RAY message RAM supports a maximum of 128 message buffers/ 8192 bytes of header and data section. In case all the configured LPdus cannot be accommodated within the message RAM either due to exceeding the maximum limit of 128 message buffers or exceeding the maximum RAM size of 8192 bytes, the hardware buffer reconfiguration mechanism of sharing a message buffer with more than one LPdu needs to be supported. The mechanism of hardware buffer reconfiguration allows sharing of a message buffer with more than one LPdu, this facilitates to accommodate additional LPdus. The maximum number of LPdus that can be mapped to a single buffer is limited to 4. This maximum limit on mapping a single buffer to the number of LPdus is necessary to limit any loss of LPdus/ frames during transmission/ reception of LPdus.

The buffer reconfiguration mechanism is realized with the support of the `Fr_17_Eray_PrepareLPdu()` API, which is enabled by setting the FR configuration parameter `FrPrepareLPduSupport` to TRUE. For LPdus to be considered for buffer reconfiguration, the communication action `PREPARE_LPDU` needs to be configured in FrIf job list of the FrIf configuration. Thus, only the LPdus within the static segment configured with `PREPARE_LPDU` communication action participate in buffer reconfiguration.

Not all the message buffers participate in the buffer configuration.

Exclusive message buffers are allocated to:

- Dynamic LPdus - LPdus within the dynamic segment
- LPdus which are dynamically reconfigurable - FrIfReconfigurable set to true
- Receive FIFO - LPdus which are part of FIFO
- LPdu which is a Key slot/ Start-up/ Sync frame
- Static LPdus (LPdus within the static segment) without `PREPARE_LPDU` action

After allocation of a single exclusive message buffer to each of these LPdus, the remaining message buffers/ RAM space participate in buffer reconfiguration and thus can be shared among the LPdus within static segment configured with `PREPARE_LPDU` communication action. The FrIf job list execution start time (FrIf parameters `FrIfCycle` and `FrIfMacrotick`) must be taken into consideration during the allocation of message buffers to different LPdus. One of the methods is to arrange the LPdus according to their FrIf job list execution start time in ascending order and then proceed with allocating it to the available message buffers.While allocating to the available message buffers for sharing with Lpdus, the parameters 'FrIfBaseCycle' and 'FrIfCycleRepetition' are considered to ensure that the allocated LPDU's doesn't have a common communication cycle.

In the FrIf configuration, the communication action `PREPARE_LPDU` needs to configured for the LPdus which are required to participate in the hardware buffer reconfiguration before a transmit/ receive operation so that before calling `Fr_17_Eray_TransmitTxLPdu()`/ `Fr_17_Eray_ReceiveRxLPdu()` APIs, the `Fr_17_Eray_PrepareLPdu()` API is invoked. This sequence of invoking the `Fr_17_Eray_PrepareLPdu()` API before `Fr_17_Eray_TransmitTxLPdu()` / `Fr_17_Eray_ReceiveRxLPdu()` API needs to be maintained by the user in order to ensure correct transmission/ reception. In addition, it has to be ensured that the `Fr_17_Eray_PrepareLPdu()` API is invoked only after the successful transmission/ reception of the previous LPdu which shares the message buffer with the LPdu which is passed to the API `Fr_17_Eray_PrepareLPdu()`. Depending on the LPdu passed to the `Fr_17_Eray_PrepareLPdu()` API, the allocated message buffer is configured according to the parameters of the related LPdu. A call to the `Fr_17_Eray_PrepareLPdu()` API does not have any impact on the LPdus which do not participate in buffer reconfiguration.

The FR module configuration generator should report error messages under the following situations:

- When the configured LPdus cannot be accommodated within the message RAM and the `PREPARE_LPDU` communication action is not configured for required LPdus (no hardware buffer reconfiguration).
- When the configured LPdus cannot be accommodated within the message RAM even though the required LPdus are configured with `PREPARE_LPDU` communication action due to the reason that maximum buffer reconfiguration limit is reached/ no available message RAM to accommodate all the LPdus.

1.1.5.2 BSW scheduler mechanism

In order to enter the READY state from the CONFIG state, the ERAY module mandates to execute an unlock sequence before writing to the `SUCC1.CMD` in the SUC Configuration Register 1. The write operation to `SUCC1.CMD` has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (`LCK.CLK`). If this write sequence is pre-empted by other read or write accesses, the Communication Controller remains in the CONFIG state and the sequence has to be repeated. Therefore, this sequence is placed within a critical section encapsulated by SchM enter and exit functions. The expected actions from the SchM enter function (`SchM_Enter_Fr_17_Eray_ConfigLockKey`) is to disable the global interrupt and within the SchM exit function (`SchM_Exit_Fr_17_Eray_ConfigLockKey`) is to enable the global interrupt.

1.1.5.3 Clock configuration

In order to control the clock divider of the kernel clock f_{CLC_ERAY}, the `FrClockDivider` configuration parameter is added within the `FrClockConfiguration` container. The clock divider $_{CLC_ERAY}$ of the kernel clock can be selected as either $f_{CLC_ERAY} = f_{SPB}$, $f_{CLC_ERAY} = f_{SPB}$ /2 or $f_{CLC_ERAY} = f_{SPB}$ /4. The parameter `FrClockDivider` only controls the kernel clock f_{CLC_ERAY} and not the sampling clock f_{SCLK}.

1.1.5.4 Input channel selection

The hardware provides the feature to select the alternate receiver input lines for both channels A and B for each of the communication controller. These alternate receiver input lines for channels A/ B can be selected using the configuration parameters `FrRxInputSelectionA` and `FrRxInputSelectionB`, which are added within the container `FrController`.

1.1.5.5 Additional receive FIFO - related parameters

The receive FIFO of the E-Ray module requires additional fields to be configured other than the FIFO configuration parameters provided by AUTOSAR specification. These additional fields are configured using the parameters `FrFrameIdRejectionFilter`, `FrFrameIdRejectionFilterMask`, `FrRejectNullFrames` and `FrRejectStaticSegment` which are added within the container `FrFifo`. These parameters together with the AUTOSAR parameters determine whether a message is rejected by the FIFO.

1.1.5.6 Configuration parameter for timeout event

As per the technical specification of the E-Ray module, the accepted commands will cause a change of register `CCSV` after at most 8 cycles of the slower of the two clocks f_{CLC_ERAY} and f_{SCLK}. In order to incorporate the time required to reflect this change, the configuration parameter `FrTimeoutDurationFactor` is added to configure the maximum time in nanoseconds for blocking function until a timeout error is raised in short term wait loops. The timeout error is issued by the DEM parameter `FR_E_CTRL_TESTRESULT`.

1.1.5.7 Get absolute timer status

The API `Fr_17_Eray_GetAbsoluteTimerIRQStatus()` does not access `SRC` register of the IR module to determine the pending status of the absolute timer interrupt. This API writes to the output parameter `Fr_IRQStatusPtr` the status of the absolute timer flag which is set whenever the absolute timer matches the conditions configured in the timer configuration register. This facilitates the user to use this API in both interrupt mode and polling mode. This may be required by the user to execute the FlexRay job list execution functions in a task context or in an ISR.

1.1.5.8 Configuration parameter for the `Fr_GetNmVector` API

A configuration parameter `FrNmVectorEnable` is added which enables/ disables the existence of the API `Fr_17_Eray_GetNmVector`. This parameter `FrNmVectorEnable` is required as the API `Fr_17_Eray_GetNmVector` is invoked by the optional FrIf module API `FrIf_GetNmVector`. The optional FrIf API `FrIf_GetNmVector` is enabled/ disabled by the FrIf configuration parameter `FrIfGetNmVectorSupport`. Hence, to provide a similar NM support in FR driver the parameter `FrNmVectorEnable` is added.

1.1.5.9 User mode support

The Tricore CPU can execute in three privilege modes namely User-0, User-1 and Supervisor. It is possible to execute the FR driver in User-1 and Supervisor modes. However, the registers that the FR module writes to can be updated directly in both the supervisory and user1 modes except for the `CLC` register. The `CLC` register is updated only by the API `Fr_17_Eray_Init()` during the INIT phase execution. There is no de-init API provided by the FR driver. Hence, the support for supervisory/ user1 mode selection is applicable only during the init phase. A configuration parameter `FrInitApiMode` is added to select the mode of operation during the init phase. The registers accessed by the runtime APIs can be written directly in both Supervisory and User1 modes. Hence no configuration parameter is provided for Supervisory / User-1 mode support during runtime phase. The selection of the operating mode is a pre-compile configuration.

1.1.5.10 Transmission Conflict Detection

The `Fr_CheckTxLPduStatus` API returns the transmit status of the LPdu, i.e whether the LPdu is transmitted or not. In addition to this, the `Fr_CheckTxLPduStatus` API can also detect the occurrence of transmission conflict when `FrTxConflictDetection` parameter is set to true. When a transmission conflict condition is detected, the `Fr_CheckTxLPduStatus` API stores the transmit status as `FR_TRANSMITTED_CONFLICT`. This feature is configurable (enable/ disable) in AUTOSAR v4.2.2 and AUTOSAR v4.4.0.

1.2 Assumptions of Use (AoU)

There are no AoUs for the FlexRay driver.

1.3 Reference information

1.3.1 Configuration interfaces

This section details the configuration container hierarchy along with their configuration parameters.
Supported configuration variant: Post-Build

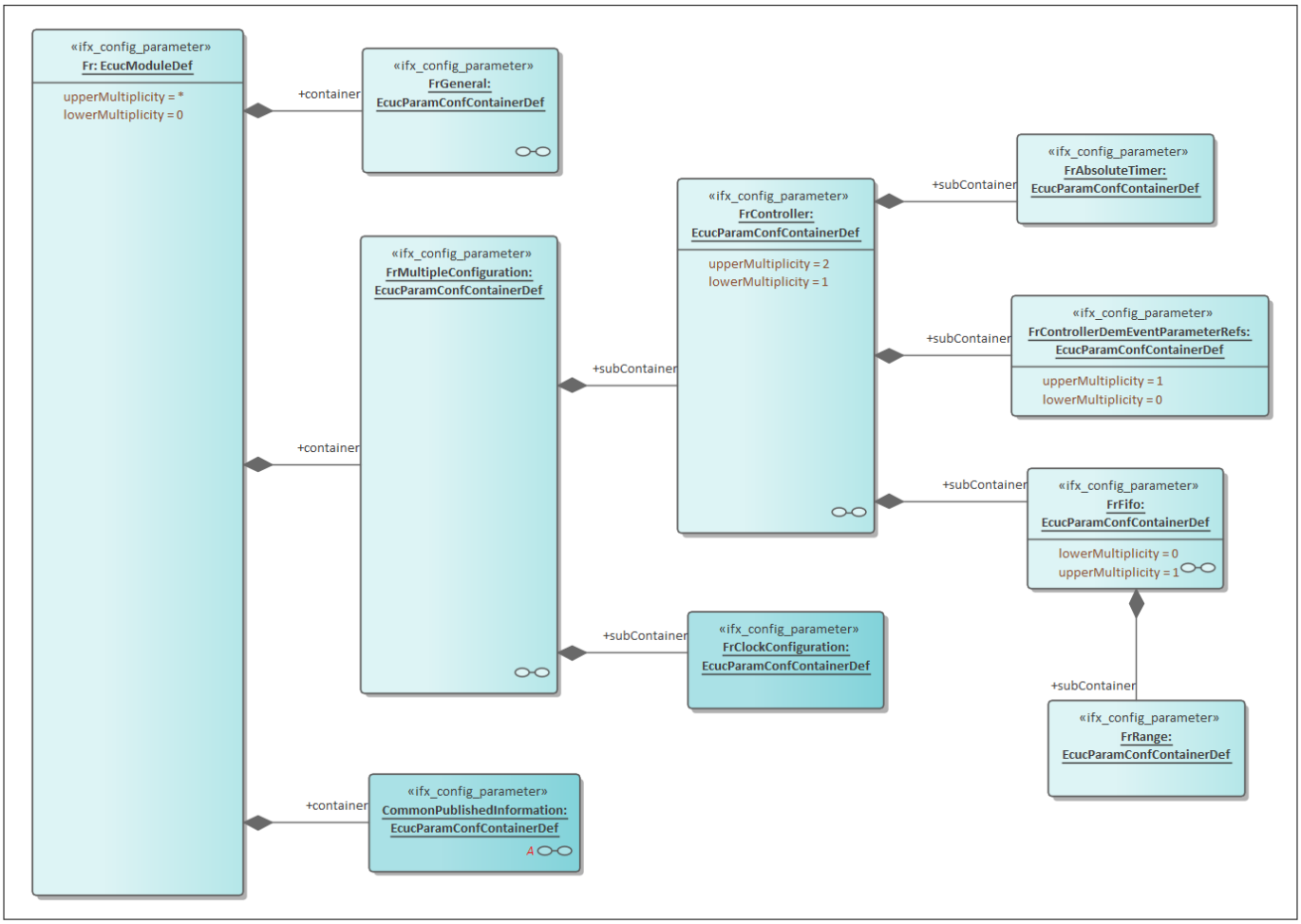

Figure 5 Container hierarchy along with their configuration parameters

1.3.1.1 Container: Fr

Configuration of the FR (FlexRay driver) module.
Post-Build Variant Multiplicity: -
Multiplicity Configuration Class: -

1.3.1.2 Container: FrGeneral

General configuration parameters of the FlexRay driver module.
Post-Build Variant Multiplicity: -
Multiplicity Configuration Class: -

