

# MCAL User Manual for COM-E package

## 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.

#### Intended audience

This document is intended for anyone using the COM-E package of the TC3xx MCAL software.

#### Document conventions

**Table 1** Conventions

Convention	Explanation
<b>Bold</b>	Emphasizes heading levels, column headings, table and figure captions, screen names, windows, dialog boxes, menus, sub-menus
<i>Italics</i>	Denote variable(s) and reference(s)
Courier	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, code snippets, file/folder names, directories, command line inputs
New	
Hyperlink	Provides quick and easy access to cross-referenced topics/sections
>	Indicates that a cascading sub-menu opens when you choose a menu item

#### Glossary of terms

**Table 2** Glossary

Term	Description
AUTOSAR	Automotive Open System Architecture
BSW	Basic software
Channel	A channel is a software exchange medium for data that are defined with the same criteria: configuration parameters, number of data elements with same size and data pointers (source, destination) or location.
CHI	Communication host interface
CMU	Clock management unit (functional block of GTM)
DEM	Diagnostics event manager (MCAL module)
DET	Development error tracer
DF_EEPROM	Data Flash dedicated for EEPROM emulation
DFDBER	Double bit error
DFlash	Data Flash
DFSBER	Single bit error
DFTBER	Triple bit error

## About this document

**Table 2**      **Glossary (continued)**

Term	Description
Driver	A driver is a BSW module located in the MCAL layer and contains the functionality to control and access an internal or an external device.
ECC	Error correction code
EEPROM	Electrically erasable and programmable ROM (read only memory)
ERAY	FlexRay IP module (hardware)
ERU	External request unit
ESR	External service request (microcontroller pin)
ETH	Ethernet
EVADC	Enhanced versatile analog-to-digital converter
EVER	Bit indicating erase verify error
Fast-Mode	Triggering the watchdog hardware has to be done with a short timeout period. This mode can be used during normal operations of the ECU. For example, the watchdog hardware is configured for the Window mode (triggering the watchdog should occur within certain minimum / maximum boundaries within the timeout period) and a timeout period of 5 ms.
FCE	Flexible CRC engine
FEE	Flash EEPROM emulation
FIFO	First in first out
GC	Garbage collection
GETH	Gigabit Ethernet MAC
GTM	Generic timer module (hardware)
HOH	Hardware object (transmit/receive) handle
HRH	Hardware receive handle
HSCT	High-speed communications tunnel
HSPDM	High-speed pulse density modulation module
I/O	Input/Output
IB	Driver defined internal buffer / channel
IFX	Infineon Technologies
ISR	Interrupt service routine
Job	A job is composed of one or several channels with the same chip select (is not released during the processing of job). A job is considered atomic and therefore cannot be interrupted by another job. A job has an assigned priority.
(Logical) block	Smallest erasable unit (4 K) as seen by the module user. Consists of one or more virtual pages.
LPdu	Datalink layer protocol data unit
LPM	Low power mode
MAC	Media access control
MC-ISAR	Microcontroller Infineon Software Architecture
MCU	Microcontroller unit

## About this document

**Table 2**                      **Glossary (continued)**

Term	Description
MII	Media independent interface
Module	The element is composed of various software units called modules. Typically each software driver is referred to as module. More explicitly, it is also referred to as software module.
MRST	Master receive slave transmit
MTL	MAC transaction layer
MTSR	Master transmit slave receive
(Normal) write mode	In this mode, the maximum amount of data that can be written with one command is 8 byte (1 Page).
NVM	AUTOSAR NVRAM manager
NVRAM	Non-volatile RAM (random access memory)
NVRAM block	Management unit as seen by the NVRAM manager
Off-Mode	Watchdog hardware is disabled / shut-down
OS	Operating system
Page	A page is an aligned group of data double words plus an ECC extension. It is the smallest unit that can be programmed. DFlash: 1 data double word (8 bytes) plus 22-bit ECC extension.
Peripheral	Hardware module used by a driver. A driver can use one or more peripherals.
PHY	Physical layer device (Ethernet transceiver)
Physical address	Address information in device-specific format (depending on the underlying Flash driver and device) that is used to access a logical block.
Physical sector	It is a combination of 256 logical sectors (in DFlash), which comprises of 1 MB memory.
PLL	Phase lock loop
Pn_xxxxx	Port n register (xxxxx is register name)
POC	Protocol operation control
PORST	Power-on reset
PVER	Bit indicating program verify error
QS	Quasi-static
QSPI	Queued serial peripheral interface
r	Read access
RGMII	Reduced gigabit media independent interface
RMII	Reduced media independent interface
rw	Read and write access
SchM	Scheduler manager (AUTOSAR module)
SCU	System control unit
Sequence	A sequence is a number of consecutive jobs to transmit but it can be rescheduled between jobs using a priority mechanism. A sequence transmission is interruptible (by another sequence transmission) or not depending on a static configuration.
SER	Source error

## About this document

**Table 2**                      **Glossary (continued)**

Term	Description
Slow-Mode	Triggering the watchdog hardware can be done with a long timeout period. This mode can be used during system start-up / initialization phase. For example, the watchdog hardware is configured for toggle mode (no constraints on the point in time at which the triggering is done) and a timeout period of 10 ms.
SLSO	Programmable slave select outputs
SMU	Safety management unit
SPB	System peripheral bus
SRC	Service request control register
SRI	System resource interconnect
TECQED	Triple-bit error correction and quad-bit error detection
TIM	Timer input module (functional block of a microcontroller peripheral)
TOM	Timer output module (functional block of a microcontroller peripheral)
Unconfigured block	Data block which is stored in the DFlash (DF_EEPROM), but is not contained in the currently active configuration
User job	User requested read/write/invalidate job
VariantLT	This variant allows a mix of pre-compile time, link-time configuration parameters. The intention of this variant is to optimize the parameters configuration for an object code delivery.
VariantPB	This variant allows a mix of pre-compile time, post-build time and link time configuration parameters. The intention of this variant is to optimize the parameters configuration for a re-loadable binary.
VarinatPC	This variant allows only pre-compile configuration parameters. The intention of this variant is to optimize the parameters configuration for a source code delivery.
Virtual address	Consisting of 16-bit block number and 16-bit offset inside the logical block.
Virtual page	May consist of one or several physical pages to ease handling of logical blocks and address calculation
w	Write access
Wordline (WL)	An aligned group of bytes. In DFLASH, 512 bytes are in the single-ended mode and 256 bytes are in the complement-sensing mode.

## Reference documents

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

- AURIX™ TC3xx MCAL User Manual for BASIC package, V1.30.0\_9.0, 2019-10-10, Infineon Technologies Munich AG
- AURIX™ TC3xx User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- AURIX™ TC38x Appendix to User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- AURIX™ TC39x-B Appendix to User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- AURIX™ TC35x Appendix to User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- AURIX™ TC37xEXT Appendix to User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- AURIX™ TC37x Appendix to User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- AURIX™ TC36x Appendix to User Manual, V1.2.0, 2019-04, Infineon Technologies Munich AG
- TC35x\_AA\_Errata\_Sheet, Rel1.2, 2019-07-19, Infineon Technologies Munich AG
- TC35x\_AB\_Errata\_Sheet, Rel1.1, 2019-07-19, Infineon Technologies Munich AG

---

**About this document**

- TC37xEXT\_AA\_Errata\_Sheet, Rel1.2, 2019-07-19, Infineon Technologies Munich AG
- TC37xEXT\_AB\_Errata\_Sheet, Rel1.1, 2019-07-19, Infineon Technologies Munich AG
- TC39x\_BC\_Errata\_Sheet, Rel1.2, 2019-07-19, Infineon Technologies Munich AG
- TC39x\_BB\_Errata\_Sheet, Rel1.3, 2019-07-19, Infineon Technologies Munich AG
- TC39x\_BA\_Errata\_Sheet, Rel1.5, 2019-07-19, Infineon Technologies Munich AG
- TC39x\_AA\_Errata\_Sheet, Rel1.8, 2019-03-29, Infineon Technologies Munich AG
- TC38x\_AD\_Errata\_Sheet, Rel1.2, 2019-07-19, Infineon Technologies Munich AG
- TC38x\_AC\_Errata\_Sheet, Rel1.2, 2019-07-19, Infineon Technologies Munich AG
- TC38x\_AB\_Errata\_Sheet, Rel1.3, 2019-07-19, Infineon Technologies Munich AG
- TC38x\_AA\_Errata\_Sheet, Rel1.5, 2019-07-19, Infineon Technologies Munich AG
- Specification of FlexRay Driver, AUTOSAR\_SWS\_FlexRayDriver.pdf, AUTOSAR Release 4.2.2
- Specification of FlexRay Interface, AUTOSAR\_SWS\_FlexRayInterface.pdf, AUTOSAR Release 4.2.2
- Specification of Ethernet Driver, AUTOSAR\_SWS\_EthernetDriver.pdf, AUTOSAR Release 4.2.2

---

**Table of contents**
**Table of contents**

	<b>About this document</b> .....	1
	<b>Table of contents</b> .....	6
<b>1</b>	<b>Eth_17_GEthMac driver</b> .....	15
1.1	User information .....	15
1.1.1	Description .....	15
1.1.2	Hardware-software mapping .....	15
1.1.2.1	GETH: primary hardware peripheral .....	15
1.1.2.2	Port: dependent hardware peripheral .....	16
1.1.2.3	SCU: dependent hardware peripheral .....	17
1.1.2.4	SRC: dependent hardware peripheral .....	17
1.1.3	File structure .....	17
1.1.3.1	C file structure .....	17
1.1.3.2	Code generator plugin files .....	19
1.1.4	Integration hints .....	20
1.1.4.1	Integration with AUTOSAR stack .....	20
1.1.4.2	Multicore and Resource Manager .....	25
1.1.4.3	MCU support .....	25
1.1.4.4	Port support .....	25
1.1.4.5	DMA support .....	27
1.1.4.6	Interrupt connections .....	28
1.1.4.7	Example usage .....	29
1.1.5	Key architectural considerations .....	32
1.1.5.1	ETH controller initialization sequence .....	32
1.1.5.2	Eth_17_GEthMac_SetControllerMode API implemented as synchronous .....	33
1.1.5.3	Development error checks added to avoid undefined behavior .....	33
1.1.5.4	Time support APIs report ETH_E_ACCESS production error .....	33
1.1.5.5	Multicore support for ETH driver .....	33
1.1.5.6	Specific hardware features used for nominal operation of ETH driver .....	34
1.2	Assumptions of Use (AoUs) .....	35
1.3	Reference information .....	36
1.3.1	Configuration interfaces .....	36
1.3.1.1	Container: CommonPublishedInformation .....	36
1.3.1.1.1	ArMajorVersion .....	36
1.3.1.1.2	ArMinorVersion .....	37
1.3.1.1.3	ArPatchVersion .....	37
1.3.1.1.4	ModuleId .....	38
1.3.1.1.5	Release .....	38
1.3.1.1.6	SwMajorVersion .....	39
1.3.1.1.7	SwMinorVersion .....	39

---

**Table of contents**

1.3.1.1.8	SwPatchVersion .....	39
1.3.1.1.9	VendorApilInfix .....	40
1.3.1.1.10	VendorId .....	40
1.3.1.2	Container: Eth .....	41
1.3.1.3	Container: EthConfigSet .....	41
1.3.1.4	Container: EthCtrlConfig .....	41
1.3.1.4.1	EthCRSDVRMIInput .....	41
1.3.1.4.2	EthCarrierSenseMIInput .....	42
1.3.1.4.3	EthCollisionMI .....	42
1.3.1.4.4	EthCtrlEnableCrcStripping .....	43
1.3.1.4.5	EthCtrlEnableMii .....	43
1.3.1.4.6	EthCtrlEnableRxInterrupt .....	44
1.3.1.4.7	EthCtrlEnableTxInterrupt .....	45
1.3.1.4.8	EthCtrlIdx .....	45
1.3.1.4.9	EthCtrlPhyAddress .....	46
1.3.1.4.10	EthCtrlRxBufLenByte .....	46
1.3.1.4.11	EthCtrlTxBufLenByte .....	47
1.3.1.4.12	EthMdioAlternateInput .....	47
1.3.1.4.13	EthOpMode .....	48
1.3.1.4.14	EthPhyInterface .....	48
1.3.1.4.15	EthRecDataValidMIInput .....	49
1.3.1.4.16	EthReceiveData0Input .....	50
1.3.1.4.17	EthReceiveData1Input .....	50
1.3.1.4.18	EthReceiveData2Input .....	51
1.3.1.4.19	EthReceiveData3Input .....	51
1.3.1.4.20	EthRefClkRMIInput .....	52
1.3.1.4.21	EthRxBufTotal .....	53
1.3.1.4.22	EthRxErrMIInput .....	53
1.3.1.4.23	EthRxclkInput .....	54
1.3.1.4.24	EthSkewRxClockDelay .....	54
1.3.1.4.25	EthSkewTxClockDelay .....	55
1.3.1.4.26	EthSpeed .....	55
1.3.1.4.27	EthTxBufTotal .....	56
1.3.1.4.28	EthTxClockMIInput .....	56
1.3.1.5	Container: EthCtrlOffloading .....	57
1.3.1.5.1	EthCtrlEnableOffloadChecksumICMP .....	57
1.3.1.5.2	EthCtrlEnableOffloadChecksumIPv4 .....	58
1.3.1.5.3	EthCtrlEnableOffloadChecksumTCP .....	58
1.3.1.5.4	EthCtrlEnableOffloadChecksumUDP .....	59
1.3.1.6	Container: EthDemEventParameterRefs .....	60
1.3.1.6.1	ETH_E_ACCESS .....	60
1.3.1.6.2	ETH_E_ALIGNMENT .....	60

---

**Table of contents**

1.3.1.6.3	ETH_E_CRC .....	61
1.3.1.6.4	ETH_E_LATECOLLISION .....	61
1.3.1.6.5	ETH_E_MULTIPLECOLLISION .....	62
1.3.1.6.6	ETH_E_OVERSIZEFRAME .....	62
1.3.1.6.7	ETH_E_RX_FRAMES_LOST .....	63
1.3.1.6.8	ETH_E_SINGLECOLLISION .....	63
1.3.1.6.9	ETH_E_UNDERSIZEFRAME .....	64
1.3.1.7	Container: EthGeneral .....	64
1.3.1.7.1	EthDevErrorDetect .....	64
1.3.1.7.2	EthGetDropCountApi .....	65
1.3.1.7.3	EthGetEtherStatsApi .....	65
1.3.1.7.4	EthGlobalTimeSupport .....	66
1.3.1.7.5	EthIndex .....	66
1.3.1.7.6	EthInitApiMode .....	67
1.3.1.7.7	EthMainFunctionPeriod .....	67
1.3.1.7.8	EthMaxCtrlsSupported .....	68
1.3.1.7.9	EthMultiCoreErrorDetect .....	68
1.3.1.7.10	EthOperationFrequency .....	69
1.3.1.7.11	EthPeripheralBusClock .....	69
1.3.1.7.12	EthRuntimeApiMode .....	70
1.3.1.7.13	EthTimeoutCount .....	71
1.3.1.7.14	EthUpdatePhysAddrFilter .....	71
1.3.1.7.15	EthVersionInfoApi .....	71
1.3.2	Functions - Type definitions .....	72
1.3.2.1	Eth_17_GEthMac_ConfigType .....	72
1.3.2.2	Eth_BufIdxType .....	72
1.3.2.3	Eth_DataType .....	73
1.3.2.4	Eth_FilterActionType .....	73
1.3.2.5	Eth_FrameType .....	73
1.3.2.6	Eth_ModeType .....	74
1.3.2.7	Eth_RateRatioType .....	74
1.3.2.8	Eth_ReturnType .....	74
1.3.2.9	Eth_RxStatusType .....	75
1.3.2.10	Eth_StateType .....	75
1.3.2.11	Eth_TimeIntDiffType .....	75
1.3.2.12	Eth_TimeStampQualType .....	76
1.3.2.13	Eth_TimeStampType .....	76
1.3.3	Functions - APIs .....	76
1.3.3.1	Eth_17_GEthMac_Init .....	76
1.3.3.2	Eth_17_GEthMac_SetControllerMode .....	77
1.3.3.3	Eth_17_GEthMac_GetControllerMode .....	78
1.3.3.4	Eth_17_GEthMac_GetPhysAddr .....	79



---

**Table of contents**

1.3.3.5	Eth_17_GEthMac_SetPhysAddr .....	80
1.3.3.6	Eth_17_GEthMac_UpdatePhysAddrFilter .....	81
1.3.3.7	Eth_17_GEthMac_WriteMii .....	83
1.3.3.8	Eth_17_GEthMac_ReadMii .....	84
1.3.3.9	Eth_17_GEthMac_GetDropCount .....	85
1.3.3.10	Eth_17_GEthMac_GetEtherStats .....	87
1.3.3.11	Eth_17_GEthMac_GetCurrentTime .....	88
1.3.3.12	Eth_17_GEthMac_EnableEgressTimeStamp .....	89
1.3.3.13	Eth_17_GEthMac_GetEgressTimeStamp .....	90
1.3.3.14	Eth_17_GEthMac_GetIngressTimeStamp .....	92
1.3.3.15	Eth_17_GEthMac_SetCorrectionTime .....	93
1.3.3.16	Eth_17_GEthMac_SetGlobalTime .....	94
1.3.3.17	Eth_17_GEthMac_ProvideTxBuffer .....	95
1.3.3.18	Eth_17_GEthMac_Transmit .....	96
1.3.3.19	Eth_17_GEthMac_Receive .....	97
1.3.3.20	Eth_17_GEthMac_TxConfirmation .....	98
1.3.3.21	Eth_17_GEthMac_GetVersionInfo .....	99
1.3.4	Notifications and Callbacks .....	100
1.3.5	Scheduled functions .....	100
1.3.5.1	Eth_17_GEthMac_MainFunction .....	100
1.3.6	Interrupt service routines .....	101
1.3.6.1	Eth_17_GEthMac_RxDmaCh0IrqHdlr .....	101
1.3.6.2	Eth_17_GEthMac_TxDmaCh0IrqHdlr .....	102
1.3.7	Error codes classification .....	103
1.3.7.1	Development errors .....	103
1.3.7.2	Production errors .....	106
1.3.7.3	Safety errors .....	107
1.3.7.4	Runtime errors .....	107
1.3.8	Deviations and limitations .....	107
1.3.8.1	Deviations .....	107
1.3.8.2	Limitations .....	108
1.3.9	Unsupported hardware features .....	109
<b>2</b>	<b>Fr_17_Eray driver .....</b>	<b>111</b>
2.1	User information .....	111
2.1.1	Description .....	111
2.1.2	Hardware-software mapping .....	111
2.1.2.1	ERAY: primary hardware peripheral .....	112
2.1.2.2	SRC: dependent hardware peripheral .....	112
2.1.2.3	Port: dependent hardware peripheral .....	113
2.1.2.4	SCU: dependent hardware peripheral .....	113
2.1.3	File structure .....	113

---

**Table of contents**

2.1.3.1	C file structure .....	113
2.1.3.2	Code generator plugin files .....	115
2.1.4	Integration hints .....	117
2.1.4.1	Integration with AUTOSAR stack .....	117
2.1.4.2	Multicore and Resource Manager .....	120
2.1.4.3	MCU support .....	120
2.1.4.4	Port support .....	121
2.1.4.5	DMA support .....	124
2.1.4.6	Interrupt connections .....	124
2.1.4.7	Example usage .....	125
2.1.5	Key architectural considerations .....	127
2.1.5.1	Buffer reconfiguration .....	127
2.1.5.2	BSW scheduler mechanism .....	128
2.1.5.3	Clock configuration .....	128
2.1.5.4	Input channel selection .....	129
2.1.5.5	Additional receive FIFO: related parameters .....	129
2.1.5.6	Configuration parameter for timeout event .....	129
2.1.5.7	Get absolute timer status .....	129
2.1.5.8	Configuration parameter for the <code>Fr_GetNmVector</code> API .....	129
2.1.5.9	User mode support .....	129
2.2	Assumptions of Use (AoUs) .....	130
2.3	Reference information .....	131
2.3.1	Configuration interfaces .....	131
2.3.1.1	Container: Fr .....	131
2.3.1.1.1	Config Variant .....	131
2.3.1.2	Container: FrGeneral .....	132
2.3.1.2.1	FrCtrlTestCount .....	132
2.3.1.2.2	FrDevErrorDetect .....	132
2.3.1.2.3	FrDisableLPduSupport .....	133
2.3.1.2.4	FrIndex .....	134
2.3.1.2.5	FrInitApiMode .....	134
2.3.1.2.6	FrNmVectorEnable .....	135
2.3.1.2.7	FrNumCtrlSupported .....	135
2.3.1.2.8	FrPrepareLPduSupport .....	136
2.3.1.2.9	FrReconfigLPduSupport .....	136
2.3.1.2.10	FrRxStringentCheck .....	137
2.3.1.2.11	FrRxStringentLengthCheck .....	137
2.3.1.2.12	FrTimeoutDurationFactor .....	138
2.3.1.2.13	FrTxConflictDetection .....	138
2.3.1.2.14	FrVersionInfoApi .....	139
2.3.1.3	Container: FrMultipleConfiguration .....	139
2.3.1.4	Container: FrController .....	139

---

**Table of contents**

2.3.1.4.1	FrCtrlIdx .....	140
2.3.1.4.2	FrPAllowHaltDueToClock .....	140
2.3.1.4.3	FrPAllowPassiveToActive .....	141
2.3.1.4.4	FrPChannels .....	141
2.3.1.4.5	FrPClusterDriftDamping .....	142
2.3.1.4.6	FrPDecodingCorrection .....	142
2.3.1.4.7	FrPDelayCompensationA .....	143
2.3.1.4.8	FrPDelayCompensationB .....	143
2.3.1.4.9	FrPExternalSync .....	144
2.3.1.4.10	FrPFallBackInternal .....	144
2.3.1.4.11	FrPKeySlotId .....	145
2.3.1.4.12	FrPKeySlotOnlyEnabled .....	145
2.3.1.4.13	FrPKeySlotUsedForStartup .....	146
2.3.1.4.14	FrPKeySlotUsedForSync .....	147
2.3.1.4.15	FrPLatestTx .....	147
2.3.1.4.16	FrPMacroInitialOffsetA .....	148
2.3.1.4.17	FrPMacroInitialOffsetB .....	148
2.3.1.4.18	FrPMicroInitialOffsetA .....	149
2.3.1.4.19	FrPMicroInitialOffsetB .....	149
2.3.1.4.20	FrPMicroPerCycle .....	150
2.3.1.4.21	FrPNmVectorEarlyUpdate .....	150
2.3.1.4.22	FrPOffsetCorrectionOut .....	151
2.3.1.4.23	FrPOffsetCorrectionStart .....	151
2.3.1.4.24	FrPPayloadLengthDynMax .....	152
2.3.1.4.25	FrPRateCorrectionOut .....	152
2.3.1.4.26	FrPSamplesPerMicrotick .....	153
2.3.1.4.27	FrPSecondKeySlotId .....	153
2.3.1.4.28	FrPTwoKeySlotMode .....	154
2.3.1.4.29	FrPWakeupChannel .....	154
2.3.1.4.30	FrPWakeupPattern .....	155
2.3.1.4.31	FrPdAcceptedStartupRange .....	156
2.3.1.4.32	FrPdListenTimeout .....	156
2.3.1.4.33	FrPdMicrotick .....	157
2.3.1.4.34	FrRxInputSelectionA .....	157
2.3.1.4.35	FrRxInputSelectionB .....	158
2.3.1.5	Container: FrAbsoluteTimer .....	158
2.3.1.5.1	FrAbsTimerIdx .....	159
2.3.1.6	Container: FrControllerDemEventParameterRefs .....	159
2.3.1.6.1	FR_E_CTRL_TESTRESULT .....	159
2.3.1.7	Container: FrFifo .....	160
2.3.1.7.1	FrAdmitWithoutMessageId .....	160
2.3.1.7.2	FrBaseCycle .....	160

---

**Table of contents**

2.3.1.7.3	FrChannels .....	161
2.3.1.7.4	FrCycleRepetition .....	162
2.3.1.7.5	FrFifoDepth .....	162
2.3.1.7.6	FrFrameIdRejectionFilter .....	163
2.3.1.7.7	FrFrameIdRejectionFilterMask .....	163
2.3.1.7.8	FrMsgIdMask .....	164
2.3.1.7.9	FrMsgIdMatch .....	164
2.3.1.7.10	FrRejectNullFrames .....	164
2.3.1.7.11	FrRejectStaticSegment .....	165
2.3.1.8	Container: FrRange .....	166
2.3.1.8.1	FrRangeMax .....	166
2.3.1.8.2	FrRangeMin .....	166
2.3.1.9	Container: FrClockConfiguration .....	167
2.3.1.9.1	FrClockDivider .....	167
2.3.1.10	Container: CommonPublishedInformation .....	167
2.3.1.10.1	ArMajorVersion .....	167
2.3.1.10.2	ArMinorVersion .....	168
2.3.1.10.3	ArPatchVersion .....	168
2.3.1.10.4	ModuleId .....	169
2.3.1.10.5	Release .....	169
2.3.1.10.6	SwMajorVersion .....	169
2.3.1.10.7	SwMinorVersion .....	170
2.3.1.10.8	SwPatchVersion .....	170
2.3.1.10.9	VendorApilInfix .....	171
2.3.1.10.10	VendorId .....	171
2.3.2	Functions - Type definitions .....	172
2.3.2.1	Fr_17_Eray_ConfigType .....	172
2.3.2.2	Fr_POCTestType .....	172
2.3.2.3	Fr_SlotModeType .....	172
2.3.2.4	Fr_ErrorModeType .....	173
2.3.2.5	Fr_WakeupStatusType .....	173
2.3.2.6	Fr_StartupStateType .....	174
2.3.2.7	Fr_POCTestStatusType .....	174
2.3.2.8	Fr_TxLPduStatusType .....	175
2.3.2.9	Fr_RxLPduStatusType .....	175
2.3.2.10	Fr_ChannelType .....	175
2.3.2.11	FR_CIDX_<CONFIGPARAM_NAME> .....	176
2.3.2.12	FR_SLOTMODE_SINGLE .....	179
2.3.3	Functions - APIs .....	179
2.3.3.1	Fr_17_Eray_Init .....	180
2.3.3.2	Fr_17_Eray_ControllerInit .....	180
2.3.3.3	Fr_17_Eray_StartCommunication .....	181

---

**Table of contents**

2.3.3.4	Fr_17_Eray_AllowColdstart .....	182
2.3.3.5	Fr_17_Eray_AllSlots .....	183
2.3.3.6	Fr_17_Eray_HaltCommunication .....	184
2.3.3.7	Fr_17_Eray_AbortCommunication .....	185
2.3.3.8	Fr_17_Eray_SendWUP .....	186
2.3.3.9	Fr_17_Eray_SetWakeupChannel .....	187
2.3.3.10	Fr_17_Eray_GetPOCStatus .....	188
2.3.3.11	Fr_17_Eray_TransmitTxLPdu .....	189
2.3.3.12	Fr_17_Eray_CancelTxLPdu .....	190
2.3.3.13	Fr_17_Eray_ReceiveRxLPdu .....	190
2.3.3.14	Fr_17_Eray_CheckTxLPduStatus .....	192
2.3.3.15	Fr_17_Eray_PrepareLPdu .....	193
2.3.3.16	Fr_17_Eray_ReconfigLPdu .....	194
2.3.3.17	Fr_17_Eray_DisableLPdu .....	195
2.3.3.18	Fr_17_Eray_GetGlobalTime .....	196
2.3.3.19	Fr_17_Eray_GetNmVector .....	197
2.3.3.20	Fr_17_Eray_GetNumOfStartupFrames .....	198
2.3.3.21	Fr_17_Eray_GetChannelStatus .....	198
2.3.3.22	Fr_17_Eray_GetClockCorrection .....	199
2.3.3.23	Fr_17_Eray_GetSyncFrameList .....	200
2.3.3.24	Fr_17_Eray_GetWakeupRxStatus .....	202
2.3.3.25	Fr_17_Eray_SetAbsoluteTimer .....	203
2.3.3.26	Fr_17_Eray_CancelAbsoluteTimer .....	204
2.3.3.27	Fr_17_Eray_EnableAbsoluteTimerIRQ .....	205
2.3.3.28	Fr_17_Eray_AckAbsoluteTimerIRQ .....	206
2.3.3.29	Fr_17_Eray_DisableAbsoluteTimerIRQ .....	207
2.3.3.30	Fr_17_Eray_GetAbsoluteTimerIRQStatus .....	207
2.3.3.31	Fr_17_Eray_GetVersionInfo .....	208
2.3.3.32	Fr_17_Eray_ReadCCConfig .....	209
2.3.4	Notifications and Callbacks .....	210
2.3.5	Scheduled functions .....	210
2.3.6	Interrupt service routines .....	210
2.3.7	Error codes classification .....	210
2.3.7.1	Development errors .....	211
2.3.7.2	Production errors .....	215
2.3.7.3	Safety errors .....	216
2.3.7.4	Runtime errors .....	216
2.3.8	Deviations and limitations .....	216
2.3.8.1	Deviations .....	216
2.3.8.2	Limitations .....	216
2.3.9	Unsupported hardware features .....	217

---

Table of contents

Revision history .....	218
Disclaimer .....	219

## Eth\_17\_GEthMac driver

# 1 Eth\_17\_GEthMac driver

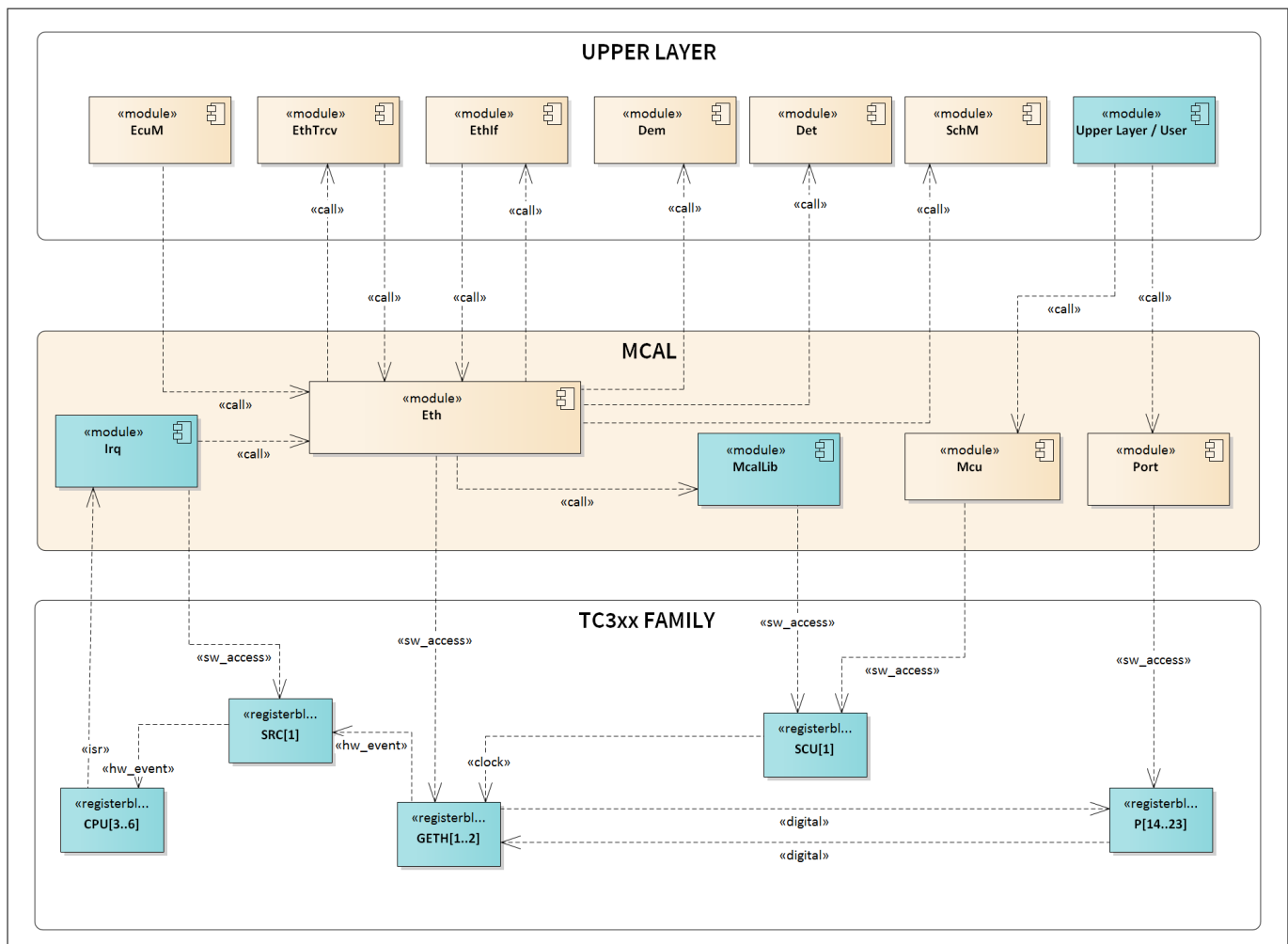
## 1.1 User information

### 1.1.1 Description

The Ethernet (ETH) driver is responsible for providing standard ETH controller services specified by AUTOSAR. This enables the upper layer (ETH interface) to access the underlying bus system in a uniform manner. The ETH driver provides functionality for configuration, initialization, data transmission and reception. The ETH driver provides optional features such as checksum offloading, time stamping, updating the physical source address, read or write interface to ETH transceiver, ETH statistics and ETH drop count.