1.3.1.2.1 FrCtrlTestCount

Table 14 Specification for FrCtrlTestCount

Name	`FrCtrlTestCount`		
Description	Maximum number of iterations the FlexRay controller hardware test is performed during controller initialization.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	1		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.2 FrDevErrorDetect

Table 15 Specification for FrDevErrorDetect

Name	`FrDevErrorDetect`		
Description	Switches the Default Error Tracer (DET) detection and notification ON or OFF. - true: enabled (ON). - false: disabled (OFF).		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.3 FrDisableLPduSupport

Table 16 Specification for FrDisableLPduSupport

Name	`FrDisableLPduSupport`		
Description	Enables or disables API function Fr_DisableLPdu. Remark: The optional APIs are disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.4 FrEcucPartitionRef

Table 17 Specification for FrEcucPartitionRef

Name	`FrEcucPartitionRef`		
Description	The parameter maps the FR driver to zero or multiple ECUC partitions to make the modules API available in this partition. The FR driver will operate as an independent instance in each of the partitions. *Note: Parameter support is added only for AUTOSAR schema compliance. This parameter is not used in code generation logic, hence this parameter is made editable false.*		
Multiplicity	0..*	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.2.5 FrExtendedLPduReporting

Table 18 Specification for FrExtendedLPduReporting

Name	`FrExtendedLPduReporting`		
Description	Enables or disables reporting of actual cycle and slot ID by Fr_TransmitTxLPdu, Fr_ReceiveRxLPdu and Fr_CheckTxLPduStatus		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.2.6 FrIndex

Table 19 Specification for FrIndex

Name	`FrIndex`		
Description	Specifies the instance Id of this module instance. If only one instance is present it should have the Id 0. Remark: Minimum instance ID is selected as the default value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.7 FrInitApiMode

Table 20 Specification for FrInitApiMode

Name	`FrInitApiMode`		
Description	This configuration parameter defines the mode in which the Init API will be used. This parameter is introduced to support the selection of the operation mode (supervisor/ user1) during the init phase. Remark: Since FR driver accesses the SFRs, it is more efficient to operate the FR driver in supervisor mode. Hence, the default mode of operation is supervisor.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	FR_MCAL_SUPERVISOR: Operating mode used is Supervisory FR_MCAL_USER1: Operating mode used is USER1		
Default value	FR_MCAL_SUPERVISOR		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.8 FrNmVectorEnable

Table 21 Specification for FrNmVectorEnable

Name	`FrNmVectorEnable`		
Description	Enables/ disables the existence of the Fr_17_Eray_GetNmVector API. This parameter is introduced to disable the network management functionality if it is not required. Remark: The optional APIs are disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-

(table continues...)

Table 21 (continued) Specification for FrNmVectorEnable

Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.9 FrNumCtrlSupported

Table 22 Specification for FrNumCtrlSupported

Name	`FrNumCtrlSupported`		
Description	Determines the maximum number of communication controllers that the driver supports. Remark: Minimum number of controllers supported is selected as the default value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 2		
Default value	1		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.10 FrPrepareLPduSupport

Table 23 Specification for FrPrepareLPduSupport

Name	`FrPrepareLPduSupport`		
Description	Enables or disables API function Fr_PrepareLPdu. Remark: The optional APIs are disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-

(table continues...)

Table 23 (continued) Specification for FrPrepareLPduSupport

Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.11 FrReconfigLPduSupport

Table 24 Specification for FrReconfigLPduSupport

Name	`FrReconfigLPduSupport`		
Description	Enables or disables API function Fr_ReconfigLPdu. Remark: The optional APIs are disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.12 FrRunTimeErrorDetect

Table 25 Specification for FrRunTimeErrorDetect

Name	`FrRunTimeErrorDetect`		
Description	Switches the Runtime Error detection and notification ON or OFF. - true: enabled (ON). - false: disabled (OFF).		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

(table continues...)

Table 25 (continued) Specification for FrRunTimeErrorDetect

Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.2.13 FrRxStringentCheck

Table 26 Specification for FrRxStringentCheck

Name	`FrRxStringentCheck`		
Description	If stringent check is enabled (true), received frames are accepted only if no slot status error occurred.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.14 FrRxStringentLengthCheck

Table 27 Specification for FrRxStringentLengthCheck

Name	`FrRxStringentLengthCheck`		
Description	If stringent length check is enabled (true), received frames are accepted only if the received payload length matches the configured payload length.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		

(table continues...)

Table 27 (continued) Specification for FrRxStringentLengthCheck

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.15 FrTimeoutDurationFactor

Table 28 Specification for FrTimeoutDurationFactor

Name	`FrTimeoutDurationFactor`		
Description	Specifies the maximum time in nanoseconds for blocking function until a timeout is raised in short term wait loops. Duration of 8 clock cycles of the slower of the two clocks fCLC_ERAY or fSCLK is to be configured for this parameter. This parameter is introduced to configure the maximum time until a timeout error (DEM parameter FR_E_CTRL_TESTRESULT) is reported. Remark: The default value of this parameter is set to 400 as an example value within the range. As per the target specification, the mentioned duration of 8 cycles is with the assumption that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Taking these assumptions into consideration, it is suggested that the user configures the time duration greater than 8 clock cycles for this parameter.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	100 - 4294967295		
Default value	400		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.16 FrTxConflictDetection

Table 29 Specification for FrTxConflictDetection

Name	`FrTxConflictDetection`		
Description	This parameter is introduced to enable/ disable the transmit conflict detection feature. If transmit conflict detection is enabled (true), the Fr_CheckTxLPduStatus API detects the occurrence of a transmission conflict. Remark: By default, this feature is not part of AUTOSAR version 4.2.2, hence default value is false to keep the transmit conflict detection feature disabled. This feature is part of AUTOSAR version 4.4.0, hence default value is true, in order to keep the transmit conflict detection feature enabled.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE/ FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.17 FrVersionInfoApi

Table 30 Specification for FrVersionInfoApi

Name	`FrVersionInfoApi`		
Description	Enables/disables the existence of the Fr_GetVersionInfo API. Remark: The optional APIs are disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-

(table continues...)

Table 30 (continued) Specification for FrVersionInfoApi

Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.3 Container: FrMultipleConfiguration

This container contains the configuration parameters and sub-containers of the AUTOSAR FR module.

The container is composed of two sub-containers FrController and FrClockConfiguration.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.4 Container: FrController

Configuration of the individual controller.

This container has three sub-containers within it - FrAbsoluteTimer, FrFifo and FrControllerDemEventParameterRefs.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.4.1 FrCtrlEcucPartitionRef

Table 31 Specification for FrCtrlEcucPartitionRef

Name	`FrCtrlEcucPartitionRef`		
Description	Maps one single Flexray controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Flexray driver is mapped to. *Note: Parameter support is added only for AUTOSAR schema compliance. This parameter is not used in code generation logic, hence this parameter is made editable false.*		
Multiplicity	0..1	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.4.2 FrCtrlIdx

Table 32 Specification for FrCtrlIdx

Name	`FrCtrlIdx`		
Description	Determines index of CC within Fr. This value will be assigned to the symbolic name derived from the short name of the FrController container. Remark: The first CC index is selected as the default value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 1		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.3 FrPAllowHaltDueToClock

Table 33 Specification for FrPAllowHaltDueToClock

Name	`FrPAllowHaltDueToClock`		
Description	Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the CC is allowed to transition to POC:halt. If set to false, the CC will not transition to the POC:halt state but will enter or remain in the POC:normal passive state (self healing would still be possible). Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL

(table continues...)

Table 33	(continued) Specification for FrPAllowHaltDueToClock
Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.4.4 FrPAllowPassiveToActive

Table 34	Specification for FrPAllowPassiveToActive		
Name	`FrPAllowPassiveToActive`		
Description	Number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to zero, the CC is not allowed to transition from POC:normal passive to POC:normal active. Remark: The default value of this parameter is set to a valid value (example value) within the range. However, in practical scenario, this value will be overwritten by the corresponding parameter value within the customer FIBEX file.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 31		
Default value	10		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.5 FrPChannels

Table 35	Specification for FrPChannels		
Name	`FrPChannels`		
Description	Channels to which the node is connected. Remark: Channel configuration should be consistent with cluster channel configuration (e.g. FlexRay CC channel configuration cannot be CHANNEL_AB when the cluster channel configuration is CHANNEL_A). The default value of this parameter (FR_CHANNEL_AB) is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef

(table continues...)

Table 35 (continued) Specification for FrPChannels

Range	FR_CHANNEL_A: Cluster uses channel A FR_CHANNEL_AB: Cluster uses channel A and B FR_CHANNEL_B: Cluster uses channel B		
Default value	FR_CHANNEL_AB		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.6 FrPClusterDriftDamping

Table 36 Specification for FrPClusterDriftDamping

Name	`FrPClusterDriftDamping`		
Description	Local cluster drift damping factor used for rate correction (Microticks). Remark: The default value of this parameter is set to a valid value (example value) within the range. However, in practical scenario, this value will be overwritten by the corresponding parameter value within the customer FIBEX file.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 20		
Default value	1		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.7 FrPDecodingCorrection

Table 37 Specification for FrPDecodingCorrection

Name	`FrPDecodingCorrection`

(table continues...)

Table 37 (continued) Specification for FrPDecodingCorrection

Description	Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point (Microticks). Remark: Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	14 - 143		
Default value	14		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.8 FrPDelayCompensationA

Table 38 Specification for FrPDelayCompensationA

Name	`FrPDelayCompensationA`		
Description	Value used to compensate for reception delays on the indicated channel. This value covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 microsecond to 0.05 microsecond (Microticks). Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 200		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.9 FrPDelayCompensationB

Table 39 Specification for FrPDelayCompensationB

Name	`FrPDelayCompensationB`		
Description	Value used to compensate for reception delays on the indicated channel. This value covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 microsecond to 0.05 microsecond (Microticks). Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 200		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.10 FrPExternalSync

Table 40 Specification for FrPExternalSync

Name	`FrPExternalSync`		
Description	Flag indicating whether the node is externally synchronized (operating as time gateway sink in a TT-E cluster) or locally synchronized. Remark: Set to false for FlexRay Protocol 2.1 Rev. A compliance. The default value is also set to false for FlexRay Protocol 2.1 Rev. A compliance.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-

(table continues...)

Table 40 (continued) Specification for FrPExternalSync

Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.11 FrPFallBackInternal

Table 41 Specification for FrPFallBackInternal

Name	`FrPFallBackInternal`		
Description	Flag indicating whether a time gateway sink node will switch to local clock operation when synchronization with the time gateway source node is lost (FrPFallBackInternal = true) or will instead go to POC:ready (FrPFallBackInternal =false). Remark: Set to false for FlexRay Protocol 2.1 Rev. A compliance. The default value is also set to false for FlexRay Protocol 2.1 Rev. A compliance.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.12 FrPKeySlotId

Table 42 Specification for FrPKeySlotId

Name	`FrPKeySlotId`		
Description	ID of the key slot, i.e., the slot used to transmit the startup frame, sync frame, or designated key slot frame. If this parameter is set to zero the node does not have a key slot. Remark: The default value is set to 1 (minimum value of the key slot) as the value 0 implies that the node does not have a key slot. However, in practical scenario, this value will be overwritten by the actual keyslot ID value from the customer FIBEX file.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 1023		
Default value	1		

(table continues...)

Table 42 (continued) Specification for FrPKeySlotId

Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.13 FrPKeySlotOnlyEnabled

Table 43 Specification for FrPKeySlotOnlyEnabled

Name	`FrPKeySlotOnlyEnabled`		
Description	Flag indicating whether or not the node will enter key slot only mode following startup. Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pSingleSlotEnabled. The default value is set to false as the key slot only feature is disabled by default. However, in practical scenario, this value will be overwritten by the corresponding parameter value from the customer FIBEX file.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.14 FrPKeySlotUsedForStartup

Table 44 Specification for FrPKeySlotUsedForStartup

Name	`FrPKeySlotUsedForStartup`

(table continues...)

Table 44 (continued) Specification for FrPKeySlotUsedForStartup

Description	Flag indicating whether the key slot is used to transmit a startup frame. If FrPKeySlotUsedForStartup is set to true then FrPKeySlotUsedForSync must also be set to true. Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR. By default, the key slot is not configured to transmit startup frame.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	FrPKeySlotId		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.15 FrPKeySlotUsedForSync

Table 45 Specification for FrPKeySlotUsedForSync

Name	`FrPKeySlotUsedForSync`		
Description	Flag indicating whether the key slot is used to transmit a sync frame. If FrPKeySlotUsedForStartup is set to true then FrPKeySlotUsedForSync must also be set to true. Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR. By default, the key slot is not configured to transmit sync frame.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	FrPKeySlotUsedForStartup, FrPKeySlotId		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.16 FrPLatestTx

Table 46 Specification for FrPLatestTx

Name	FrPLatestTx		
Description	Number of the last minislot in which a frame transmission can start in the dynamic segment. Remark: Upper limit 7980 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 7980		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.17 FrPMacroInitialOffsetA

Table 47 Specification for FrPMacroInitialOffsetA

Name	FrPMacroInitialOffsetA		
Description	Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration (Macroticks). Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	2 - 68		
Default value	2		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.18 FrPMacroInitialOffsetB

Table 48 Specification for FrPMacroInitialOffsetB

Name	`FrPMacroInitialOffsetB`		
Description	Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration (Macroticks). Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	2 - 68		
Default value	2		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.19 FrPMicroInitialOffsetA

Table 49 Specification for FrPMicroInitialOffsetA

Name	`FrPMicroInitialOffsetA`		
Description	Number of microticks between the closest macrotick boundary described by pMacroInitialOffsetA and the secondary time reference point. The parameter depends on FrPDelayCompensationA and therefore it has to be set independently for each channel (Microticks). Remark: The minimum value of the parameter is restricted to 1 by Hardware errata FlexRay_AI.092. The default value is set to the supported minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 239		
Default value	1		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL

(table continues...)

Table 49 (continued) Specification for FrPMicroInitialOffsetA

Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.4.20 FrPMicroInitialOffsetB

Table 50 Specification for FrPMicroInitialOffsetB

Name	`FrPMicroInitialOffsetB`		
Description	Number of microticks between the closest macrotick boundary described by pMacroInitialOffsetB and the secondary time reference point. The parameter depends on FrPDelayCompensationB and therefore it has to be set independently for each channel (Microticks). Remark: The minimum value of the parameter is restricted to 1 by Hardware errata FlexRay_AI.092. The default value is set to the supported minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 239		
Default value	1		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.21 FrPMicroPerCycle

Table 51 Specification for FrPMicroPerCycle

Name	`FrPMicroPerCycle`		
Description	Nominal number of microticks in the communication cycle of the local node. If nodes have different microtick durations this number will differ from node to node (Microticks). Remark: Upper limit 640000 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	640 - 640000		
Default value	640		

(table continues...)

Table 51 (continued) Specification for FrPMicroPerCycle

Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.22 FrPNmVectorEarlyUpdate

Table 52 Specification for FrPNmVectorEarlyUpdate

Name	`FrPNmVectorEarlyUpdate`		
Description	Flag indicating when the update of the Network Management Vector in the CHI will take place. If FrPNmVectorEarlyUpdate is set to false, the update will take place after the NIT. If FrPNmVectorEarlyUpdate is set to true, the update will take place after the end of the static segment. Remark: Set to false for FlexRay Protocol 2.1 Rev. A compliance. The default value is also set to false for FlexRay Protocol 2.1 Rev. A compliance.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.23 FrPOffsetCorrectionOut

Table 53 Specification for FrPOffsetCorrectionOut

Name	`FrPOffsetCorrectionOut`

(table continues...)

Table 53 (continued) Specification for FrPOffsetCorrectionOut

Description	Magnitude of the maximum permissible offset correction value (Microticks). Remark: Upper limit 15567 for FlexRay Protocol 2.1 Rev A compliance. However, hardware supports Upper limit of 15266. The default value is set to the minimum value of this parameter.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	13 - 15266		
Default value	13		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.24 FrPOffsetCorrectionStart

Table 54 Specification for FrPOffsetCorrectionStart

Name	`FrPOffsetCorrectionStart`		
Description	Start of the offset correction phase within the NIT, expressed as the number of macroticks from the start of cycle (Macroticks). Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gOffsetCorrectionStart. Remark: Lower limit 9 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	9 - 15999		
Default value	9		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.25 FrPPayloadLengthDynMax

Table 55 Specification for FrPPayloadLengthDynMax

Name	FrPPayloadLengthDynMax		
Description	Maximum payload length for dynamic frames (16 bit words). Remark: The default value is set to the minimum value of this parameter.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 127		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.26 FrPRateCorrectionOut

Table 56 Specification for FrPRateCorrectionOut

Name	FrPRateCorrectionOut		
Description	Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle (Microticks). Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift. Upper limit 1923 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	2 - 1923		
Default value	2		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.27 FrPSamplesPerMicrotick

Table 57 Specification for FrPSamplesPerMicrotick

Name	`FrPSamplesPerMicrotick`		
Description	Number of samples per microtick. Remark: Set to N2SAMPLES for 10 Mbps baudrate. This parameter is disabled for configuration in GUI.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	N1SAMPLES: 1 sample N2SAMPLES: 2 samples N4SAMPLES: 4 samples		
Default value	N2SAMPLES		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.28 FrPSecondKeySlotId

Table 58 Specification for FrPSecondKeySlotId

Name	`FrPSecondKeySlotId`		
Description	ID of the second key slot, in which a second startup frame will be sent when operating as a coldstart node in a TT-L or TT-D cluster. If this parameter is set to zero the node does not have a second key slot. Remark: Set to 0 for FlexRay Protocol 2.1 Rev A compliance. The default value is also set to 0 for FlexRay Protocol 2.1 Rev A compliance.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 1023		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		

(table continues...)

Table 58 (continued) Specification for FrPSecondKeySlotId

Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.4.29 FrPTwoKeySlotMode

Table 59 Specification for FrPTwoKeySlotMode

Name	`FrPTwoKeySlotMode`		
Description	Flag indicating whether node operates as a coldstart node in a TT-E or TT-L cluster. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and FrPKeySlotUsedForStartup must also be set to true. If FrPExternalSync is set to true then FrPTwoKeySlotMode must also be set to true. Remark: Set to false for FlexRay Protocol 2.1 Rev A compliance. The default value is also set to false for FlexRay Protocol 2.1 Rev A compliance.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.30 FrPWakeupChannel

Table 60 Specification for FrPWakeupChannel

Name	`FrPWakeupChannel`		
Description	Channel used by the node to send a wakeup pattern. FrPWakeupChannel must be selected from among the channels configured by FrPChannels. Remark: The value of this parameter should be consistent with PChannels configuration. For example: If FrPChannels is FR_CHANNEL_A, the PWakeupChannel cannot be FR_CHANNEL_B. The default value of this parameter (FR_CHANNEL_A) is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef

(table continues...)

Table 60 (continued) Specification for FrPWakeupChannel

Range	FR_CHANNEL_A: channel A FR_CHANNEL_B: channel B		
Default value	FR_CHANNEL_A		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	FrPChannels		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.31 FrPWakeupPattern

Table 61 Specification for FrPWakeupPattern

Name	`FrPWakeupPattern`		
Description	Number of repetitions of the wakeup symbol that are combined to form a wakeup pattern when the node enters the POC:wakeup send state. Remark: Lower limit 2 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	2 - 63		
Default value	2		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.32 FrPdAcceptedStartupRange

Table 62 Specification for FrPdAcceptedStartupRange

Name	`FrPdAcceptedStartupRange`

(table continues...)

Table 62 (continued) Specification for FrPdAcceptedStartupRange

Description	Expanded range of measured clock deviation allowed for startup frames during integration (Microticks). Remark: Upper limit 1875 for FlexRay Protocol 2.1 Rev A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 1875		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.33 FrPdListenTimeout

Table 63 Specification for FrPdListenTimeout

Name	`FrPdListenTimeout`		
Description	Value for the startup listen timeout and wakeup listen timeout. Although this parameter is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster (Microticks). Remark: Upper limit 1283846 for FlexRay Protocol 2.1 Rev. A compliance. The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1284 - 1283846		
Default value	1284		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.34 FrPdMicrotick

Table 64 Specification for FrPdMicrotick

Name	`FrPdMicrotick`		
Description	Duration of a microtick. Remark: Set to T25NS for 10 Mbps baud rate. This parameter is disabled for configuration in GUI. The default value is also set to T25NS for 10 Mbps baud rate.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	T100NS: 100 nanosecond T12_5NS: 12.5 nanosecond T200NS: 200 nanosecond T25NS: 25 nanosecond T50NS: 50 nanosecond		
Default value	T25NS		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.35 FrRxInputSelectionA

Table 65 Specification for FrRxInputSelectionA

Name	`FrRxInputSelectionA`		
Description	Provides alternate Port Pin selection for FlexRay Receive input line for Channel A. This parameter is introduced to provide support for the selection of the alternate receiver input line for channel A. Remark: The default value of this parameter is set as per device supported first data line.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	FR_RXSELx_PORTy_z: This Channel A receiver input varies as per RXSEL data line x, port number y and pin number z. For example FR_RXSEL0_PORT14_8.		
Default value	As per device supported first data line		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-

(table continues...)

Table 65 (continued) Specification for FrRxInputSelectionA

Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.36 FrRxInputSelectionB

Table 66 Specification for FrRxInputSelectionB

Name	`FrRxInputSelectionB`		
Description	Provides alternate Port Pin selection for FlexRay Receive input line for Channel B. This parameter is introduced to provide support for the selection of the alternate receiver input line for channel B. Remark: The default value of this parameter is set as per device supported first data line.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	FR_RXSELx_PORTy_z: This Channel B receiver input varies as per RXSEL data line x, port number y and pin number z. For example FR_RXSEL0_PORT14_7.		
Default value	As per device supported first data line		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5 Container: FrAbsoluteTimer

Specifies the absolute timer configuration parameters of the Fr.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.5.1 FrAbsTimerIdx

Table 67 Specification for FrAbsTimerIdx

Name	`FrAbsTimerIdx`

(table continues...)

Table 67 (continued) Specification for FrAbsTimerIdx

Description	Contains the index of an absolute timer contained in FR on a certain FlexRay CC. Remark: The default value of this parameter is set to index 0 as only one absolute timer per CC is supported by the hardware.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 0		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6 Container: FrControllerDemEventParameterRefs

Container for the references to DemEventParameter elements which will be invoked using the Mcal_Wrapper_Dem_ReportErrorStatus API for AS422 and Mcal_Wrapper_Dem_SetEventStatus for AS440 in case the corresponding Production error occurs. The EventId is taken from the DemEventId value of the referenced DemEventParameter. The standardized errors are provided in the container and can be extended by vendor specific error references.

Post-Build Variant Multiplicity: TRUE

Multiplicity Configuration Class: Post-Build

1.3.1.6.1 FR_E_CTRL_TESTRESULT

Table 68 Specification for FR_E_CTRL_TESTRESULT

Name	FR_E_CTRL_TESTRESULT		
Description	Reference to DEM event Id that is reported for FlexRay controller hardware test failure. If this parameter is not configured, no event reporting happens. This DEM event Id is also reported when there is a timeout in hardware response. Remark: Since the name of the dependent container is user configurable, the default value is kept as NULL.		
Multiplicity	0..1	**Type**	EcucSymbolicNameReferenceDef
Range	Reference to Node: DemEventParameter		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE

(table continues...)

Table 68 (continued) Specification for FR_E_CTRL_TESTRESULT

Value configuration class	Post-Build	**Multiplicity configuration class**	Post-Build
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7 Container: FrFifo

One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.

This container has a sub-container FrRange within it.

Deviation:

The upper multiplicity of the container is limited to 1 as the hardware supports only one configurable receive FIFO.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

1.3.1.7.1 FrAdmitWithoutMessageId

Table 69 Specification for FrAdmitWithoutMessageId

Name	`FrAdmitWithoutMessageId`		
Description	Determines whether or not frames received in the dynamic segment that does not contain a message ID will be admitted into the FIFO. Remark: This parameter is not used for implementation and disabled for configuration in GUI. The default value of this parameter is set to false as it is disabled.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.2 FrBaseCycle

Table 70 Specification for FrBaseCycle

Name	`FrBaseCycle`		
Description	FIFO cycle counter acceptance criteria. Remark: The default value of this parameter is set to minimum value. However, in practical scenario, this value will be overwritten by the corresponding parameter value from the customer FIBEX file.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 63		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.3 FrChannels

Table 71 Specification for FrChannels

Name	`FrChannels`		
Description	FIFO channel admittance criteria. Remark: The default value of this parameter is set to FR_CHANNEL_A. However, in practical scenario, this value will be overwritten by the corresponding parameter value from the customer FIBEX file.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	FR_CHANNEL_A: Frames received on channel A FR_CHANNEL_AB: Frames received on channel A and B FR_CHANNEL_B: Frames received on channel B		
Default value	FR_CHANNEL_A		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.4 FrCycleRepetition

Table 72 Specification for FrCycleRepetition

Name	`FrCycleRepetition`		
Description	FIFO cycle counter acceptance criteria. Valid values are 1,2,4,8,16,32,64 for FlexRay Protocol 2.1. Remark: The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 64		
Default value	1		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.5 FrFifoDepth

Table 73 Specification for FrFifoDepth

Name	`FrFifoDepth`		
Description	FrFifoDepth configures the maximum number of receive frames which can be contained in the FIFO. Remark: The FifoDepth maximum value is limited to 127 due to hardware constraints. The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 127		
Default value	1		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.6 FrFrameIdRejectionFilter

Table 74 Specification for FrFrameIdRejectionFilter

Name	`FrFrameIdRejectionFilter`		
Description	FIFO FrameId rejection criteria, frame ID to be rejected by the FIFO. This parameter is introduced to support the configuration of the rejection criteria of the hardware FIFO. Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 2047		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.7 FrFrameIdRejectionFilterMask

Table 75 Specification for FrFrameIdRejectionFilterMask

Name	`FrFrameIdRejectionFilterMask`		
Description	FIFO FrameId rejection mask, For all the bits that are 0, corresponding bits are considered in the FrFrameIdRejectionFilter for FrameId rejection. This parameter is introduced to support the configuration of the rejection criteria of the hardware FIFO. Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 2047		
Default value	2047		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.8 FrMsgIdMask

Table 76 Specification for FrMsgIdMask

Name	FrMsgIdMask		
Description	FIFO message identifier acceptance criteria (mask filter). This parameter is disabled for configuration as these are not supported by the hardware, so ignored for configuration. Remark: The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.9 FrMsgIdMatch

Table 77 Specification for FrMsgIdMatch

Name	FrMsgIdMatch		
Description	FIFO message identifier acceptance criteria (match filter). This parameter is disabled for configuration as these are not supported by the hardware so ignored for configuration. Remark: The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.10 FrRejectNullFrames

Table 78 Specification for FrRejectNullFrames

Name	`FrRejectNullFrames`		
Description	Determines whether or not null frames received are considered for FIFO. This parameter is introduced to support the configuration of the rejection criteria of the hardware FIFO. Remark: The default value of this parameter is set to true to conform to the AUTOSAR requirement that only non-null frames must be accepted.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.11 FrRejectStaticSegment

Table 79 Specification for FrRejectStaticSegment

Name	`FrRejectStaticSegment`		
Description	Determines whether or not frames received in the static segment are considered for FIFO. This parameter is introduced to support the configuration of the rejection criteria of the hardware FIFO. Remark: The default value of this parameter is set to false as the optional feature is disabled by default.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-

(table continues...)