### 1.1.2 Hardware-software mapping

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



**Figure 1** Mapping of hardware-software interfaces

#### 1.1.2.1 GETH: primary hardware peripheral

##### Hardware functional features

The ETH driver uses the GETH for initialization, configuration and data transmission.

---

**Eth\_17\_GEthMac driver**

The key hardware functional features used by the driver are:

- Data transmission speed supported are 10, 100 and 1000 Mbps
- PHY interfaces supported are MII, RMII and RGMII
- Full duplex and half duplex modes are supported for data transmission
- Preamble and start of packet data (SFD) insertion/deletion
- Automatic CRC and pad generation/stripping options
- Up to 32 layer 2 (MAC) address filtering
- ETH frame time stamp (supports IEEE 1588-2008 for precision networked clock synchronization)
- Checksum offloading for IPV4, ICMP, TCP and UDP frames
- Embedded DMA in the ETH controller is used for data exchanges between the ETH controller and the system memory
- Single DMA channel and single MTL queue from GETH MAC hardware is used
- MTL queue is built from FIFO memory available in GETH MAC for transmission and reception of size 4 Kbyte and 8 Kbyte respectively. Entire FIFO memory can be split to be configured upto 4 queues. Each queue can be configured of different size in multiples of 256 bytes. ETH driver is designed to configure one transmit queue of size 4 Kbyte and one receive queue of size 8 Kbyte
- Standard ETH of frame size 1518 bytes
- Destination address filters
- Broadcast frames are always allowed irrespective of the filter status
- GETH is implemented as a 32 bit peripheral. Nevertheless it is connected to 64 bit wide bus (SRI)

The unsupported features of the ETH driver are:

- Single and double VLAN tagged frames
- Jumbo frames
- Source address filters are not supported
- Multiple DMA channels
- Multiple priority based for queues
- Loopback mode
- Energy efficient ETH (EEE)
- Pulse per second output
- Promiscuous mode

**Users of the hardware**

The ETH driver exclusively utilizes the GETH module.

**Hardware diagnostic features**

- SMU alarms configured for the GETH are not monitored by the ETH driver
- ETH packet drop counts and packet statistics are available through the ETH driver API services

**Hardware events**

The ETH driver uses the following hardware events from the GETH IP:

- Packet transmission complete
- Packet reception complete

**1.1.2.2 Port: dependent hardware peripheral****Hardware functional features**



---

**Eth\_17\_GEthMac driver**

The MII/RMII/RGMII and MDIO signals are routed to the transceiver through the port pads. These signals 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 ETH driver.

**1.1.2.3 SCU: dependent hardware peripheral****Hardware functional features**

The ETH driver depends on the SCU IP for the clock, ENDINIT and reset functionalities. The driver requires the fSPB, fSRI and fGETH clock signals for functioning. The fGETH defines the basic frequency for the GETH kernel. The fGETH is independent to fSPB and allows the GETH to operate at a constant baud rate.

**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 ETH driver.

**Hardware events**

Not applicable.

**1.1.2.4 SRC: dependent hardware peripheral****Hardware functional features**

The ETH driver depends on the interrupt router for raising an interrupt to the CPU based on the transmit and receive events, which indicates successful packet transmission and reception respectively.

**Users of the hardware**

The interrupt router is configured either by the IRQ driver or the user software. The ETH driver does not administer any functional block of the interrupt router.

**Hardware diagnostic features**

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

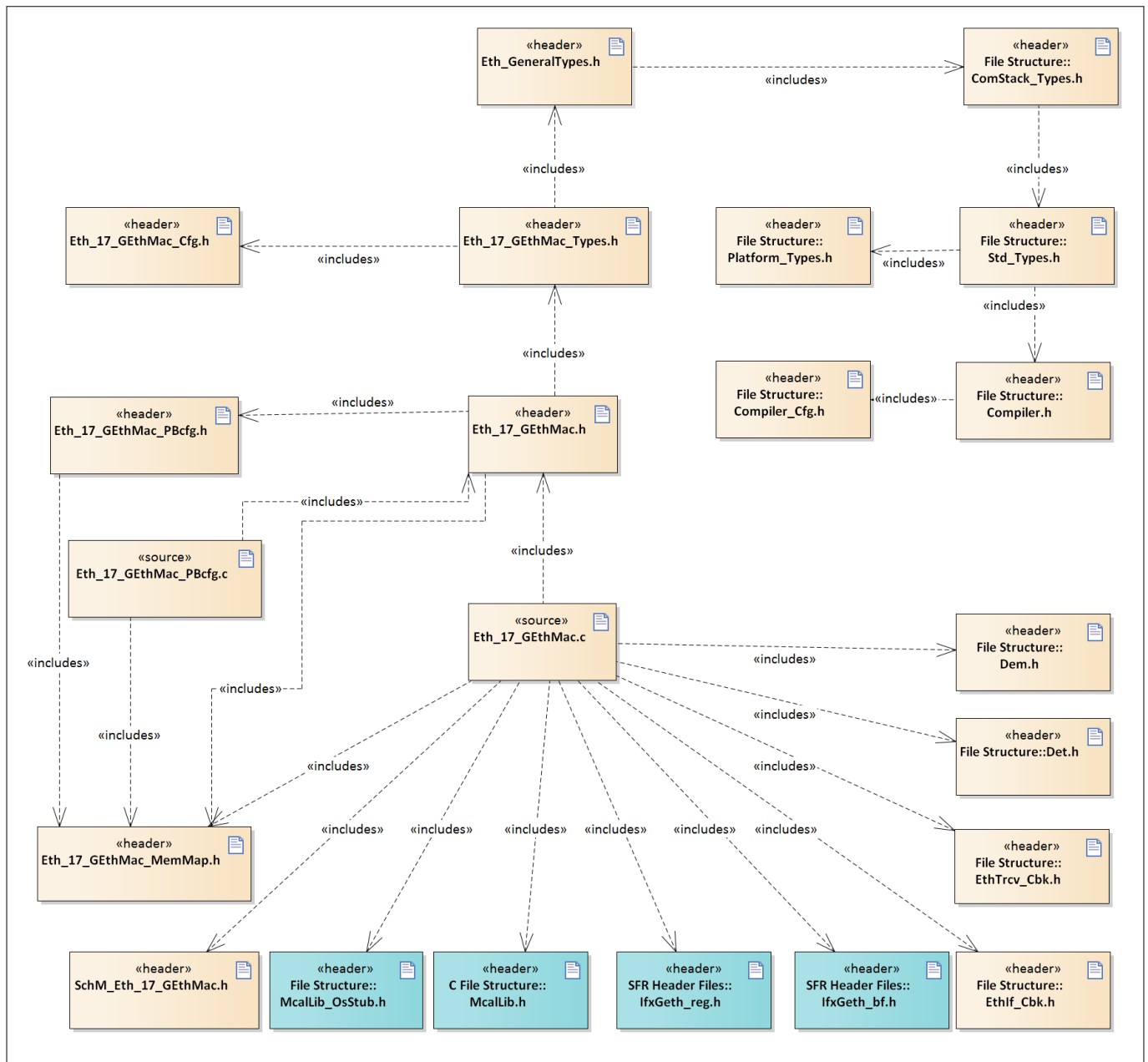
**Hardware events**

The interrupt events raised by the interrupt router are serviced by the CPU. The ETH driver provides interrupt handlers as software interfaces, which must be invoked from the ISR.

**1.1.3 File structure****1.1.3.1 C file structure**

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

### Eth\_17\_GEthMac driver



**Figure 2 C file structure**

**Table 3 C file structure**

File name	Description
ComStack_Types.h	Type Definition for Com stack
Compiler.h	Provides abstraction from compiler-specific keywords
Compiler_Cfg.h	Configuration header file for compiler abstraction
Dem.h	Provides the exported interfaces of Diagnostic Event Manager
Det.h	Provides the exported interfaces of Development Error Tracer
EthIf_Cbk.h	Contains the declarations of the callback functions to ETH Interface (EthIf) module

## Eth\_17\_GEthMac driver

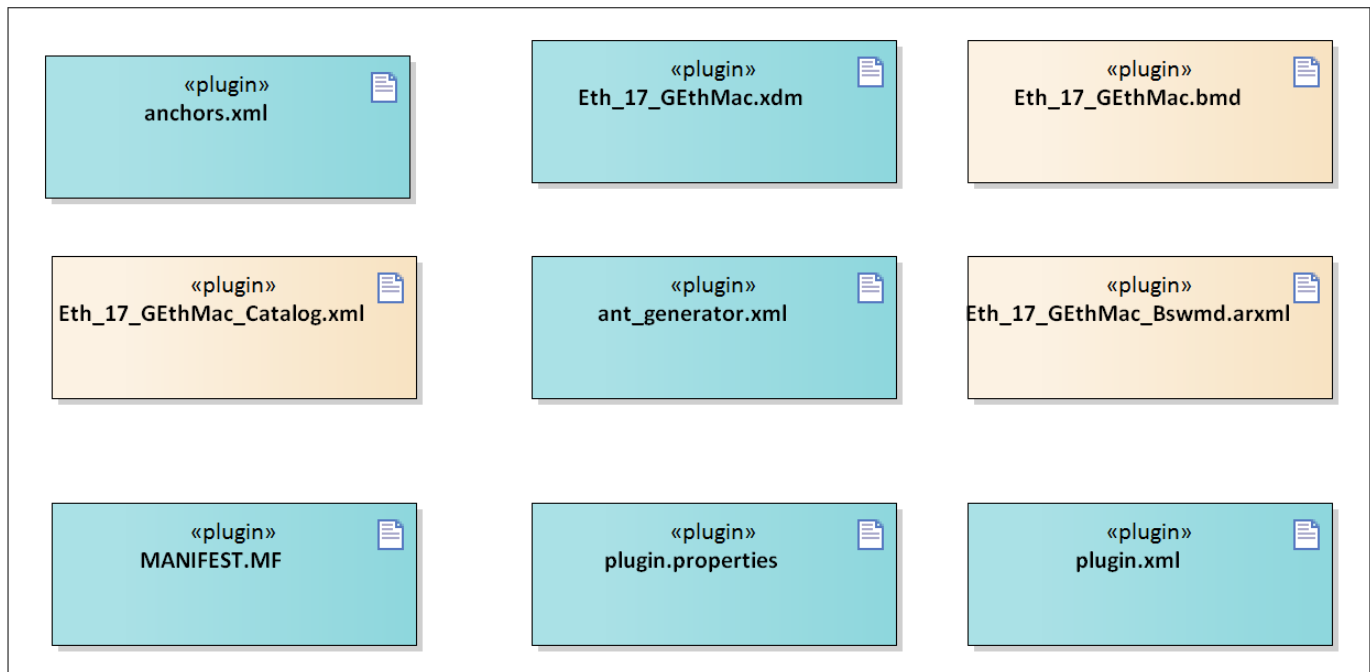
**Table 3 C file structure (continued)**

File name	Description
EthTrcv_Cbk.h	Contains the declarations of the callback functions to ETH Transceiver (EthTrcv) module
Eth_17_GEthMac.c	File (Static) containing implementation of APIs
Eth_17_GEthMac.h	Header file (Static) defining prototypes of data structures, APIs and interrupt handlers
Eth_17_GEthMac_Cfg.h	Header file (Generated) containing constants and pre-processor macros as #defines
Eth_17_GEthMac_MemMap.h	File (Static) containing the memory section definitions used by the ETH driver
Eth_17_GEthMac_PBcfg.c	File (Generated) containing definition of the configuration data structures
Eth_17_GEthMac_PBcfg.h	File (Generated) containing declaration of the post-build configuration data structures of ETH driver
Eth_17_GEthMac_Types.h	ETH driver specific type declaration file
Eth_GeneralTypes.h	ETH specific type declaration file as defined by AUTOSAR
IfxGeth_bf.h	SFR header file for GETH
IfxGeth_reg.h	SFR header file for GETH
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.
Platform_Types.h	Platform-specific type declaration file as defined by AUTOSAR
SchM_Eth_17_GEthMac.h	Export header for SchM functions of the ETH
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 ETH driver.

## Eth\_17\_GEthMac driver



**Figure 3** Code generator plugin files

**Table 4** Code generator plugin files

File name	Description
Eth_17_GEthMac.bmd	AUTOSAR format XML data model schema file
Eth_17_GEthMac.xdm	Tresos format XML data model schema file
Eth_17_GEthMac_Bswmd.arxml	AUTOSAR format module description file
Eth_17_GEthMac_Catalog.xml	AUTOSAR format catalog file
MANIFEST.MF	Tresos plugin support file containing the metadata for the ETH driver
anchors.xml	Tresos anchors support file for the ETH driver
ant_generator.xml	Tresos support file to generate and rename multiple post-build configuration when using variation point
plugin.properties	Tresos plugin support file for the ETH driver
plugin.xml	Tresos plugin support file for the ETH driver

### 1.1.4 Integration hints

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

#### 1.1.4.1 Integration with AUTOSAR stack

This section lists the modules, which are not part of MCAL, but are required to integrate the ETH 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, the EcuM is used for initialization and de-initialization of the

---

**Eth\_17\_GEthMac driver**

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.

- **ETH interface (EthIf)**

The EthIf module is part of the AUTOSAR stack that provides upper layers, a hardware independent interface, to the ETH communication system comprising multiple different ETH controllers.

The ETH driver uses the APIs of EthIf to provide transmit confirmation, indicate successful reception and indicate successful ETH controller mode change. The files' `EthIf_Cbk.c` and `EthIf_Cbk.h` are provided as stub code and needs to be replaced with complete EthIf module during integration phase.

- **ETH transceiver driver (EthTrcv)**

The EthTrcv module is part of the AUTOSAR stack that provides upper layers a hardware independent interface comprising multiple equal transceivers.

The ETH driver uses APIs of EthTrcv to indicate successful Media independent (MII) read and write access.

The files `EthTrcv_Cbk.c` and `EthTrcv_Cbk.h` are provided as stub code and needs to be replaced with complete EthTrcv module during integration phase.

- **Memory mapping**

Memory mapping is a concept from AUTOSAR that allows relocation of text, variables, constants and configuration data to user-specific memory regions. In order to achieve this, all the relocatable elements of the driver are encapsulated in different memory-section macros. These macros are defined in the file `Eth_17_GEthMac_MemMap.h`. The 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

## Eth\_17\_GEthMac driver

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

```

/**** GLOBAL DATA SECTION for CORE[x] , x(Core Id)= 0..5 ****/
#if defined ETH_17_GETHMAC_START_SEC_VAR_CLEARED_QM_CORE0_UNSPECIFIED
/* User Pragma here */
#undef ETH_17_GETHMAC_START_SEC_VAR_CLEARED_QM_CORE0_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_STOP_SEC_VAR_CLEARED_QM_CORE0_UNSPECIFIED
/* User Pragma here */
#undef ETH_17_GETHMAC_STOP_SEC_VAR_CLEARED_QM_CORE0_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_START_SEC_VAR_TXBUFFER_CLEARED_QM_CORE0_32
/* User Pragma here */
#undef ETH_17_GETHMAC_START_SEC_VAR_TXBUFFER_CLEARED_QM_CORE0_32
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_STOP_SEC_VAR_TXBUFFER_CLEARED_QM_CORE0_32
/* User Pragma here */
#undef ETH_17_GETHMAC_STOP_SEC_VAR_TXBUFFER_CLEARED_QM_CORE0_32
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_START_SEC_VAR_RXBUFFER_CLEARED_QM_CORE0_32
/* User Pragma here */
#undef ETH_17_GETHMAC_START_SEC_VAR_RXBUFFER_CLEARED_QM_CORE0_32
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_STOP_SEC_VAR_RXBUFFER_CLEARED_QM_CORE0_32
/* User Pragma here */
#undef ETH_17_GETHMAC_STOP_SEC_VAR_RXBUFFER_CLEARED_QM_CORE0_32
#undef MEMMAP_ERROR

/**** ETH MODULE CONFIG DATA ****/
#elif defined ETH_17_GETHMAC_START_SEC_CONFIG_DATA_QM_GLOBAL_UNSPECIFIED
/* User Pragma here */
#undef ETH_17_GETHMAC_START_SEC_CONFIG_DATA_QM_GLOBAL_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_STOP_SEC_CONFIG_DATA_QM_GLOBAL_UNSPECIFIED
/* User Pragma here */
#undef ETH_17_GETHMAC_STOP_SEC_CONFIG_DATA_QM_GLOBAL_UNSPECIFIED
#undef MEMMAP_ERROR

/**** CODE SECTION ****/
#elif defined ETH_17_GETHMAC_START_SEC_CODE_QM_GLOBAL
/* User Pragma here */
#undef ETH_17_GETHMAC_START_SEC_CODE_QM_GLOBAL
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_STOP_SEC_CODE_QM_GLOBAL
/* User Pragma here */
#undef ETH_17_GETHMAC_STOP_SEC_CODE_QM_GLOBAL
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_START_SEC_CODE_QM_LOCAL
/* User Pragma here */
#undef ETH_17_GETHMAC_START_SEC_CODE_QM_LOCAL
#undef MEMMAP_ERROR
#elif defined ETH_17_GETHMAC_STOP_SEC_CODE_QM_LOCAL

```

---

**Eth\_17\_GEthMac driver**

```
/* User Pragma here */
#undef ETH_17_GETHMAC_STOP_SEC_CODE_QM_LOCAL
#undef MEMMAP_ERROR

#endif

#if defined MEMMAP_ERROR
#error "Eth_17_EthMac_MemMap.h, wrong pragma command"
#endif
```

- **DET**

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

- **DEM**

The DEM module is a part of the AUTOSAR stack that handles all the production errors reported by the BSW modules. The ETH driver reports all the production errors to the DEM modules through the `Dem_ReportErrorStatus()` API. The user of the ETH driver must process all the production errors (fail / pass) reported to the DEM module through the `Dem_ReportErrorStatus()` API.

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

- **SchM**

The SchM module is a part of the RTE that manages the BSW Scheduler. The ETH driver uses the exclusive areas defined in the `SchM_Eth_17_GEthMac.c` file to protect the SFRs and variables from concurrent accesses from different threads. The SchMs identified for the ETH driver are:

- TransmitData
- UpdateGlobalTime

The `SchM_Eth_17_GEthMac.h` and `SchM_Eth_17_GEthMac.c` files are provided in the MCAL package as an example code and needs to be updated by the integrator. The user must implement the SchM functions

---

**Eth\_17\_GEthMac driver**

defined by the ETH driver as **suspend / resume** of interrupts for the CPU on which the API is invoked. A sample implementation of the SchM functions are shown as follows:

```
/**** Sample implementation of SchM_Eth_17_GEthMac.c ****/
#include "Os.h"

void SchM_Enter_Eth_17_GEthMac_TransmitData(void)
{
    SuspendAllInterrupts(); /* Suspend CPU core interrupt */
}

void SchM_Exit_Eth_17_GEthMac_TransmitData(void)
{
    ResumeAllInterrupts(); /* Resume CPU core interrupt */
}

void SchM_Enter_Eth_17_UpdateGlobalTime(void)
{
    SuspendAllInterrupts(); /* Suspend CPU core interrupt */
}

void SchM_Exit_Eth_17_UpdateGlobalTime(void)
{
    ResumeAllInterrupts(); /* Resume CPU core interrupt */
}
```

- **Safety error**

The ETH driver does not report any safety errors.

- **Notifications and callbacks**

The ETH driver itself does not implement any notifications. However, the ETH driver reports transmit confirmation, successful reception and controller mode change through notification functions of the EthIf module and successful Media independent accesses(MII) read / write access through notification functions of EthTrcv module.

- **Operating system (OS)**

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



## Eth\_17\_GEthMac driver

### 1.1.4.2 Multicore and Resource Manager

The ETH driver supports execution of its APIs simultaneously from all CPU cores. The user should allocate ETH controllers to the CPU cores at pre-compile time using the Resource Manager module. The following are the key points to be considered with respect to multicore in the driver:

- ETH controller can be allocated to CPU cores at pre-compile time. For example, `EthCtrlConfig_0`, `EthCtrlConfig_1`.
- It must be ensured that ETH controller id passed as parameter while invoking an API, belonging to the same core.
- DETs are raised in case APIs are invoked with mismatch of core and controller id.
- Locating constants, variables and configuration data to correct memory space should be done by the user. Memory sections are marked GLOBAL (common to all cores) and CORE[x](specific to a CPU core). The following should be considered by the user to ensure better performance of the driver:

#### Code section:

The executable code of ETH driver is placed under single MemMap section. It can be relocated to any PFlash.

#### Data section:

The RAM variable memory sections marked as specific to core, should be re-located to the DSPR/DLMU of the same core. The sections marked as global should be relocated to the non-cached LMU region.

#### Configuration data and constants:

The configuration data section sections marked as specific to core, should be re-located to the PFLASH of the same core. The sections marked as global should be relocated to the PFlash of the master core.

*Note: Relocating code, data and constants to a distant memory space would impact execution timings.*

### 1.1.4.3 MCU support

The ETH driver is dependent on the MCU driver for the generation of fSRI, fSPB and fGETH clocks. The initialization of the ETH driver must be started only after completing the MCU initialization. The fGETH defines the application clock frequency for the Gigabit ETH Kernel. The fGETH is independent to fSPB and allows the Gigabit ETH to operate at a constant baud rate (frequency). To configure fSRI, fSPB and fGETH clock frequencies, update the parameters `McuSRIFrequency`, `McuSPBFrequency` and `McuGethFrequency` from MCU driver configuration. The following must be considered while configuring MCU driver in EB Tresos:

- In the ETH controller, due to the requirement of simultaneous read and write transfers from the FIFO memories, the application clock frequency (fGETH) shall be such that the data transfer bandwidth of memory (fSRI) must be at least twice the bandwidth as that of the application clock frequency (fGETH). That is, fSRI must be greater than or equal to  $2 * fGETH$ .
- MCU initialization must be completed prior to invoking of the ETH driver initialization.

### 1.1.4.4 Port support

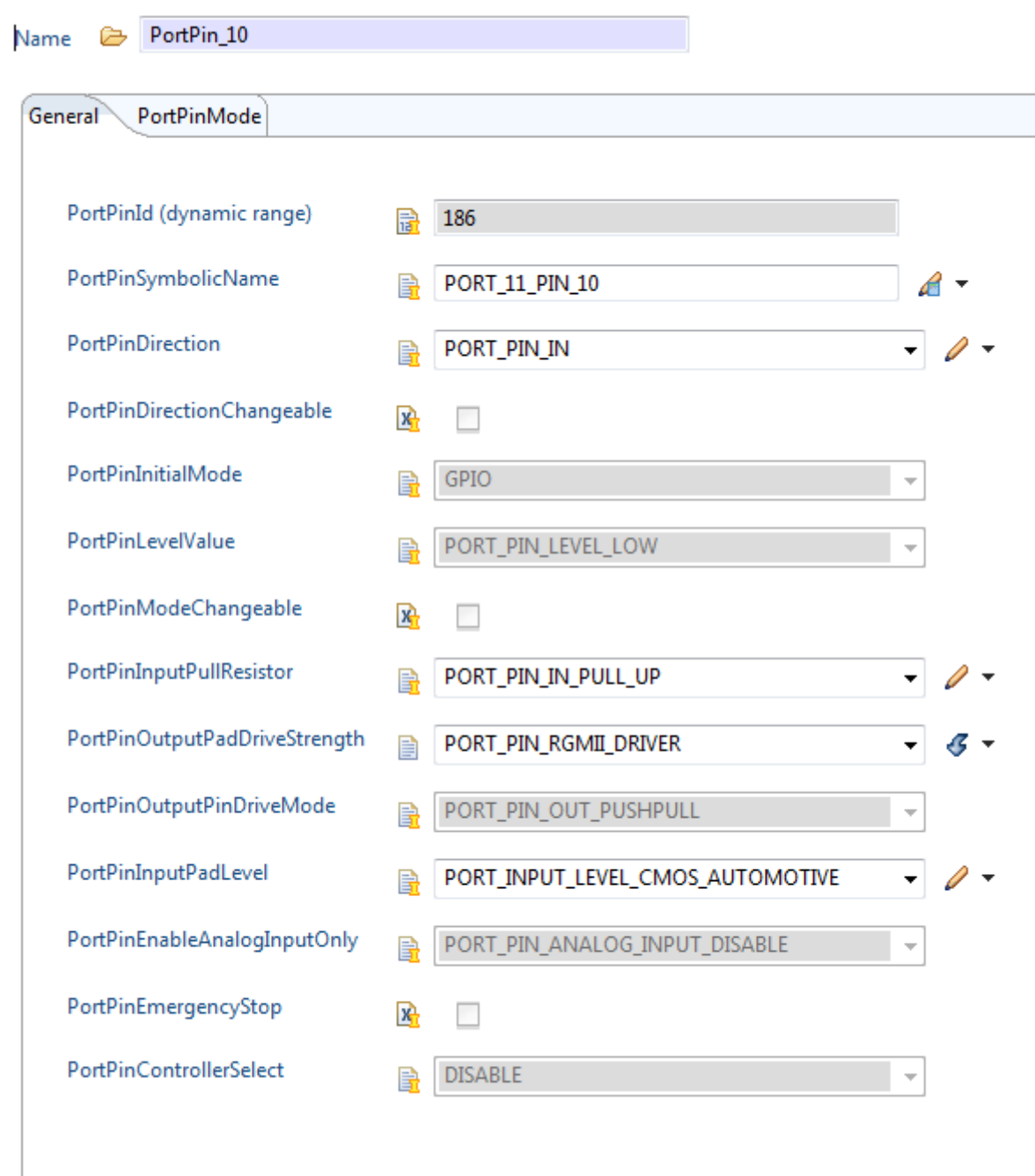
The PORT driver configures the port pins of the entire microcontroller. The user must configure port pins used by the ETH driver through the PORT configuration and initialize the PORT driver prior to invoking of the ETH driver initialization. The following must be considered while configuring PORT driver in EB Tresos:

- Configure all PORT pins that are used in the ETH driver for MII/RMII/RGMII and MDIO interface with PHY. That is, parameters such as `PortPinDirection` (input or output), `PortPinInitialMode` ( as GPIO for input pin or corresponding ALT option for output pins) and so on.

## Eth\_17\_GEthMac driver

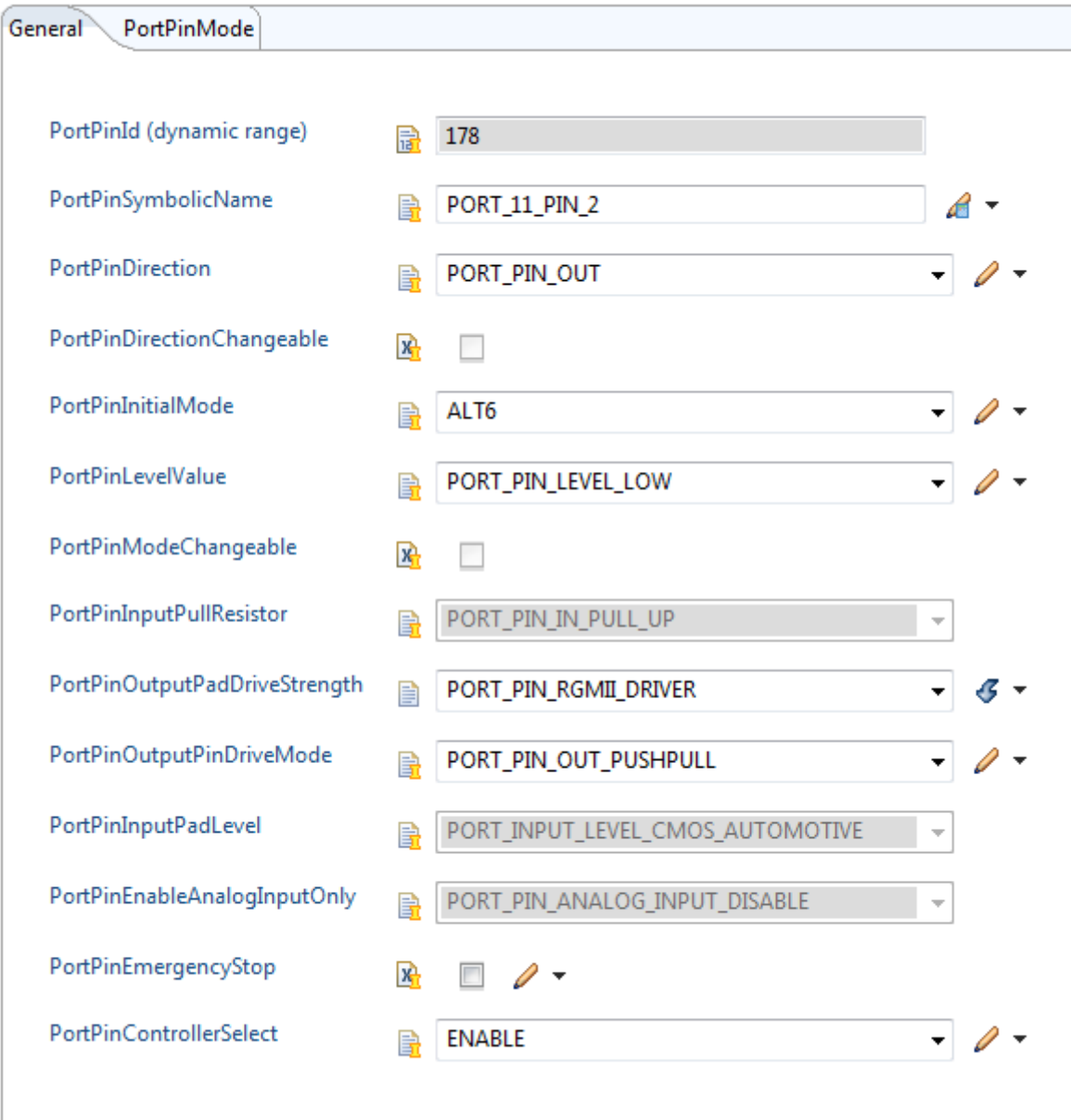
- For all output and input pins used by the ETH driver for MII/RMII/RGMII interface, the value of parameter `PortPinOutputPadDriveStrength` shall be configured as `PORT_PIN_RGMII_DRIVER`.
- For all output pins used by the ETH driver, the parameter `PortPinControllerSelect` shall be selected as `ENABLE`.