Table 79 (continued) Specification for FrRejectStaticSegment

Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8 Container: FrRange

FIFO Frame Id range acceptance criteria.

This container is ignored for configuration.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.8.1 FrRangeMax

Table 80 Specification for FrRangeMax

Name	`FrRangeMax`		
Description	Last FrameId of this range that will be accepted by the FIFO. This parameter is ignored for configuration. Remark: The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 2047		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.2 FrRangeMin

Table 81 Specification for FrRangeMin

Name	`FrRangeMin`

(table continues...)

Table 81 (continued) Specification for FrRangeMin

Description	First FrameId of this range that will be accepted by the FIFO. This parameter is ignored for configuration. Remark: The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 2047		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9 Container: FrClockConfiguration

Clock Configuration of the individual controller.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.9.1 FrClockDivider

Table 82 Specification for FrClockDivider

Name	`FrClockDivider`		
Description	Local clock divider. This parameter is introduced to control the clock divider of the kernel clock fCLC_ERAY. Remark: FrClockDivider parameter must not be configured to a value of 3 (reserved value). The default value of this parameter is set to the minimum value.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 4		
Default value	1		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10 Container: CommonPublishedInformation

This section describes the parameters published by the FlexRay driver module.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.10.1 ArMajorVersion

Table 83 Specification for ArMajorVersion

Name	`ArMajorVersion`		
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	4		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.2 ArMinorVersion

Table 84 Specification for ArMinorVersion

Name	`ArMinorVersion`		
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per the selected Autosar Version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.3 ArPatchVersion

Table 85 Specification for ArPatchVersion

Name	`ArPatchVersion`		
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per the selected Autosar Version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.4 ModuleId

Table 86 Specification for ModuleId

Name	`ModuleId`		
Description	Module ID of the FR module from Module List.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	81		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.5 Release

Table 87 Specification for Release

Name	`Release`
Description	This parameter indicates the TC3xx device derivative used for the implementation.

(table continues...)

Table 87 (continued) Specification for Release

Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	As per hardware derivative		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.6 SwMajorVersion

Table 88 Specification for SwMajorVersion

Name	`SwMajorVersion`		
Description	Major version number of the vendor specific implementation of the module.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.7 SwMinorVersion

Table 89 Specification for SwMinorVersion

Name	`SwMinorVersion`		
Description	Minor version number of the vendor specific implementation of the module.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per driver		

(table continues...)

Table 89 (continued) Specification for SwMinorVersion

Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.8 SwPatchVersion

Table 90 Specification for SwPatchVersion

Name	`SwPatchVersion`		
Description	Patch level version number of the vendor specific implementation of the module.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.9 VendorApiInfix

Table 91 Specification for VendorApiInfix

Name	`VendorApiInfix`		
Description	This parameter is used to specify the vendor specific name.		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	Eray		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

(table continues...)

Table 91 (continued) Specification for VendorApiInfix

Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.10.10 VendorId

Table 92 Specification for VendorId

Name	`VendorId`		
Description	Vendor ID of the dedicated implementation of the FR module according to the AUTOSAR vendor list.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	17		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.2 Functions - Type definitions

This chapter lists out all the data types of the FlexRay driver.

1.3.2.1 Fr_SlotAssignmentType

Table 93 Specification for Fr_SlotAssignmentType

Syntax	`Fr_SlotAssignmentType`	
Type	Structure	
File	`Fr_GeneralTypes.h`	
Range	uint8 Cycle	Cycle in which the frame is transmitted/ received.
	uint16 SlotId	Slot Id of the frame
	Fr_ChannelType ChannelId	Channel of the frame

(table continues...)

Table 93 (continued) Specification for Fr_SlotAssignmentType

Description	This structure contains information about the assignment of a FlexRay frame to a cycle, Channel ID and slot ID
Source	AUTOSAR
Autosar Version	Applicable for Autosar version 4.4.0.

1.3.2.2 Fr_17_Eray_ConfigType

Table 94 Specification for Fr_17_Eray_ConfigType

Syntax	`Fr_17_Eray_ConfigType`	
Type	Structure	
File	`Fr_17_Eray.h`	
Range	--	The elements of the data structure are specific to the microcontroller.
Description	This type contains the implementation-specific post build configuration structure of the FlexRay driver. Detailed description is available in the design document.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.3 Fr_POCStateType

Table 95 Specification for Fr_POCStateType

Syntax	`Fr_POCStateType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_POCSTATE_CONFIG	Config state
	1 - FR_POCSTATE_DEFAULT_CONFIG	Default config state
	2 - FR_POCSTATE_HALT	Halt state
	3 - FR_POCSTATE_NORMAL_ACTIVE	Normal active state
	4 - FR_POCSTATE_NORMAL_PASSIVE	Normal passive state
	5 - FR_POCSTATE_READY	Ready state
	6 - FR_POCSTATE_STARTUP	Startup state
	7 - FR_POCSTATE_WAKEUP	Wakeup state
Description	Represents the FlexRay controller POC states.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.4 Fr_SlotModeType

Table 96 Specification for Fr_SlotModeType

Syntax	`Fr_SlotModeType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_SLOTMODE_KEYSLOT	Single slot mode
	1 - FR_SLOTMODE_ALL_PENDING	All pending slot mode
	2 - FR_SLOTMODE_ALL	All slot mode
Description	Represents the FlexRay controller slotmodes.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.5 Fr_ErrorModeType

Table 97 Specification for Fr_ErrorModeType

Syntax	`Fr_ErrorModeType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_ERRORMODE_ACTIVE	Active error mode
	1 - FR_ERRORMODE_PASSIVE	Passive error mode
	2 - FR_ERRORMODE_COMM_HALT	Communication halted error mode
Description	Represents the FlexRay controller error modes.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.6 Fr_WakeupStatusType

Table 98 Specification for Fr_WakeupStatusType

Syntax	`Fr_WakeupStatusType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_WAKEUP_UNDEFINED	Wakeup state - undefined
	1 - FR_WAKEUP_RECEIVED_HEADER	Wakeup state - received header
	2 - FR_WAKEUP_RECEIVED_WUP	Wakeup state - received wakeup pattern
	3 - FR_WAKEUP_COLLISION_HEADER	Wakeup state - collision header
	4 - FR_WAKEUP_COLLISION_WUP	Wakeup state - collision wakeup pattern
	5 - FR_WAKEUP_COLLISION_UNKNOWN	Wakeup state - collision unknown

(table continues...)

Table 98 (continued) Specification for Fr_WakeupStatusType

	6 - FR_WAKEUP_TRANSMITTED	Wakeup state - transmitted
Description	Represents the FlexRay controller wakeup status types.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.7 Fr_StartupStateType

Table 99 Specification for Fr_StartupStateType

Syntax	`Fr_StartupStateType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_STARTUP_UNDEFINED	Startup state - undefined
	1 - FR_STARTUP_COLDSTART_LISTEN	Startup state - coldstart listen
	2 - FR_STARTUP_INTEGRATION_COLDSTART_CHECK	Startup state - integration coldstart check
	3 - FR_STARTUP_COLDSTART_JOIN	Startup state - coldstart join
	4 - FR_STARTUP_COLDSTART_COLLISION_RESOLUTION	Startup state - collision resolution
	5 - FR_STARTUP_COLDSTART_CONSISTENCY_CHECK	Startup state - consistency check
	6 - FR_STARTUP_INTEGRATION_LISTEN	Startup state - integration listen
	7 - FR_STARTUP_INITIALIZE_SCHEDULE	Startup state - initialize schedule
	8 - FR_STARTUP_INTEGRATION_CONSISTENCY_CHECK	Startup state - integration consistency check
	9 - FR_STARTUP_COLDSTART_GAP	Startup state - coldstart gap
	10 - FR_STARTUP_EXTERNAL_STARTUP	External startup
Description	Represents the FlexRay controller startup status types.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.8 Fr_POCStatusType

Table 100 Specification for Fr_POCStatusType

Syntax	`Fr_POCStatusType`

(table continues...)

Table 100 (continued) Specification for Fr_POCStatusType

Type	Structure	
File	`Fr_GeneralTypes.h`	
Range	Fr_POCStateType State	POC state
	boolean Freeze	Freeze bit
	boolean CHIHaltRequest	CHI Halt request bit
	boolean CHIReadyRequest	CHI Ready request bit
	boolean ColdstartNoise	Coldstart noise bit
	Fr_SlotModeType SlotMode	Slot mode
	Fr_ErrorModeType ErrorMode	Error mode
	Fr_WakeupStatusType WakeupStatus	Wakeup state
	Fr_StartupStateType StartupState	Startup state
Description	Represents the FlexRay controller POC-Status information.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.9 Fr_TxLPduStatusType

Table 101 Specification for Fr_TxLPduStatusType

Syntax	`Fr_TxLPduStatusType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_TRANSMITTED	LSdu was transmitted
	1 - FR_TRANSMITTED_CONFLICT	A transmission conflict has occurred
	2 - FR_NOT_TRANSMITTED	LSdu was not transmitted
Description	Represents the LSdu TX status.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.10 Fr_RxLPduStatusType

Table 102 Specification for Fr_RxLPduStatusType

Syntax	`Fr_RxLPduStatusType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_RECEIVED	LSdu was received
	1 - FR_NOT_RECEIVED	LSdu was not received

(table continues...)

Table 102 (continued) Specification for Fr_RxLPduStatusType

	2 - FR_RECEIVED_MORE_DATA_AVAILABLE	FIFO is not empty
Description	Represents the LSdu TX status.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.11 Fr_ChannelType

Table 103 Specification for Fr_ChannelType

Syntax	`Fr_ChannelType`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0/1 - FR_CHANNEL_A	FlexRay channel A. The numeric value for this element is 0 for Autosar version 4.2.2 and 1 for Autosar version 4.4.0
	1/2 - FR_CHANNEL_B	FlexRay channel B The numeric value for this element is 1 for Autosar version 4.2.2 and 2 for Autosar version 4.4.0
	2/3 - FR_CHANNEL_AB	FlexRay channel A and B The numeric value for this element is 2 for Autosar version 4.2.2 and 3 for Autosar version 4.4.0
Description	Represents the FlexRay channels.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.12 FR_CIDX_[CONFIGPARAM_NAME]

Table 104 Specification for FR_CIDX_[CONFIGPARAM_NAME]

Syntax	`FR_CIDX_[CONFIGPARAM_NAME]`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_CIDX_GDCYCLE	Maps to configuration parameter - FrIfGdCycle
	1 - FR_CIDX_PMICROPERCYCLE	Maps to configuration parameter - FrPMicroPerCycle
	2 - FR_CIDX_PDLISTENTIMEOUT	Maps to configuration parameter - FrPdListenTimeout
	3 - FR_CIDX_GMACROPERCYCLE	Maps to configuration parameter - FrIfGMacroPerCycle

(table continues...)

Table 104 (continued) Specification for FR_CIDX_[CONFIGPARAM_NAME]

	4 - FR_CIDX_GDMACROTICK	Maps to configuration parameter - FrIfGdMacrotick
	5 - FR_CIDX_GNUMBEROFMINISLOTS	Maps to configuration parameter - FrIfGNumberOfMinislots
	6 - FR_CIDX_GNUMBEROFSTATICSLOTS	Maps to configuration parameter - FrIfGNumberOfStaticSlots
	7 - FR_CIDX_GDNIT	Maps to configuration parameter - FrIfGdNit
	8 - FR_CIDX_GDSTATICSLOT	Maps to configuration parameter - FrIfGdStaticSlot
	9 - FR_CIDX_GDWAKEUPRXWINDOW	Maps to configuration parameter - FrIfGdWakeupRxWindow
	10 - FR_CIDX_PKEYSLOTID	Maps to configuration parameter - FrPKeySlotId
	11 - FR_CIDX_PLATESTTX	Maps to configuration parameter - FrPLatestTx
	12 - FR_CIDX_POFFSETCORRECTIONOUT	Maps to configuration parameter - FrPOffsetCorrectionOut
	13 - FR_CIDX_POFFSETCORRECTIONSTART	Maps to configuration parameter - FrPOffsetCorrectionStart
	14 - FR_CIDX_PRATECORRECTIONOUT	Maps to configuration parameter - FrPRateCorrectionOut
	15 - FR_CIDX_PSECONDKEYSLOTID	Maps to configuration parameter - FrPSecondKeySlotId
	16 - FR_CIDX_PDACCEPTEDSTARTUPRANGE	Maps to configuration parameter - FrPdAcceptedStartupRange
	17 - FR_CIDX_GCOLDSTARTATTEMPTS	Maps to configuration parameter - FrIfGColdStartAttempts
	18 - FR_CIDX_GCYCLECOUNTMAX	Maps to configuration parameter - FrIfGCycleCountMax
	19 - FR_CIDX_GLISTENNOISE	Maps to configuration parameter - FrIfGListenNoise
	20 - FR_CIDX_GMAXWITHOUTCLOCKCORRECTFATAL	Maps to configuration parameter - FrIfGMaxWithoutClockCorrectFatal
	21 - FR_CIDX_GMAXWITHOUTCLOCKCORRECTPASSIVE	Maps to configuration parameter - FrIfGMaxWithoutClockCorrectPassive
	22 - FR_CIDX_GNETWORKMANAGEMENTVECTORLENGTH	Maps to configuration parameter - FrIfGNetworkManagementVectorLength

(table continues...)

Table 104 (continued) Specification for FR_CIDX_[CONFIGPARAM_NAME]

	23 - FR_CIDX_GPAYLOADLENGTHSTATIC	Maps to configuration parameter - FrIfGPayloadLengthStatic
	24 - FR_CIDX_GSYNCFRAMEIDCOUNTMAX	Maps to configuration parameter - FrIfGSyncFrameIDCountMax
	25 - FR_CIDX_GDACTIONPOINTOFFSET	Maps to configuration parameter - FrIfGdActionPointOffset
	26 - FR_CIDX_GDBIT	Maps to configuration parameter - FrIfGdBit
	27 - FR_CIDX_GDCASRXLOWMAX	Maps to configuration parameter - FrIfGdCasRxLowMax
	28 - FR_CIDX_GDDYNAMICSLOTIDLEPHASE	Maps to configuration parameter - FrIfGdDynamicSlotIdlePhase
	29 - FR_CIDX_GDMINISLOTACTIONPOINTOFFSET	Maps to configuration parameter - FrIfGdMiniSlotActionPointOffset
	30 - FR_CIDX_GDMINISLOT	Maps to configuration parameter - FrIfGdMinislot
	31 - FR_CIDX_GDSAMPLECLOCKPERIOD	Maps to configuration parameter - FrIfGdSampleClockPeriod
	32 - FR_CIDX_GDSYMBOLWINDOW	Maps to configuration parameter - FrIfGdSymbolWindow
	33 - FR_CIDX_GDSYMBOLWINDOWACTIONPOINTOFFSET	Maps to configuration parameter - FrIfGdSymbolWindowActionPointOffset
	34 - FR_CIDX_GDTSSTRANSMITTER	Maps to configuration parameter - FrIfGdTssTransmitter
	35 - FR_CIDX_GDWAKEUPRXIDLE	Maps to configuration parameter - FrIfGdWakeupRxIdle
	36 - FR_CIDX_GDWAKEUPRXLOW	Maps to configuration parameter - FrIfGdWakeupRxLow
	37 - FR_CIDX_GDWAKEUPTXACTIVE	Maps to configuration parameter - FrIfGdWakeupTxActive
	38 - FR_CIDX_GDWAKEUPTXIDLE	Maps to configuration parameter - FrIfGdWakeupTxIdle
	39 - FR_CIDX_PALLOWPASSIVETOACTIVE	Maps to configuration parameter - FrPAllowPassiveToActive
	40 - FR_CIDX_PCHANNELS	Maps to configuration parameter - FrPChannels
	41 - FR_CIDX_PCLUSTERDRIFTDAMPING	Maps to configuration parameter - FrPClusterDriftDamping
	42 - FR_CIDX_PDECODINGCORRECTION	Maps to configuration parameter - FrPDecodingCorrection

(table continues...)

Table 104 (continued) Specification for FR_CIDX_[CONFIGPARAM_NAME]

	43 - FR_CIDX_PDELAYCOMPENSATIONA	Maps to configuration parameter - FrPDelayCompensationA
	44 - FR_CIDX_PDELAYCOMPENSATIONB	Maps to configuration parameter - FrPDelayCompensationB
	45 - FR_CIDX_PMACROINITIALOFFSETA	Maps to configuration parameter - FrPMacroInitialOffsetA
	46 - FR_CIDX_PMACROINITIALOFFSETB	Maps to configuration parameter - FrPMacroInitialOffsetB
	47 - FR_CIDX_PMICROINITIALOFFSETA	Maps to configuration parameter - FrPMicroInitialOffsetA
	48 - FR_CIDX_PMICROINITIALOFFSETB	Maps to configuration parameter - FrPMicroInitialOffsetB
	49 - FR_CIDX_PPAYLOADLENGTHDYNMAX	Maps to configuration parameter - FrPPayloadLengthDynMax
	50 - FR_CIDX_PSAMPLESPERMICROTICK	Maps to configuration parameter - FrPSamplesPerMicrotick
	51 - FR_CIDX_PWAKEUPCHANNEL	Maps to configuration parameter - FrPWakeupChannel
	52 - FR_CIDX_PWAKEUPPATTERN	Maps to configuration parameter - FrPWakeupPattern
	53 - FR_CIDX_PDMICROTICK	Maps to configuration parameter - FrPdMicrotick
	54 - FR_CIDX_GDIGNOREAFTERTX	Maps to configuration parameter - FrIfGdIgnoreAfterTx
	55 - FR_CIDX_PALLOWHALTDUETOCLOCK	Maps to configuration parameter - FrPAllowHaltDueToClock
	56 - FR_CIDX_PEXTERNALSYNC	Maps to configuration parameter - FrPExternalSync
	57 - FR_CIDX_PFALLBACKINTERNAL	Maps to configuration parameter - FrPFallBackInternal
	58 - FR_CIDX_PKEYSLOTONLYENABLED	Maps to configuration parameter - FrPKeySlotOnlyEnabled
	59 - FR_CIDX_PKEYSLOTUSEDFORSTARTUP	Maps to configuration parameter - FrPKeySlotUsedForStartup
	60 - FR_CIDX_PKEYSLOTUSEDFORSYNC	Maps to configuration parameter - FrPKeySlotUsedForSync
	61 - FR_CIDX_PNMVECTOREARLYUPDATE	Maps to configuration parameter - FrPNmVectorEarlyUpdate
	62 - FR_CIDX_PTWOKEYSLOTMODE	Maps to configuration parameter - FrPTwoKeySlotMode

(table continues...)

Table 104 (continued) Specification for FR_CIDX_[CONFIGPARAM_NAME]

Description	List of Macros (indices) that can be passed into API function Fr_ReadCCConfig as parameter Fr_ConfigParamIdx.
Source	AUTOSAR
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.2.13 FR_SLOTMODE_SINGLE

Table 105 Specification for FR_SLOTMODE_SINGLE

Syntax	`FR_SLOTMODE_SINGLE`	
Type	Enumeration	
File	`Fr_GeneralTypes.h`	
Range	0 - FR_SLOTMODE_KEYSLOT	Mapping via preprocessor macro
Description	The preprocessor macro FR_SLOTMODE_SINGLE maps to value of the FR_SLOTMODE_KEYSLOT.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3 Functions - APIs

This section lists all the APIs of the FlexRay driver.

1.3.3.1 Fr_17_Eray_Init

Table 106 Specification for `Fr_17_Eray_Init` API

Syntax	`void Fr_17_Eray_Init ( const Fr_17_Eray_ConfigType * const Fr_ConfigPtr )`	
Service ID	0x1C	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant	
Parameters (in)	Fr_ConfigPtr	Pointer to FR module configuration structure
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-

(table continues...)

Table 106 (continued) Specification for `Fr_17_Eray_Init` API

Description	Initializes the Fr. This function internally stores the configuration address to enable subsequent API calls to access the configuration.
Source	AUTOSAR
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED
Configuration dependencies	-
User hints	None
SFR accessed	ERAY_CLC(rw), ERAY_MHDS(r), ERAY_SUCC1(rw), SCU_CCUCON0(r), SCU_EICON0(rw), SCU_OSCCON(r), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.2 Fr_17_Eray_ControllerInit

Table 107 Specification for `Fr_17_Eray_ControllerInit` API

Syntax	`Std_ReturnType Fr_17_Eray_ControllerInit ( const uint8 Fr_CtrlIdx )`	
Service ID	0x00	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Initializes a FlexRay Communication Controller.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INV_CTRL_IDX, FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED	
Configuration dependencies	-	

(table continues...)

Table 107 (continued) Specification for `Fr_17_Eray_ControllerInit` API

User hints	None
SFR accessed	ERAY_CCSV(r), ERAY_CUST1(rw), ERAY_EIER(ex_w), ERAY_EIR(ex_w), ERAY_FCL(ex_w), ERAY_FRF(ex_w), ERAY_FRFM(ex_w), ERAY_GTUC01(rw), ERAY_GTUC02(rw), ERAY_GTUC03(rw), ERAY_GTUC04(rw), ERAY_GTUC05(rw), ERAY_GTUC06(rw), ERAY_GTUC07(rw), ERAY_GTUC08(rw), ERAY_GTUC09(rw), ERAY_GTUC10(rw), ERAY_GTUC11(rw), ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_ILE(ex_w), ERAY_LCK(w), ERAY_MHDC(rw), ERAY_MHDS(w), ERAY_MRC(w), ERAY_NEMC(rw), ERAY_PRTC1(rw), ERAY_PRTC2(rw), ERAY_SIER(w), ERAY_SILS(ex_w), ERAY_SIR(w), ERAY_SUCC1(rw), ERAY_SUCC2(rw), ERAY_SUCC3(rw), ERAY_WRHS1(w), ERAY_WRHS2(w), ERAY_WRHS3(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.3 Fr_17_Eray_StartCommunication

Table 108 Specification for `Fr_17_Eray_StartCommunication` API

Syntax	`Std_ReturnType Fr_17_Eray_StartCommunication ( const uint8 Fr_CtrlIdx )`	
Service ID	0x03	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Starts communication. Initiates the startup procedure within the FlexRay CC.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_POCSTATE	

(table continues...)

Table 108 (continued) Specification for `Fr_17_Eray_StartCommunication` API

Configuration dependencies	-
User hints	None
SFR accessed	ERAY_CCSV(r), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.4 Fr_17_Eray_AllowColdstart

Table 109 Specification for `Fr_17_Eray_AllowColdstart` API

Syntax	`Std_ReturnType Fr_17_Eray_AllowColdstart ( const uint8 Fr_CtrlIdx )`	
Service ID	0x23	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Invokes the CC CHI command ALLOW_COLDSTART.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_POCSTATE	
Configuration dependencies	-	
User hints	The user has to ensure that the controller configuration parameter FrPKeySlotId is configured to a non-zero slot value and a Tx frame is configured for this slot.	

(table continues...)

Table 109 (continued) Specification for `Fr_17_Eray_AllowColdstart` API

SFR accessed	ERAY_CCSV(r), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.5 Fr_17_Eray_AllSlots

Table 110 Specification for `Fr_17_Eray_AllSlots` API

Syntax	`Std_ReturnType  Fr_17_Eray_AllSlots`
	`(`
	`  const uint8 Fr_CtrlIdx`
	`)`	

Service ID	0x24	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Invokes the CC CHI command ALL_SLOTS, which requests a switch from key slot only mode to all slots transmission mode at the beginning of the next communication cycle.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_POCSTATE	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.6 Fr_17_Eray_HaltCommunication

Table 111 Specification for `Fr_17_Eray_HaltCommunication` API

Syntax	`Std_ReturnType Fr_17_Eray_HaltCommunication ( const uint8 Fr_CtrlIdx )`	
Service ID	0x04	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Invokes the CC CHI command HALT, which requests the halt state which will be reached by the end of the current FlexRay communication cycle but might not be reached immediately.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_POCSTATE	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.7 Fr_17_Eray_AbortCommunication

Table 112 Specification for `Fr_17_Eray_AbortCommunication` API

Syntax	`Std_ReturnType Fr_17_Eray_AbortCommunication ( const uint8 Fr_CtrlIdx )`
Service ID	0x05

(table continues...)

Table 112 (continued) Specification for `Fr_17_Eray_AbortCommunication` API

Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Invokes the CC CHI command FREEZE, which immediately aborts communication (if active) and changes to the POC:halt state from any previous POCState.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.8 Fr_17_Eray_SendWUP

Table 113 Specification for `Fr_17_Eray_SendWUP` API

Syntax	`Std_ReturnType  Fr_17_Eray_SendWUP`
	`(`
	`  const uint8 Fr_CtrlIdx`
	`)`	

Service ID	0x06	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.

(table continues...)

Table 113 (continued) Specification for `Fr_17_Eray_SendWUP` API

Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Invokes the CC CHI command WAKEUP, which initiates the wakeup transmission procedure on the configured FlexRay channel.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_POCSTATE	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.9 Fr_17_Eray_SetWakeupChannel

Table 114 Specification for `Fr_17_Eray_SetWakeupChannel` API

Syntax	`Std_ReturnType  Fr_17_Eray_SetWakeupChannel`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const Fr_ChannelType Fr_ChnlIdx`
	`)`	

Service ID	0x07	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_ChnlIdx	Index of FlexRay CC within the context of the FlexRay driver. Index of FlexRay channel within the context of the FlexRay CC Fr_CtrlIdx. Valid values are FR_CHANNEL_A and FR_CHANNEL_B.
Parameters (out)	-	-
Parameters (in - out)	-	-

(table continues...)

Table 114 (continued) Specification for `Fr_17_Eray_SetWakeupChannel` API

Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Sets a wakeup channel.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_POCSTATE, FR_17_ERAY_E_INV_CHNL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_LCK(w), ERAY_SUCC1(rw), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.10 Fr_17_Eray_GetPOCStatus

Table 115 Specification for `Fr_17_Eray_GetPOCStatus` API

Syntax	`Std_ReturnType  Fr_17_Eray_GetPOCStatus`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  Fr_POCStatusType * const Fr_POCStatusPtr`
	`)`	

Service ID	0x0a	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_POCStatusPtr	Address the output value is stored to.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the POC status.	
Source	AUTOSAR	

(table continues...)