Refer to the following sample configurations for the PORT driver:



Parameter	Value
PortPinId (dynamic range)	186
PortPinSymbolicName	PORT_11_PIN_10
PortPinDirection	PORT_PIN_IN
PortPinDirectionChangeable	<input type="checkbox"/>
PortPinInitialMode	GPIO
PortPinLevelValue	PORT_PIN_LEVEL_LOW
PortPinModeChangeable	<input type="checkbox"/>
PortPinInputPullResistor	PORT_PIN_IN_PULL_UP
PortPinOutputPadDriveStrength	PORT_PIN_RGMII_DRIVER
PortPinOutputPinDriveMode	PORT_PIN_OUT_PUSH_PULL
PortPinInputPadLevel	PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE
PortPinEnableAnalogInputOnly	PORT_PIN_ANALOG_INPUT_DISABLE
PortPinEmergencyStop	<input type="checkbox"/>
PortPinControllerSelect	DISABLE

**Figure 4** Input pin configuration

**Eth\_17\_GEthMac driver**


The screenshot shows the 'PortPinMode' configuration window with the following settings:

Property	Value
PortPinId (dynamic range)	178
PortPinSymbolicName	PORT_11_PIN_2
PortPinDirection	PORT_PIN_OUT
PortPinDirectionChangeable	<input type="checkbox"/>
PortPinInitialMode	ALT6
PortPinLevelValue	PORT_PIN_LEVEL_LOW
PortPinModeChangeable	<input type="checkbox"/>
PortPinInputPullResistor	PORT_PIN_IN_PULL_UP
PortPinOutputPadDriveStrength	PORT_PIN_RGMII_DRIVER
PortPinOutputPinDriveMode	PORT_PIN_OUT_PUSH_PULL
PortPinInputPadLevel	PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE
PortPinEnableAnalogInputOnly	PORT_PIN_ANALOG_INPUT_DISABLE
PortPinEmergencyStop	<input type="checkbox"/>
PortPinControllerSelect	ENABLE

**Figure 5 Output pin configuration**

### 1.1.4.5 DMA support

The ETH controller has its own DMA in hardware and the ETH driver implements the necessary code to exercise this DMA. Therefore, the ETH driver does not use any services of the MCAL DMA available in the TC3xx device.

The following must be ensured by the user for proper functioning of the DMA controller:

- Address space 0xD and 0xC shall not be used for DMA-related usage. MemMap sections allocating memory in the scratch pad RAM shall always generate global address instead of local addresses.
- The memory address accessed by DMA shall be placed in non-cached memory.

*Note:* The variables defined in the ETH driver for transmit buffer, receive buffer, transmit DMA descriptor list and receive DMA descriptor list will be accessed by the DMA controller.

---

**Eth\_17\_GEthMac driver****1.1.4.6 Interrupt connections**

The interrupt connections of the ETH driver are described in this section.

- **Packet transmission complete interrupt from DMA channel-0 for controller Id 0**

When ETH transmission is configured in the interrupt mode and a requested packet transmission is completed, interrupt will be generated. In the ETH controller, the service request line number SRC\_GETH2 is used for transmission complete interrupt. User must ensure that the interrupt handler provided by ETH driver is called when packet transmission complete interrupt occurs. A sample invocation for transmission complete interrupt from DMA channel-0 is as follows:

```
/* Include Eth header file Eth.h */
#include "Eth_17_GEthMac.h"

/*****TX Interrupt from DMA Channel-0, Controller Id 0*****/
ISR(ETHSR2_ISR)
{
    /* Enable Global Interrupts */
    ENABLE();
    /* Invoke Irq handler from GETH module file */
    Eth_17_GEthMac_TxDmaCh0IrqHdlr(0);
}
```

- **Packet receive complete interrupt from DMA channel-0 for controller Id 0**

When ETH reception is configured in the interrupt mode and a packet is received by ETH controller, interrupt will be generated. In the ETH controller, service request line number SRC\_GETH6 is used for receive complete interrupt. User must ensure that the interrupt handler provided by the ETH driver is called when packet receive complete interrupt occurs. A sample invocation for receive complete interrupt from DMA channel-0 is as follows:

```
/* Include Eth header file Eth.h */
#include "Eth_17_GEthMac.h"

/*****RX Interrupt from DMA Channel-0, Controller Id 0*****/
ISR(ETHSR6_ISR)
{
    /* Enable Global Interrupts */
    ENABLE();
    /* Invoke Irq handler from GETH module file */
    Eth_17_GEthMac_RxDmaCh0IrqHdlr(0);
}
```

---

**Eth\_17\_GEthMac driver****1.1.4.7 Example usage****Configuration**

ETH driver must be configured before usage and configuration files are generated and made available during the software build process. To configure ETH driver, the following guidelines shall be followed properly.

*Note: User of the ETH driver must ensure that the `EthCtrlRxBufLenByte` parameter is configured as expected in the ETH bus (including the broadcast frames if any). If received packet size is more than what is configured, then such packets will be ignored by the ETH driver and receive notification will not be called.*

**Step1:** In the MCU driver, configure the following system clocks: fSRI, fSPB and fGETH.

**Step2:** In the PORT driver, for all the port pins that are used in the ETH driver as interface (MII/RMII/RGMII and MDIO) with PHY, configure the same in the PORT driver.

**Step3:** If the ETH driver is configured in the interrupt mode (for transmit and receive), configure the interrupt priority, type of service and interrupt type in IRQ driver. ETH driver uses interrupt line SRC\_GETH2 for transmission and SRC\_GETH6 for receive.

*Note:* Ensure MAC address is unique if more than one controller is configured.

**Step4:** In the ETH driver, select the required API configuration such as PHY interface (MII/RMII/RGMII), ETH speed (10/100/1000 Mbps), ETH operation mode (Half/Full duplex), alternate input selection and so on.

**Step5:** In the Resource Manager, allocate controller with Id 0 to Core0. (If this is not configured, by default the controller is assumed to be allocated to the master core as per the Resource Manager module).

**Initialization**

---

**Eth\_17\_GEthMac driver**

The code sequence for initializing the ETH driver is as follows from the core (core 0) in which the controller is configured:

```
#include "Eth_17_GEthMac.h"
#include "Port.h"
#include "Mcu.h"
#include "McalLib.h"
#include "Irq.h"

uint32 CoreId;

/* MCU Initialization */
Mcu_Init(&Mcu_Config);
Mcu_InitClock(0U);
while(Mcu_GetPllStatus() != MCU_PLL_LOCKED);
Mcu_DistributePllClock();

/* Port Initialization */
Port_Init(&Port_Config);

/* Ethernet driver Initialization */
Eth_17_GEthMac_Init(&Eth_Config);

/*Obtain the current core Id*/
CoreId = Mcal_GetCpuIndex();

/*To check if Ethernet driver is initialized for the current core*/
if(*(Eth_17_GEthMac_CoreInitStatus[CoreId]) == (uint32)ETH_STATE_INIT)
{
    /*Successful initialization*/
}

/* Perform Ethernet transceiver initialization */
```

**Setting controller mode**

The code sequence to change the mode of the ETH controller from ETH\_MODE\_DOWN state to ETH\_MODE\_ACTIVE is as follows:

```
Eth_17_GEthMac_SetControllerMode(0, ETH_MODE_ACTIVE);
```

**Set physical address (MAC address)**

The code sequence to set the MAC address of the ETH controller is as follows:

```
Eth_17_GEthMac_SetPhysAddr(0, &MacAddress[0]);
```

**Get physical address (MAC address)**

## Eth\_17\_GEthMac driver

The code sequence to get the MAC address of the ETH controller is as follows:

```
Eth_17_GEthMac_GetPhysAddr((0, &MacAddressRead[0]));
```

### Configure ETH transceiver (PHY) device

The code sequence to configure ETH Transceiver (PHY) by using Eth\_17\_GEthMac\_ReadMii and Eth\_17\_GEthMac\_WriteMii APIs is as follows:

```
uint16 phy_id1 = 0;
uint16 phy_id2 = 0;
uint16 reg_value = 0;

/* Read Device ID from Ethernet transceiver(PHY) */
Eth_17_GEthMac_ReadMii( 0, 1, MII_PHYSID1, &phy_id1);
Eth_17_GEthMac_ReadMii( 0, 1, MII_PHYSID2, &phy_id2);

/* Configure Link Speed and Duplex mode in Ethernet transceiver (PHY) */
reg_value = BMCR_SPEED100 | BMCR_FULLDPLX;
reset_status = Eth_17_GEthMac_WriteMii(0, 1, MII_BMCR, reg_value);
```

### Transmit frame

The code sequence to transmit an ETH frame in the polling mode is as follows:

```
/* Request the data buffer for frame Transmission */
Eth_17_GEthMac_ProvideTxBuffer(0, BufIdxPtr, BufPtr, LenBytePtr)

/* Application Layer fill the buffer with frame data*/

/* Transmit already filled buffer using the BuffIdx*/
Eth_17_GEthMac_Transmit(0, BufIdx,FrameType, TxConfirmation, LenByte,
PhysAddrPtr)
.....

/* Call TX confirmation API to check if the packet transmission is completed */
Eth_17_GEthMac_TxConfirmation(0);
```

### Receive frame

The received ETH frames are transferred to the driver buffer automatically. In polling mode, the application may call Eth\_17\_GEthMac\_Receive API to get indication of all the filled buffers. In the interrupt mode, the ETH driver will call receive indication from receive ISR context. The code sequence to receive ETH frames in polling mode is as follows:

```
/* Poll for received frames */
Eth_17_GEthMac_Receive(0, RxStatusPtr);
```

## Eth\_17\_GEthMac driver

### 1.1.5 Key architectural considerations

#### 1.1.5.1 ETH controller initialization sequence

The AUTOSAR specifications for the ETH and the ETH transceiver(PHY) drivers are conceived in a way that both shall work independently. As per AUTOSAR specification, the ETH controller and PHY initialization sequencing shall be as follows:

1. Invoke `Eth_17_GEthMac_Init()`. This will initialize MAC controller and select PHY interface (MDIO and MII).
2. Invoke `EthTrcv_Init()`. This will communicate to PHY using PHY interface (MDIO) and prepare the PHY to operate in the required mode.

Note: Since the PHY interface (MDIO) is managed by the ETH driver, `EthTrcv_Init()` will use `Eth_WriteMii()` and `Eth_ReadMii()` API services provided by ETH driver.

But the said sequence does not fit with the Gigabit ETH MAC IP used in the TC3xx controller. Until the clock signal from the PHY is stable at MAC input, the MAC cannot finish PHY interface (MII) selection. Otherwise, this will cause unpredictable behavior. Hence the ETH controller initialization by ETH driver is split into two parts.

Part 1: The `Eth_17_GEthMac_Init()` API enables the module and prepares the MDIO interface.

Note: The `Eth_17_GEthMac_Init()` API will not choose the MII interface because it is probable that the clock from the PHY is not yet available at the MAC input pin. The `Eth_17_GEthMac_Init()` API initializes all the controllers allocated to the core from which it is being invoked and core initialization status is set to success. If there is a failure in initializing any one of the controllers then ETH\_E\_ACCESS DEM is reported as PREFAILED for the failed controller and core initialization status is set to unsuccessful.

Part 2: The `Eth_17_GEthMac_SetControllerMode()` API performs the following actions:

- (i) Flush the transmit and receive queue 0 (since only queue 0 is used in the driver design)
- (ii) Disables/ masks the module interrupts.
- (iii) If the configured mode is RGMII, set the mode to MII (`GETH_GPCTL.EPR = 000b`) and initialize the skew timing to 0. If the configured mode is not RGMII, select the PHY interface to either RMII or MII based on the configuration.
- (iv) Perform a kernel reset (expecting that `EthTrcv_Init()` is already called and the clock signal from the PHY is already available at the MAC input pin) and then wait for the necessary fSPB cycles before proceeding further.
- (v) Set the PHY interface to the configured mode and also select the alternate inputs. In RGMII mode, set the Tx / Rx timing skew as per the configuration.
- (vi) Apply software reset to DMA and then complete the ETH controller initialization for the requested controller only (that is, only for the controller index which is passed as a parameter to the `Eth_17_GEthMac_SetControllerMode()` API).

This action is performed only once when this `Eth_17_GEthMac_SetControllerMode()` API is called for the first time after `Eth_17_GEthMac_Init()` API.

Note:

- Since `Eth_17_GEthMac_SetControllerMode()` should be invoked before initiating any data transmit or receive operation, this change in the ETH controller initialization sequence does not affect the user application. Therefore, the calling sequences mentioned in AUTOSAR for the ETH driver remains the same.
- `Eth_17_GEthMac_SetControllerMode()` API performs the second stage of initialization only for the controller index passed as the input parameter. If the second stage of initialization for the controller is successful, then the controller initialization status is set to success.
- Runtime APIs reports the development error `ETH_17_GETHMAC_E_NOT_INITIALIZED`, if indexed controller initialization or core initialization is unsuccessful.



## Eth\_17\_GEthMac driver

### 1.1.5.2 Eth\_17\_GEthMac\_SetControllerMode API implemented as synchronous

AUTOSAR specification mentions the `Eth_17_GEthMac_SetControllerMode()` API as asynchronous function. Since the ETH controller in TC3xx supports the change of the controller mode synchronously, the `Eth_17_GEthMac_SetControllerMode()` API is implemented as synchronous.

### 1.1.5.3 Development error checks added to avoid undefined behavior

The ETH driver performs the following error checks which are not explicitly mentioned in the product requirement. These error checks are introduced to avoid any undefined behavior from the ETH driver. If DET error check is enabled:

- The `Eth_17_GEthMac_SetControllerMode()` API will check the `CtrlMode` parameter for being valid. If the check fails, the API will report the `ETH_17_E_INV_PARAM` DET and return `E_NOT_OK`.
- The `Eth_17_GEthMac_UpdatePhysAddrFilter()` API will check the `Action` parameter for being valid. If the check fails, the API will report `ETH_17_E_INV_PARAM` DET and return `E_NOT_OK`.
- The `Eth_17_GEthMac_Transmit()` API will check the `LenByte` parameter for being valid. If the value of `LenByte` is more than granted through the `Eth_17_GEthMac_ProvideTxBuffer()`, then the API `Eth_17_GEthMac_Transmit()` will report `ETH_17_E_INV_PARAM` DET and return `E_NOT_OK`. If the value of `LenByte` is less than granted through the `Eth_17_GEthMac_ProvideTxBuffer()` API, then the API `Eth_17_GEthMac_Transmit()` will proceed with transmission of ETH packet of length `LenByte`.
- The APIs `Eth_17_GEthMac_EnableEgressTimeStamp()` and `Eth_17_GEthMac_GetEgressTimeStamp()` will check the `BufIdx` parameter for being valid. If the `BufIdx` is within total number of buffer configured and if `BufIdx` same as the one allocated to application by ETH driver from the `Eth_17_GEthMac_ProvideTxBuffer()` API. If aforementioned checks fail, the API will report `ETH_17_E_INV_PARAM` DET.

### 1.1.5.4 Time support APIs report ETH\_E\_ACCESS production error

Any timer update operation in the ETH controller needs a feedback from register bits to ensure that requested operation is completed. If the requested operation is not completed within a configured time limit, time out occurs and `ETH_E_ACCESS` production error is reported from following APIs:

`Eth_17_GEthMac_SetGlobalTime`

`Eth_17_GEthMac_SetCorrectionTime`

**Note:** The `EthTimeoutCount` configuration parameter is used to configure the time limit.

### 1.1.5.5 Multicore support for ETH driver

ETH controllers are allowed to be configured to any of the cores as per the hardware availability. The APIs of the ETH driver can be classified as concurrent safe for different controllers.

**Note:** For certain TC3xx devices, more than one ETH controller can be present. In a multicore environment, each controller can be allocated to different cores or the same core based on the application requirement. A core specific status of the driver is maintained using a global variable `Eth_17_GEthMac_CoreInitStatus` and is available to the user for debugging purposes. The `Eth_17_GEthMac_Init` API updates the core specific initialization status from `ETH_STATE_UNINIT` to `ETH_STATE_INIT` after a successful execution from the core in which it is being invoked. `ETH_E_CORE_NOT_CONFIGURED` DET is reported if the `Eth_17_GEthMac_Init` API is invoked from a core in which no controller is allocated. `ETH_17_GETHMAC_E_INV_CTRL_IDX` DET is reported if runtime

## Eth\_17\_GEthMac driver

*APIs with controller index as input parameter are invoked from a core to which no controller or if the requested controller is not allocated.*

### 1.1.5.6 Specific hardware features used for nominal operation of ETH driver

To implement the ETH driver as per the product requirement, the following hardware-specific features are used in ETH driver.

- Alternative pin selection:  
For the selected input pins of the ETH controller, the ETH controller hardware provides alternative hardware port pins and the ETH driver software should select one pin (which is used in hardware design) from available alternative input pin options. To meet this requirement, the ETH driver provides the following pre-compile configuration parameters under `EthCtrlConfig` container.  
`EthMdioAlternateInput`, `EthRxclkInput`, `EthRxErrMIIInput`, `EthCarrierSenseMIIInput`,  
`EthRecDataValidMIIInput`, `EthTxClockMIIInput`, `EthCollisionMII`, `EthRefClkRMIIInput`,  
`EthCRSDVRMIIInput`, `EthReceiveData0Input`, `EthReceiveData1Input`, `EthReceiveData2Input` and  
`EthReceiveData3Input`.
- Value of gigabit ETH MAC Kernel frequency:  
The ETH controller uses the timer which is part of the ETH controller for implementing the requirement of global time support APIs. To configure, this internal timer, the ETH driver need to read the value of Gigabit ETH MAC Kernel frequency which is configured in the MCU driver. To extract this value, the ETH driver provides a pre-compile configuration parameter `EthOperationFrequency` in the `EthGeneral` container.
- Value of system peripheral bus frequency:  
To configure, MDIO interface clock, the ETH driver need to read the value of the system peripheral bus frequency which is configured in the MCU driver. To extract this value, the ETH driver provides a pre-compile configuration parameter `EthPeripheralBusClock` in the `EthGeneral` container.
- FIFO space and DMA channel:  
The ETH controller has transmit/receive FIFO space shared by multiple queues and multiple transmit/receive DMA channels. The ETH driver uses one transmit queue of maximum FIFO size and one transmit DMA channel for the transmit operation and one receive queue of maximum FIFO size and one receive DMA channel for the receive operation.
- Configuration of transmit and receive timing skew in RGMII mode:  
`EthSkewTxClockDelay` and `EthSkewRxClockDelay` configuration parameters under the `EthCtrlConfig` container are added to configure the transmit and receive clock delay for skew timing. This is applicable only in the RGMII mode. In the MII and RMII modes, both parameters are not active.

## **1.2 Assumptions of Use (AoUs)**

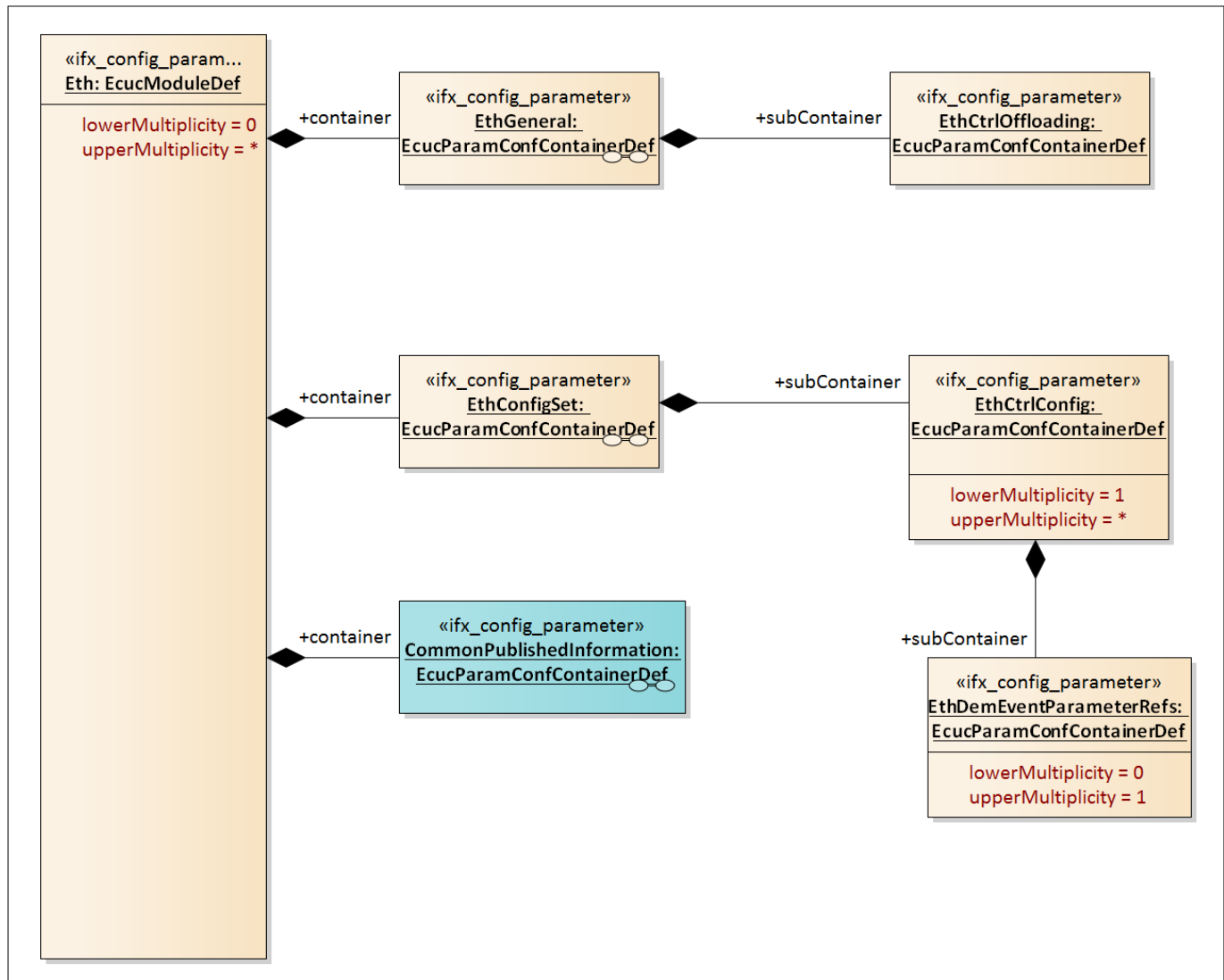
There are no AoUs for the driver.

## Eth\_17\_GEthMac driver

### 1.3 Reference information

#### 1.3.1 Configuration interfaces

This section details the configuration container hierarchy along with their configuration parameters.



**Figure 6** Container hierarchy along with their configuration parameters

##### 1.3.1.1 Container: CommonPublishedInformation

Container contains the common published information of the ETH driver

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

##### 1.3.1.1.1 ArMajorVersion

**Table 5** Specification for ArMajorVersion

Name	ArMajorVersion
------	----------------

**Eth\_17\_GEthMac driver**
**Table 5 Specification for ArMajorVersion (continued)**

<b>Description</b>	Provides the major version of the AUTOSAR specification.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	4		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.1.2 ArMinorVersion**
**Table 6 Specification for ArMinorVersion**

<b>Name</b>	ArMinorVersion		
<b>Description</b>	Provides the minor version of the AUTOSAR specification.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	2		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.1.3 ArPatchVersion**
**Table 7 Specification for ArPatchVersion**

<b>Name</b>	ArPatchVersion		
<b>Description</b>	Provides the patch version of the AUTOSAR specification.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	2		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-

**Eth\_17\_GEthMac driver**
**Table 7 Specification for ArPatchVersion (continued)**

<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.1.4 ModuleId**
**Table 8 Specification for ModuleId**

<b>Name</b>	ModuleId		
<b>Description</b>	Provides the module Id.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 65535		
<b>Default value</b>	88		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.1.5 Release**
**Table 9 Specification for Release**

<b>Name</b>	Release		
<b>Description</b>	Indicates the TC3xx device derivative used for the implementation.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucStringParamDef
<b>Range</b>	String		
<b>Default value</b>	As per the hardware derivative		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Eth\_17\_GEthMac driver

### 1.3.1.1.6 SwMajorVersion

**Table 10 Specification for SwMajorVersion**

<b>Name</b>	SwMajorVersion		
<b>Description</b>	Provides the major version of the software.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	As per the software version		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.1.7 SwMinorVersion

**Table 11 Specification for SwMinorVersion**

<b>Name</b>	SwMinorVersion		
<b>Description</b>	Provides the minor version of the software.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	As per the software version		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.1.8 SwPatchVersion

**Table 12 Specification for SwPatchVersion**

<b>Name</b>	SwPatchVersion		
<b>Description</b>	Provides the patch version of the software.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		

## Eth\_17\_GEthMac driver

**Table 12 Specification for SwPatchVersion (continued)**

<b>Default value</b>	As per the software version		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.1.9 VendorApiInfix

**Table 13 Specification for VendorApiInfix**

<b>Name</b>	VendorApiInfix		
<b>Description</b>	Provides the VendorApiInfix.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucStringParamDef
<b>Range</b>	String		
<b>Default value</b>	GEthMac		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.1.10 VendorId

**Table 14 Specification for VendorId**

<b>Name</b>	VendorId		
<b>Description</b>	Provides the vendor Id		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 65535		
<b>Default value</b>	17		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-



**Eth\_17\_GEthMac driver**
**Table 14 Specification for VendorId (continued)**

<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.2 Container: Eth**

Configuration of the individual ETH controller.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**1.3.1.3 Container: EthConfigSet**

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

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**1.3.1.4 Container: EthCtrlConfig**

Configuration of the individual ETH controller. Note: The multiplicity of EthCtrlConfig is device dependent. It is 1 to maximum number of controllers available.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: Pre-Compile

**1.3.1.4.1 EthCRSDVRMIInput**
**Table 15 Specification for EthCRSDVRMIInput**

<b>Name</b>	EthCRSDVRMIInput		
<b>Description</b>	Selects one of the four supported pins for the ETH carrier sense/data valid combi-signal for RMII. The availability of the port pins is dependent on the micro-controller package. Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as RMII. Note 2: The default option is ALT <sub>x</sub> _SELECT_NONE where, x indicates the alternate input select. ALT <sub>x</sub> _SELECT_NONE indicates that no port pin is selected. User must choose a suitable alternate port pin available for the device.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALT <sub>x</sub> _SELECT_PX <sub>y</sub> _Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALT <sub>x</sub> _SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-

**Eth\_17\_GEthMac driver**
**Table 15 Specification for EthCRSDVRMIInput (continued)**

<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.2 EthCarrierSenseMIInput**
**Table 16 Specification for EthCarrierSenseMIInput**

<b>Name</b>	EthCarrierSenseMIInput		
<b>Description</b>	<p>Selects one of the two supported pins for the ETH carrier sense MII.</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as MII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.3 EthCollisionMIInput**
**Table 17 Specification for EthCollisionMIInput**

<b>Name</b>	EthCollisionMIInput		
<b>Description</b>	<p>Selects one of the four supported pins for collision for MII.</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as MII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef

**Eth\_17\_GEthMac driver**
**Table 17 Specification for EthCollisionMII (continued)**

<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.4 EthCtrlEnableCrcStripping**
**Table 18 Specification for EthCtrlEnableCrcStripping**

<b>Name</b>	EthCtrlEnableCrcStripping		
<b>Description</b>	<p>Includes or excludes the length of the checksum in the received frame length reported to the upper layer.</p> <p>Note 1: If this parameter is enabled (that is set to TRUE), the ETH driver excludes the length of the checksum in the received frame length.</p> <p>Note 2: The default value of this parameter is kept as FALSE. Therefore, the application will receive complete ETH frame including checksum field.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.5 EthCtrlEnableMii**
**Table 19 Specification for EthCtrlEnableMii**

<b>Name</b>	EthCtrlEnableMii		
<b>Description</b>	Enables/disables MII-/RMII-/RGMII-based APIs for transceiver access. Note: The optional APIs are disabled by default to minimize the executable code size. If there are more than one		

**Eth\_17\_GEthMac driver**
**Table 19 Specification for EthCtrlEnableMii (continued)**

	controller for the selected device then the value of this parameter must be same across the controllers.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.6 EthCtrlEnableRxInterrupt**
**Table 20 Specification for EthCtrlEnableRxInterrupt**

<b>Name</b>	EthCtrlEnableRxInterrupt		
<b>Description</b>	Enables/disables the receive interrupt. If the receive interrupt is disabled, the reception will work in the polling mode.  Note: The default value of this parameter is kept as FALSE and, therefore, the ETH driver can be functional without configuring the interrupt module by default.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Eth\_17\_GEthMac driver

### 1.3.1.4.7 EthCtrlEnableTxInterrupt

**Table 21 Specification for EthCtrlEnableTxInterrupt**

<b>Name</b>	EthCtrlEnableTxInterrupt		
<b>Description</b>	Enables / Disables transmit interrupt. If it is disabled, transmission will work in polling mode. Note: The default value of this parameter kept as false and hence the ETH driver can be functional without configuring the interrupt module in default.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.4.8 EthCtrlIdx

**Table 22 Specification for EthCtrlIdx**

<b>Name</b>	EthCtrlIdx		
<b>Description</b>	Specifies the instance ID of the configured controller. This value is assigned to the symbolic name derived from the short name of the EthCtrlConfig. Note: EthCtrlIdx varies from 0 to maximum number of controllers available.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	0		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	ECU
<b>Dependency</b>	-		

## Eth\_17\_GEthMac driver

### 1.3.1.4.9 EthCtrlPhyAddress

**Table 23 Specification for EthCtrlPhyAddress**

<b>Name</b>	EthCtrlPhyAddress		
<b>Description</b>	<p>Specifies the unique 48-bit physical address (MAC address) of the ETH controller in network byte order.</p> <p>Regular Expression: [0-9a-fA-F]{2}[:-][0-9a-fA-F]{2}{5}</p> <p>Allowed characters are [a-f, A-F, 0-9] and each pair should be separated by symbols : or -. For example, 00:A0:C9:14:C8:29</p> <p>Note 1: As per AUTOSAR, multiplicity is 0-1. However, in the ETH driver the multiplicity is implemented as 1-1. Therefore, Post-Build variant multiplicity is also false.</p> <p>Note 2: The default value of this parameter is kept to match the Vendor ID of MAC is Infineon's ID. If there are more than one ETH controller, then the same default value is retained. It is the responsibility of the user to provide a unique MAC address as per the application needs (if the MAC address is not unique then an error is prompted to the user to provide a unique MAC address).</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucStringParamDef
<b>Range</b>	String		
<b>Default value</b>	00:03:19:00:00:01		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.4.10 EthCtrlRxBufLenByte

**Table 24 Specification for EthCtrlRxBufLenByte**

<b>Name</b>	EthCtrlRxBufLenByte		
<b>Description</b>	<p>Limits the maximum transmit buffer length (frame length) in bytes. This configured length includes ETH frame header and frame checksum (total 18 bytes).</p> <p>Note 1: During configuration, the buffer size should be (Header + Payload data + CRC).</p> <p>Note 2: Maximum length of one ETH frame packet is 1522. (1504(Payload) + 14(Header) + 4(CRC) = 1522).</p> <p>Note 3: The default value for this parameter is kept as 1522 because the maximum size of one ETH frame is 1522.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 1522		
<b>Default value</b>	1522		

**Eth\_17\_GEthMac driver**
**Table 24 Specification for EthCtrlRxBufLenByte (continued)**

<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.11 EthCtrlTxBufLenByte**
**Table 25 Specification for EthCtrlTxBufLenByte**

<b>Name</b>	EthCtrlTxBufLenByte		
<b>Description</b>	Limits the maximum transmit buffer length (frame length) in bytes. This configured length includes ETH frame header and frame checksum (Total 18 Bytes). Note 1: During configuration, the buffer size shall be (Header + Payload data + CRC). Note 2: Maximum length of one ETH frame packet is 1522. (1504(Payload) + 14(Header) + 4(CRC) = 1522). Note 3: Since the maximum size of one ETH frame is 1522, the default value for this parameter is kept as 1522.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 1522		
<b>Default value</b>	1522		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.12 EthMdioAlternateInput**
**Table 26 Specification for EthMdioAlternateInput**

<b>Name</b>	EthMdioAlternateInput		
<b>Description</b>	Selects one of the four supported pins for the MDIO signal. The availability of port pins is dependent on the micro-controller package. Note: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected. User must choose a suitable alternate port pin available for the device.		

**Eth\_17\_GEthMac driver**
**Table 26 Specification for EthMdioAlternateInput (continued)**

<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.13 EthOpMode**
**Table 27 Specification for EthOpMode**

<b>Name</b>	EthOpMode		
<b>Description</b>	Specifies the mode of operation (FULLDUPLEX/HALFDUPLEX).		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	FULLDUPLEX: Full-duplex mode HALFDUPLEX: Half-duplex mode		
<b>Default value</b>	FULLDUPLEX		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.14 EthPhyInterface**
**Table 28 Specification for EthPhyInterface**

<b>Name</b>	EthPhyInterface		
<b>Description</b>	Specifies the interface used between the MAC and PHY. The availability of modes is dependent on the micro-controller package. Note 1: Since the commonly used application is RMII, the default value of this parameter is selected as RMII.		