Table 115 (continued) Specification for `Fr_17_Eray_GetPOCStatus` API	
Error handling	FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_PARAM_POINTER
Configuration dependencies	-
User hints	None
SFR accessed	ERAY_CCEV(ex_r), ERAY_CCSV(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.11 Fr_17_Eray_TransmitTxLPdu

Table 116 Specification for `Fr_17_Eray_TransmitTxLPdu` API		
Syntax	`Std_ReturnType  Fr_17_Eray_TransmitTxLPdu ( const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx, const uint8 * const Fr_LSduPtr, const uint8 Fr_LSduLength )`	
Service ID	0x0b	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx Fr_LSduPtr Fr_LSduLength	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame. This reference points to a buffer where the assembled LSdu to be transmitted within this LPdu is stored at. Determines the length of the data (in Bytes) to be transmitted.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Transmits data on the FlexRay network.	
Source	AUTOSAR	

(table continues...)

Table 116 (continued) Specification for `Fr_17_Eray_TransmitTxLPdu` API

Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FR_17_ERAY_E_INV_LENGTH
Configuration dependencies	-
User hints	-
SFR accessed	ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_OBCM(w), ERAY_OBCR(rw), ERAY_RDHS2(r), ERAY_WRDS(ex_w), ERAY_WRHS1(w), ERAY_WRHS2(w), ERAY_WRHS3(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar version 4.2.2.

1.3.3.12 Fr_17_Eray_TransmitTxLPdu

Table 117 Specification for `Fr_17_Eray_TransmitTxLPdu` API

Syntax	`Std_ReturnType Fr_17_Eray_TransmitTxLPdu ( const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx, const uint8 * const Fr_LSduPtr, const uint8 Fr_LSduLength, Fr_SlotAssignmentType * const Fr_SlotAssignmentPtr )`	
Service ID	0x0b	
Sync/Async	Asynchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx Fr_LSduPtr Fr_LSduLength	Index of FlexRay CC within the context of the FlexRay driver This index is used to Uniquely identify a FlexRay frame This reference points to a buffer where the assembled LSdu to be transmitted within this LPdu is stored at Determines the length of data (in bytes) to be transmitted
Parameters (out)	Fr_SlotAssignmentPtr	This reference points to the memory location where the actual cycle, slot id, and channel of the frame identified by Fr_LPduIdx shall be stored. A NULL_PTR indicates the the information is not required by the caller.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors

(table continues...)

Table 117 (continued) Specification for `Fr_17_Eray_TransmitTxLPdu` API

Description	Transmit data on the FlexRay network
Source	AUTOSAR
Error handling	FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INV_LENGTH, FR_17_ERAY_E_INIT_FAILED, FR_E_CTRL_TESTRESULT
Configuration dependencies	-
User hints	None
SFR accessed	ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_OBCM(w), ERAY_OBCR(rw), ERAY_RDHS1(r), ERAY_RDHS2(r), ERAY_WRDS(ex_w), ERAY_WRHS1(w), ERAY_WRHS2(w), ERAY_WRHS3(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar version 4.4.0.

1.3.3.13 Fr_17_Eray_CancelTxLPdu

Table 118 Specification for `Fr_17_Eray_CancelTxLPdu` API

Syntax	`Std_ReturnType Fr_17_Eray_CancelTxLPdu ( const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx )`	
Service ID	0x2d	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Cancels the already pending transmission of an LPdu contained in the physical transmit resource (e.g. message buffer) of the controller.	
Source	AUTOSAR	

(table continues...)

Table 118 (continued) Specification for `Fr_17_Eray_CancelTxLPdu` API

Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX
Configuration dependencies	-
User hints	None
SFR accessed	ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_TXRQ1(r), ERAY_TXRQ2(r), ERAY_TXRQ3(r), ERAY_TXRQ4(r), ERAY_WRHS1(w), ERAY_WRHS2(w), ERAY_WRHS3(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.14 Fr_17_Eray_ReceiveRxLPdu

Table 119 Specification for `Fr_17_Eray_ReceiveRxLPdu` API

Syntax	```Std_ReturnType Fr_17_Eray_ReceiveRxLPdu (const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx, uint8 * const Fr_LSduPtr, Fr_RxLPduStatusType * const Fr_RxLPduStatusPtr, uint8 * const Fr_LSduLengthPtr)```	
Service ID	0x0c	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame.
Parameters (out)	Fr_LSduPtr Fr_RxLPduStatusPtr Fr_LSduLengthPtr	This reference points to the buffer where the LSdu to be received must be stored. This reference points to the memory location where the status of the LPdu must be stored. This reference points to the memory location where the length of the LSdu (in bytes) must be stored. This length represents the number of bytes copied to Fr_LSduPtr.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors

(table continues...)

Table 119 (continued) Specification for `Fr_17_Eray_ReceiveRxLPdu` API

Description	Receives data from the FlexRay network.
Source	AUTOSAR
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FR_17_ERAY_E_PARAM_POINTER, FRIF_E_LPDU_SLOTSTATUS
Configuration dependencies	-
User hints	None
SFR accessed	ERAY_FSR(ex_r), ERAY_MBS(r), ERAY_MRC(r), ERAY_NDAT1(ex_r), ERAY_NDAT2(ex_r), ERAY_NDAT3(ex_r), ERAY_NDAT4(ex_r), ERAY_OBCM(w), ERAY_OBCR(rw), ERAY_RDDS(r), ERAY_RDHS2(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar version 4.2.2.

1.3.3.15 Fr_17_Eray_ReceiveRxLPdu

Table 120 Specification for `Fr_17_Eray_ReceiveRxLPdu` API

Syntax	```Std_ReturnType Fr_17_Eray_ReceiveRxLPdu (const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx, uint8 * const Fr_LSduPtr, Fr_RxLPduStatusType * const Fr_RxLPduStatusPtr, uint8 * const Fr_LSduLengthPtr, Fr_SlotAssignmentType * const Fr_SlotAssignmentPtr)```	
Service ID	0x0c	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver
	Fr_LPduIdx	This index is used to uniquely identify a FlexRay frame

(table continues...)

Table 120 (continued) Specification for `Fr_17_Eray_ReceiveRxLPdu` API

Parameters (out)	Fr_LSduPtr Fr_RxLPduStatusPtr Fr_LSduLengthPtr Fr_SlotAssignmentPtr	This reference points to the buffer where the LSdu to be received shall be stored This reference points to the memory location where the status of the LPdu shall be stored This reference points to the memory location where the length of the LSdu (in bytes) shall be stored. This length represents the number of bytes copied to Fr_LSduPtr. This reference points to the memory location where the actual cycle, slot ID, and channel of the frame identified by Fr_LPduIdx shall be stored. A NULL_PTR indicates that the information is not required by the caller.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Receives data from FlexRay network	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FR_E_CTRL_TESTRESULT, FRIF_E_LPDU_SLOTSTATUS	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_FSR(ex_r), ERAY_MBS(r), ERAY_MRC(r), ERAY_NDAT1(ex_r), ERAY_NDAT2(ex_r), ERAY_NDAT3(ex_r), ERAY_NDAT4(ex_r), ERAY_OBCM(w), ERAY_OBCR(rw), ERAY_RDDS(r), ERAY_RDHS1(r), ERAY_RDHS2(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar version 4.4.0.	

1.3.3.16 Fr_17_Eray_CheckTxLPduStatus

Table 121 Specification for `Fr_17_Eray_CheckTxLPduStatus` API

Syntax	`Std_ReturnType  Fr_17_Eray_CheckTxLPduStatus`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint16 Fr_LPduIdx,`
	`  Fr_TxLPduStatusType * const Fr_TxLPduStatusPtr`
	`)`

Service ID	0x0d

(table continues...)

Table 121 (continued) Specification for `Fr_17_Eray_CheckTxLPduStatus` API

Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame
Parameters (out)	Fr_TxLPduStatusPtr	This reference is used to store the transmit status of the LPdu
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Checks the transmit status of the LSdu. *Note: When the FrTxConflictDetection parameter is configured to true, this API provides the status FR_TRANSMITTED_CONFLICT when the transmission conflict has occurred.*	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FRIF_E_LPDU_SLOTSTATUS	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_MBS(r), ERAY_OBCM(rw), ERAY_OBCR(rw), ERAY_TXRQ1(r), ERAY_TXRQ2(r), ERAY_TXRQ3(r), ERAY_TXRQ4(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar version 4.2.2.	

1.3.3.17 Fr_17_Eray_CheckTxLPduStatus

Table 122 Specification for `Fr_17_Eray_CheckTxLPduStatus` API

Syntax	`Std_ReturnType  Fr_17_Eray_CheckTxLPduStatus`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint16 Fr_LPduIdx,`
	`  Fr_TxLPduStatusType * const Fr_TxLPduStatusPtr,`
	`  Fr_SlotAssignmentType * const Fr_SlotAssignmentPtr`
	`)`

Service ID	0x0d
Sync/Async	Synchronous

(table continues...)

Table 122 (continued) Specification for `Fr_17_Eray_CheckTxLPduStatus` API

Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx	Index of FlexRay CC within the context of the FlexRay Driver. This index is used to uniquely identify a FlexRay frame
Parameters (out)	Fr_TxLPduStatusPtr Fr_SlotAssignmentPtr	This reference is used to store the transmit status of the LPdu This reference points to the memory location where the actual cycle, slot ID, and channel of the frame identified by Fr_LPduIdx shall be stored. A NULL_PTR indicates that the information is not required by the caller.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: API call finished successfully. E_NOT_OK: API call aborted due to errors.
Description	Checks the transmit status of the LSdu. *Note: When the FrTxConflictDetection parameter is configured to false, this API does not provide the status FR_TRANSMITTED_CONFLICT when the transmission conflict has occurred.*	
Source	AUTOSAR	
Error handling	FRIF_E_LPDU_SLOTSTATUS, FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_MBS(r), ERAY_OBCM(rw), ERAY_OBCR(rw), ERAY_RDHS1(r), ERAY_TXRQ1(r), ERAY_TXRQ2(r), ERAY_TXRQ3(r), ERAY_TXRQ4(r), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar version 4.4.0.	

1.3.3.18 Fr_17_Eray_PrepareLPdu

Table 123 Specification for `Fr_17_Eray_PrepareLPdu` API

Syntax	`Std_ReturnType  Fr_17_Eray_PrepareLPdu`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint16 Fr_LPduIdx`
	`)`

Service ID	0x1f
Sync/Async	Synchronous

(table continues...)

Table 123 (continued) Specification for `Fr_17_Eray_PrepareLPdu` API

Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Prepares a LPdu. *Note: If the function Fr_17_Eray_PrepareLPdu is called multiple times for a reconfigurable receive LPdu, the received data may be missed. Hence application should make sure that for a reconfigurable receive LPdu, PrepareLPdu API would be called only when receive is not started.*	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FR_E_CTRL_TESTRESULT	
Configuration dependencies	FrPrepareLPduSupport	
User hints	None	
SFR accessed	ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_WRHS1(w), ERAY_WRHS2(w), ERAY_WRHS3(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.19 Fr_17_Eray_ReconfigLPdu

Table 124 Specification for `Fr_17_Eray_ReconfigLPdu` API

Syntax	`Std_ReturnType  Fr_17_Eray_ReconfigLPdu`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint16 Fr_LPduIdx,`
	`  const uint16 Fr_FrameId,`
	`  const Fr_ChannelType Fr_ChnlIdx,`
	`  const uint8 Fr_CycleRepetition,`
	`  const uint8 Fr_CycleOffset,`
	`  const uint8 Fr_PayloadLength,`
	`  const uint16 Fr_HeaderCRC`
	`)`

(table continues...)

Table 124 (continued) Specification for `Fr_17_Eray_ReconfigLPdu` API

Service ID	0x25	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx Fr_FrameId Fr_ChnlIdx Fr_CycleRepetition Fr_CycleOffset Fr_PayloadLength Fr_HeaderCRC	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame FlexRay Frame ID the FrIf_LPdu must be configured to. FlexRay Channel the FrIf_LPdu must be configured to. Cycle Repetition part of the cycle filter mechanism FrIf_LPdu must be configured to. Cycle Offset part of the cycle filter mechanism FrIf_LPdu must be configured to. Payloadlength in units of bytes the FrIf_LPduIdx must be configured to. Header CRC the FrIf_LPdu must be configured to.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Reconfigures a given LPdu according to the parameters (FrameId, Channel, CycleRepetition, CycleOffset, PayloadLength, HeaderCRC) at runtime.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX, FR_17_ERAY_E_INV_CHNL_IDX, FR_17_ERAY_E_INV_HEADERCRC, FR_17_ERAY_E_INV_CYCLE, FR_17_ERAY_E_INV_LENGTH	
Configuration dependencies	FrReconfigLPduSupport	
User hints	None	
SFR accessed	ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_WRHS1(w), ERAY_WRHS2(w), ERAY_WRHS3(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.20 Fr_17_Eray_DisableLPdu

Table 125 Specification for `Fr_17_Eray_DisableLPdu` API

Syntax	`Std_ReturnType Fr_17_Eray_DisableLPdu ( const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx )`	
Service ID	0x26	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_LPduIdx	Index of FlexRay CC within the context of the FlexRay driver. This index is used to uniquely identify a FlexRay frame
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Disables the hardware resource of an LPdu for transmission/ reception.	
Source	AUTOSAR	
Error handling	FR_E_CTRL_TESTRESULT, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_LPDU_IDX	
Configuration dependencies	FrDisableLPduSupport	
User hints	None	
SFR accessed	ERAY_IBCM(w), ERAY_IBCR(rw), ERAY_WRHS1(w), STM_TIM0(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.21 Fr_17_Eray_GetGlobalTime

Table 126 Specification for `Fr_17_Eray_GetGlobalTime` API

Syntax	`Std_ReturnType Fr_17_Eray_GetGlobalTime ( const uint8 Fr_CtrlIdx, uint8 * const Fr_CyclePtr, uint16 * const Fr_MacroTickPtr )`	
Service ID	0x10	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_CyclePtr Fr_MacroTickPtr	Address where the current FlexRay communication cycle value should be stored. Address where the current macrotick value should be stored.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the current global FlexRay time.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_MTCCV(ex_r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.22 Fr_17_Eray_GetNmVector

Table 127 Specification for `Fr_17_Eray_GetNmVector` API

Syntax	`Std_ReturnType Fr_17_Eray_GetNmVector ( const uint8 Fr_CtrlIdx, uint8 * const Fr_NmVectorPtr )`	
Service ID	0x22	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_NmVectorPtr	Address where the NmVector of the last communication cycle should be stored.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the network management vector of the last communication cycle.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	FrNmVectorEnable	
User hints	None	
SFR accessed	ERAY_CCSV(r), ERAY_NMV(ex_r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.23 Fr_17_Eray_GetNumOfStartupFrames

Table 128 Specification for `Fr_17_Eray_GetNumOfStartupFrames` API

Syntax	`Std_ReturnType Fr_17_Eray_GetNumOfStartupFrames ( const uint8 Fr_CtrlIdx, uint8 * const Fr_NumOfStartupFramesPtr )`

(table continues...)

Table 128 (continued) Specification for `Fr_17_Eray_GetNumOfStartupFrames` API

Service ID	0x27	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_NumOfStartupFramesPtr	Address where the number of startup frames seen within the last even/odd cycle pair should be stored.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the current number of startup frames seen on the cluster. For FlexRay 2.1 Rev A compliant hardware, the driver always assumes 2 startup frames available.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_CCSV(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.24 Fr_17_Eray_GetChannelStatus

Table 129 Specification for `Fr_17_Eray_GetChannelStatus` API

Syntax	`Std_ReturnType  Fr_17_Eray_GetChannelStatus`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  uint16 * const Fr_ChannelAStatusPtr,`
	`  uint16 * const Fr_ChannelBStatusPtr`
	`)`

Service ID	0x28
Sync/Async	Synchronous
Safety Level	Refer to the release notes for the safety related info

(table continues...)

Table 129 (continued) Specification for `Fr_17_Eray_GetChannelStatus` API

Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_ChannelAStatusPtr Fr_ChannelBStatusPtr	Address where the bitcoded channel A status information must be stored. Address where the bitcoded channel B status information must be stored.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the channel status information. The function gets the aggregated channel status, NIT status, symbol window status and then resets the aggregated channel status information.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_ACS(rw), ERAY_CCSV(r), ERAY_SWNIT(ex_r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.25 Fr_17_Eray_GetClockCorrection

Table 130 Specification for `Fr_17_Eray_GetClockCorrection` API

Syntax	`Std_ReturnType  Fr_17_Eray_GetClockCorrection`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  sint16 * const Fr_RateCorrectionPtr,`
	`  sint32 * const Fr_OffsetCorrectionPtr`
	`)`

Service ID	0x29
Sync/Async	Synchronous
Safety Level	Refer to the release notes for the safety related info
Re-entrancy	Non Reentrant for the same device

(table continues...)

Table 130 (continued) Specification for `Fr_17_Eray_GetClockCorrection` API

Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_RateCorrectionPtr Fr_OffsetCorrectionPtr	Address where the current rate correction value must be stored. Address where the current offset correction value must be stored.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the current clock correction values (vRateCorrection and vOffsetCorrection).	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_OCV(ex_r), ERAY_RCV(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.26 Fr_17_Eray_GetSyncFrameList

Table 131 Specification for `Fr_17_Eray_GetSyncFrameList` API

Syntax	`Std_ReturnType  Fr_17_Eray_GetSyncFrameList`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint8 Fr_ListSize,`
	`  uint16 * const Fr_ChannelAEvenListPtr,`
	`  uint16 * const Fr_ChannelBEvenListPtr,`
	`  uint16 * const Fr_ChannelAOddListPtr,`
	`  uint16 * const Fr_ChannelBOddListPtr`
	`)`

Service ID	0x2a
Sync/Async	Synchronous
Safety Level	Refer to the release notes for the safety related info
Re-entrancy	Non Reentrant for the same device

(table continues...)

Table 131 (continued) Specification for `Fr_17_Eray_GetSyncFrameList` API

Parameters (in)	Fr_CtrlIdx Fr_ListSize	Index of FlexRay CC within the context of the FlexRay driver. Size of the arrays passed via parameters: Fr_ChannelAEvenListPtr Fr_ChannelBEvenListPtr Fr_ChannelAOddListPtr Fr_ChannelBOddListPtr. The service must ensure to not write more entries into those arrays than granted by this parameter.
Parameters (out)	Fr_ChannelAEvenListPtr Fr_ChannelBEvenListPtr Fr_ChannelAOddListPtr Fr_ChannelBOddListPtr	Address the list of syncframes on channel A within the even communication cycle is written to. The exact number of elements written to the list is limited by parameter Fr_ListSize. Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen. Address the list of syncframes on channel B within the even communication cycle is written to. The exact number of elements written to the list is limited by parameter Fr_ListSize. Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen. Address the list of syncframes on channel A within the odd communication cycle is written to. The exact number of elements written to the list is limited by parameter Fr_ListSize. Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen. Address the list of syncframes on channel B within the odd communication cycle is written to. The exact number of elements written to the list is limited by parameter Fr_ListSize. Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets a list of syncframes received or transmitted on channel A and channel B via the even and odd communication cycle.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_FRAMELIST_SIZE	
Configuration dependencies	-	
User hints	None	

(table continues...)

Table 131 (continued) Specification for `Fr_17_Eray_GetSyncFrameList` API

SFR accessed	ERAY_ESID(ex_r), ERAY_OSID(ex_r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.27 Fr_17_Eray_GetWakeupRxStatus

Table 132 Specification for `Fr_17_Eray_GetWakeupRxStatus` API

Syntax	`Std_ReturnType  Fr_17_Eray_GetWakeupRxStatus`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  uint8 * const Fr_WakeupRxStatusPtr`
	`)`	

Service ID	0x2b	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
Parameters (out)	Fr_WakeupRxStatusPtr	Address where bit coded wakeup reception status must be stored. Bit 0: Wakeup received on channel A indicator Bit 1: Wakeup received on channel B indicator Bit 2-7: Unused
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Gets the wakeup received information from the FlexRay controller. After the wakeup received information is read, this function resets the wakeup received indication status information.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	

(table continues...)

Table 132 (continued) Specification for `Fr_17_Eray_GetWakeupRxStatus` API

SFR accessed	ERAY_SIR(rw) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.28 Fr_17_Eray_SetAbsoluteTimer

Table 133 Specification for `Fr_17_Eray_SetAbsoluteTimer` API

Syntax	`Std_ReturnType  Fr_17_Eray_SetAbsoluteTimer`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint8 Fr_AbsTimerIdx,`
	`  const uint8 Fr_Cycle,`
	`  const uint16 Fr_Offset`
	`)`	

Service ID	0x11	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_AbsTimerIdx Fr_Cycle Fr_Offset	Index of FlexRay CC within the context of the FlexRay driver. Index of absolute timer within the context of the FlexRay CC. Absolute cycle the timer elapses in. Offset within cycle Fr_Cycle in units of macrotick the timer elapses at.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Sets the absolute FlexRay timer according to the parameters Fr_Cycle and Fr_Offset.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INV_TIMER_IDX, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_INV_CYCLE, FR_17_ERAY_E_INV_OFFSET, FR_17_ERAY_E_INV_POCSTATE	
Configuration dependencies	-	
User hints	None	

(table continues...)

Table 133 (continued) Specification for `Fr_17_Eray_SetAbsoluteTimer` API

SFR accessed	ERAY_CCSV(r), ERAY_GTUC02(r), ERAY_T0C(w) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.29 Fr_17_Eray_CancelAbsoluteTimer

Table 134 Specification for `Fr_17_Eray_CancelAbsoluteTimer` API

Syntax	`Std_ReturnType  Fr_17_Eray_CancelAbsoluteTimer`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint8 Fr_AbsTimerIdx`
	`)`	

Service ID	0x13	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_AbsTimerIdx	Index of FlexRay CC within the context of the FlexRay driver. Index of absolute timer within the context of the FlexRay CC.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Stops an absolute timer.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INV_TIMER_IDX, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_SIER(w), ERAY_T0C(w) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.30 Fr_17_Eray_EnableAbsoluteTimerIRQ

Table 135 Specification for `Fr_17_Eray_EnableAbsoluteTimerIRQ` API

Syntax	`Std_ReturnType Fr_17_Eray_EnableAbsoluteTimerIRQ ( const uint8 Fr_CtrlIdx, const uint8 Fr_AbsTimerIdx )`	
Service ID	0x15	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_AbsTimerIdx	Index of FlexRay CC within the context of the FlexRay driver. Index of absolute timer within the context of the FlexRay CC.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Enables the interrupt line of an absolute timer.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INV_TIMER_IDX, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_SIES(ex_w), ERAY_SIR(w) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.31 Fr_17_Eray_AckAbsoluteTimerIRQ

Table 136 Specification for `Fr_17_Eray_AckAbsoluteTimerIRQ` API

Syntax	`Std_ReturnType Fr_17_Eray_AckAbsoluteTimerIRQ ( const uint8 Fr_CtrlIdx, const uint8 Fr_AbsTimerIdx )`
Service ID	0x17

(table continues...)

Table 136 (continued) Specification for `Fr_17_Eray_AckAbsoluteTimerIRQ` API

Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_AbsTimerIdx	Index of FlexRay CC within the context of the FlexRay driver. Index of absolute timer within the context of the FlexRay CC.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Resets the interrupt condition of an absolute timer.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INV_TIMER_IDX, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_SIR(w) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.32 Fr_17_Eray_DisableAbsoluteTimerIRQ

Table 137 Specification for `Fr_17_Eray_DisableAbsoluteTimerIRQ` API

Syntax	`Std_ReturnType  Fr_17_Eray_DisableAbsoluteTimerIRQ`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint8 Fr_AbsTimerIdx`
	`)`	

Service ID	0x19	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_AbsTimerIdx	Index of FlexRay CC within the context of the FlexRay driver. Index of absolute timer within the context of the FlexRay CC.

(table continues...)

Table 137 (continued) Specification for `Fr_17_Eray_DisableAbsoluteTimerIRQ` API

Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Disables the interrupt line of an absolute timer.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INV_TIMER_IDX, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX	
Configuration dependencies	-	
User hints	None	
SFR accessed	ERAY_SIER(w) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.33 Fr_17_Eray_GetAbsoluteTimerIRQStatus

Table 138 Specification for `Fr_17_Eray_GetAbsoluteTimerIRQStatus` API

Syntax	`Std_ReturnType  Fr_17_Eray_GetAbsoluteTimerIRQStatus`
	`(`
	`  const uint8 Fr_CtrlIdx,`
	`  const uint8 Fr_AbsTimerIdx,`
	`  boolean * const Fr_IRQStatusPtr`
	`)`	

Service ID	0x20	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_AbsTimerIdx	Index of FlexRay CC within the context of the FlexRay driver. Index of absolute timer within the context of the FlexRay CC.
Parameters (out)	Fr_IRQStatusPtr	Address the output value is stored to.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors

(table continues...)

Table 138 (continued) Specification for `Fr_17_Eray_GetAbsoluteTimerIRQStatus` API

Description	Gets the IRQ status of an absolute timer.
Source	AUTOSAR
Error handling	FR_17_ERAY_E_INV_TIMER_IDX, FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_PARAM_POINTER
Configuration dependencies	-
User hints	None.
SFR accessed	ERAY_SIR(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.34 Fr_17_Eray_GetVersionInfo

Table 139 Specification for `Fr_17_Eray_GetVersionInfo` API

Syntax	`void Fr_17_Eray_GetVersionInfo ( Std_VersionInfoType * const VersioninfoPtr )`	
Service ID	0x1b	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Reentrant	
Parameters (in)	-	-
Parameters (out)	VersioninfoPtr	Address where the version information of the FR module must be stored.
Parameters (in - out)	-	-
Return	void	-
Description	Returns the version information of the FR module. The version information includes: - Module Id - Vendor Id - Vendor specific version numbers.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_PARAM_POINTER	

(table continues...)

Table 139 (continued) Specification for `Fr_17_Eray_GetVersionInfo` API

Configuration dependencies	FrVersionInfoApi
User hints	None
SFR accessed	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.35 Fr_17_Eray_ReadCCConfig

Table 140 Specification for `Fr_17_Eray_ReadCCConfig` API

Syntax	`Std_ReturnType Fr_17_Eray_ReadCCConfig ( const uint8 Fr_CtrlIdx, const uint8 Fr_ConfigParamIdx, uint32 * const Fr_ConfigParamValuePtr )`	
Service ID	0x2e	
Sync/Async	Synchronous	
Safety Level	Refer to the release notes for the safety related info	
Re-entrancy	Non Reentrant for the same device	
Parameters (in)	Fr_CtrlIdx Fr_ConfigParamIdx	Index of FlexRay CC within the context of the FlexRay driver. Index that identifies the configuration parameter to read. See macros FR_CIDX_[config_parameter_name].
Parameters (out)	Fr_ConfigParamValuePtr	Address the output value is stored to.
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
Description	Reads a FlexRay protocol configuration parameter for a particular FlexRay controller out of the configuration of the module.	
Source	AUTOSAR	
Error handling	FR_17_ERAY_E_INIT_FAILED, FR_17_ERAY_E_INV_CONFIG_IDX, FR_17_ERAY_E_INV_CTRL_IDX, FR_17_ERAY_E_PARAM_POINTER	
Configuration dependencies	-	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.4 Notifications and Callbacks

The FlexRay driver does not provide any notification and callbacks.