**Eth\_17\_GEthMac driver**
**Table 28 Specification for EthPhyInterface (continued)**

	Note 2: Availability of the Ethernet controller speed with respect to the PHY interface 1000/100/10MBPS depends on the device.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	MII: Media Independent Interface RGMII: Reduced Gigabit Media Independent Interface RMII: Reduced Media Independent Interface		
<b>Default value</b>	RMII		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.15 EthRecDataValidMIIInput**
**Table 29 Specification for EthRecDataValidMIIInput**

<b>Name</b>	EthRecDataValidMIIInput		
<b>Description</b>	Selects one of the four supported pins for the ETH Receive Data Valid MII. The availability of the port pins is dependent on the micro-controller package. Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as MII. Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected. User must choose a suitable alternate port pin available for the device.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

## Eth\_17\_GEthMac driver

### 1.3.1.4.16 EthReceiveData0Input

**Table 30 Specification for EthReceiveData0Input**

<b>Name</b>	EthReceiveData0Input		
<b>Description</b>	<p>Selects one of the four supported pins for receive data 0 for MII, RMII and RGMII (RGMII can use RXD0A only).</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

### 1.3.1.4.17 EthReceiveData1Input

**Table 31 Specification for EthReceiveData1Input**

<b>Name</b>	EthReceiveData1Input		
<b>Description</b>	<p>Selects one of the four supported pins for receive data 1 for MII, RMII and RGMII (RGMII can use RXD1A only).</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-

**Eth\_17\_GEthMac driver**
**Table 31 Specification for EthReceiveData1Input (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.18 EthReceiveData2Input**
**Table 32 Specification for EthReceiveData2Input**

<b>Name</b>	EthReceiveData2Input		
<b>Description</b>	<p>Selects one of the four supported pins for receive data 2 for MII and RGMII (RGMII can use RXD2A only).</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is invalid only if the EthPhyInterface parameter is selected as RMII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.19 EthReceiveData3Input**
**Table 33 Specification for EthReceiveData3Input**

<b>Name</b>	EthReceiveData3Input		
<b>Description</b>	<p>Selects one of the four supported pins for receive data 3 for MII and RGMII (RGMII can use RXD3A only).</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is invalid only if the EthPhyInterface parameter is selected as RMII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p>		

**Eth\_17\_GEthMac driver**
**Table 33 Specification for EthReceiveData3Input (continued)**

	User must choose a suitable alternate port pin available for the device.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.20 EthRefClkRMIIInput**
**Table 34 Specification for EthRefClkRMIIInput**

<b>Name</b>	EthRefClkRMIIInput		
<b>Description</b>	<p>Selects one of the four supported pins for reference clock input for RMII.</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as RMII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

## Eth\_17\_GEthMac driver

### 1.3.1.4.21 EthRxBufTotal

**Table 35 Specification for EthRxBufTotal**

<b>Name</b>	EthRxBufTotal		
<b>Description</b>	<p>Configures the number of receive buffers.</p> <p>Note: Total buffer size in the RAM that is reserved by the ETH driver for receive packets is calculated as EthCtrlRxBufLenByte*EthRxBufTotal.</p> <p>Note: By default, the number of buffers reserved is kept as 4 and, therefore, up to 4 packets can be received without overflow.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	4		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.4.22 EthRxErrMIIInput

**Table 36 Specification for EthRxErrMIIInput**

<b>Name</b>	EthRxErrMIIInput		
<b>Description</b>	<p>Selects one of the four supported pins for the ETH Receive Error MII.</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as MII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL

**Eth\_17\_GEthMac driver**
**Table 36 Specification for EthRxErrMIIInput (continued)**

<b>Dependency</b>	EthPhyInterface
-------------------	-----------------

**1.3.1.4.23 EthRxclkInput**
**Table 37 Specification for EthRxclkInput**

<b>Name</b>	EthRxclkInput		
<b>Description</b>	<p>Selects one of the four supported pins for the ETH receive clock for MII and RGMII (RGMII can use RXCLKA only).</p> <p>The availability of the port pins is dependent on the micro-controller package</p> <p>Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as MII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.24 EthSkewRxClockDelay**
**Table 38 Specification for EthSkewRxClockDelay**

<b>Name</b>	EthSkewRxClockDelay		
<b>Description</b>	<p>Specifies the receive clock delay in the RGMII mode for Transmit Skew Timing.</p> <p>Note: The minimum value is kept as the default value for this parameter and, therefore, by default the clock delay is 0.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 15		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-

**Eth\_17\_GEthMac driver**
**Table 38 Specification for EthSkewRxClockDelay (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.25 EthSkewTxClockDelay**
**Table 39 Specification for EthSkewTxClockDelay**

<b>Name</b>	EthSkewTxClockDelay		
<b>Description</b>	Specifies the transmit clock delay in RGMII mode for transmit skew timing. Note: Minimum value is kept as default value for this parameter and hence in default the clock delay is zero.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 15		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.4.26 EthSpeed**
**Table 40 Specification for EthSpeed**

<b>Name</b>	EthSpeed		
<b>Description</b>	Selects the speed of the ETH controller. Note 1: Since the commonly used application is 100 Mbps, therefore, the default value of this parameter is selected as 100 Mbps. Note 2: Availability of the 1000/100/10MBPS depends on the device.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ETH_1000MBPS: Data transfer rate is 1000 Mbps ETH_100MBPS: Data transfer rate is 100 Mbps ETH_10MBPS: Data transfer rate is 10 Mbps		
<b>Default value</b>	ETH_100MBPS		

**Eth\_17\_GEthMac driver**
**Table 40 Specification for EthSpeed (continued)**

<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.27 EthTxBufTotal**
**Table 41 Specification for EthTxBufTotal**

<b>Name</b>	EthTxBufTotal		
<b>Description</b>	<p>Configures the number of transmit buffers.</p> <p>Note: Total buffer size in the RAM that is reserved by the ETH driver for transmission packet is calculated as EthCtrlTxBufLenByte*EthTxBufTotal.</p> <p>Note: By default, the number of buffers reserved is kept as 4 and, therefore, up to 4 transmit request can be made without waiting for transmit completion.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	4		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.4.28 EthTxClockMIIInput**
**Table 42 Specification for EthTxClockMIIInput**

<b>Name</b>	EthTxClockMIIInput		
<b>Description</b>	<p>Selects one of the four supported pins for the transmit clock input for MII.</p> <p>The availability of the port pins is dependent on the micro-controller package.</p> <p>Note 1: This parameter is valid only if the EthPhyInterface parameter is selected as MII.</p> <p>Note 2: The default option is ALTx_SELECT_NONE where, x indicates the alternate input select. ALTx_SELECT_NONE indicates that no port pin is selected.</p> <p>User must choose a suitable alternate port pin available for the device.</p>		



**Eth\_17\_GEthMac driver**
**Table 42 Specification for EthTxClockMIIInput (continued)**

<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ALTx_SELECT_PXy_Yz: Xy and Yz are port and pin number respectively which depends on the device variant.		
<b>Default value</b>	ALTx_SELECT_NONE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthPhyInterface		

**1.3.1.5 Container: EthCtrlOffloading**

Configuration of hardware checksum offloading features. Individual enabling of hardware offload functionality for CRC checksum( for IPV4, UDP, TCP, ICMP frames) is not possible due to hardware limitation. Enabling of any one of the below configuration parameter EthCtrlEnableOffloadChecksumIPV4, EthCtrlEnableOffloadChecksumUDP, EthCtrlEnableOffloadChecksumTCP, EthCtrlEnableOffloadChecksumICMP would enable the checksum offload functionality for IPV4, UDP, TCP and ICMP. This is a deviation to AUTOSAR requirements.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**1.3.1.5.1 EthCtrlEnableOffloadChecksumICMP**
**Table 43 Specification for EthCtrlEnableOffloadChecksumICMP**

<b>Name</b>	EthCtrlEnableOffloadChecksumICMP		
<b>Description</b>	<p>Enables/disables checksum offloading of IPv4, TCP, UDP and ICMP frames for both transmission (that is, calculating and inserting checksum in the transmitted frames at the hardware level) and reception (that is, checking for checksum mismatch at the hardware level for the received frames).</p> <p>Note: This is a deviation from the AUTOSAR requirement. The GETHMAC hardware supports either checksum offloading can be enabled for all types of packets or disable all types of packets.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		

**Eth\_17\_GEthMac driver**
**Table 43 Specification for EthCtrlEnableOffloadChecksumICMP (continued)**

<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthCtrlEnableOffloadChecksumUDP, EthCtrlEnableOffloadChecksumTCP, EthCtrlEnableOffloadChecksumIPv4		

**1.3.1.5.2 EthCtrlEnableOffloadChecksumIPv4**
**Table 44 Specification for EthCtrlEnableOffloadChecksumIPv4**

<b>Name</b>	EthCtrlEnableOffloadChecksumIPv4		
<b>Description</b>	<p>It enables/disables checksum offloading of IPv4, TCP, UDP, ICMP frames for both transmission (That is calculation and insertion of checksum in the transmitted frames at hardware level) and reception (That is checksum calculation at hardware level for received frames to check for checksum mismatch).</p> <p>Note: This is a deviation from the AUTOSAR requirement. The GETHMAC hardware supports either checksum offloading can be enabled for all types of packets or disable all types of packets.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthCtrlEnableOffloadChecksumUDP, EthCtrlEnableOffloadChecksumTCP, EthCtrlEnableOffloadChecksumICMP		

**1.3.1.5.3 EthCtrlEnableOffloadChecksumTCP**
**Table 45 Specification for EthCtrlEnableOffloadChecksumTCP**

<b>Name</b>	EthCtrlEnableOffloadChecksumTCP
-------------	---------------------------------

**Eth\_17\_GEthMac driver**
**Table 45 Specification for EthCtrlEnableOffloadChecksumTCP (continued)**

<b>Description</b>	It enables/disables checksum offloading of IPv4, TCP, UDP and ICMP frames for both transmission (that is, calculating and inserting checksum in the transmitted frames at the hardware level) and reception (that is, checksum calculation at the hardware level for received frames to check for checksum mismatch).  Note: This is a deviation from the AUTOSAR requirement. The GETHMAC hardware supports either checksum offloading can be enabled for all types of packets or disable all types of packets.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthCtrlEnableOffloadChecksumUDP, EthCtrlEnableOffloadChecksumIPv4, EthCtrlEnableOffloadChecksumICMP		

**1.3.1.5.4 EthCtrlEnableOffloadChecksumUDP**
**Table 46 Specification for EthCtrlEnableOffloadChecksumUDP**

<b>Name</b>	EthCtrlEnableOffloadChecksumUDP		
<b>Description</b>	It enables/disables checksum offloading of IPv4, TCP, UDP, ICMP frames for both transmission (That is calculation and insertion of checksum in the transmitted frames at hardware level) and reception (That is checksum calculation at hardware level for received frames to check for checksum mismatch).  Note: This is a deviation from the AUTOSAR requirement. The GETHMAC hardware supports either checksum offloading can be enabled for all types of packets or disable all types of packets.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-

**Eth\_17\_GEthMac driver**
**Table 46 Specification for EthCtrlEnableOffloadChecksumUDP (continued)**

<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthCtrlEnableOffloadChecksumTCP, EthCtrlEnableOffloadChecksumIPv4, EthCtrlEnableOffloadChecksumICMP		

**1.3.1.6 Container: EthDemEventParameterRefs**

This is a container for the references to the DemEventParameter elements, which are invoked using the Dem\_ReportErrorStatus() API in case the corresponding errors occur. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor-specific error references. Post-Build Variant Multiplicity: TRUE Multiplicity Configuration Class: Post-Build

Post-Build Variant Multiplicity: TRUE

Multiplicity Configuration Class: Post-Build

**1.3.1.6.1 ETH\_E\_ACCESS**
**Table 47 Specification for ETH\_E\_ACCESS**

<b>Name</b>	ETH_E_ACCESS		
<b>Description</b>	Provides preference to the DemEventParameter, which is issued when the error controller access fails.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.2 ETH\_E\_ALIGNMENT**
**Table 48 Specification for ETH\_E\_ALIGNMENT**

<b>Name</b>	ETH_E_ALIGNMENT		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the error alignment error occurs.		

**Eth\_17\_GEthMac driver**
**Table 48 Specification for ETH\_E\_ALIGNMENT (continued)**

<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.3 ETH\_E\_CRC**
**Table 49 Specification for ETH\_E\_CRC**

<b>Name</b>	ETH_E_CRC		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the error CRC failure occurs.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.4 ETH\_E\_LATECOLLISION**
**Table 50 Specification for ETH\_E\_LATECOLLISION**

<b>Name</b>	ETH_E_LATECOLLISION		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the ETH late frame collision event occurs. Note: This is applicable only in the half-duplex mode.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef

**Eth\_17\_GEthMac driver**
**Table 50 Specification for ETH\_E\_LATECOLLISION (continued)**

<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.5 ETH\_E\_MULTIPLECOLLISION**
**Table 51 Specification for ETH\_E\_MULTIPLECOLLISION**

<b>Name</b>	ETH_E_MULTIPLECOLLISION		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the ETH multiple frame collision event occurs. Note: This is applicable only in the half-duplex mode.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.6 ETH\_E\_OVERSIZEFRAME**
**Table 52 Specification for ETH\_E\_OVERSIZEFRAME**

<b>Name</b>	ETH_E_OVERSIZEFRAME		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the over-sized frame error occurs.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		

**Eth\_17\_GEthMac driver**
**Table 52 Specification for ETH\_E\_OVERSIZEFRAME (continued)**

<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.7 ETH\_E\_RX\_FRAMES\_LOST**
**Table 53 Specification for ETH\_E\_RX\_FRAMES\_LOST**

<b>Name</b>	ETH_E_RX_FRAMES_LOST		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the error receive frames lost error occurs.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.8 ETH\_E\_SINGLECOLLISION**
**Table 54 Specification for ETH\_E\_SINGLECOLLISION**

<b>Name</b>	ETH_E_SINGLECOLLISION		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the ETH single frame collision event occurs. Note: This is applicable only in the half-duplex mode.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE

**Eth\_17\_GEthMac driver**
**Table 54 Specification for ETH\_E\_SINGLECOLLISION (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.6.9 ETH\_E\_UNDERSIZEFRAME**
**Table 55 Specification for ETH\_E\_UNDERSIZEFRAME**

<b>Name</b>	ETH_E_UNDERSIZEFRAME		
<b>Description</b>	Provides reference to the DemEventParameter, which is issued when the under-sized frame error occurs.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7 Container: EthGeneral**

General configuration of the ETH driver module

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**1.3.1.7.1 EthDevErrorDetect**
**Table 56 Specification for EthDevErrorDetect**

<b>Name</b>	EthDevErrorDetect		
<b>Description</b>	Enables or disables the DET detection and reporting.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	TRUE		



**Eth\_17\_GEthMac driver**
**Table 56 Specification for EthDevErrorDetect (continued)**

<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.2 EthGetDropCountApi**
**Table 57 Specification for EthGetDropCountApi**

<b>Name</b>	EthGetDropCountApi		
<b>Description</b>	Enables or disables the Eth_17_GEthMac_GetDropCount() API. Note: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.3 EthGetEtherStatsApi**
**Table 58 Specification for EthGetEtherStatsApi**

<b>Name</b>	EthGetEtherStatsApi		
<b>Description</b>	Enables or disables the Eth_17_GEthMac_GetEtherStats() API. Note: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		

**Eth\_17\_GEthMac driver**
**Table 58 Specification for EthGetEtherStatsApi (continued)**

<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.4 EthGlobalTimeSupport**
**Table 59 Specification for EthGlobalTimeSupport**

<b>Name</b>	EthGlobalTimeSupport		
<b>Description</b>	Enables or disables the following GlobalTime APIs. Eth_17_GEthMac_GetCurrentTime() Eth_17_GEthMac_EnableEgressTimeStamp() Eth_17_GEthMac_GetEgressTimeStamp() Eth_17_GEthMac_GetIngressTimeStamp() Eth_17_GEthMac_SetCorrectionTime() Eth_17_GEthMac_SetGlobalTime() Note: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.5 EthIndex**
**Table 60 Specification for EthIndex**

<b>Name</b>	EthIndex
<b>Description</b>	Specifies the ID of this module instance. If only one instance is present it should have an ID value of 0.

## Eth\_17\_GEthMac driver

**Table 60 Specification for EthIndex (continued)**

	Note: Since there is only one instance present in the TC3xx controller, the default value is kept as 0.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	0		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.7.6 EthInitApiMode

**Table 61 Specification for EthInitApiMode**

<b>Name</b>	EthInitApiMode		
<b>Description</b>	Defines the mode in which the Init() API is used. Note: Since the ETH driver accesses the SFRs, therefore, it is more efficient to operate the ETH driver in the Supervisor mode. Hence, the default mode of operation is Supervisor.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ETH_MCAL_SUPERVISOR: Operating mode used is Supervisory. The access to supervisor mode registers is abstracted via McalLib module. McalLib routes the call to OS APIs. ETH_MCAL_USER1: Operating mode used is User-1		
<b>Default value</b>	ETH_MCAL_SUPERVISOR		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthRuntimeApiMode		

### 1.3.1.7.7 EthMainFunctionPeriod

**Table 62 Specification for EthMainFunctionPeriod**

<b>Name</b>	EthMainFunctionPeriod
-------------	-----------------------

**Eth\_17\_GEthMac driver**
**Table 62 Specification for EthMainFunctionPeriod (continued)**

<b>Description</b>	Specifies the period of main function Eth_17_GEthMac_MainFunction in seconds. ETH driver does not require this information but the BSW schedule uses this information.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucFloatParamDef
<b>Range</b>	0 - 10		
<b>Default value</b>	0.005		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.8 EthMaxCtrlsSupported**
**Table 63 Specification for EthMaxCtrlsSupported**

<b>Name</b>	EthMaxCtrlsSupported		
<b>Description</b>	Limits the total number of supported controllers. This parameter is disabled for configuration because the ETH controllers available depends on the device variant.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	1 - maximum controllers available for the device		
<b>Default value</b>	1		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.9 EthMultiCoreErrorDetect**
**Table 64 Specification for EthMultiCoreErrorDetect**

<b>Name</b>	EthMultiCoreErrorDetect		
<b>Description</b>	<p>This parameter enables or disables the Multi core related default error tracer (Det) detection and reporting. It is applicable only when DETs are enabled.</p> <p>Note: By default the value of this parameter is set to TRUE since it is dependent on EthDevErrorDetect parameter.</p>		

**Eth\_17\_GEthMac driver**
**Table 64 Specification for EthMultiCoreErrorDetect (continued)**

<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	TRUE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	EthDevErrorDetect		

**1.3.1.7.10 EthOperationFrequency**
**Table 65 Specification for EthOperationFrequency**

<b>Name</b>	EthOperationFrequency		
<b>Description</b>	<p>Contains reference to the fGETH (basic frequency for the Gigabit ETH kernel) frequency value contained in the MCU module (in the Mcu/McuModuleConfiguration/McuClockSettingConfig/McuClockReferencePointConfig container). This parameter is to calculate the value required to be initialized in the register bits GETH_MAC_SUB_SECOND_INCREMENT.B.SSINC to operate timer correctly.</p> <p>Note: Since the dependent container is user configurable, the default value of this parameter is kept as NULL.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucReferenceDef
<b>Range</b>	Reference to Node: McuClockReferencePointConfig		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.11 EthPeripheralBusClock**
**Table 66 Specification for EthPeripheralBusClock**

<b>Name</b>	EthPeripheralBusClock
-------------	-----------------------

**Eth\_17\_GEthMac driver**
**Table 66 Specification for EthPeripheralBusClock (continued)**

<b>Description</b>	Contains reference to the fSPB (System Peripheral Bus) frequency value contained in the MCU module (in the Mcu/McuModuleConfiguration/McuClockSettingConfig/McuClockReferencePointConfig container). This parameter to program 2.5 MHz clock value of the MDIO interface.  Note: Since the dependent container is user configurable, the default value of this parameter is kept as NULL.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucReferenceDef
<b>Range</b>	Reference to Node: McuClockReferencePointConfig		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**1.3.1.7.12 EthRuntimeApiMode**
**Table 67 Specification for EthRuntimeApiMode**

<b>Name</b>	EthRuntimeApiMode		
<b>Description</b>	Provides the mode in which the Runtime API is used.  Note: Since the ETH driver accesses the SFRs, it is more efficient to operate the ETH driver in the Supervisor mode. Therefore, the default mode of operation is supervisor.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	ETH_MCAL_SUPERVISOR: Operating mode used is Supervisory. The access to supervisor mode registers is abstracted via McalLib module. McalLib routes the call to OS APIs. ETH_MCAL_USER1: Operating mode used is User-1.		
<b>Default value</b>	ETH_MCAL_SUPERVISOR		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Eth\_17\_GEthMac driver

### 1.3.1.7.13 EthTimeoutCount

**Table 68** Specification for EthTimeoutCount

<b>Name</b>	EthTimeoutCount		
<b>Description</b>	Specifies the maximum waiting time in nanoseconds for hardware timeout errors. Note: The maximum value is kept as the default value for this parameter.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	100 - 4294967295		
<b>Default value</b>	4294967295		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.7.14 EthUpdatePhysAddrFilter

**Table 69** Specification for EthUpdatePhysAddrFilter

<b>Name</b>	EthUpdatePhysAddrFilter		
<b>Description</b>	Enables or disables the API, Eth_17_GEthMac_UpdatePhysAddrFilter. Note: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.1.7.15 EthVersionInfoApi

**Table 70** Specification for EthVersionInfoApi

<b>Name</b>	EthVersionInfoApi
-------------	-------------------

## Eth\_17\_GEthMac driver

**Table 70 Specification for EthVersionInfoApi (continued)**

<b>Description</b>	Enables or disables the version info API, Eth_17_GEthMac_GetVersionInfo. Note: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 1.3.2 Functions - Type definitions

#### 1.3.2.1 Eth\_17\_GEthMac\_ConfigType

**Table 71 Specification for Eth\_17\_GEthMac\_ConfigType**

<b>Syntax</b>	Eth_17_GEthMac_ConfigType	
<b>Type</b>	Structure	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	--	The elements of the data structure are specific to the micro-controller
<b>Description</b>	Defines the type for data structure containing the set of configuration parameters required for initializing the ETH driver and controller	
<b>Source</b>	AUTOSAR	

#### 1.3.2.2 Eth\_BufIdxType

**Table 72 Specification for Eth\_BufIdxType**

<b>Syntax</b>	Eth_BufIdxType		
<b>Type</b>	uint32		
<b>File</b>	Eth_GeneralTypes.h		
<b>Range</b>	0x00000000 - 0xFFFFFFFF	ETH buffer identifier	
<b>Description</b>	ETH buffer identifier type		



## Eth\_17\_GEthMac driver

**Table 72 Specification for Eth\_BufIdxType (continued)**

<b>Source</b>	AUTOSAR
---------------	---------

### 1.3.2.3 Eth\_DataType

**Table 73 Specification for Eth\_DataType**

<b>Syntax</b>	Eth_DataType	
<b>Type</b>	uint8	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0-255	One byte data
<b>Description</b>	This type defines the ETH data type used for data transmission and reception.	
<b>Source</b>	AUTOSAR	

### 1.3.2.4 Eth\_FilterActionType

**Table 74 Specification for Eth\_FilterActionType**

<b>Syntax</b>	Eth_FilterActionType	
<b>Type</b>	Enumeration	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0 - ETH_ADD_TO_FILTER	Add the MAC address to the filter, that is, allow reception
	1 - ETH_REMOVE_FROM_FILTER	Remove the MAC address from the filter, that is, reception is blocked in the lower layer.
<b>Description</b>	The Eth_FilterActionType enumeration type describes the action to be taken for the MAC address given in *PhysAddrPtr of API function Eth_UpdatePhysAddrFilter()	
<b>Source</b>	AUTOSAR	

### 1.3.2.5 Eth\_FrameType

**Table 75 Specification for Eth\_FrameType**

<b>Syntax</b>	Eth_FrameType	
<b>Type</b>	uint16	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0x0000 - 0xFFFF	ETH frame type used in the ETH frame header
<b>Description</b>	This type defines the ETH frame type used in the ETH frame header	
<b>Source</b>	AUTOSAR	

## Eth\_17\_GEthMac driver

### 1.3.2.6 Eth\_ModeType

**Table 76** Specification for Eth\_ModeType

<b>Syntax</b>	Eth_ModeType	
<b>Type</b>	Enumeration	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0 - ETH_MODE_DOWN	Controller disabled
	1 - ETH_MODE_ACTIVE	Controller enabled
<b>Description</b>	This type defines the controller modes	
<b>Source</b>	AUTOSAR	

### 1.3.2.7 Eth\_RateRatioType

**Table 77** Specification for Eth\_RateRatioType

<b>Syntax</b>	Eth_RateRatioType	
<b>Type</b>	Structure	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	Eth_TimeIntDiffType IngressTimeStampDelta	IngressTimeStampSync2 - IngressTimeStampSync1
	Eth_TimeIntDiffType OriginTimeStampDelta	OriginTimeStampSync2 - OriginTimeStampSync1
<b>Description</b>	Variables of this type are used to express frequency ratios	
<b>Source</b>	AUTOSAR	

### 1.3.2.8 Eth\_ReturnType

**Table 78** Specification for Eth\_ReturnType

<b>Syntax</b>	Eth_ReturnType	
<b>Type</b>	Enumeration	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0 - ETH_OK	Success
	1 - ETH_E_NOT_OK	General failure
	2 - ETH_E_NO_ACCESS	ETH hardware access failure
<b>Description</b>	ETH Driver specific return type	
<b>Source</b>	AUTOSAR	

## Eth\_17\_GEthMac driver

### 1.3.2.9 Eth\_RxStatusType

**Table 79** Specification for Eth\_RxStatusType

<b>Syntax</b>	Eth_RxStatusType	
<b>Type</b>	Enumeration	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0 - ETH_RECEIVED	ETH frame has been received, no further frames available
	1 - ETH_NOT_RECEIVED	ETH frame has not been received, no further frames available
	2 - ETH_RECEIVED_MORE_DATA_AVAILABLE	ETH frame has been received, more frames are available
<b>Description</b>	Used as OUT parameter in the Eth_Receive() API that indicates whether a frame has been received and if so, whether more frames are available or frames are lost.	
<b>Source</b>	AUTOSAR	

### 1.3.2.10 Eth\_StateType

**Table 80** Specification for Eth\_StateType

<b>Syntax</b>	Eth_StateType	
<b>Type</b>	Enumeration	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0 - ETH_STATE_UNINIT	Driver is not yet configured
	1 - ETH_STATE_INIT	Driver is configured
<b>Description</b>	Status supervision is used for development error detection. The state will be available for debugging.	
<b>Source</b>	AUTOSAR	

### 1.3.2.11 Eth\_TimeIntDiffType

**Table 81** Specification for Eth\_TimeIntDiffType

<b>Syntax</b>	Eth_TimeIntDiffType	
<b>Type</b>	Structure	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	Eth_TimeStampType diff	Time difference
	boolean sign	Positive (True) / negative (False) time
<b>Description</b>	Variables of this type are used to express time differences	
<b>Source</b>	AUTOSAR	

## Eth\_17\_GEthMac driver

### 1.3.2.12 Eth\_TimeStampQualType

**Table 82** Specification for Eth\_TimeStampQualType

<b>Syntax</b>	Eth_TimeStampQualType	
<b>Type</b>	Enumeration	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	0 - ETH_VALID	0- Valid time stamp
	1 - ETH_INVALID	1- Invalid time stamp
	2 - ETH_UNCERTAIN	2-Uncertain time stamp
<b>Description</b>	Quality information regarding the evaluated time stamp	
<b>Source</b>	AUTOSAR	

### 1.3.2.13 Eth\_TimeStampType

**Table 83** Specification for Eth\_TimeStampType

<b>Syntax</b>	Eth_TimeStampType	
<b>Type</b>	Structure	
<b>File</b>	Eth_GeneralTypes.h	
<b>Range</b>	uint32 nanoseconds	Nanoseconds part of the time
	uint32 seconds	32 bit LSB of the 48 bits seconds part of the time
	uint16 secondsHi	16 bit MSB of the 48 bits seconds part of the time
<b>Description</b>	Variables of this type are used for expressing time stamps including relative time and absolute calendar time. The absolute time starts at 1970-01-01. 0 to 281474976710655s == 3257812230d (0xFFFF FFFF FFFF) 0 to 999999999ns (0x3B9A C9FF) invalid value in nanoseconds:(0x3B9A CA00) to (0x3FFF FFFF) Bit 30 and 31 reserved, default: 0	
<b>Source</b>	AUTOSAR	

## 1.3.3 Functions - APIs

### 1.3.3.1 Eth\_17\_GEthMac\_Init

**Table 84** Specification for Eth\_17\_GEthMac\_Init API

<b>Syntax</b>	<pre>void Eth_17_GEthMac_Init (     const Eth_17_GEthMac_ConfigType * const CfgPtr )</pre>
<b>Service ID</b>	0x01
<b>Sync/Async</b>	Synchronous

**Eth\_17\_GEthMac driver**
**Table 84 Specification for Eth\_17\_GEthMac\_Init API (continued)**

<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CfgPtr	Points to the implementation specific structure
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	This function enables the module, resets the kernel and prepares the MDIO interface. Note: The initialization is performed only for the controllers allocated the core from which Eth_17_GEthMac_Init API is begin invoked.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_INIT_FAILED: Invalid configuration set selection. ETH_17_GETMAC_E_CORE_NOT_CONFIGURED: ETH controller not configured to the core. Runtime Errors: None DEM: ETH_E_ACCESS: ETH controller access failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	none	

**1.3.3.2 Eth\_17\_GEthMac\_SetControllerMode**
**Table 85 Specification for Eth\_17\_GEthMac\_SetControllerMode API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_SetControllerMode (     const uint8 CtrlIdx,     const Eth_ModeType CtrlMode )</pre>	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx CtrlMode	Index of the ETH controller within the context of the ETH driver Mode of the controller

**Eth\_17\_GEthMac driver**
**Table 85 Specification for Eth\_17\_GEthMac\_SetControllerMode API (continued)**

<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: success E_NOT_OK: Controller mode could not be changed
<b>Description</b>	<p>This function performs two actions:</p> <p>Action 1: It chooses the selected MII and completes the ETH controller initialization only for the controller ID passed as the input parameter. This action is done only once when this API is called for the first time after Eth_17_GEthMac_Init.</p> <p>Action 2: It enables or disables the ETH controller with controller ID passed as the input parameter.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_INV_PARAM: Invalid parameter.</p> <p>Runtime Errors: None</p> <p>DEM:</p> <p>ETH_E_ACCESS: ETH controller access failure</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None.	

**1.3.3.3 Eth\_17\_GEthMac\_GetControllerMode**
**Table 86 Specification for Eth\_17\_GEthMac\_GetControllerMode API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_GetControllerMode (     const uint8 CtrlIdx,     Eth_ModeType * const CtrlModePtr )</pre>
<b>Service ID</b>	0x04
<b>Sync/Async</b>	Synchronous

## Eth\_17\_GEthMac driver

**Table 86 Specification for Eth\_17\_GEthMac\_GetControllerMode API (continued)**

<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the ETH Driver
<b>Parameters (out)</b>	CtrlModePtr	ETH_MODE_DOWN: the controller is disabled ETH_MODE_ACTIVE: the controller is enabled
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: success E_NOT_OK: controller mode could not be obtained
<b>Description</b>	Obtains the state of the indexed controller	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None.	

### 1.3.3.4 Eth\_17\_GEthMac\_GetPhysAddr

**Table 87 Specification for Eth\_17\_GEthMac\_GetPhysAddr API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_GetPhysAddr (     const uint8 CtrlIdx,     uint8 * const PhysAddrPtr )</pre>	
<b>Service ID</b>	0x08	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	

**Eth\_17\_GEthMac driver**
**Table 87 Specification for Eth\_17\_GEthMac\_GetPhysAddr API (continued)**

<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of ETH Controller within the context of the ETH driver.
<b>Parameters (out)</b>	PhysAddrPtr	Physical source address (MAC address) in the network byte order.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	Obtains the physical source address used by the indexed controller	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None Note: All DET IDs are also reported as safety errors.	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None.	

**1.3.3.5 Eth\_17\_GEthMac\_SetPhysAddr**
**Table 88 Specification for Eth\_17\_GEthMac\_SetPhysAddr API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_SetPhysAddr (     const uint8 CtrlIdx,     const uint8 * const PhysAddrPtr )</pre>	
<b>Service ID</b>	0x13	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same CtrlIdx, reentrant for different	



**Eth\_17\_GEthMac driver**
**Table 88 Specification for Eth\_17\_GEthMac\_SetPhysAddr API (continued)**

<b>Parameters (in)</b>	CtrlIdx PhysAddrPtr	Index of the ETH controller within the context of the ETH driver. Pointer to memory containing the physical source address (MAC address) in the network byte order.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	Sets the physical source address used by the indexed controller	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	-	

**1.3.3.6 Eth\_17\_GEthMac\_UpdatePhysAddrFilter**
**Table 89 Specification for Eth\_17\_GEthMac\_UpdatePhysAddrFilter API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_UpdatePhysAddrFilter (     const uint8 CtrlIdx,     const uint8 * const PhysAddrPtr,     const Eth_FilterActionType Action )</pre>
<b>Service ID</b>	0x12
<b>Sync/Async</b>	Synchronous
<b>ASIL Level</b>	QM
<b>Re-entrancy</b>	Non Reentrant for the same CtrlIdx, Reentrant for different

**Eth\_17\_GEthMac driver**
**Table 89 Specification for Eth\_17\_GEthMac\_UpdatePhysAddrFilter API (continued)**

<b>Parameters (in)</b>	CtrlIdx PhysAddrPtr Action	Index of the ETH controller within the context of the ETH driver Pointer to the memory containing the physical destination address (MAC address) in the network byte order. This is the multicast destination address of the layer 2 ETH frame. Add or remove the address from the ETH controllers filter.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: filter is successfully changed E_NOT_OK: filter could not be changed
<b>Description</b>	<p>Add or remove the MAC address from the hardware filters</p> <p>The filtering is only done based on the destination address of the received ETH frame.</p> <p>If the physical source address (MAC address) is set to FF:FF:FF:FF:FF:FF, this will completely open the filter.</p> <p>If the physical source address (MAC address) is set to 00:00:00:00:00:00, this will cause to reduce the filter, to the controller's unique unicast MAC address and end promiscuous mode when turned on.</p> <p>A broadcast frame will always be allowed to pass the filter irrespective of the filter state.</p> <p>The Eth_17_GEthMac_UpdatePhysAddrFilter() function is available only when EthUpdatePhysAddrFilter is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_INV_PARAM: Invalid parameter.</p> <p>Runtime Errors: None</p> <p>DEM: None</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	EthUpdatePhysAddrFilter	
<b>User hints</b>	-	

## Eth\_17\_GEthMac driver

### 1.3.3.7 Eth\_17\_GEthMac\_WriteMii

**Table 90 Specification for Eth\_17\_GEthMac\_WriteMii API**

<b>Syntax</b>	<pre> Eth_ReturnType  Eth_17_GEthMac_WriteMii (     const uint8 CtrlIdx,     const uint8 TrcvIdx,     const uint8 RegIdx,     const uint16 RegVal ) </pre>	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx TrcvIdx RegIdx RegVal	Index of ETH Controller within the context of the ETH driver Index of the transceiver on the RGMII/RMII/MII Index of the transceiver register on the RGMII/RMII/MII Value to be written into the indexed register
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Eth_ReturnType	ETH_OK: Service accepted ETH_E_NOT_OK: Service denied ETH_E_NO_ACCESS: ETH transceiver access failure
<b>Description</b>	Configures or writes a transceiver register with the requested value The Eth_17_GEthMac_WriteMii() function is available only when EthCtrlEnableMii is enabled.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	EthCtrlEnableMii	

## Eth\_17\_GEthMac driver

**Table 90 Specification for Eth\_17\_GEthMac\_WriteMii API (continued)**

<b>User hints</b>	None.
-------------------	-------

### 1.3.3.8 Eth\_17\_GEthMac\_ReadMii

**Table 91 Specification for Eth\_17\_GEthMac\_ReadMii API**

<b>Syntax</b>	<pre> Eth_ReturnType Eth_17_GEthMac_ReadMii (     const uint8 CtrlIdx,     const uint8 TrcvIdx,     const uint8 RegIdx,     uint16 * const RegValPtr ) </pre>	
<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx TrcvIdx RegIdx	Index of the controller within the context of the ETH driver Index of the transceiver on the RGMII/RMII/MII Index of the transceiver register on the RGMII/RMII/MII
<b>Parameters (out)</b>	RegValPtr	Filled with the register content of the indexed register
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Eth_ReturnType	ETH_OK: service accepted ETH_E_NOT_OK: service denied ETH_E_NO_ACCESS: ETH transceiver access failure
<b>Description</b>	<p>Reads a transceiver register.</p> <p>The Eth_17_GEthMac_ReadMii() function is available only when EthCtrlEnableMii is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>Runtime Errors: None</p> <p>DEM: None</p>	

## Eth\_17\_GEthMac driver

**Table 91 Specification for Eth\_17\_GEthMac\_ReadMii API (continued)**

	Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	EthCtrlEnableMii
<b>User hints</b>	None.

### 1.3.3.9 Eth\_17\_GEthMac\_GetDropCount

**Table 92 Specification for Eth\_17\_GEthMac\_GetDropCount API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_GetDropCount (     const uint8 CtrlIdx,     const uint8 CountValues,     uint32 * const DropCount )</pre>	
<b>Service ID</b>	0x14	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx CountValues	Index of the controller within the context of the ETH driver Maximal number of values that can be written from DropCount. Note: As per the AUTOSAR specification 4.2.2, the CountValues parameter is IN-OUT. But the parameter is not a pointer. This is recognized as an error and completely reworked for AUTOSAR 4.3 based on Bugzilla 68804. To keep the compatibility of the interface, the CountValues parameter will stay as variable and will not be changed to a pointer. Hence, this parameter will be used as only IN and will not be considered for OUT.
<b>Parameters (out)</b>	DropCount	A pointer to an array where the drop count values of different errors are written
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: success E_NOT_OK: drop counter could not be obtained
<b>Description</b>	Reads a list with drop counter values of the corresponding controller. In the TC3xx devices, the list DropCount[] contains the following values in the given order, where the maximal possible value denotes an invalid value. For example if this counter is not available: - Dropped packets due to buffer overrun - Dropped packets due to CRC errors	

**Eth\_17\_GEthMac driver**
**Table 92 Specification for Eth\_17\_GEthMac\_GetDropCount API (continued)**

	<ul style="list-style-type: none"> <li>- Number of undersize packets which were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well formed. (see IETF RFC 1757)</li> <li>- Number of oversize packets which are longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed. (see IETF RFC 1757)</li> <li>- Number of alignment errors, that is, packets which are received and are not an integral number of octets in length and do not pass the CRC.</li> <li>- SQE test error according to IETF RFC1643 dot3StatsSQETestErrors</li> <li>- The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. (see IETF RFC 2233 ifInDiscards)</li> <li>- Total number of erroneous in-bound packets</li> <li>- The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. (see IETF RFC 2233 ifOutDiscards)</li> <li>- total number of erroneous outbound packets</li> <li>- Single collision frames: A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision. (see IETF RFC1643 dot3StatsSingleCollisionFrames)</li> <li>- Multiple collision frames: A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision. (see IETF RFC1643 dot3StatsMultipleCollisionFrames)</li> <li>- Number of deferred transmission: A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy. (see IETF RFC1643 dot3StatsDeferredTransmissions)</li> <li>- Number of late collisions: The number of times that a collision is detected on a particular interface later than 512 bit-times into the transmission of a packet. (see IETF RFC1643 dot3StatsLateCollisions)</li> <li>- The following positions in the list can contain hardware dependent counter values</li> </ul> <p>Note: From above list, item numbers 6 (SQE test error) is not supported in ETH driver. Corresponding this error count, a value 0xFFFFFFFF (ETH_COUNTER_NOT_AVAILABLE) will be filled in DropCount array. Collision related count will be available only if mode of operation is half duplex.</p> <p>The EthGetDropCountApi() function is available only when EthGetDropCountApi is enabled.</p>
<b>Source</b>	AUTOSAR
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p>

**Eth\_17\_GEthMac driver**
**Table 92 Specification for Eth\_17\_GEthMac\_GetDropCount API (continued)**

	Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	EthGetDropCountApi
<b>User hints</b>	-

**1.3.3.10 Eth\_17\_GEthMac\_GetEtherStats**
**Table 93 Specification for Eth\_17\_GEthMac\_GetEtherStats API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_GetEtherStats (     const uint8 CtrlIdx,     uint32 * const etherStats )</pre>	
<b>Service ID</b>	0x15	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the ETH driver
<b>Parameters (out)</b>	etherStats	List of values according to IETF RFC 2819 (Remote Network Monitoring Management Information Base)
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: success E_NOT_OK: drop counter could not be obtained
<b>Description</b>	Returns the following, according to IETF RFC2819, where the maximal possible value will denote an invalid value, for example, if this counter is not available: <ul style="list-style-type: none"> <li>- etherStatsDropEvents</li> <li>- etherStatsOctets</li> <li>- etherStatsPkts</li> <li>- etherStatsBroadcastPkts</li> <li>- etherStatsMulticastPkts</li> <li>- etherStatsCrcAlignErrors</li> <li>- etherStatsUndersizePkts</li> <li>- etherStatsOversizePkts</li> <li>- etherStatsFragments</li> <li>- etherStatsJabbers</li> </ul>	

**Eth\_17\_GEthMac driver**
**Table 93 Specification for Eth\_17\_GEthMac\_GetEtherStats API (continued)**

	<ul style="list-style-type: none"> <li>- etherStatsCollisions</li> <li>- etherStatsPkts64Octets</li> <li>- etherStatsPkts65to127Octets</li> <li>- etherStatsPkts128to255Octets</li> <li>- etherStatsPkts256to511Octets</li> <li>- etherStatsPkts512to1023Octets</li> <li>- etherStatsPkts1024to1518Octets</li> </ul> <p>Note: In the above list, items which are not available are filled with value 0xFFFFFFFF (ETH_COUNTER_NOT_AVAILABLE). Collision-related status is available only if the mode of operation is half duplex.</p> <p>The Eth_17_GEthMac_GetEtherStats() function is available only when EthGetEtherStatsApi is enabled.</p>
<b>Source</b>	AUTOSAR
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>Runtime Errors: None</p> <p>DEM: None</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>
<b>Configuration dependencies</b>	EthGetEtherStatsApi
<b>User hints</b>	-

**1.3.3.11 Eth\_17\_GEthMac\_GetCurrentTime**
**Table 94 Specification for Eth\_17\_GEthMac\_GetCurrentTime API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_GetCurrentTime (     const uint8 CtrlIdx,     Eth_TimeStampQualType * const timeQualPtr,     Eth_TimeStampType * const timeStampPtr )</pre>
<b>Service ID</b>	0x16
<b>Sync/Async</b>	Synchronous



**Eth\_17\_GEthMac driver**
**Table 94 Specification for Eth\_17\_GEthMac\_GetCurrentTime API (continued)**

<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the ETH driver
<b>Parameters (out)</b>	timeQualPtr timeStampPtr	Quality of hardware time stamp, for example, based on current drift.  Note: Since the TC38xx ETH controller does not provide a quality information, the reported value is always valid.  Current time stamp
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: successful E_NOT_OK: failed
<b>Description</b>	Returns a time value from the hardware timer registers. The Eth_17_GEthMac_GetCurrentTime() function is available only when EthGlobalTimeSupport is enabled.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	EthGlobalTimeSupport	
<b>User hints</b>	None.	

**1.3.3.12 Eth\_17\_GEthMac\_EnableEgressTimeStamp**
**Table 95 Specification for Eth\_17\_GEthMac\_EnableEgressTimeStamp API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_EnableEgressTimeStamp (     const uint8 CtrlIdx,</pre>
---------------	---

**Eth\_17\_GEthMac driver**
**Table 95 Specification for Eth\_17\_GEthMac\_EnableEgressTimeStamp API (continued)**

	<pre>const uint8 BufIdx )</pre>	
<b>Service ID</b>	0x17	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx BufIdx	Index of the controller within the context of the ETH driver Index of the message buffer, where application expects egress time stamping
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	<p>Activates egress time stamping on a dedicated message object( or message buffer)</p> <p>The Eth_17_GEthMac_EnableEgressTimeStamp() function is available only when EthGlobalTimeSupport is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_INV_PARAM: Invalid parameter.</p> <p>Runtime Errors: None</p> <p>DEM: None</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	EthGlobalTimeSupport	
<b>User hints</b>	None.	

**1.3.3.13 Eth\_17\_GEthMac\_GetEgressTimeStamp**
**Table 96 Specification for Eth\_17\_GEthMac\_GetEgressTimeStamp API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_GetEgressTimeStamp (</pre>
---------------	---

**Eth\_17\_GEthMac driver**
**Table 96 Specification for Eth\_17\_GEthMac\_GetEgressTimeStamp API (continued)**

	<pre> const uint8 CtrlIdx, const uint8 BufIdx, Eth_TimeStampQualType * const timeQualPtr, Eth_TimeStampType * const timeStampPtr ) </pre>	
<b>Service ID</b>	0x18	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx BufIdx	Index of the controller within the context of the ETH driver Index of the message buffer, where application expects egress time stamping
<b>Parameters (out)</b>	timeQualPtr timeStampPtr	Quality of hardware time stamp, for example based on current drift Current time stamp
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	<p>Reads back the egress time stamp on a dedicated message object. It must be called within the TxConfirmation() function.</p> <p>The Eth_17_GEthMac_GetEgressTimeStamp() function is available only when EthGlobalTimeSupport is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_INV_PARAM: Invalid parameter.</p> <p>Runtime Errors: None</p> <p>DEM: None</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	EthGlobalTimeSupport	
<b>User hints</b>	None.	

## Eth\_17\_GEthMac driver

### 1.3.3.14 Eth\_17\_GEthMac\_GetIngressTimeStamp

**Table 97** Specification for Eth\_17\_GEthMac\_GetIngressTimeStamp API

<b>Syntax</b>	<pre>void Eth_17_GEthMac_GetIngressTimeStamp (     const uint8 CtrlIdx,     const Eth_DataType * const DataPtr,     Eth_TimeStampQualType * const timeQualPtr,     Eth_TimeStampType * const timeStampPtr )</pre>	
<b>Service ID</b>	0x19	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx DataPtr	Index of the controller within the context of the ETH driver Pointer to the message buffer, where application expects ingress time stamping Note: Since the ETH driver does not need content of message buffer for extracting time stamp, this parameter is not used in ETH driver design
<b>Parameters (out)</b>	timeQualPtr timeStampPtr	Quality of hardware time stamp, for example based on current drift Current time stamp
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	<p>Reads back the ingress time stamp on a dedicated message object. It must be called within the RxIndication() function.</p> <p>The Eth_17_GEthMac_GetIngressTimeStamp() function is available only when EthGlobalTimeSupport is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	

## Eth\_17\_GEthMac driver

**Table 97 Specification for Eth\_17\_GEthMac\_GetIngressTimeStamp API (continued)**

<b>Configuration dependencies</b>	EthGlobalTimeSupport
<b>User hints</b>	None.

### 1.3.3.15 Eth\_17\_GEthMac\_SetCorrectionTime

**Table 98 Specification for Eth\_17\_GEthMac\_SetCorrectionTime API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_SetCorrectionTime (     const uint8 CtrlIdx,     const Eth_TimeIntDiffType * const timeOffsetPtr,     const Eth_RateRatioType * const rateRatioPtr )</pre>	
<b>Service ID</b>	0x1a	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx timeOffsetPtr rateRatioPtr	Index of the controller within the context of the ETH driver Offset between time stamp grandmaster and time stamp by local clock: $(\text{OriginTimeStampSync} - \text{IngressTimeStampSync}) + \text{Pdelay}$ Time elements to calculate and to modify the ratio of the frequency of the grandmaster in relation to the frequency of the local clock with: $\text{ratio} = \text{OriginTimeStampDelta} / \text{IngressTimeStampDelta}$
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	None.
<b>Description</b>	<p>Allows the time slave to adjust the local ETH reference clock in the hardware. This function updates or offsets the ETH driver timer as per timeOffsetPtr and corrects the time difference by correcting the timer clock as per rateRatioPtr.</p> <p>The Eth_17_GEthMac_SetCorrectionTime() function is available only when EthGlobalTimeSupport is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.	

**Eth\_17\_GEthMac driver**
**Table 98 Specification for Eth\_17\_GEthMac\_SetCorrectionTime API (continued)**

	<p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>Runtime Errors: None</p> <p>DEM:</p> <p>ETH_E_ACCESS: ETH controller access failure</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>
<b>Configuration dependencies</b>	EthGlobalTimeSupport
<b>User hints</b>	None.

**1.3.3.16 Eth\_17\_GEthMac\_SetGlobalTime**
**Table 99 Specification for Eth\_17\_GEthMac\_SetGlobalTime API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_SetGlobalTime (     const uint8 CtrlIdx,     const Eth_TimeStampType * const timeStampPtr )</pre>	
<b>Service ID</b>	0x1b	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx timeStampPtr	Index of the controller within the context of the ETH driver New time stamp
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: successful E_NOT_OK: failed
<b>Description</b>	<p>Allows the time master to adjust the global ETH reference clock in the hardware.</p> <p>We can use this method to set a global time base on the ETH in general or to synchronize the global ETH time base with another time base, for example, FlexRay.</p> <p>The Eth_17_GEthMac_SetGlobalTime() function is available only when EthGlobalTimeSupport is enabled.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET:	

## Eth\_17\_GEthMac driver

**Table 99 Specification for Eth\_17\_GEthMac\_SetGlobalTime API (continued)**

	<p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>Runtime Errors: None</p> <p>DEM:</p> <p>ETH_E_ACCESS: ETH controller access failure</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>
<b>Configuration dependencies</b>	EthGlobalTimeSupport
<b>User hints</b>	None.

### 1.3.3.17 Eth\_17\_GEthMac\_ProvideTxBuffer

**Table 100 Specification for Eth\_17\_GEthMac\_ProvideTxBuffer API**

<b>Syntax</b>	<pre>BufReq_ReturnType Eth_17_GEthMac_ProvideTxBuffer (     const uint8 CtrlIdx,     Eth_BufIdxType * const BufIdxPtr,     uint8 ** const BufPtr,     uint16 * const LenBytePtr )</pre>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the ETH controller within the context of the ETH driver
<b>Parameters (out)</b>	BufIdxPtr	Index to the granted buffer resource. To be used for subsequent requests
	BufPtr	Pointer to the granted buffer
<b>Parameters (in - out)</b>	LenBytePtr	IN: desired length in bytes, OUT: granted length in bytes.
<b>Return</b>	BufReq_ReturnType	<p>BUFREQ_OK: buffer provided successfully</p> <p>BUFREQ_E_NOT_OK: API call aborted due to development errors</p> <p>BUFREQ_E_BUSY: all buffers are used</p>

**Eth\_17\_GEthMac driver**
**Table 100 Specification for Eth\_17\_GEthMac\_ProvideTxBuffer API (continued)**

	BUFREQ_E_OVFL: requested buffer too large
<b>Description</b>	Provides access to a transmit buffer of the specified controller
<b>Source</b>	AUTOSAR
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>Runtime Errors: None</p> <p>DEM: None</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None.

**1.3.3.18 Eth\_17\_GEthMac\_Transmit**
**Table 101 Specification for Eth\_17\_GEthMac\_Transmit API**

<b>Syntax</b>	<pre>Std_ReturnType Eth_17_GEthMac_Transmit (     const uint8 CtrlIdx,     const Eth_BufIdxType BufIdx,     const Eth_FrameType FrameType ,     const boolean TxConfirmation,     const uint16 LenByte,     const uint8 * const PhysAddrPtr )</pre>	
<b>Service ID</b>	0xA	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the ETH driver
	BufIdx	Index of the buffer resource
	FrameType	ETH frame type
	TxConfirmation	Activates transmission confirmation
	LenByte	Data length in byte



**Eth\_17\_GEthMac driver**
**Table 101 Specification for Eth\_17\_GEthMac\_Transmit API (continued)**

	PhysAddrPtr	Physical target address (MAC address) in the network byte order
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK: success E_NOT_OK: transmission failed
<b>Description</b>	Triggers transmission of a previously filled transmit buffer	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_INV_PARAM: Invalid parameter. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list. ETH_17_GETHMAC_E_INV_MODE: Invalid controller mode. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	-	

**1.3.3.19 Eth\_17\_GEthMac\_Receive**
**Table 102 Specification for Eth\_17\_GEthMac\_Receive API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_Receive (     const uint8 CtrlIdx,     Eth_RxStatusType * const RxStatusPtr )</pre>
<b>Service ID</b>	0xB
<b>Sync/Async</b>	Synchronous
<b>ASIL Level</b>	QM
<b>Re-entrancy</b>	Non Reentrant

**Eth\_17\_GEthMac driver**
**Table 102 Specification for Eth\_17\_GEthMac\_Receive API (continued)**

<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the ETH driver
<b>Parameters (out)</b>	RxStatusPtr	Indicates whether a frame has been received and if so, whether more frames are available or frames got lost.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	<p>Triggers frame reception.</p> <p>The Eth_17_GEthMac_Receive function reads the next frame from the receive buffers. This function passes the received frame to the ETH interface using the EthIf_RxIndication callback function and indicates if there are more frames in the receive buffers through RxStatusPtr.</p> <p>When calling the EthIf_RxIndication callback function, the broadcast frames are indicated to the ETH interface.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized.</p> <p>Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.</p> <p>ETH_17_GETHMAC_E_INV_MODE: Invalid controller mode.</p> <p>ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index.</p> <p>Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core.</p> <p>Runtime Errors: None</p> <p>DEM: None</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	-	

**1.3.3.20 Eth\_17\_GEthMac\_TxConfirmation**
**Table 103 Specification for Eth\_17\_GEthMac\_TxConfirmation API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_TxConfirmation (     const uint8 CtrlIdx )</pre>
<b>Service ID</b>	0xC
<b>Sync/Async</b>	Synchronous
<b>ASIL Level</b>	QM

**Eth\_17\_GEthMac driver**
**Table 103 Specification for Eth\_17\_GEthMac\_TxConfirmation API (continued)**

<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the ETH driver
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	Triggers frame transmission confirmation	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_INV_MODE: Invalid controller mode. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	-	

**1.3.3.21 Eth\_17\_GEthMac\_GetVersionInfo**
**Table 104 Specification for Eth\_17\_GEthMac\_GetVersionInfo API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_GetVersionInfo (     Std_VersionInfoType * const VersionInfoPtr )</pre>	
<b>Service ID</b>	0xD	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Reentrant	
<b>Parameters (in)</b>	-	-

**Eth\_17\_GEthMac driver**
**Table 104 Specification for Eth\_17\_GEthMac\_GetVersionInfo API (continued)**

<b>Parameters (out)</b>	VersionInfoPtr	Version information of this module
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	Returns the version information of the ETH driver.  The Eth_17_GEthMac_GetVersionInfo() function is available only when, EthVersionInfoApi() is enabled.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_PARAM_POINTER: Invalid pointer in parameter list.  Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	EthVersionInfoApi	
<b>User hints</b>	None.	

### 1.3.4 Notifications and Callbacks

The ETH driver does not support any notification and callbacks.

### 1.3.5 Scheduled functions

#### 1.3.5.1 Eth\_17\_GEthMac\_MainFunction

**Table 105 Specification for Eth\_17\_GEthMac\_MainFunction API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_MainFunction (     void )</pre>	
<b>Service ID</b>	0x1C	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	-	-

**Eth\_17\_GEthMac driver**
**Table 105 Specification for Eth\_17\_GEthMac\_MainFunction API (continued)**

<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	This function checks for the controller errors and lost frames.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: None Runtime Errors: None DEM: ETH_E_RX_FRAMES_LOST: ETH frames lost ETH_E_CRC: CRC failure ETH_E_OVERSIZEFRAME : Frame size overflow ETH_E_UNDERSIZEFRAME: Frame size underflow ETH_E_ALIGNMENT: Frame alignment error ETH_E_SINGLECOLLISION: Single frame collision ETH_E_LATECOLLISION: Late frame collision ETH_E_MULTIPLECOLLISION: Multiple frame collision Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None.	

## 1.3.6 Interrupt service routines

### 1.3.6.1 Eth\_17\_GEthMac\_RxDmaCh0IrqHdlr

**Table 106 Specification for Eth\_17\_GEthMac\_RxDmaCh0IrqHdlr API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_RxDmaCh0IrqHdlr (     const uint8 CtrlIdx )</pre>	
<b>Service ID</b>	0x10	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same CtrlIdx, reentrant for different	

**Eth\_17\_GEthMac driver**
**Table 106 Specification for Eth\_17\_GEthMac\_RxDmaCh0IrqHdlr API (continued)**

<b>Parameters (in)</b>	CtrlIdx	Controller Index
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	None.
<b>Description</b>	IRQ handler for the frame reception interrupt and receive buffer unavailable interrupt from the receive DMA channel-0 for the controller with ID passed as the input parameter. Note: The ETH driver is not handling any error-related interrupts.	
<b>Source</b>	IFX	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. Runtime Errors: None DEM: None Safety Errors: None Note: All DET IDs are also reported as safety errors.	
<b>Configuration dependencies</b>	EthCtrlEnableRxInterrupt	
<b>User hints</b>	None.	

**1.3.6.2 Eth\_17\_GEthMac\_TxDmaCh0IrqHdlr**
**Table 107 Specification for Eth\_17\_GEthMac\_TxDmaCh0IrqHdlr API**

<b>Syntax</b>	<pre>void Eth_17_GEthMac_TxDmaCh0IrqHdlr (     const uint8 CtrlIdx )</pre>
<b>Service ID</b>	0x11
<b>Sync/Async</b>	Synchronous
<b>ASIL Level</b>	QM
<b>Re-entrancy</b>	Non Reentrant for the same CtrlIdx, reentrant for different

**Eth\_17\_GEthMac driver**
**Table 107 Specification for Eth\_17\_GEthMac\_TxDmaCh0IrqHdlr API (continued)**

<b>Parameters (in)</b>	CtrlIdx	Controller Index
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	None.
<b>Description</b>	IRQ handler for the frame transmission interrupt from transmits DMA channel-0 for the controller with ID passed as the input parameter. Note: The ETH driver does not handle any error-related interrupts.	
<b>Source</b>	IFX	
<b>Error handling</b>	DET: ETH_17_GETHMAC_E_NOT_INITIALIZED: ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core. ETH_17_GETHMAC_E_INV_CTRL_IDX: Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is not allocated to the current core. Runtime Errors: None DEM: None Safety Errors: None Note: All DET IDs are also reported as safety errors.	
<b>Configuration dependencies</b>	EthCtrlEnableTxInterrupt	
<b>User hints</b>	None.	

### 1.3.7 Error codes classification

This section explains various error types and their corresponding source APIs.

#### 1.3.7.1 Development errors

The following table lists all the development errors reported by the driver.

**Table 108 Description of development errors reported**

Description	Source	Error code and value	Applicable APIs
ETH controller not configured to the core.	IFX	ETH_17_GETMAC_E_CORE_NOT_CONFIGURED=0x64	Eth_17_GEthMac_Init
Invalid controller index. Note: This DET is reported if the controller index is invalid/if the controller is	AUTOSAR	ETH_17_GETHMAC_E_INV_CTRL_IDX=0x01	Eth_17_GEthMac_RxDmaCh0IrqHdlr, Eth_17_GEthMac_TxDmaCh0IrqHdlr,

**Eth\_17\_GEthMac driver**
**Table 108 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
not allocated to the current core.			Eth_17_GEthMac_GetEtherStats, Eth_17_GEthMac_GetDropCount, Eth_17_GEthMac_SetCorrectionTime, Eth_17_GEthMac_SetGlobalTime, Eth_17_GEthMac_GetIngressTimeStamp, Eth_17_GEthMac_GetEgressTimeStamp, Eth_17_GEthMac_EnableEgressTimeStamp, Eth_17_GEthMac_GetCurrentTime, Eth_17_GEthMac_TxConfirmation, Eth_17_GEthMac_Receive, Eth_17_GEthMac_Transmit, Eth_17_GEthMac_UpdatePhysAddrFilter, Eth_17_GEthMac_SetPhysAddr, Eth_17_GEthMac_ProvideTxBuffer, Eth_17_GEthMac_GetPhysAddr, Eth_17_GEthMac_GetControllerMode, Eth_17_GEthMac_WriteMii, Eth_17_GEthMac_ReadMii, Eth_17_GEthMac_SetControllerMode
ETH driver and controller is not initialized. Note: This DET is reported if Eth_17_GEthMac_Init() API is not called before invoking runtime APIs and if controller is not configured to the current core.	AUTOSAR	ETH_17_GETHMAC_E_NOT_INITIALIZED=0x02	Eth_17_GEthMac_TxDmaCh0IrqHdlr, Eth_17_GEthMac_RxDmaCh0IrqHdlr, Eth_17_GEthMac_GetEtherStats, Eth_17_GEthMac_GetDropCount, Eth_17_GEthMac_SetCorrectionTime, Eth_17_GEthMac_SetGlob



**Eth\_17\_GEthMac driver**
**Table 108 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
			alTime, Eth_17_GEthMac_GetIngr essTimeStamp, Eth_17_GEthMac_GetEgre ssTimeStamp, Eth_17_GEthMac_EnableE gressTimeStamp, Eth_17_GEthMac_GetCurr entTime, Eth_17_GEthMac_TxConfi rmation, Eth_17_GEthMac_Receive, Eth_17_GEthMac_Transmi t, Eth_17_GEthMac_Update PhysAddrFilter , Eth_17_GEthMac_SetPhys Addr, Eth_17_GEthMac_Provide TxBuffer, Eth_17_GEthMac_GetPhys Addr, Eth_17_GEthMac_GetCont rollerMode, Eth_17_GEthMac_WriteMii , Eth_17_GEthMac_ReadMii , Eth_17_GEthMac_SetCont rollerMode
Invalid pointer in parameter list.	AUTOSAR	ETH_17_GETHMAC_E_PARAM_POINT ER=0x03	Eth_17_GEthMac_SetPhys Addr, Eth_17_GEthMac_GetCont rollerMode, Eth_17_GEthMac_GetEthe rStats, Eth_17_GEthMac_GetDro pCount, Eth_17_GEthMac_SetCorr ectionTime, Eth_17_GEthMac_SetGlob alTime, Eth_17_GEthMac_GetIngr essTimeStamp, Eth_17_GEthMac_GetEgre ssTimeStamp, Eth_17_GEthMac_GetCurr entTime,

**Eth\_17\_GEthMac driver**
**Table 108 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
			Eth_17_GEthMac_Transmit, Eth_17_GEthMac_UpdatePhysAddrFilter, Eth_17_GEthMac_GetVersionInfo, Eth_17_GEthMac_ProvideTxBuffer, Eth_17_GEthMac_GetPhysAddr, Eth_17_GEthMac_ReadMii
Invalid parameter.	AUTOSAR	ETH_17_GETHMAC_E_INV_PARAM=0x04	Eth_17_GEthMac_UpdatePhysAddrFilter, Eth_17_GEthMac_SetControllerMode, Eth_17_GEthMac_GetEgressTimeStamp, Eth_17_GEthMac_EnableEgressTimeStamp, Eth_17_GEthMac_Transmit
Invalid controller mode.	AUTOSAR	ETH_17_GETHMAC_E_INV_MODE=0x05	Eth_17_GEthMac_TxConfirmation, Eth_17_GEthMac_Receive, Eth_17_GEthMac_Transmit
Invalid configuration set selection.	AUTOSAR	ETH_17_GETHMAC_E_INIT_FAILED=0x20	Eth_17_GEthMac_Init

**1.3.7.2 Production errors**

The following table lists all the production errors reported by the driver.

**Table 109 Description of production errors reported**

Description	Source	Error code and value	Applicable APIs
ETH controller access failure.	AUTOSAR	ETH_E_ACCESS=Value Assigned by DEM	Eth_17_GEthMac_SetCorrectionTime, Eth_17_GEthMac_SetGlobalTime, Eth_17_GEthMac_Init, Eth_17_GEthMac_SetControllerMode
Frame alignment error.	AUTOSAR	ETH_E_ALIGNMENT=Value Assigned by DEM	Eth_17_GEthMac_MainFunction

**Eth\_17\_GEthMac driver**
**Table 109 Description of production errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
CRC failure.	AUTOSAR	ETH_E_CRC=Value Assigned by DEM	Eth_17_GEthMac_MainFunction
Late frame collision.	AUTOSAR	ETH_E_LATECOLLISION=Value Assigned by DEM	Eth_17_GEthMac_MainFunction
Multiple frame collision.	AUTOSAR	ETH_E_MULTIPLECOLLISION=Value Assigned by DEM	Eth_17_GEthMac_MainFunction
Frame size overflow.	AUTOSAR	ETH_E_OVERSIZEFRAME =Value Assigned by DEM	Eth_17_GEthMac_MainFunction
ETH frames lost.	AUTOSAR	ETH_E_RX_FRAMES_LOST=Value Assigned by DEM	Eth_17_GEthMac_MainFunction
Single frame collision.	AUTOSAR	ETH_E_SINGLECOLLISION=Value Assigned by DEM	Eth_17_GEthMac_MainFunction
Frame size underflow.	AUTOSAR	ETH_E_UNDERSIZEFRAME=Value Assigned by DEM	Eth_17_GEthMac_MainFunction

### 1.3.7.3 Safety errors

The driver does not report any safety errors.