1.3.5 Scheduled functions

The FlexRay driver does not provide any scheduled functions.

1.3.6 Interrupt service routines

The FlexRay driver does not provide any interrupt handlers.

1.3.7 Callout

The FlexRay driver does not provide any callout function.

1.3.8 Errors Handling

This section describes the various errors reported by the FlexRay driver.

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers.	AUTOSAR	0x01	DET	0x01	DET
FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure	AUTOSAR	Value Assigned by DEM	Production Error	Value Assigned by DEM	Production Error
FRIF_E_LPDU_SLOTSTATUS: Flexray Protocol communication error - Slot Error (configured in FrIf Module)	AUTOSAR	Value Assigned by DEM	Production Error	Value Assigned by DEM	Production Error
FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list.	AUTOSAR	0x02	DET	0x02	DET
FR_17_ERAY_E_INV_OFFSET: Parameter offset exceeds bounds.	AUTOSAR	0x03	DET	0x03	DET
FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index.	AUTOSAR	0x04	DET	0x04	DET
FR_17_ERAY_E_INV_CHNL_IDX: Invalid channel index.	AUTOSAR	0x05	DET	0x05	DET
FR_17_ERAY_E_INV_CYCLE: Parameter cycle exceeds 63.	AUTOSAR	0x06	DET	0x06	DET
FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.	AUTOSAR	0x08	DET	0x08	DET

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state.	AUTOSAR	0x09	DET	0x09	RUNTIME
FR_17_ERAY_E_INV_LENGTH: Payload length parameter has an invalid value.	AUTOSAR	0x0A	DET	0x0A	DET
FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index.	AUTOSAR	0x0B	DET	0x0B	DET
FR_17_ERAY_E_INV_HEADERCRC: Invalid FlexRay header CRC.	AUTOSAR	0x0C	DET	0x0C	DET
FR_17_ERAY_E_INV_CONFIG_IDX: Invalid value passed as parameter Fr_ConfigParamIdx.	AUTOSAR	0x0D	DET	0x0D	DET
FR_17_ERAY_E_INV_FRAMELIST_SIZE: Invalid framelist size value.	AUTOSAR	0x0E	DET	0x0E	DET

1.3.9 Deviations and limitations

This section describes the deviations and limitations of the FlexRay driver.

1.3.9.1 Deviations

This section describes the deviations of the FlexRay driver.

1.3.9.1.1 Software specification deviations

This section describes the deviations from software specification.

Table 141 Known deviations

Reference	Deviation
For all requirements related to Production/Runtime errors	Reporting of Production error: Dem_ReportErrorStatus is done through Mcal_Wrapper_Dem_ReportErrorStatus interface for AUTOSAR 4.2.2 and Dem_SetEventStatus is done through Mcal_Wrapper_Dem_SetEventStatus interface for AUTOSAR 4.4.0. Reporting of Runtime error: Det_ReportRuntimeError is done through Mcal_Wrapper_Det_ReportRuntimeError interface. This is applicable for AUTOSAR 4.4.0. All production and runtime related datatypes and modified interfaces inclusion shall be done via Mcal_Wrapper.h

1.3.9.1.2 AMDC Violations

The FlexRay driver does not have any AMDC violations.

1.3.9.1.3 VSMD Violations

This section describes the violations reported by the EB VSMD checker tool with respect to AUTOSAR.

Table 142 Violations reported by VSMD checker tool for EB03

Rule ID:	EB03
VSMD Node(s):	/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrControllerDemEventParameterRefs / AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrControllerDemEventParameterRefs/ FR_E_CTRL_TESTRESULT /AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrCtrlEcucPartitionRef
Description:	The StMD node has LOWER-MULTIPLICITY=0 and UPPER-MULTIPLICITY=1. The VSMD-node shall get the OPTIONAL-attribute instead of creating a list!
Additional Information:	-

Table 143 Violations reported by VSMD checker tool for EB09

Rule ID:	EB09
VSMD Node(s):	/AURIX2G/EcucDefs/Fr
Description:	EB specific rule to check consistency of parameter postBuildVariantUsed.
Additional Information:	-

Table 144 Violations reported by VSMD checker tool for EcucSws_1014

Rule ID:	EcucSws_1014
VSMD Node(s):	/AURIX2G/EcucDefs/Fr/AURIX2G/EcucDefs/Fr/ FrGeneral/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ FrPdMicrotick
Description:	Additional vendor specific parameter definitions (using ParameterTypes), container definitions and references shall be added to the VSMD according to the alphabetical order.
Additional Information:	-

Table 145 Violations reported by VSMD checker tool for EcucSws_1035

Rule ID:	EcucSws_1035

(table continues...)

Table 145 (continued) Violations reported by VSMD checker tool for EcucSws_1035

VSMD Node(s):	/AURIX2G/EcucDefs/Fr/AURIX2G/EcucDefs/Fr/ FrGeneral/AURIX2G/EcucDefs/Fr/FrGeneral/ FrCtrlTestCount/AURIX2G/EcucDefs/Fr/FrGeneral/ FrDevErrorDetect/AURIX2G/EcucDefs/Fr/FrGeneral/ FrDisableLPduSupport/AURIX2G/EcucDefs/Fr/ FrGeneral/FrEcucPartitionRef/AURIX2G/EcucDefs/Fr/ FrGeneral/FrExtendedLPduReporting/AURIX2G/ EcucDefs/Fr/FrGeneral/FrIndex/AURIX2G/EcucDefs/Fr/ FrGeneral/FrNumCtrlSupported/AURIX2G/ EcucDefs/Fr/FrGeneral/FrPrepareLPduSupport/ AURIX2G/EcucDefs/Fr/FrGeneral/ FrReconfigLPduSupport/AURIX2G/EcucDefs/Fr/ FrGeneral/FrRxStringentCheck/AURIX2G/EcucDefs/Fr/ FrGeneral/FrRxStringentLengthCheck/AURIX2G/ EcucDefs/Fr/FrGeneral/FrVersionInfoApi/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrAbsoluteTimer/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrAbsoluteTimer/FrAbsTimerIdx/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ FrControllerDemEventParameterRefs/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ FrControllerDemEventParameterRefs/ FR_E_CTRL_TESTRESULT/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrCtrlEcucPartitionRef/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrCtrlIdx/ AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrFifo/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrAdmitWithoutMessageId/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrBaseCycle/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrChannels/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrCycleRepetition/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrFifoDepth/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrMsgIdMask/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/ FrMsgIdMatch/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrFifo/FrRange/ AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrFifo/FrRange/FrRangeMax/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/

(table continues...)

Table 145 (continued) Violations reported by VSMD checker tool for EcucSws_1035

	FrFifo/FrRange/FrRangeMin/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPAllowHaltDueToClock/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPAllowPassiveToActive/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrPChannels/ AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrPClusterDriftDamping/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ FrPDecodingCorrection/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPDelayCompensationA/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPDelayCompensationB/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPExternalSync/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPFallBackInternal/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrPKeySlotId/ AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrPKeySlotOnlyEnabled/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ FrPKeySlotUsedForStartup/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPKeySlotUsedForSync/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/FrPLatestTx/ AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/ FrController/FrPMacroInitialOffsetA/AURIX2G/ EcucDefs/Fr/FrMultipleConfiguration/FrController/ FrPMacroInitialOffsetB/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPMicroInitialOffsetA/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPMicroInitialOffsetB/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPMicroPerCycle/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPNmVectorEarlyUpdate/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPOffsetCorrectionOut/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPOffsetCorrectionStart/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPPayloadLengthDynMax/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPRateCorrectionOut/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/ FrPSamplesPerMicrotick/AURIX2G/EcucDefs/Fr/ FrMultipleConfiguration/FrController/

(table continues...)

Table 145 (continued) Violations reported by VSMD checker tool for EcucSws_1035

	FrPSecondKeySlotId/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/FrController/FrPTwoKeySlotMode/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/FrController/FrPWakeupChannel/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/FrController/FrPWakeupPattern/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/FrController/FrPdAcceptedStartupRange/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/FrController/FrPdListenTimeout/AURIX2G/EcucDefs/Fr/FrMultipleConfiguration/FrController/FrPdMicrotick
Description:	For Containers, Parameters and References elements UUID must be unique (also between StMD and VSMD).
Additional Information:	-

Table 146 Violations reported by VSMD checker tool for EcucSws_2101

Rule ID:	EcucSws_2101
VSMD Node(s):	/AURIX2G/EcucDefs/Fr/POST_BUILD_VARIANT_USED
Description:	For each ConfigurationVariant supported by the ModuleDef, there must be one ImplementationConfigClass element. In VSMD, the ImplementationConfigClass is mandatory.
Additional Information:	-

Table 147 Violations reported by VSMD checker tool for EcucSws_6003

Rule ID:	EcucSws_6003
VSMD Node(s):	/AURIX2G/EcucDefs/Fr
Description:	The SHORT-NAME of the AR-PACKAGEs of StMD and VSMD must be different to ensure a unique SHORT-NAME-path.
Additional Information:	-

Table 148 Violations reported by VSMD checker tool for TpsEcuc_06051_ASR41

Rule ID:	TpsEcuc_06051_ASR41
VSMD Node(s):	/AURIX2G/EcucDefs/Fr/POST_BUILD_VARIANT_USED
Description:	The implementationConfigClass of an EcucParameterDef or EcucAbstractReferenceDef in VSMD shall be the same or higher (where PreCompile configuration class is considered to be the lowest and PostBuild the highest) as in StMD with respect to the selected subset defined by the actually implemented supportedConfigVariant.
Additional Information:	-

1.3.9.2 Limitations

This section describes the limitations of the FlexRay driver.

Table 149 Known limitations

Reference	Limitation
Hardware buffer reconfiguration feature	When the number of LPdus exceed 128 or the total message RAM size required for LPdus exceeds the hardware message RAM size, then the LPdus which are part of the static segment (except the keyslot frame) configured with the communication action PREPARE_LPDU in the FrIf configuration are considered for hardware buffer reconfiguration. The hardware buffers which participate in buffer reconfiguration feature are allocated to a maximum of 4 LPdus.
Configuration of Key slot LPdu in FrIf module	If a key slot frame is present for a node, then key slot frame should be configured with the parameter `FrIfLPduIdx` set to a value 0 within the FrIf configuration. This is required for the generation of the correct data offset addresses within the FR driver.
LPDU count in all Post Build configurations	Pre compile macro 'FR_17_ERAY_MSG_BUFF_COUNT_MAX_0' considers the LPdus count only in the present configuration and not the max value across all Post-build configurations. Hint: By configuring the variant which is having maximum number of configured LPdus as the last one, the macro FR_17_ERAY_MSG_BUFF_COUNT_MAX_0 will be generated with max number of LPdus count. However user has to verify this manually by checking the macro value.

Revision history

Table 150 Revision History

Date	Version	Description
2024-08-07	8.0	Document is released
2024-07-22	7.1	In Section 1.3.9.2: usage of Number of supported LPdus information is removed from Limitation Section.
2023-06-30	7.0	Document is released
2023-06-12	6.1	• Updated the section 1.1.2.Hardware-software mapping to include Mcal_Wrapper module and removed Dem module. • Updated the Section 1.1.3.C file Structure to include Mcal_Wrapper.h and removed Dem.h. • In Section 1.1.4.1.Integration with AUTOSAR stack, the following points are modified - Instead of DEM Module, Mcal_Wrapper Section is added. - Moved Runtime Error description from DET to Mcal_Wrapper Module. • Updated the Description of 1.3.1.6.FrControllerDemEventParameterRefs container with Mcal_Wrapper API information. • DEM has been modified to Production error in Section: 1.3.8.Error Handling. • Updated the section 1.3.9.1.1: Software Specification Deviations for Autosar requirements. - Updated Reference from "SWS_FR_00099: Rte_Dem_Types.h" to "For all requirements related to Production/Runtime errors". - Updated Description to add Mcal_Wrapper Module Information. • ASIL Level has been updated to Safety level in Section 1.3.3.Functions - APIs.
2022-08-05	6.0	Document is released
2022-08-05	5.1	Updated section 1.1.5.1 Buffer Reconfiguration with dependency to FrIfBaseCycle and FrIfCycleRepetition
2021-11-04	5.0	Document is released
2021-11-03	4.1	'Mapping of hardware-software interfaces' figure is corrected
2021-10-29	4.0	Document is released
2021-10-28	3.1	Config variant attribute table information is removed and added this information in 'Configuration interfaces' section.
2021-03-16	3.0	Document is released
2021-03-16	2.1	Updated Software Specification deviations section
2020-11-27	2.0	Document is released
2020-11-11	1.1	Updated Integration for AUTOSAR stack for DEM section
2020-08-13	1.0	Document is released.

(table continues...)

Table 150 (continued) Revision History

2020-08-04	0.1	• Initial version • FR driver chapter moved from MC-ISAR_TC3xx_UM_COM-E to this document • Added Transmission Conflict Detection in key architectural considerations • Updated Port Support section for TC3E7 device in Integration hints • Updated AMDC and VSMD violations in Deviations section

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-08-07

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2024 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-ocr1484806431059

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.