### 1.3.7.4 Runtime errors

The driver does not report any runtime errors.

## 1.3.8 Deviations and limitations

The section describes the deviations and limitations from software specification.

### 1.3.8.1 Deviations

The section describes the deviations from software specification.

**Table 110 Known deviations**

Reference	Deviation
AUTOSAR requirements [SWS_Eth_00216] ,[SWS_Eth_00217].	Individual enabling of hardware checksum offload functionality for IPV4, UDP, TCP, ICMP frames is not possible due to hardware limitation. Enabling any one of the following configuration parameters: EthCtrlEnableOffloadChecksumIPV4, EthCtrlEnableOffloadChecksumUDP, EthCtrlEnableOffloadChecksumTCP, EthCtrlEnableOffloadChecksumICMP enables the checksum offload functionality. This is a deviation from the AUTOSAR requirements [SWS_Eth_00216] , [SWS_Eth_00217].

**Eth\_17\_GEthMac driver**
**Table 110 Known deviations (continued)**

Reference	Deviation
AUTOSAR requirement [SWS_Eth_00226], has wrong syntax for second parameter CountValues.	In AUTOSAR specification, as per the syntax of the <code>Eth_17_GEthMac_GetDropCount()</code> API, the datatype of <code>CountValues</code> parameter shall be <code>uint8</code> . However, this parameter is mentioned as <code>InOut</code> in the AUTOSAR specification. Since the parameter is not a pointer type, it cannot be used as <code>Out</code> parameter. Hence this parameter is used only as <code>In</code> parameter in the Ethernet driver.
AUTOSAR requirement [SWS_EthIf_00085], has incorrect description of parameter <code>LenByte</code> , which is not meaningful.	The Ethernet driver will call <code>EthIf_RxIndication</code> to indicate a successful reception. <code>LenByte</code> provided by Ethernet driver to <code>EthIf_RxIndication</code> will be the length of payload. But as per [SWS_EthIf_00085], this should be length of the received frame bytes which includes frame header length of 14 bytes (including target MAC address, source MAC address, and type) and payload length, but this is not meaningful since the <code>DataPtr</code> points to payload. AUTOSAR Bugzilla is created to change the description of <code>LenByte</code> . Refer <a href="https://www.autosar.org/bugzilla/show_bug.cgi?id=76835">https://www.autosar.org/bugzilla/show_bug.cgi?id=76835</a>
AUTOSAR header file inclusion requirement for ETH module.	As per the AUTOSAR requirement, <code>Dem.h</code> shall be included in the <code>Eth_17_GEthMac.c</code> file. The Ethernet module configuration structure defined in <code>Eth_17_GEthMac.h</code> file refer the data type <code>Dem_EventIdType</code> from <code>Dem</code> module. Hence to avoid compilation error, <code>Dem.h</code> is included in <code>Eth_17_GEthMac.h</code> file.
Transmit and receive buffer size configuration parameters.	For the parameters <code>EthCtrlTxBufLenByte</code> , <code>EthCtrlRxBufLenByte</code> , <code>EthRxBufTotal</code> and <code>EthTxBufTotal</code> the value configuration class is pre-compile and post-build variant value is <code>FALSE</code> .
AUTOSAR requirements [SWS_Eth_00058], [SWS_Eth_00064].	As per AUTOSAR specification, the return type of <code>Eth_17_GEthMac_WriteMii()</code> and <code>Eth_17_GEthMac_ReadMii()</code> APIs shall be <code>Std_ReturnType</code> . But the Ethernet driver implemented the return type as <code>Eth_ReturnType</code> . This is due to an error in Autosar specification that includes a third return value <code>ETH_E_NO_ACCESS</code> which is not part of <code>Std_ReturnType</code> . Hence the return values mentioned in the specification are implemented using <code>Eth_ReturnType</code> type.

### 1.3.8.2 Limitations

The section describes the limitations from software specification.

**Table 111 Known limitations**

Reference	Limitation
Ethernet transceivers intermittently fails to transmit first Ethernet packet transmitted from Ethernet MAC.	Ethernet driver testing is performed using Triboard which is an evaluation board from Infineon. It is observed that the transceivers used in Triboard intermittently fail to transmit first packet transmitted from Ethernet MAC. But it is tested and confirmed that all packets which are sent from Ethernet MAC are transmitted

**Eth\_17\_GEthMac driver**
**Table 111 Known limitations (continued)**

Reference	Limitation
	<p>successfully to Ethernet transceivers. Using a qualified Ethernet transceiver subsystem (hardware and driver software) should resolve this behavior.</p> <p>The work around followed while testing the Ethernet driver is, after the transceiver is initialized a delay of 3 to 4 seconds is added in test code.</p>
Ethernet transmit intermittently fails when configured in RGMII/10mbps mode.	Work around: None.
Eth_17_GEthMac_Transmit() API does not work if BufIdx parameter is not passed in same sequence as it is provided.	<p>Eth_17_GEthMac_Transmit() API is designed to work in such a way that BufIdx parameter passed to this API shall be in same sequence as BufIdx is allocated by calling Eth_17_GEthMac_ProvideTxBuffer() API.</p> <p>Example for right usage of this API is given below</p> <p>Step1- Invoke Eth_17_GEthMac_ProvideTxBuffer() API - Assume that BufIdx1 is allocated to application from this API</p> <p>Step2- Invoke Eth_17_GEthMac_ProvideTxBuffer() API - Assume that BufIdx2 is allocated to application from this API</p> <p>Step3- Invoke Eth_17_GEthMac_Transmit() API with parameter as BufIdx1</p> <p>Step4- Invoke Eth_17_GEthMac_Transmit() API with parameter as BufIdx2</p> <p>In this case the limitation is, If Step4 is followed before Step3, then the Eth_17_GEthMac_Transmit() API does not work. The Eth_17_GEthMac_Transmit() API is designed in this way to make use of a feature from Ethernet controller hardware that it automatically manages linked list circular buffer. Hence Ethernet driver performance is increased by avoiding this feature in software.</p>
Usage of Compiler library in Ethernet driver for compilation of global time APIs.	<p>If the pre-compile configuration parameter EthGlobalTimeSupport is enabled in Ethernet driver configuration, then the Ethernet driver will use the compiler's floating point library for implementation of double precision floating point in global time APIs. Infineon has not validated the compiler's floating library used by the Ethernet driver.</p>

### 1.3.9 Unsupported hardware features

The following hardware features of the Ethernet controller are not currently supported by the ETH driver:

- Single and double VLAN tagged frames
- Jumbo frames
- Source address filters are not supported
- Multiple DMA channels
- Multiple priority based for queues
- Loopback mode

---

**Eth\_17\_GEthMac driver**

- Energy Efficient ETH (EEE)
- Pulse per second output
- Promiscuous mode

## 2 Fr\_17\_Eray driver

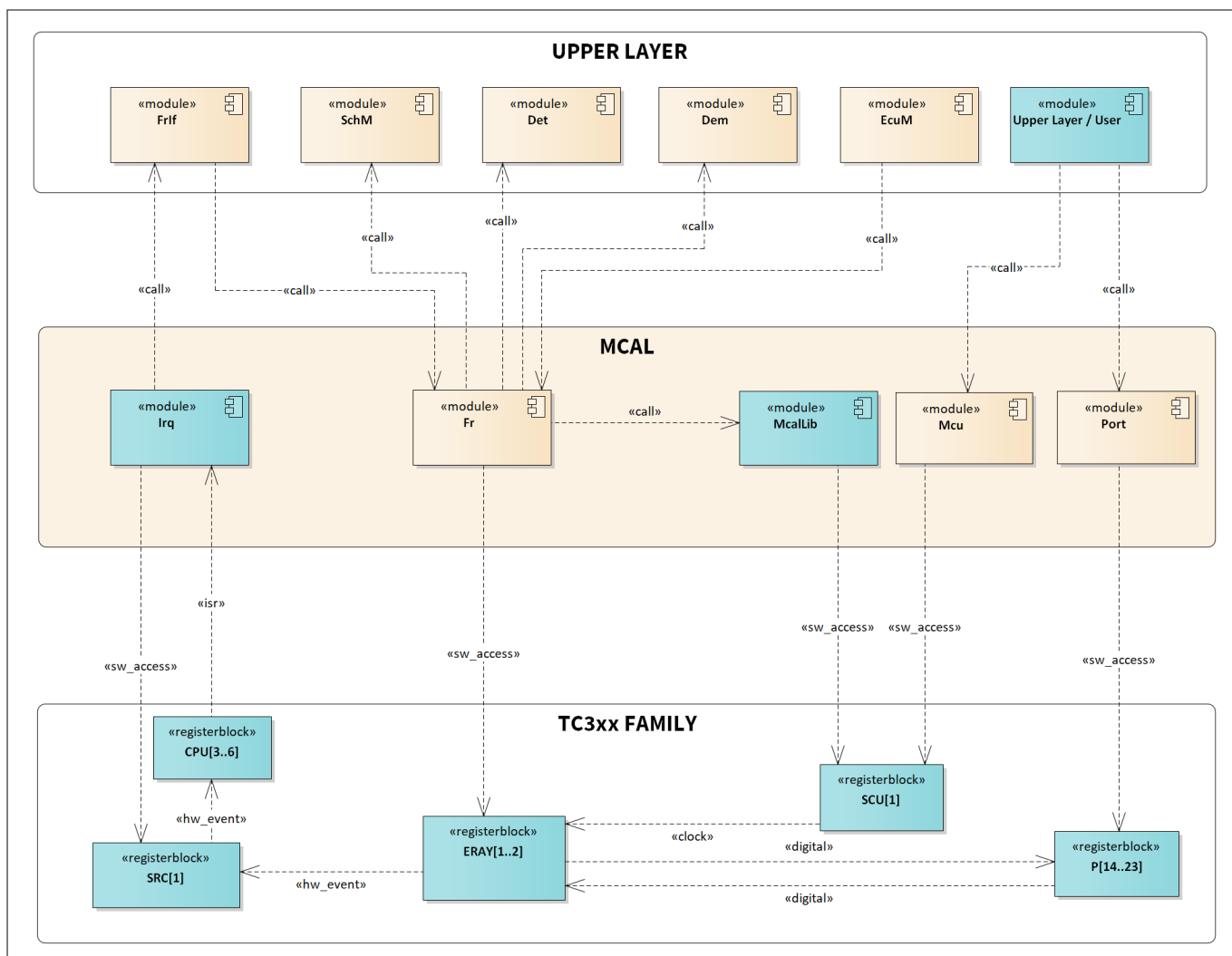
### 2.1 User information

#### 2.1.1 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.

#### 2.1.2 Hardware-software mapping

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



**Figure 7 Mapping of hardware-software interfaces**

### **2.1.2.1 ERAY: primary hardware peripheral**

#### **Hardware functional features**

The FR driver uses the ERAY driver 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
- Interrupt service requests are not handled by the FR driver, but they are expected to be handled by the FlexRay Interface(FrIf) module

### **2.1.2.2 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.



---

**Fr\_17\_Eray driver****2.1.2.3 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.

**2.1.2.4 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.

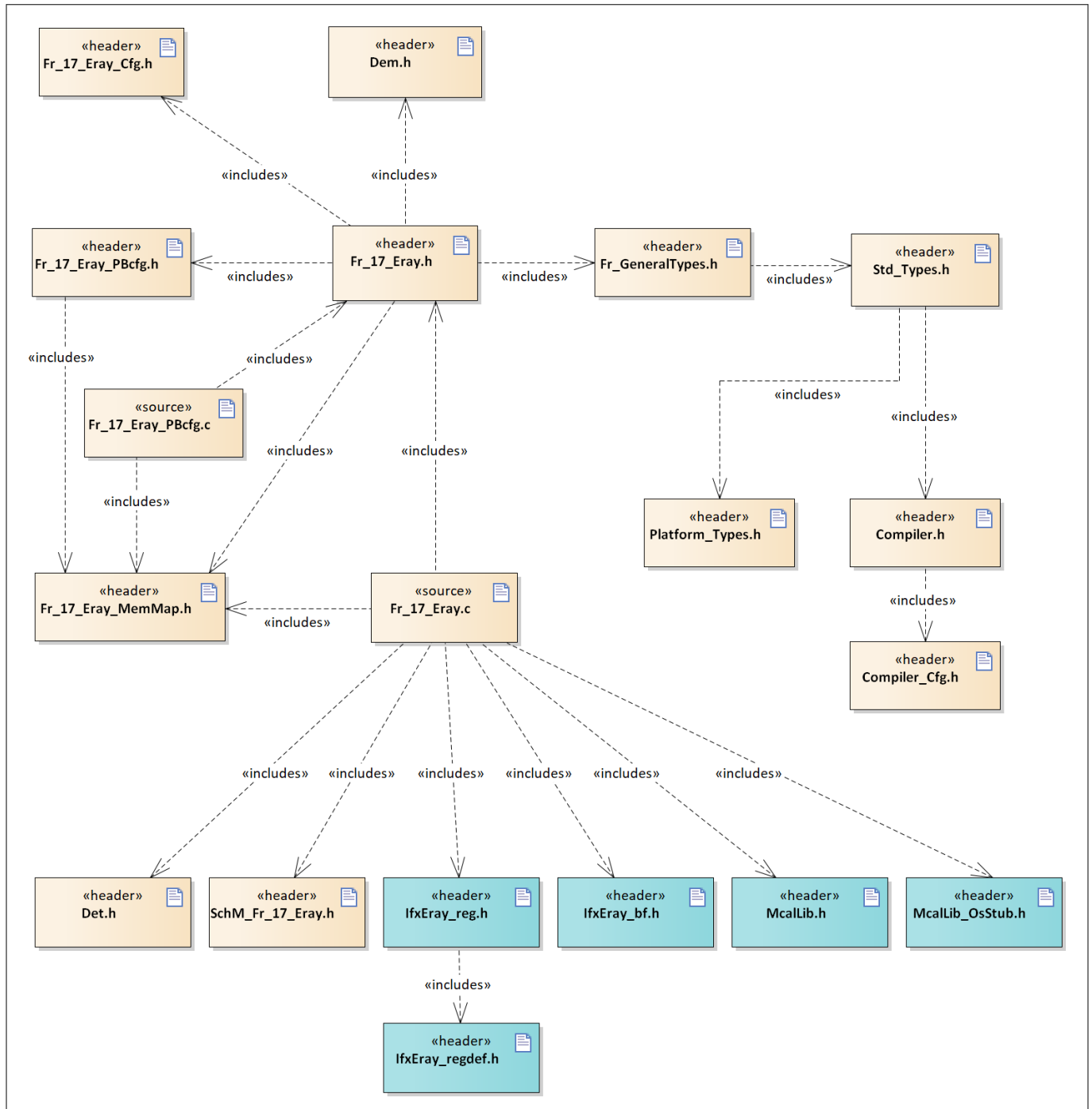
**Hardware events**

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

**2.1.3 File structure****2.1.3.1 C file structure**

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

### Fr\_17\_Eray driver



**Figure 8 C file structure**

**Table 112 C file structure**

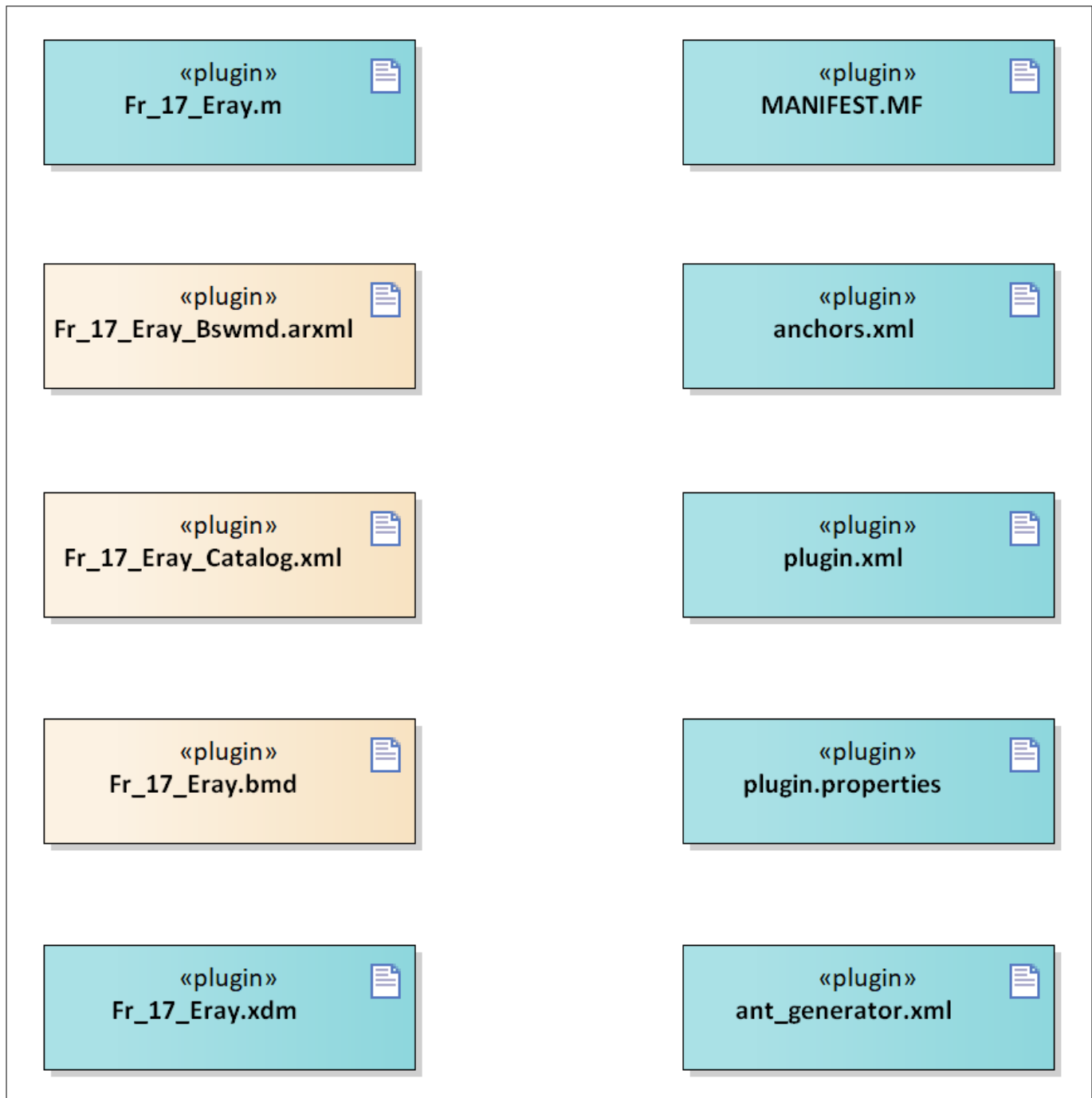
File name	Description
Compiler.h	Provides abstraction from compiler-specific keywords
Compiler_Cfg.h	Configuration header file for compiler abstraction
Dem.h	Provides the exported interfaces of Diagnostic Event Manager
Det.h	Provides the exported interfaces of Development Error Tracer

**Fr\_17\_Eray driver**
**Table 112 C file structure (continued)**

File name	Description
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 the Tricore™. This shall be included by other drivers to call OS APIs.
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.

### 2.1.3.2 Code generator plugin files

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

**Fr\_17\_Eray driver**

**Figure 9 Code generator plugin files**
**Table 113 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 driver**
**Table 113**      **Code generator plugin files (continued)**

File name	Description
Fr_17_Eray_Catalog.xml	AUTOSAR format catalog file
MANIFEST.MF	Tresos plugin support file containing the metadata 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

## 2.1.4 Integration hints

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

### 2.1.4.1 Integration with AUTOSAR stack

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

## Fr\_17\_Eray driver

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 and runtime 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.

- **DEM**

The DEM module is a part of the AUTOSAR stack that handles all the production errors reported by the BSW modules. The FR driver reports all the production errors to the DEM modules through the

## Fr\_17\_Eray driver

Dem\_ReportErrorStatus() API. The user of the FR driver must process all the production errors (fail / pass) reported to the DEM module through the Dem\_ReportErrorStatus() API.

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

**Note:** *Re-entrancy of the APIs Fr\_17\_Eray\_ControllerInit, Fr\_17\_Eray\_StartCommunication, Fr\_17\_Eray\_AllowColdstart, Fr\_17\_Eray\_AllSlots, Fr\_17\_Eray\_HaltCommunication, Fr\_17\_Eray\_AbortCommunication, Fr\_17\_Eray\_SendWUP, Fr\_17\_Eray\_SetWakeupChannel, Fr\_17\_Eray\_TransmitTxLPdu, Fr\_17\_Eray\_CancelTxLPdu, Fr\_17\_Eray\_ReceiveRxLPdu, Fr\_17\_Eray\_CheckTxLPduStatus, Fr\_17\_Eray\_PrepareLPdu, Fr\_17\_Eray\_ReconfigLPdu and Fr\_17\_Eray\_DisableLPdu is dependent on the re-entrancy of Dem\_ReportErrorStatus(). As per their design, the module APIs are reentrant for different device. However, in case API Dem\_ReportErrorStatus() is implemented as non reentrant, the APIs inherit the property of the same.*

- **SchM**

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

In the 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 be updated by the integrator. The user must implement the SchM functions

## Fr\_17\_Eray driver

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 (OS)**

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.

The 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.

### 2.1.4.2 Multicore and Resource Manager

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

### 2.1.4.3 MCU support

The FR 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:



## Fr\_17\_Eray driver

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

### 2.1.4.4 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 114** Connectivity of I/O signals for FR controller 0 Channel A - TC39x, TC38x, TC357 and TC37x devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0A0	P14.8	In	<code>FrRxInputSelectionA - value FR_RXSEL0</code>
RXD0A1	P11.9	In	<code>FrRxInputSelectionA - value FR_RXSEL1</code>
RXD0A2	P02.1	In	<code>FrRxInputSelectionA - value FR_RXSEL2</code>
RXD0A3	P14.1	In	<code>FrRxInputSelectionA - value FR_RXSEL3</code>
TXD0A	P02.0, P11.3, P14.10, P14.0	Out	Not applicable
TXEN0A	P02.4, P11.6, P14.9	Out	Not applicable

**Table 115** Connectivity of I/O signals for FR controller 0 Channel B - TC39x, TC38x, TC357 and TC37x devices

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD0B0	P14.7	In	<code>FrRxInputSelectionB - value FR_RXSEL0</code>
RXD0B1	P11.10	In	<code>FrRxInputSelectionB - value FR_RXSEL1</code>
RXD0B2	P02.3	In	<code>FrRxInputSelectionB - value FR_RXSEL2</code>
RXD0B3	P14.1	In	<code>FrRxInputSelectionB - value FR_RXSEL3</code>

**Fr\_17\_Eray driver**
**Table 115**      **Connectivity of I/O signals for FR controller 0 Channel B - TC39x, TC38x, TC357 and TC37x devices (continued)**

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
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 116**      **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 117**      **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 118**      **Connectivity of I/O signals for FR controller 1 Channel A - TC397, TC397 ADAS and TC387 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	--	--

**Fr\_17\_Eray driver**
**Table 118**      **Connectivity of I/O signals for FR controller 1 Channel A - TC397, TC397 ADAS and TC387 devices (continued)**

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
RXD1A3	No Connection	--	--
TXD1A	P14.10	Out	Not applicable
TXEN1A	P14.9	Out	Not applicable

**Table 119**      **Connectivity of I/O signals for FR controller 1 Channel B - TC397, TC397 ADAS and TC387 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 120**      **Connectivity of I/O signals for FR controller 0 Channel A - TC356 device**

Interface signals	Port lines	I/O direction	Configuration parameter provided in FR driver
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 121**      **Connectivity of I/O signals for FR controller 0 Channel B - TC356 device**

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

---

**Fr\_17\_Eray driver**
**Table 121**                      **Connectivity of I/O signals for FR controller 0 Channel B - TC356 device (continued)**

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

### 2.1.4.5                      DMA support

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

### 2.1.4.6                      Interrupt connections

The FR 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.*

## Fr\_17\_Eray driver

### 2.1.4.7 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.

## Fr\_17\_Eray driver

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.

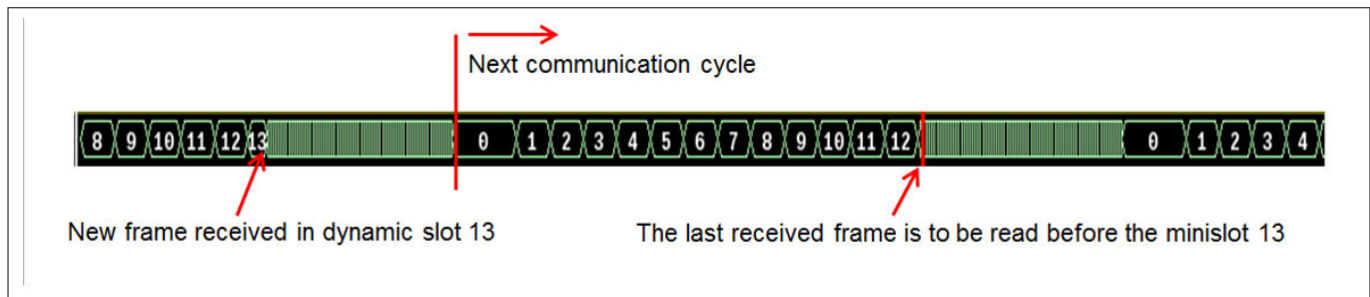
- **Absolute timer mode**

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.

**Fr\_17\_Eray driver**

**Figure 10**      **Frame received in dynamic segment**

- **Transmission conflict detection feature**

A configuration parameter `FrTxConflictDetection` is added to enable/ disable the detection of the transmission conflict. 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`.

## 2.1.5 Key architectural considerations

### 2.1.5.1 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.



## Fr\_17\_Eray driver

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 (`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.

In the `FrIf` configuration, the communication action `PREPARE_LPDU` needs to be 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.

### 2.1.5.2 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.

### 2.1.5.3 Clock configuration

In order to control the clock divider of the kernel clock `fCLC_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 `fCLC_ERAY = fSPB`, `fCLC_ERAY = fSPB / 2` or `fCLC_ERAY = fSPB / 4`. The parameter `FrClockDivider` only controls the kernel clock `fCLC_ERAY` and not the sampling clock `fSCLK`.



---

**Fr\_17\_Eray driver**

#### 2.1.5.4 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`.

#### 2.1.5.5 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.

#### 2.1.5.6 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 `fCLC_ERAY` and `fSCLK`. 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`.

#### 2.1.5.7 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.

#### 2.1.5.8 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.

#### 2.1.5.9 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.

## **2.2 Assumptions of Use (AoUs)**

There are no AoUs for the driver.

## Fr\_17\_Eray driver

## 2.3 Reference information

### 2.3.1 Configuration interfaces

This section details the configuration container hierarchy along with their configuration parameters.

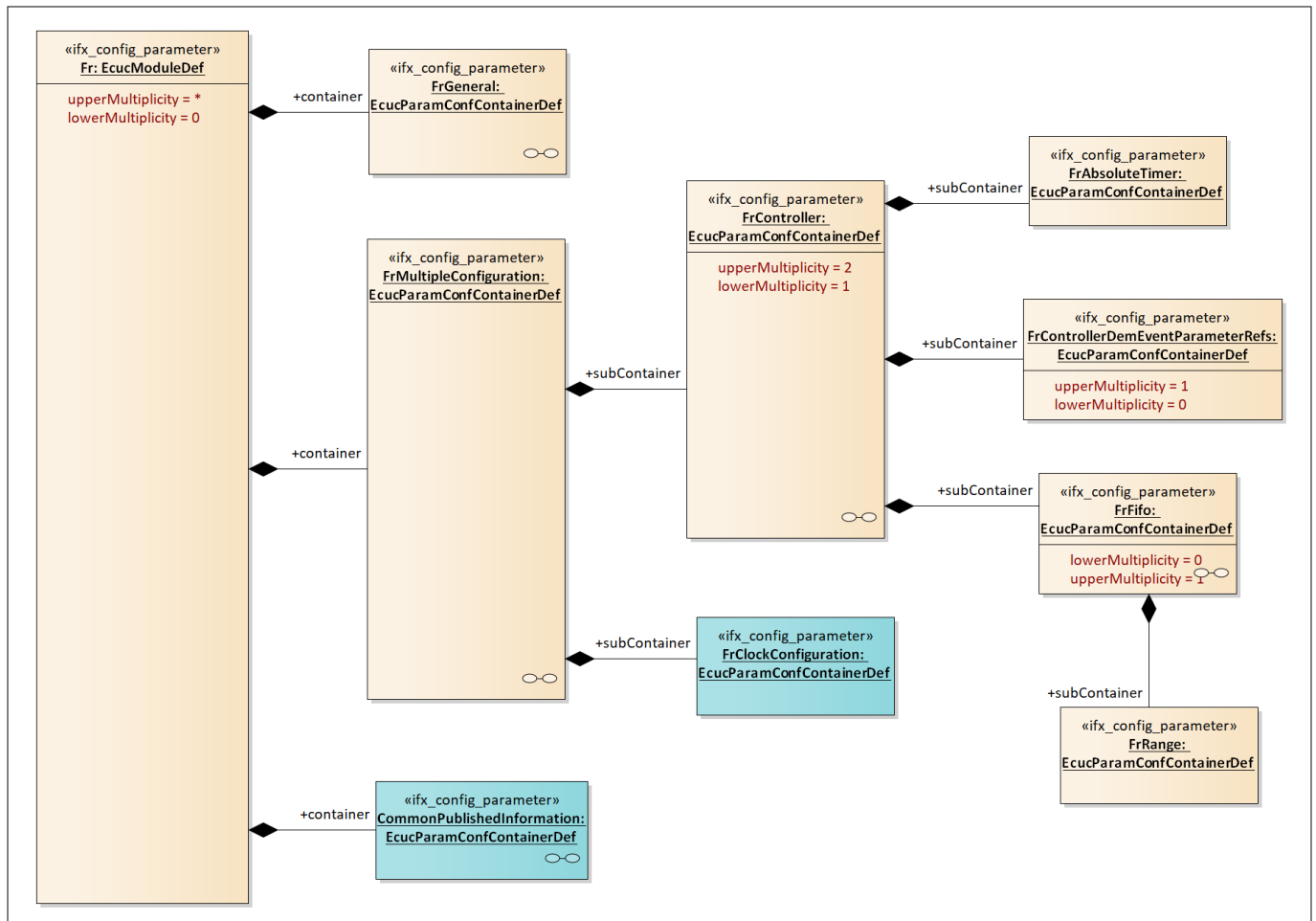


Figure 11 Container hierarchy along with their configuration parameters

#### 2.3.1.1 Container: Fr

Configuration of the FR (FlexRay driver) module.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

##### 2.3.1.1.1 Config Variant

Table 122 Specification for Config Variant

<b>Name</b>	Config Variant
<b>Description</b>	<p>Selects the config-variant for the FR module.</p> <p>This parameter is introduced to identify the config-variant supported by the module.</p> <p>Remark: The config-variant supported is PostBuild.</p>

**Fr\_17\_Eray driver**
**Table 122 Specification for Config Variant (continued)**

<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	VariantPostBuild: Post Build Support		
<b>Default value</b>	VariantPostBuild		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.2 Container: FrGeneral**

General configuration parameters of the FlexRay driver module.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**2.3.1.2.1 FrCtrlTestCount**
**Table 123 Specification for FrCtrlTestCount**

<b>Name</b>	FrCtrlTestCount		
<b>Description</b>	Maximum number of iterations the FlexRay controller hardware test is performed during controller initialization.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	1		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.2.2 FrDevErrorDetect**
**Table 124 Specification for FrDevErrorDetect**

<b>Name</b>	FrDevErrorDetect
-------------	------------------

**Fr\_17\_Eray driver**
**Table 124 Specification for FrDevErrorDetect (continued)**

<b>Description</b>	Switches the Default Error Tracer (DET) detection and notification ON or OFF. - true: enabled (ON). - false: disabled (OFF). Remark: The default value of this parameter is set to true so that the development error check is enabled by default to conform to the AUTOSAR requirement.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	TRUE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.2.3 FrDisableLPduSupport**
**Table 125 Specification for FrDisableLPduSupport**

<b>Name</b>	FrDisableLPduSupport		
<b>Description</b>	Enables or disables API function Fr_DisableLPdu. Remark: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.2.4 FrIndex

**Table 126 Specification for FrIndex**

<b>Name</b>	FrIndex		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	0		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.2.5 FrInitApiMode

**Table 127 Specification for FrInitApiMode**

<b>Name</b>	FrInitApiMode		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	FR_MCAL_SUPERVISOR: Operating mode used is Supervisory FR_MCAL_USER1: Operating mode used is USER1		
<b>Default value</b>	FR_MCAL_SUPERVISOR		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.2.6 FrNmVectorEnable

**Table 128 Specification for FrNmVectorEnable**

<b>Name</b>	FrNmVectorEnable		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.2.7 FrNumCtrlSupported

**Table 129 Specification for FrNumCtrlSupported**

<b>Name</b>	FrNumCtrlSupported		
<b>Description</b>	Determines the maximum number of communication controllers that the driver supports. Remark: Minimum number of controllers supported is selected as the default value.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	1 - 2		
<b>Default value</b>	1		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.2.8 FrPrepareLPduSupport

**Table 130** Specification for FrPrepareLPduSupport

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

### 2.3.1.2.9 FrReconfigLPduSupport

**Table 131** Specification for FrReconfigLPduSupport

<b>Name</b>	FrReconfigLPduSupport		
<b>Description</b>	Enables or disables API function Fr_ReconfigLPdu. Remark: The optional APIs are disabled by default to minimize the executable code size.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		



## Fr\_17\_Eray driver

## 2.3.1.2.10 FrRxStringentCheck

Table 132 Specification for FrRxStringentCheck

<b>Name</b>	FrRxStringentCheck		
<b>Description</b>	If stringent check is enabled (true), received frames are accepted only if no slot status error occurred.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## 2.3.1.2.11 FrRxStringentLengthCheck

Table 133 Specification for FrRxStringentLengthCheck

<b>Name</b>	FrRxStringentLengthCheck		
<b>Description</b>	If stringent length check is enabled (true), received frames are accepted only if the received payload length matches the configured payload length.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**Fr\_17\_Eray driver**
**2.3.1.2.12 FrTimeoutDurationFactor**
**Table 134 Specification for FrTimeoutDurationFactor**

<b>Name</b>	FrTimeoutDurationFactor		
<b>Description</b>	<p>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.</p> <p>This parameter is introduced to configure the maximum time until a timeout error (DEM parameter FR_E_CTRL_TESTRESULT) is reported.</p> <p>Remark: The default value of this parameter is set to 400 as an example value within the range.</p> <p>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.</p> <p>Taking these assumptions into consideration, it is suggested that the user configures the time duration greater than 8 clock cycles for this parameter.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	100 - 4294967295		
<b>Default value</b>	400		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.2.13 FrTxConflictDetection**
**Table 135 Specification for FrTxConflictDetection**

<b>Name</b>	FrTxConflictDetection		
<b>Description</b>	<p>If transmit conflict detection is enabled (true), the Fr_CheckTxLPduStatus API detects the occurrence of a transmission conflict.</p> <p>This parameter is introduced to enable/ disable the transmit conflict detection feature.</p> <p>Remark: The default value is false, in order to keep the transmit conflict detection feature disabled by default.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		

**Fr\_17\_Eray driver**
**Table 135 Specification for FrTxConflictDetection (continued)**

<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.2.14 FrVersionInfoApi**
**Table 136 Specification for FrVersionInfoApi**

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

**2.3.1.3 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: -

**2.3.1.4 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

## Fr\_17\_Eray driver

### 2.3.1.4.1 FrCtrlIdx

**Table 137 Specification for FrCtrlIdx**

<b>Name</b>	FrCtrlIdx		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 1		
<b>Default value</b>	0		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.4.2 FrPAllowHaltDueToClock

**Table 138 Specification for FrPAllowHaltDueToClock**

<b>Name</b>	FrPAllowHaltDueToClock		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**Fr\_17\_Eray driver**
**2.3.1.4.3 FrPAllowPassiveToActive**
**Table 139 Specification for FrPAllowPassiveToActive**

<b>Name</b>	FrPAllowPassiveToActive		
<b>Description</b>	<p>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.</p> <p>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.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 31		
<b>Default value</b>	10		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.4 FrPChannels**
**Table 140 Specification for FrPChannels**

<b>Name</b>	FrPChannels		
<b>Description</b>	<p>Channels to which the node is connected.</p> <p>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).</p> <p>The default value of this parameter (FR_CHANNEL_AB) is set to the reset value of the corresponding bit-field within the SFR.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	<p>FR_CHANNEL_A: Cluster uses channel A</p> <p>FR_CHANNEL_AB: Cluster uses channel A and B</p> <p>FR_CHANNEL_B: Cluster uses channel B</p>		
<b>Default value</b>	FR_CHANNEL_AB		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-

**Fr\_17\_Eray driver**
**Table 140 Specification for FrPChannels (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.5 FrPClusterDriftDamping**
**Table 141 Specification for FrPClusterDriftDamping**

<b>Name</b>	FrPClusterDriftDamping		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 20		
<b>Default value</b>	1		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.6 FrPDecodingCorrection**
**Table 142 Specification for FrPDecodingCorrection**

<b>Name</b>	FrPDecodingCorrection		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	14 - 143		
<b>Default value</b>	14		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-

**Fr\_17\_Eray driver**
**Table 142 Specification for FrPDecodingCorrection (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.7 FrPDelayCompensationA**
**Table 143 Specification for FrPDelayCompensationA**

<b>Name</b>	FrPDelayCompensationA		
<b>Description</b>	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 0.0125us to 0.05us (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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 200		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.8 FrPDelayCompensationB**
**Table 144 Specification for FrPDelayCompensationB**

<b>Name</b>	FrPDelayCompensationB		
<b>Description</b>	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 0.0125us to 0.05us (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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 200		

**Fr\_17\_Eray driver**
**Table 144 Specification for FrPDelayCompensationB (continued)**

<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.9 FrPExternalSync**
**Table 145 Specification for FrPExternalSync**

<b>Name</b>	FrPExternalSync		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.10 FrPFallBackInternal**
**Table 146 Specification for FrPFallBackInternal**

<b>Name</b>	FrPFallBackInternal		
<b>Description</b>	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.		



**Fr\_17\_Eray driver**
**Table 146 Specification for FrPFallBackInternal (continued)**

<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.11 FrPKeySlotId**
**Table 147 Specification for FrPKeySlotId**

<b>Name</b>	FrPKeySlotId		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 1023		
<b>Default value</b>	1		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.12 FrPKeySlotOnlyEnabled**
**Table 148 Specification for FrPKeySlotOnlyEnabled**

<b>Name</b>	FrPKeySlotOnlyEnabled		
<b>Description</b>	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.		

**Fr\_17\_Eray driver**
**Table 148 Specification for FrPKeySlotOnlyEnabled (continued)**

	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.13 FrPKeySlotUsedForStartup**
**Table 149 Specification for FrPKeySlotUsedForStartup**

<b>Name</b>	FrPKeySlotUsedForStartup		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	FrPKeySlotId		

## Fr\_17\_Eray driver

### 2.3.1.4.14 FrPKeySlotUsedForSync

**Table 150 Specification for FrPKeySlotUsedForSync**

<b>Name</b>	FrPKeySlotUsedForSync		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	FrPKeySlotUsedForStartup, FrPKeySlotId		

### 2.3.1.4.15 FrPLatestTx

**Table 151 Specification for FrPLatestTx**

<b>Name</b>	FrPLatestTx		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 7980		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.4.16 FrPMacroInitialOffsetA

**Table 152 Specification for FrPMacroInitialOffsetA**

<b>Name</b>	FrPMacroInitialOffsetA		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	2 - 68		
<b>Default value</b>	2		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.4.17 FrPMacroInitialOffsetB

**Table 153 Specification for FrPMacroInitialOffsetB**

<b>Name</b>	FrPMacroInitialOffsetB		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	2 - 68		
<b>Default value</b>	2		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.4.18 FrPMicroInitialOffsetA

**Table 154 Specification for FrPMicroInitialOffsetA**

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

### 2.3.1.4.19 FrPMicroInitialOffsetB

**Table 155 Specification for FrPMicroInitialOffsetB**

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

**Fr\_17\_Eray driver**
**Table 155 Specification for FrPmInitialOffsetB (continued)**

<b>Dependency</b>	-
-------------------	---

**2.3.1.4.20 FrPmPerCycle**
**Table 156 Specification for FrPmPerCycle**

<b>Name</b>	FrPmPerCycle		
<b>Description</b>	<p>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).</p> <p>Remark: Upper limit 640000 for FlexRay Protocol 2.1 Rev A compliance.</p> <p>The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	640 - 640000		
<b>Default value</b>	640		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.21 FrPNmVectorEarlyUpdate**
**Table 157 Specification for FrPNmVectorEarlyUpdate**

<b>Name</b>	FrPNmVectorEarlyUpdate		
<b>Description</b>	<p>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.</p> <p>Remark: Set to false for FlexRay Protocol 2.1 Rev. A compliance.</p> <p>The default value is also set to false for FlexRay Protocol 2.1 Rev. A compliance.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-

**Fr\_17\_Eray driver**
**Table 157 Specification for FrPNmVectorEarlyUpdate (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.22 FrPOffsetCorrectionOut**
**Table 158 Specification for FrPOffsetCorrectionOut**

<b>Name</b>	FrPOffsetCorrectionOut		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	13 - 15266		
<b>Default value</b>	13		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.23 FrPOffsetCorrectionStart**
**Table 159 Specification for FrPOffsetCorrectionStart**

<b>Name</b>	FrPOffsetCorrectionStart		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	9 - 15999		
<b>Default value</b>	9		

**Fr\_17\_Eray driver**
**Table 159 Specification for FrPOffsetCorrectionStart (continued)**

<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.24 FrPPayloadLengthDynMax**
**Table 160 Specification for FrPPayloadLengthDynMax**

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

**2.3.1.4.25 FrPRateCorrectionOut**
**Table 161 Specification for FrPRateCorrectionOut**

<b>Name</b>	FrPRateCorrectionOut		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	2 - 1923		
<b>Default value</b>	2		



---

**Fr\_17\_Eray driver**
**Table 161 Specification for FrPRateCorrectionOut (continued)**

<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.26 FrPSamplesPerMicrotick**
**Table 162 Specification for FrPSamplesPerMicrotick**

<b>Name</b>	FrPSamplesPerMicrotick		
<b>Description</b>	Number of samples per microtick. Remark: Set to N2SAMPLES for 10 Mbps baudrate. This parameter is disabled for configuration in GUI.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	N1SAMPLES: 1 sample N2SAMPLES: 2 samples N4SAMPLES: 4 samples		
<b>Default value</b>	N2SAMPLES		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.27 FrPSecondKeySlotId**
**Table 163 Specification for FrPSecondKeySlotId**

<b>Name</b>	FrPSecondKeySlotId		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef

**Fr\_17\_Eray driver**
**Table 163 Specification for FrPSecondKeySlotId (continued)**

<b>Range</b>	0 - 1023		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.28 FrPTwoKeySlotMode**
**Table 164 Specification for FrPTwoKeySlotMode**

<b>Name</b>	FrPTwoKeySlotMode		
<b>Description</b>	<p>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 FrPEternalSync is set to true then FrPTwoKeySlotMode must also be set to true.</p> <p>Remark: Set to false for FlexRay Protocol 2.1 Rev A compliance.</p> <p>The default value is also set to false for FlexRay Protocol 2.1 Rev A compliance.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.29 FrPWakeupChannel**
**Table 165 Specification for FrPWakeupChannel**

<b>Name</b>	FrPWakeupChannel
<b>Description</b>	<p>Channel used by the node to send a wakeup pattern.</p> <p>FrPWakeupChannel must be selected from among the channels configured by FrPChannels.</p>

**Fr\_17\_Eray driver**
**Table 165 Specification for FrPWakeupChannel (continued)**

	<p>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.</p> <p>The default value of this parameter (FR_CHANNEL_A) is set to the reset value of the corresponding bit-field within the SFR.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	FR_CHANNEL_A: channel A FR_CHANNEL_B: channel B		
<b>Default value</b>	FR_CHANNEL_A		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	FrPChannels		

**2.3.1.4.30 FrPWakeupPattern**
**Table 166 Specification for FrPWakeupPattern**

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

## Fr\_17\_Eray driver

### 2.3.1.4.31 FrPdAcceptedStartupRange

**Table 167 Specification for FrPdAcceptedStartupRange**

<b>Name</b>	FrPdAcceptedStartupRange		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 1875		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.4.32 FrPdListenTimeout

**Table 168 Specification for FrPdListenTimeout**

<b>Name</b>	FrPdListenTimeout		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	1284 - 1283846		
<b>Default value</b>	1284		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.4.33 FrPdMicrotick

**Table 169 Specification for FrPdMicrotick**

<b>Name</b>	FrPdMicrotick		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	T100NS: 100 ns T12_5NS: 12.5 ns T200NS: 200 ns T25NS: 25 ns T50NS: 50 ns		
<b>Default value</b>	T25NS		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.4.34 FrRxInputSelectionA

**Table 170 Specification for FrRxInputSelectionA**

<b>Name</b>	FrRxInputSelectionA		
<b>Description</b>	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 (FR_RXSEL0) is set to the reset value of the corresponding bit-field within the SFR.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	FR_RXSEL0: Channel A receiver input RXDA0 selected FR_RXSEL1: Channel A receiver input RXDA1 selected FR_RXSEL2: Channel A receiver input RXDA2 selected FR_RXSEL3: Channel A receiver input RXDA3 selected		
<b>Default value</b>	FR_RXSEL0		

**Fr\_17\_Eray driver**
**Table 170 Specification for FrRxInputSelectionA (continued)**

<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.4.35 FrRxInputSelectionB**
**Table 171 Specification for FrRxInputSelectionB**

<b>Name</b>	FrRxInputSelectionB		
<b>Description</b>	<p>Provides alternate Port Pin selection for FlexRay Receive input line for Channel B.</p> <p>This parameter is introduced to provide support for the selection of the alternate receiver input line for channel B.</p> <p>Remark: The default value of this parameter (FR_RXSEL0) is set to the reset value of the corresponding bit-field within the SFR.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	FR_RXSEL0: Channel B receiver input RXDB0 selected FR_RXSEL1: Channel B receiver input RXDB1 selected FR_RXSEL2: Channel B receiver input RXDB2 selected FR_RXSEL3: Channel B receiver input RXDB3 selected		
<b>Default value</b>	FR_RXSEL0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.5 Container: FrAbsoluteTimer**

Specifies the absolute timer configuration parameters of the Fr.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

## Fr\_17\_Eray driver

### 2.3.1.5.1 FrAbsTimerIdx

**Table 172 Specification for FrAbsTimerIdx**

<b>Name</b>	FrAbsTimerIdx		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 0		
<b>Default value</b>	0		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.6 Container: FrControllerDemEventParameterRefs

Container for the references to DemEventParameter elements which will be invoked using the Dem\_ReportErrorStatus API in case the corresponding 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

### 2.3.1.6.1 FR\_E\_CTRL\_TESTRESULT

**Table 173 Specification for FR\_E\_CTRL\_TESTRESULT**

<b>Name</b>	FR_E_CTRL_TESTRESULT		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1	<b>Type</b>	EcucSymbolicNameReferenceDef
<b>Range</b>	Reference to Node: DemEventParameter		
<b>Default value</b>	NULL		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	TRUE

**Fr\_17\_Eray driver**
**Table 173 Specification for FR\_E\_CTRL\_TESTRESULT (continued)**

<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	Post-Build
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.7 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

**2.3.1.7.1 FrAdmitWithoutMessageId**
**Table 174 Specification for FrAdmitWithoutMessageId**

<b>Name</b>	FrAdmitWithoutMessageId		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.7.2 FrBaseCycle**
**Table 175 Specification for FrBaseCycle**

<b>Name</b>	FrBaseCycle
-------------	-------------



**Fr\_17\_Eray driver**
**Table 175 Specification for FrBaseCycle (continued)**

<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 63		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.7.3 FrChannels**
**Table 176 Specification for FrChannels**

<b>Name</b>	FrChannels		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucEnumerationParamDef
<b>Range</b>	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		
<b>Default value</b>	FR_CHANNEL_A		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**Fr\_17\_Eray driver**
**2.3.1.7.4 FrCycleRepetition**
**Table 177 Specification for FrCycleRepetition**

<b>Name</b>	FrCycleRepetition		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	1 - 64		
<b>Default value</b>	1		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.7.5 FrFifoDepth**
**Table 178 Specification for FrFifoDepth**

<b>Name</b>	FrFifoDepth		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	1 - 127		
<b>Default value</b>	1		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.1.7.6 FrFrameIdRejectionFilter

**Table 179 Specification for FrFrameIdRejectionFilter**

<b>Name</b>	FrFrameIdRejectionFilter		
<b>Description</b>	<p>FIFO FrameId rejection criteria, frame ID to be rejected by the FIFO.</p> <p>This parameter is introduced to support the configuration of the rejection criteria of the hardware FIFO.</p> <p>Remark: The default value of this parameter is set to the reset value of the corresponding bit-field within the SFR.</p>		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 2047		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.7.7 FrFrameIdRejectionFilterMask

**Table 180 Specification for FrFrameIdRejectionFilterMask**

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

## Fr\_17\_Eray driver

### 2.3.1.7.8 FrMsgIdMask

**Table 181 Specification for FrMsgIdMask**

<b>Name</b>	FrMsgIdMask		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 65535		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.7.9 FrMsgIdMatch

**Table 182 Specification for FrMsgIdMatch**

<b>Name</b>	FrMsgIdMatch		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 65535		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.7.10 FrRejectNullFrames

**Table 183 Specification for FrRejectNullFrames**

<b>Name</b>	FrRejectNullFrames
-------------	--------------------

**Fr\_17\_Eray driver**
**Table 183 Specification for FrRejectNullFrames (continued)**

<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	TRUE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.7.11 FrRejectStaticSegment**
**Table 184 Specification for FrRejectStaticSegment**

<b>Name</b>	FrRejectStaticSegment		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucBooleanParamDef
<b>Range</b>	TRUE FALSE		
<b>Default value</b>	FALSE		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**Fr\_17\_Eray driver**
**2.3.1.8 Container: FrRange**

FIFO Frame Id range acceptance criteria. This container is ignored for configuration.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**2.3.1.8.1 FrRangeMax**
**Table 185 Specification for FrRangeMax**

<b>Name</b>	FrRangeMax		
<b>Description</b>	Last Frameld 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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 2047		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.8.2 FrRangeMin**
**Table 186 Specification for FrRangeMin**

<b>Name</b>	FrRangeMin		
<b>Description</b>	First Frameld 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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 2047		
<b>Default value</b>	0		
<b>Post-build variant value</b>	TRUE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Post-Build	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	AUTOSAR_ECUC	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**Fr\_17\_Eray driver**
**2.3.1.9 Container: FrClockConfiguration**

Clock Configuration of the individual controller.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**2.3.1.9.1 FrClockDivider**
**Table 187 Specification for FrClockDivider**

<b>Name</b>	FrClockDivider		
<b>Description</b>	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.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	1 - 4		
<b>Default value</b>	1		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Pre-Compile	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10 Container: CommonPublishedInformation**

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

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

**2.3.1.10.1 ArMajorVersion**
**Table 188 Specification for ArMajorVersion**

<b>Name</b>	ArMajorVersion		
<b>Description</b>	Major version number of AUTOSAR specification on which the appropriate implementation is based on.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	4		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-

---

**Fr\_17\_Eray driver**
**Table 188 Specification for ArMajorVersion (continued)**

<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10.2 ArMinorVersion**
**Table 189 Specification for ArMinorVersion**

<b>Name</b>	ArMinorVersion		
<b>Description</b>	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	2		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10.3 ArPatchVersion**
**Table 190 Specification for ArPatchVersion**

<b>Name</b>	ArPatchVersion		
<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	2		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL



## Fr\_17\_Eray driver

**Table 190**      **Specification for ArPatchVersion (continued)**

<b>Dependency</b>	-
-------------------	---

### 2.3.1.10.4      **ModuleId**

**Table 191**      **Specification for ModuleId**

<b>Name</b>	ModuleId		
<b>Description</b>	Module ID of the FR module from Module List.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 65535		
<b>Default value</b>	81		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.10.5      **Release**

**Table 192**      **Specification for Release**

<b>Name</b>	Release		
<b>Description</b>	This parameter indicates the TC3xx device derivative used for the implementation.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucStringParamDef
<b>Range</b>	String		
<b>Default value</b>	As per hardware derivative		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

### 2.3.1.10.6      **SwMajorVersion**

**Table 193**      **Specification for SwMajorVersion**

<b>Name</b>	SwMajorVersion
-------------	----------------

## Fr\_17\_Eray driver

**Table 193**      **Specification for SwMajorVersion (continued)**

<b>Description</b>	Major version number of the vendor specific implementation of the module.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	As per driver		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10.7      SwMinorVersion****Table 194**      **Specification for SwMinorVersion**

<b>Name</b>	SwMinorVersion		
<b>Description</b>	Minor version number of the vendor specific implementation of the module.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	As per driver		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10.8      SwPatchVersion****Table 195**      **Specification for SwPatchVersion**

<b>Name</b>	SwPatchVersion		
<b>Description</b>	Patch level version number of the vendor specific implementation of the module.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 255		
<b>Default value</b>	As per driver		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-

**Fr\_17\_Eray driver**
**Table 195 Specification for SwPatchVersion (continued)**

<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10.9 VendorApiInfix**
**Table 196 Specification for VendorApiInfix**

<b>Name</b>	VendorApiInfix		
<b>Description</b>	This parameter is used to specify the vendor specific name.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucStringParamDef
<b>Range</b>	String		
<b>Default value</b>	Eray		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

**2.3.1.10.10 VendorId**
**Table 197 Specification for VendorId**

<b>Name</b>	VendorId		
<b>Description</b>	Vendor ID of the dedicated implementation of the FR module according to the AUTOSAR vendor list.		
<b>Multiplicity</b>	1..1	<b>Type</b>	EcucIntegerParamDef
<b>Range</b>	0 - 65535		
<b>Default value</b>	17		
<b>Post-build variant value</b>	FALSE	<b>Post-build variant multiplicity</b>	-
<b>Value configuration class</b>	Published-Information	<b>Multiplicity configuration class</b>	-
<b>Origin</b>	IFX	<b>Scope</b>	LOCAL
<b>Dependency</b>	-		

## Fr\_17\_Eray driver

### 2.3.2 Functions - Type definitions

#### 2.3.2.1 Fr\_17\_Eray\_ConfigType

**Table 198** Specification for Fr\_17\_Eray\_ConfigType

<b>Syntax</b>	Fr_17_Eray_ConfigType	
<b>Type</b>	Structure	
<b>File</b>	Fr_17_Eray.h	
<b>Range</b>	--	The elements of the data structure are specific to the microcontroller.
<b>Description</b>	This type contains the implementation-specific post build configuration structure of the FlexRay driver. Detailed description is available in the design document.	
<b>Source</b>	AUTOSAR	

#### 2.3.2.2 Fr\_POCStateType

**Table 199** Specification for Fr\_POCStateType

<b>Syntax</b>	Fr_POCStateType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	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
<b>Description</b>	Represents the FlexRay controller POC states.	
<b>Source</b>	AUTOSAR	

#### 2.3.2.3 Fr\_SlotModeType

**Table 200** Specification for Fr\_SlotModeType

<b>Syntax</b>	Fr_SlotModeType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	0 - FR_SLOTMODE_KEYSLLOT	Single slot mode

**Fr\_17\_Eray driver**
**Table 200 Specification for Fr\_SlotModeType (continued)**

	1 - FR_SLOTMODE_ALL_PENDING	All pending slot mode
	2 - FR_SLOTMODE_ALL	All slot mode
<b>Description</b>	Represents the FlexRay controller slotmodes.	
<b>Source</b>	AUTOSAR	

**2.3.2.4 Fr\_ErrorModeType**
**Table 201 Specification for Fr\_ErrorModeType**

<b>Syntax</b>	Fr_ErrorModeType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	0 - FR_ERRORMODE_ACTIVE	Active error mode
	1 - FR_ERRORMODE_PASSIVE	Passive error mode
	2 - FR_ERRORMODE_COMM_HALT	Communication halted error mode
<b>Description</b>	Represents the FlexRay controller error modes.	
<b>Source</b>	AUTOSAR	

**2.3.2.5 Fr\_WakeupStatusType**
**Table 202 Specification for Fr\_WakeupStatusType**

<b>Syntax</b>	Fr_WakeupStatusType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	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
	6 - FR_WAKEUP_TRANSMITTED	Wakeup state - transmitted
<b>Description</b>	Represents the FlexRay controller wakeup status types.	
<b>Source</b>	AUTOSAR	

**Fr\_17\_Eray driver**
**2.3.2.6 Fr\_StartupStateType**
**Table 203 Specification for Fr\_StartupStateType**

<b>Syntax</b>	Fr_StartupStateType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	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
<b>Description</b>	Represents the FlexRay controller startup status types.	
<b>Source</b>	AUTOSAR	

**2.3.2.7 Fr\_POCTestStatusType**
**Table 204 Specification for Fr\_POCTestStatusType**

<b>Syntax</b>	Fr_POCTestStatusType	
<b>Type</b>	Structure	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	Fr_POCTestStatusType 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\_17\_Eray driver**
**Table 204 Specification for Fr\_POCStatusType (continued)**

	Fr_SlotModeType SlotMode	Slot mode
	Fr_ErrorModeType ErrorMode	Error mode
	Fr_WakeupStatusType WakeupStatus	Wakeup state
	Fr_StartupStateType StartupState	Startup state
<b>Description</b>	Represents the FlexRay controller POC-Status information.	
<b>Source</b>	AUTOSAR	

**2.3.2.8 Fr\_TxLPduStatusType**
**Table 205 Specification for Fr\_TxLPduStatusType**

<b>Syntax</b>	Fr_TxLPduStatusType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	0 - FR_TRANSMITTED	LSdu was transmitted
	1 - FR_TRANSMITTED_CONFLICT	A transmission conflict has occurred
	2 - FR_NOT_TRANSMITTED	LSdu was not transmitted
<b>Description</b>	Represents the LSdu TX status.	
<b>Source</b>	AUTOSAR	

**2.3.2.9 Fr\_RxLPduStatusType**
**Table 206 Specification for Fr\_RxLPduStatusType**

<b>Syntax</b>	Fr_RxLPduStatusType	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	0 - FR_RECEIVED	LSdu was received
	1 - FR_NOT_RECEIVED	LSdu was not received
	2 - FR_RECEIVED_MORE_DATA_AVAILABLE	FIFO is not empty
<b>Description</b>	Represents the LSdu TX status.	
<b>Source</b>	AUTOSAR	

**2.3.2.10 Fr\_ChannelType**
**Table 207 Specification for Fr\_ChannelType**

<b>Syntax</b>	Fr_ChannelType
<b>Type</b>	Enumeration

**Fr\_17\_Eray driver**
**Table 207 Specification for Fr\_ChannelType (continued)**

<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	0 - FR_CHANNEL_A	FlexRay channel A
	1 - FR_CHANNEL_B	FlexRay channel B
	2 - FR_CHANNEL_AB	FlexRay channel A and B
<b>Description</b>	Represents the FlexRay channels.	
<b>Source</b>	AUTOSAR	

**2.3.2.11 FR\_CIDX\_<CONFIGPARAM\_NAME>**
**Table 208 Specification for FR\_CIDX\_<CONFIGPARAM\_NAME>**

<b>Syntax</b>	FR_CIDX_<CONFIGPARAM_NAME>	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	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
	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_GDSTATICSLLOT	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



**Fr\_17\_Eray driver**
**Table 208 Specification for FR\_CIDX\_<CONFIGPARAM\_NAME> (continued)**

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
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

**Fr\_17\_Eray driver**
**Table 208 Specification for FR\_CIDX\_<CONFIGPARAM\_NAME> (continued)**

32 - FR_CIDX_GDSYMBOLWINDOW	Maps to configuration parameter - FrIfGdSymbolWindow
33 - FR_CIDX_GDSYMBOLWINDOWACTIONPOINT OFFSET	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
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

**Fr\_17\_Eray driver**
**Table 208 Specification for FR\_CIDX\_<CONFIGPARAM\_NAME> (continued)**

	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
<b>Description</b>	List of Macros (indices) that can be passed into API function Fr_ReadCCConfig as parameter Fr_ConfigParamIdx.	
<b>Source</b>	AUTOSAR	

**2.3.2.12 FR\_SLOTMODE\_SINGLE**
**Table 209 Specification for FR\_SLOTMODE\_SINGLE**

<b>Syntax</b>	FR_SLOTMODE_SINGLE	
<b>Type</b>	Enumeration	
<b>File</b>	Fr_GeneralTypes.h	
<b>Range</b>	0 - FR_SLOTMODE_KEYSLLOT	Mapping via preprocessor macro
<b>Description</b>	The preprocessor macro FR_SLOTMODE_SINGLE maps to value of the FR_SLOTMODE_KEYSLLOT.	
<b>Source</b>	AUTOSAR	

**2.3.3 Functions - APIs**

This section lists all the APIs of the FR driver.

## Fr\_17\_Eray driver

### 2.3.3.1 Fr\_17\_Eray\_Init

**Table 210 Specification for Fr\_17\_Eray\_Init API**

<b>Syntax</b>	<pre>void Fr_17_Eray_Init (     const Fr_17_Eray_ConfigType * const Fr_ConfigPtr )</pre>	
<b>Service ID</b>	0x1C	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	Fr_ConfigPtr	Pointer to FR module configuration structure
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	Initializes the Fr. This function internally stores the configuration address to enable subsequent API calls to access the configuration.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.</p> <p>Runtime Errors: None</p> <p>DEM:</p> <p>FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

### 2.3.3.2 Fr\_17\_Eray\_ControllerInit

**Table 211 Specification for Fr\_17\_Eray\_ControllerInit API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_ControllerInit (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x00	
<b>Sync/Async</b>	Synchronous	

**Fr\_17\_Eray driver**
**Table 211 Specification for Fr\_17\_Eray\_ControllerInit API (continued)**

<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Initializes a FlexRay Communication Controller.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.3 Fr\_17\_Eray\_StartCommunication**
**Table 212 Specification for Fr\_17\_Eray\_StartCommunication API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_StartCommunication (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Asynchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.

**Fr\_17\_Eray driver**
**Table 212 Specification for Fr\_17\_Eray\_StartCommunication API (continued)**

<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Starts communication. Initiates the startup procedure within the FlexRay CC.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCTSTATE: FR CC is not in the expected POC state. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.4 Fr\_17\_Eray-AllowColdstart**
**Table 213 Specification for Fr\_17\_Eray-AllowColdstart API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray-AllowColdstart (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x23	
<b>Sync/Async</b>	Asynchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-

**Fr\_17\_Eray driver**
**Table 213 Specification for Fr\_17\_Eray\_AllowColdstart API (continued)**

<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Invokes the CC CHI command ALLOW_COLDSTART.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	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.	

**2.3.3.5 Fr\_17\_Eray\_AllSlots**
**Table 214 Specification for Fr\_17\_Eray\_AllSlots API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_AllSlots (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x24	
<b>Sync/Async</b>	Asynchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	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.	

**Fr\_17\_Eray driver**
**Table 214 Specification for Fr\_17\_Eray\_AllSlots API (continued)**

<b>Source</b>	AUTOSAR
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.6 Fr\_17\_Eray\_HaltCommunication**
**Table 215 Specification for Fr\_17\_Eray\_HaltCommunication API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_HaltCommunication (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Asynchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	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.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.	



**Fr\_17\_Eray driver**
**Table 215 Specification for Fr\_17\_Eray\_HaltCommunication API (continued)**

	FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.7 Fr\_17\_Eray\_AbortCommunication**
**Table 216 Specification for Fr\_17\_Eray\_AbortCommunication API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_AbortCommunication (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Invokes the CC CHI command FREEZE, which immediately aborts communication (if active) and changes to the POC:halt state from any previous POCState.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM:	

**Fr\_17\_Eray driver**
**Table 216 Specification for Fr\_17\_Eray\_AbortCommunication API (continued)**

	FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.8 Fr\_17\_Eray\_SendWUP**
**Table 217 Specification for Fr\_17\_Eray\_SendWUP API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_SendWUP (     const uint8 Fr_CtrlIdx )</pre>	
<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Asynchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Invokes the CC CHI command WAKEUP, which initiates the wakeup transmission procedure on the configured FlexRay channel.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	

**Fr\_17\_Eray driver**
**Table 217 Specification for Fr\_17\_Eray\_SendWUP API (continued)**

<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.9 Fr\_17\_Eray\_SetWakeupChannel**
**Table 218 Specification for Fr\_17\_Eray\_SetWakeupChannel API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_SetWakeupChannel (     const uint8 Fr_CtrlIdx,     const Fr_ChannelType Fr_ChnlIdx )</pre>	
<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Sets a wakeup channel.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state. FR_17_ERAY_E_INV_CHNL_IDX: Invalid channel index. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	

**Fr\_17\_Eray driver**
**Table 218 Specification for Fr\_17\_Eray\_SetWakeupChannel API (continued)**

<b>User hints</b>	None
-------------------	------

**2.3.3.10 Fr\_17\_Eray\_GetPOCStatus**
**Table 219 Specification for Fr\_17\_Eray\_GetPOCStatus API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetPOCStatus (     const uint8 Fr_CtrlIdx,     Fr_POCTestType * const Fr_POCTestPtr )</pre>	
<b>Service ID</b>	0x0a	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	Fr_POCTestPtr	Address the output value is stored to.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Gets the POC status.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CHNL_IDX: Invalid channel index. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_POCTEST: FR CC is not in the expected POC state. FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**Fr\_17\_Eray driver**
**2.3.3.11 Fr\_17\_Eray\_TransmitTxLPdu**
**Table 220 Specification for Fr\_17\_Eray\_TransmitTxLPdu API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_TransmitTxLPdu (     const uint8 Fr_CtrlIdx,     const uint16 Fr_LPduIdx,     const uint8 * const Fr_LSduPtr,     const uint8 Fr_LSduLength )</pre>	
<b>Service ID</b>	0x0b	
<b>Sync/Async</b>	Asynchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Transmits data on the FlexRay network.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index. FR_17_ERAY_E_INV_LENGTH: Payload length parameter has an invalid value.  Runtime Errors: None  DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure  Safety Errors: None  <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	-	

**Fr\_17\_Eray driver**
**2.3.3.12 Fr\_17\_Eray\_CancelTxLPdu**
**Table 221 Specification for Fr\_17\_Eray\_CancelTxLPdu API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_CancelTxLPdu (     const uint8 Fr_CtrlIdx,     const uint16 Fr_LPduIdx )</pre>	
<b>Service ID</b>	0x2d	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Cancels the already pending transmission of an LPdu contained in the physical transmit resource (e.g. message buffer) of the controller.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.13 Fr\_17\_Eray\_ReceiveRxLPdu**
**Table 222 Specification for Fr\_17\_Eray\_ReceiveRxLPdu API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_ReceiveRxLPdu (</pre>
---------------	--

**Fr\_17\_Eray driver**
**Table 222 Specification for Fr\_17\_Eray\_ReceiveRxLPdu API (continued)**

	<pre> const uint8 Fr_CtrlIdx, const uint16 Fr_LPduIdx, uint8 * const Fr_LSduPtr, Fr_RxLPduStatusType * const Fr_RxLPduStatusPtr, uint8 * const Fr_LSduLengthPtr ) </pre>	
<b>Service ID</b>	0x0c	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	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.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Receives data from the FlexRay network.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index. FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure FRIF_E_LPDU_SLOTSTATUS: Flexray Protocol communication error - Slot Error (configured in FrIf Module) Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**Fr\_17\_Eray driver**
**2.3.3.14 Fr\_17\_Eray\_CheckTxLPduStatus**
**Table 223 Specification for Fr\_17\_Eray\_CheckTxLPduStatus API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_CheckTxLPduStatus (     const uint8 Fr_CtrlIdx,     const uint16 Fr_LPduIdx,     Fr_TxLPduStatusType * const Fr_TxLPduStatusPtr )</pre>	
<b>Service ID</b>	0x0d	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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
<b>Parameters (out)</b>	Fr_TxLPduStatusPtr	This reference is used to store the transmit status of the LPdu
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	<p>Checks the transmit status of the LSdu.</p> <p>Note: When the FrTxConflictDetection parameter is configured to true, this API provides the status FR_TRANSMITTED_CONFLICT when the transmission conflict has occurred.</p>	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	<p>DET:</p> <p>FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list.</p> <p>FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.</p> <p>FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index.</p> <p>FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index.</p> <p>Runtime Errors: None</p> <p>DEM:</p> <p>FRIF_E_LPDU_SLOTSTATUS: Flexray Protocol communication error - Slot Error (configured in FrIf Module)</p> <p>Safety Errors: None</p> <p><i>Note: All DET IDs are also reported as safety errors.</i></p>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	



**Fr\_17\_Eray driver**
**2.3.3.15 Fr\_17\_Eray\_PrepareLPdu**
**Table 224 Specification for Fr\_17\_Eray\_PrepareLPdu API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_PrepareLPdu (     const uint8 Fr_CtrlIdx,     const uint16 Fr_LPduIdx )</pre>	
<b>Service ID</b>	0x1f	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	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.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None Note: All DET IDs are also reported as safety errors.	
<b>Configuration dependencies</b>	FrPrepareLPduSupport	
<b>User hints</b>	None	

**Fr\_17\_Eray driver**
**2.3.3.16 Fr\_17\_Eray\_ReconfigLPdu**
**Table 225 Specification for Fr\_17\_Eray\_ReconfigLPdu API**

<b>Syntax</b>	<pre>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 )</pre>	
<b>Service ID</b>	0x25	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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 FrLf_LPdu must be configured to. FlexRay Channel the FrLf_LPdu must be configured to. Cycle Repetition part of the cycle filter mechanism FrLf_LPdu must be configured to. Cycle Offset part of the cycle filter mechanism FrLf_LPdu must be configured to. Payloadlength in units of bytes the FrLf_LPduIdx must be configured to. Header CRC the FrLf_LPdu must be configured to.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Reconfigures a given LPdu according to the parameters (FrameId, Channel, CycleRepetition, CycleOffset, PayloadLength, HeaderCRC) at runtime.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index. FR_17_ERAY_E_INV_CHNL_IDX: Invalid channel index.	

**Fr\_17\_Eray driver**
**Table 225 Specification for Fr\_17\_Eray\_ReconfigLPdu API (continued)**

	FR_17_ERAY_E_INV_HEADERCRC: Invalid FlexRay header CRC. FR_17_ERAY_E_INV_CYCLE: Parameter cycle exceeds 63. FR_17_ERAY_E_INV_LENGTH: Payload length parameter has an invalid value. Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	FrReconfigLPduSupport
<b>User hints</b>	None

**2.3.3.17 Fr\_17\_Eray\_DisableLPdu**
**Table 226 Specification for Fr\_17\_Eray\_DisableLPdu API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_DisableLPdu (     const uint8 Fr_CtrlIdx,     const uint16 Fr_LPduIdx )</pre>	
<b>Service ID</b>	0x26	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Disables the hardware resource of an LPdu for transmission/ reception.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_LPDU_IDX: Invalid LPdu index.	

**Fr\_17\_Eray driver**
**Table 226 Specification for Fr\_17\_Eray\_DisableLPdu API (continued)**

	Runtime Errors: None DEM: FR_E_CTRL_TESTRESULT: FlexRay controller hardware test failure Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	FrDisableLPduSupport
<b>User hints</b>	None

**2.3.3.18 Fr\_17\_Eray\_GetGlobalTime**
**Table 227 Specification for Fr\_17\_Eray\_GetGlobalTime API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetGlobalTime (     const uint8 Fr_CtrlIdx,     uint8 * const Fr_CyclePtr,     uint16 * const Fr_MacroTickPtr )</pre>	
<b>Service ID</b>	0x10	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	Fr_CyclePtr Fr_MacroTickPtr	Address where the current FlexRay communication cycle value should be stored. Address where the current macrotick value should be stored.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Gets the current global FlexRay time.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM: None	

**Fr\_17\_Eray driver**
**Table 227 Specification for Fr\_17\_Eray\_GetGlobalTime API (continued)**

	Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.19 Fr\_17\_Eray\_GetNmVector**
**Table 228 Specification for Fr\_17\_Eray\_GetNmVector API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetNmVector (     const uint8 Fr_CtrlIdx,     uint8 * const Fr_NmVectorPtr )</pre>	
<b>Service ID</b>	0x22	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	Fr_NmVectorPtr	Address where the NmVector of the last communication cycle should be stored.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Gets the network management vector of the last communication cycle.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	FrNmVectorEnable	
<b>User hints</b>	None	

**Fr\_17\_Eray driver**
**2.3.3.20 Fr\_17\_Eray\_GetNumOfStartupFrames**
**Table 229 Specification for Fr\_17\_Eray\_GetNumOfStartupFrames API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetNumOfStartupFrames (     const uint8 Fr_CtrlIdx,     uint8 * const Fr_NumOfStartupFramesPtr )</pre>	
<b>Service ID</b>	0x27	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	Fr_NumOfStartupFramesPtr	Address where the number of startup frames seen within the last even/odd cycle pair should be stored.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	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.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.21 Fr\_17\_Eray\_GetChannelStatus**
**Table 230 Specification for Fr\_17\_Eray\_GetChannelStatus API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetChannelStatus (</pre>
---------------	---

**Fr\_17\_Eray driver**
**Table 230 Specification for Fr\_17\_Eray\_GetChannelStatus API (continued)**

	<pre> const uint8 Fr_CtrlIdx, uint16 * const Fr_ChannelAStatusPtr, uint16 * const Fr_ChannelBStatusPtr ) </pre>	
<b>Service ID</b>	0x28	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	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.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	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.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index.  Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.22 Fr\_17\_Eray\_GetClockCorrection**
**Table 231 Specification for Fr\_17\_Eray\_GetClockCorrection API**

<b>Syntax</b>	<pre> Std_ReturnType Fr_17_Eray_GetClockCorrection (     const uint8 Fr_CtrlIdx,     sint16 * const Fr_RateCorrectionPtr, </pre>
---------------	--

**Fr\_17\_Eray driver**
**Table 231 Specification for Fr\_17\_Eray\_GetClockCorrection API (continued)**

	<pre>sint32 * const Fr_OffsetCorrectionPtr )</pre>	
<b>Service ID</b>	0x29	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	Fr_RateCorrectionPtr Fr_OffsetCorrectionPtr	Address where the current rate correction value must be stored. Address where the current offset correction value must be stored.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Gets the current clock correction values (vRateCorrection and vOffsetCorrection).	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.23 Fr\_17\_Eray\_GetSyncFrameList**
**Table 232 Specification for Fr\_17\_Eray\_GetSyncFrameList API**

<b>Syntax</b>	<pre>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 )</pre>
---------------	---



**Fr\_17\_Eray driver**
**Table 232 Specification for Fr\_17\_Eray\_GetSyncFrameList API (continued)**

<b>Service ID</b>	0x2a	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx Fr_ListSize	<p>Index of FlexRay CC within the context of the FlexRay driver.</p> <p>Size of the arrays passed via parameters:  Fr_ChannelAEvenListPtr  Fr_ChannelBEvenListPtr  Fr_ChannelAOddListPtr  Fr_ChannelBOddListPtr.</p> <p>The service must ensure to not write more entries into those arrays than granted by this parameter.</p>
<b>Parameters (out)</b>	Fr_ChannelAEvenListPtr Fr_ChannelBEvenListPtr Fr_ChannelAOddListPtr Fr_ChannelBOddListPtr	<p>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.</p> <p>Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen.</p> <p>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.</p> <p>Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen.</p> <p>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.</p> <p>Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen.</p> <p>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.</p> <p>Unused list elements are filled with the value 0 to indicate that no more syncframe has been seen.</p>
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Gets a list of syncframes received or transmitted on channel A and channel B via the even and odd communication cycle.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.	

**Fr\_17\_Eray driver**
**Table 232 Specification for Fr\_17\_Eray\_GetSyncFrameList API (continued)**

	FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_FRAMELIST_SIZE: Invalid framelist size value. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.24 Fr\_17\_Eray\_GetWakeupRxStatus**
**Table 233 Specification for Fr\_17\_Eray\_GetWakeupRxStatus API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetWakeupRxStatus (     const uint8 Fr_CtrlIdx,     uint8 * const Fr_WakeupRxStatusPtr )</pre>	
<b>Service ID</b>	0x2b	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	Fr_CtrlIdx	Index of FlexRay CC within the context of the FlexRay driver.
<b>Parameters (out)</b>	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
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	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.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.	

**Fr\_17\_Eray driver**
**Table 233 Specification for Fr\_17\_Eray\_GetWakeupRxStatus API (continued)**

	FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.25 Fr\_17\_Eray\_SetAbsoluteTimer**
**Table 234 Specification for Fr\_17\_Eray\_SetAbsoluteTimer API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_SetAbsoluteTimer (     const uint8 Fr_CtrlIdx,     const uint8 Fr_AbsTimerIdx,     const uint8 Fr_Cycle,     const uint16 Fr_Offset )</pre>	
<b>Service ID</b>	0x11	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Sets the absolute FlexRay timer according to the parameters Fr_Cycle and Fr_Offset.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized.	

**Fr\_17\_Eray driver**
**Table 234 Specification for Fr\_17\_Eray\_SetAbsoluteTimer API (continued)**

	FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_INV_CYCLE: Parameter cycle exceeds 63. FR_17_ERAY_E_INV_OFFSET: Parameter offset exceeds bounds. FR_17_ERAY_E_INV_POCSTATE: FR CC is not in the expected POC state.  Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.26 Fr\_17\_Eray\_CancelAbsoluteTimer**
**Table 235 Specification for Fr\_17\_Eray\_CancelAbsoluteTimer API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_CancelAbsoluteTimer (     const uint8 Fr_CtrlIdx,     const uint8 Fr_AbsTimerIdx )</pre>	
<b>Service ID</b>	0x13	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Stops an absolute timer.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None	

**Fr\_17\_Eray driver**
**Table 235 Specification for Fr\_17\_Eray\_CancelAbsoluteTimer API (continued)**

	DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>
<b>Configuration dependencies</b>	-
<b>User hints</b>	None

**2.3.3.27 Fr\_17\_Eray\_EnableAbsoluteTimerIRQ**
**Table 236 Specification for Fr\_17\_Eray\_EnableAbsoluteTimerIRQ API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_EnableAbsoluteTimerIRQ (     const uint8 Fr_CtrlIdx,     const uint8 Fr_AbsTimerIdx )</pre>	
<b>Service ID</b>	0x15	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Enables the interrupt line of an absolute timer.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	

**Fr\_17\_Eray driver**
**Table 236 Specification for Fr\_17\_Eray\_EnableAbsoluteTimerIRQ API (continued)**

<b>User hints</b>	None
-------------------	------

**2.3.3.28 Fr\_17\_Eray\_AckAbsoluteTimerIRQ**
**Table 237 Specification for Fr\_17\_Eray\_AckAbsoluteTimerIRQ API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_AckAbsoluteTimerIRQ (     const uint8 Fr_CtrlIdx,     const uint8 Fr_AbsTimerIdx )</pre>	
<b>Service ID</b>	0x17	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Resets the interrupt condition of an absolute timer.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**Fr\_17\_Eray driver**
**2.3.3.29 Fr\_17\_Eray\_DisableAbsoluteTimerIRQ**
**Table 238 Specification for Fr\_17\_Eray\_DisableAbsoluteTimerIRQ API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_DisableAbsoluteTimerIRQ (     const uint8 Fr_CtrlIdx,     const uint8 Fr_AbsTimerIdx )</pre>	
<b>Service ID</b>	0x19	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	-	-
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Disables the interrupt line of an absolute timer.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

**2.3.3.30 Fr\_17\_Eray\_GetAbsoluteTimerIRQStatus**
**Table 239 Specification for Fr\_17\_Eray\_GetAbsoluteTimerIRQStatus API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_GetAbsoluteTimerIRQStatus (     const uint8 Fr_CtrlIdx,     const uint8 Fr_AbsTimerIdx,</pre>	
---------------	--	--

**Fr\_17\_Eray driver**
**Table 239 Specification for Fr\_17\_Eray\_GetAbsoluteTimerIRQStatus API (continued)**

	<pre>boolean * const Fr_IRQStatusPtr )</pre>	
<b>Service ID</b>	0x20	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	
<b>Parameters (in)</b>	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.
<b>Parameters (out)</b>	Fr_IRQStatusPtr	Address the output value is stored to.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Gets the IRQ status of an absolute timer.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INV_TIMER_IDX: Parameter timer index exceeds number of available timers. FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None.	

**2.3.3.31 Fr\_17\_Eray\_GetVersionInfo**
**Table 240 Specification for Fr\_17\_Eray\_GetVersionInfo API**

<b>Syntax</b>	<pre>void Fr_17_Eray_GetVersionInfo (     Std_VersionInfoType * const VersioninfoPtr )</pre>	
<b>Service ID</b>	0x1b	
<b>Sync/Async</b>	Synchronous	



**Fr\_17\_Eray driver**
**Table 240 Specification for Fr\_17\_Eray\_GetVersionInfo API (continued)**

<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Reentrant	
<b>Parameters (in)</b>	-	-
<b>Parameters (out)</b>	VersionInfoPtr	Address where the version information of the FR module must be stored.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	void	-
<b>Description</b>	Returns the version information of the FR module. The version information includes: - Module Id - Vendor Id - Vendor specific version numbers.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	FrVersionInfoApi	
<b>User hints</b>	None	

**2.3.3.32 Fr\_17\_Eray\_ReadCCConfig**
**Table 241 Specification for Fr\_17\_Eray\_ReadCCConfig API**

<b>Syntax</b>	<pre>Std_ReturnType Fr_17_Eray_ReadCCConfig (     const uint8 Fr_CtrlIdx,     const uint8 Fr_ConfigParamIdx,     uint32 * const Fr_ConfigParamValuePtr )</pre>	
<b>Service ID</b>	0x2e	
<b>Sync/Async</b>	Synchronous	
<b>ASIL Level</b>	QM	
<b>Re-entrancy</b>	Non Reentrant for the same device	

**Fr\_17\_Eray driver**
**Table 241 Specification for Fr\_17\_Eray\_ReadCCConfig API (continued)**

<b>Parameters (in)</b>	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>.
<b>Parameters (out)</b>	Fr_ConfigParamValuePtr	Address the output value is stored to.
<b>Parameters (in - out)</b>	-	-
<b>Return</b>	Std_ReturnType	E_OK : API call finished successfully E_NOT_OK : API call aborted due to errors
<b>Description</b>	Reads a FlexRay protocol configuration parameter for a particular FlexRay controller out of the configuration of the module.	
<b>Source</b>	AUTOSAR	
<b>Error handling</b>	DET: FR_17_ERAY_E_INIT_FAILED: FR module was not initialized. FR_17_ERAY_E_INV_CONFIG_IDX: Invalid value passed as parameter Fr_ConfigParamIdx. FR_17_ERAY_E_INV_CTRL_IDX: Invalid controller index. FR_17_ERAY_E_PARAM_POINTER: Invalid pointer in parameter list. Runtime Errors: None DEM: None Safety Errors: None <i>Note: All DET IDs are also reported as safety errors.</i>	
<b>Configuration dependencies</b>	-	
<b>User hints</b>	None	

### 2.3.4 Notifications and Callbacks

The FR driver does not support any notification and callbacks.

### 2.3.5 Scheduled functions

The FR driver is executed in the context of the FlexRay interface and has no function to be scheduled.

### 2.3.6 Interrupt service routines

The FR driver does not support any interrupt handlers.

### 2.3.7 Error codes classification

This section explains various error types and their corresponding source APIs.

**Fr\_17\_Eray driver**
**2.3.7.1 Development errors**

The following table lists all the development errors reported by the driver.

**Table 242 Description of development errors reported**

Description	Source	Error code and value	Applicable APIs
Parameter timer index exceeds number of available timers.	AUTOSAR	FR_17_ERAY_E_INV_TIMER_IDX=0x01	Fr_17_Eray_CancelAbsoluteTimer, Fr_17_Eray_SetAbsoluteTimer, Fr_17_Eray_EnableAbsoluteTimerIRQ, Fr_17_Eray_AckAbsoluteTimerIRQ, Fr_17_Eray_DisableAbsoluteTimerIRQ, Fr_17_Eray_GetAbsoluteTimerIRQStatus
Invalid pointer in parameter list.	AUTOSAR	FR_17_ERAY_E_PARAM_POINTER=0x02	Fr_17_Eray_CheckTxLPduStatus, Fr_17_Eray_ReceiveRxLPdu, Fr_17_Eray_TransmitTxLPdu, Fr_17_Eray_GetPOCStatus, , Fr_17_Eray_ReadCCConfig, , Fr_17_Eray_GetVersionInfo, Fr_17_Eray_GetAbsoluteTimerIRQStatus, Fr_17_Eray_GetClockCorrection, Fr_17_Eray_GetNumOfStartupFrames, Fr_17_Eray_GetGlobalTime, Fr_17_Eray_GetNmVector, Fr_17_Eray_GetChannelStatus, Fr_17_Eray_GetSyncFrameList, Fr_17_Eray_GetWakeupRxStatus
Parameter offset exceeds bounds.	AUTOSAR	FR_17_ERAY_E_INV_OFFSET=0x03	Fr_17_Eray_SetAbsoluteTimer
Invalid controller index.	AUTOSAR	FR_17_ERAY_E_INV_CTRL_IDX=0x04	Fr_17_Eray_DisableLPdu, Fr_17_Eray_ReconfigLPdu, ,

**Fr\_17\_Eray driver**
**Table 242 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
			Fr_17_Eray_PrepareLPdu, Fr_17_Eray_CheckTxLPdu Status, Fr_17_Eray_ReceiveRxLPd u, Fr_17_Eray_CancelTxLPdu , Fr_17_Eray_TransmitTxLP du, Fr_17_Eray_SendWUP, Fr_17_Eray_GetPOCStatus , Fr_17_Eray_SetWakeupCh annel, Fr_17_Eray_AbortCommu nication, Fr_17_Eray_HaltCommuni cation, Fr_17_Eray_AllSlots, Fr_17_Eray_AllowColdstar t, Fr_17_Eray_StartCommun ication, Fr_17_Eray_ReadCCConfig , Fr_17_Eray_GetAbsoluteTi merIRQStatus, Fr_17_Eray_DisableAbsol uteTimerIRQ, Fr_17_Eray_AckAbsoluteT imerIRQ, Fr_17_Eray_EnableAbsolu teTimerIRQ, Fr_17_Eray_CancelAbsolu teTimer, Fr_17_Eray_SetAbsoluteTi mer, Fr_17_Eray_GetWakeupRx Status, Fr_17_Eray_GetSyncFram eList, Fr_17_Eray_GetClockCorr ection, Fr_17_Eray_GetChannelSt atus, Fr_17_Eray_GetNumOfSta rtupFrames, Fr_17_Eray_GetNmVector, Fr_17_Eray_GetGlobalTim

**Fr\_17\_Eray driver**
**Table 242 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
			e, Fr_17_Eray_ControllerInit
Invalid channel index.	AUTOSAR	FR_17_ERAY_E_INV_CHNL_IDX=0x05	Fr_17_Eray_ReconfigLPdu, Fr_17_Eray_SetWakeupChannel, Fr_17_Eray_GetPOCStatus
Parameter cycle exceeds 63.	AUTOSAR	FR_17_ERAY_E_INV_CYCLE=0x06	Fr_17_Eray_ReconfigLPdu, Fr_17_Eray_SetAbsoluteTimer
FR module was not initialized.	AUTOSAR	FR_17_ERAY_E_INIT_FAILED=0x08	Fr_17_Eray_Init, Fr_17_Eray_DisableLPdu, Fr_17_Eray_ReconfigLPdu, Fr_17_Eray_PrepareLPdu, Fr_17_Eray_CheckTxLPduStatus, Fr_17_Eray_ReceiveRxLPdu, Fr_17_Eray_CancelTxLPdu, Fr_17_Eray_TransmitTxLPdu, Fr_17_Eray_SendWUP, Fr_17_Eray_GetPOCStatus, Fr_17_Eray_SetWakeupChannel, Fr_17_Eray_AbortCommunication, Fr_17_Eray_HaltCommunication, Fr_17_Eray_AllSlots, Fr_17_Eray_AllowColdstart, Fr_17_Eray_StartCommunication, Fr_17_Eray_ControllerInit, Fr_17_Eray_ReadCCConfig, Fr_17_Eray_GetAbsoluteTimerIRQStatus, Fr_17_Eray_DisableAbsoluteTimerIRQ, Fr_17_Eray_AckAbsoluteTimerIRQ, Fr_17_Eray_EnableAbsolu

**Fr\_17\_Eray driver**
**Table 242 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
			teTimerIRQ, Fr_17_Eray_CancelAbsoluteTimer, Fr_17_Eray_SetAbsoluteTimer, Fr_17_Eray_GetWakeupRxStatus, Fr_17_Eray_GetSyncFrameList, Fr_17_Eray_GetClockCorrection, Fr_17_Eray_GetChannelStatus, Fr_17_Eray_GetNumOfStartupFrames, Fr_17_Eray_GetNmVector, Fr_17_Eray_GetGlobalTime
FR CC is not in the expected POC state.	AUTOSAR	FR_17_ERAY_E_INV_POCSTATE=0x09	Fr_17_Eray_SendWUP, Fr_17_Eray_GetPOCStatus, Fr_17_Eray_SetWakeupChannel, Fr_17_Eray_HaltCommunication, Fr_17_Eray_AllSlots, Fr_17_Eray_AllowColdstart, Fr_17_Eray_StartCommunication, Fr_17_Eray_SetAbsoluteTimer
Payload length parameter has an invalid value.	AUTOSAR	FR_17_ERAY_E_INV_LENGTH=0x0A	Fr_17_Eray_ReconfigLPdu, Fr_17_Eray_TransmitTxLPdu
Invalid LPdu index.	AUTOSAR	FR_17_ERAY_E_INV_LPDU_IDX=0x0B	Fr_17_Eray_DisableLPdu, Fr_17_Eray_ReconfigLPdu, Fr_17_Eray_PrepareLPdu, Fr_17_Eray_CheckTxLPduStatus, Fr_17_Eray_ReceiveRxLPdu, Fr_17_Eray_CancelTxLPdu

**Fr\_17\_Eray driver**
**Table 242 Description of development errors reported (continued)**

Description	Source	Error code and value	Applicable APIs
			Fr_17_Eray_TransmitTxLPdu
Invalid FlexRay header CRC.	AUTOSAR	FR_17_ERAY_E_INV_HEADERCRC=0x0C	Fr_17_Eray_ReconfigLPdu
Invalid value passed as parameter Fr_ConfigParamIdx.	AUTOSAR	FR_17_ERAY_E_INV_CONFIG_IDX=0x0D	Fr_17_Eray_ReadCCConfig
Invalid framelist size value.	AUTOSAR	FR_17_ERAY_E_INV_FRAMELIST_SIZE=0x0E	Fr_17_Eray_GetSyncFrameList

**2.3.7.2 Production errors**

The following table lists all the production errors reported by the driver.

**Table 243 Description of production errors reported**

Description	Source	Error code and value	Applicable APIs
FlexRay controller hardware test failure.	AUTOSAR	FR_E_CTRL_TESTRESULT=Value Assigned by DEM	Fr_17_Eray_PrepareLPdu, Fr_17_Eray_DisableLPdu, Fr_17_Eray_ReconfigLPdu, Fr_17_Eray_ReceiveRxLPdu, Fr_17_Eray_CancelTxLPdu, Fr_17_Eray_TransmitTxLPdu, Fr_17_Eray_SetWakeupChannel, Fr_17_Eray_SendWUP, Fr_17_Eray_AbortCommunication, Fr_17_Eray_HaltCommunication, Fr_17_Eray_AllSlots, Fr_17_Eray_AllowColdstart, Fr_17_Eray_StartCommunication, Fr_17_Eray_ControllerInit, Fr_17_Eray_Init
Flexray Protocol communication error - Slot Error (configured in FrIf Module).	AUTOSAR	FRIF_E_LPDU_SLOTSTATUS=Value Assigned by DEM	Fr_17_Eray_CheckTxLPduStatus, Fr_17_Eray_ReceiveRxLPdu

---

**Fr\_17\_Eray driver**

### 2.3.7.3 Safety errors

The driver does not report any safety errors.

### 2.3.7.4 Runtime errors

The driver does not report any runtime errors.

## 2.3.8 Deviations and limitations

The section describes the deviations and limitations from software specification.

### 2.3.8.1 Deviations

There are no deviations for the FR driver.

### 2.3.8.2 Limitations

The section describes the limitations from software specification.

**Table 244 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 <code>PREPARE_LPDU</code> in the <code>FrIf</code> configuration are considered for hardware buffer reconfiguration. The hardware buffers which participate in buffer reconfiguration feature are allocated to a maximum of 4 LPdus.
Number of supported LPdus	The maximum number of LPdus that are supported by the FR driver is limited to 255. Hence, it has to be ensured that the number of LPdus in the <code>FrIf</code> configuration should not exceed the limit of 255.
Configuration of Key slot LPdu in <code>FrIf</code> module	If a key slot frame is present for a node, then key slot frame should be configured with the parameter <code>FrIfLPduIdx</code> set to a value 0 within the <code>FrIf</code> configuration. This is required for the generation of the correct data offset addresses within the FR driver.
LPdu count in all Post Build configurations	The pre-compile macro <code>FR_17_ERAY_MSG_BUFF_COUNT_MAX_0/FR_17_ERAY_MSG_BUFF_COUNT_MAX_1</code> generated in <code>Fr_17_Eray_Cfg.h</code> considers the LPdu count only in the present configuration and not the maximum 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 <code>FR_17_ERAY_MSG_BUFF_COUNT_MAX_0/FR_17_ERAY_MSG_BUFF_COUNT_MAX_1</code> will be generated with max number of LPdu count. However user has to verify this manually by checking the macro value.



### **2.3.9            Unsupported hardware features**

The following hardware features of ERAY are not supported:

- Relative timer
- Stop watch functionality

## Revision history

### Revision history

Major changes since the last revision

Date	Version	Description
2019-10-10	1.30.0_8.0	<ul style="list-style-type: none"> <li>Eth_17_GEthMac               <ul style="list-style-type: none"> <li>Key Architectural consideration for ETH controller initialization sequence is updated.</li> <li>Configuration parameters EthCtrlRxBufLenByte and EthCtrlTxBufLenByte is updated.</li> </ul> </li> <li>Fr_17_Eray               <ul style="list-style-type: none"> <li>Deviations and limitations section is updated.</li> </ul> </li> </ul>
2019-08-05	7.0	Reference to the BASIC User Manual is updated.
2019-07-26	6.0	Reference to the BASIC User Manual is updated.
2019-07-23	5.0	<ul style="list-style-type: none"> <li>Eth_17_GEthMac               <ul style="list-style-type: none"> <li>Limitations and deviations section is updated.</li> </ul> </li> <li>Fr_17_Eray               <ul style="list-style-type: none"> <li>FrTxConflictDetection configuration parameter is added.</li> <li>Example usage for the transmission conflict detection feature is added.</li> </ul> </li> </ul>
2019-04-22	4.0	Added support for the TC37xA and TC37xA_ED devices.
2019-04-12	3.0	<ul style="list-style-type: none"> <li>Added support for the TC35xA device.</li> <li>ETH driver updated for dual controller support.</li> </ul>
2019-02-04	2.0	Updated the <i>Integration hints</i> and <i>Reference information</i> for all modules.
2018-10-12	1.0	Initial version.

## Trademarks

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

**Edition 2019-10-10**

**Published by**  
**Infineon Technologies AG**  
**81726 Munich, Germany**

**© 2019 Infineon Technologies AG**  
**All Rights Reserved.**

**Do you have a question about any**  
**aspect of this document?**  
**Email: [erratum@infineon.com](mailto:erratum@infineon.com)**

**Document reference**  
**IFX-yol1559809191002**

## IMPORTANT NOTICE

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

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