

User Manual

for S32K1 ETH Driver

Document Number: UM2ETHASR4.4 Rev0000R1.0.1 Rev. 1.0

1 Revision History	2
2 Introduction	3
2.1 Supported Derivatives	3
2.2 Overview	4
2.3 About This Manual	5
2.4 Acronyms and Definitions	6
2.5 Reference List	6
3 Driver	8
3.1 Requirements	8
3.2 Driver Design Summary	8
3.3 Hardware Resources	9
3.4 Deviations from Requirements	9
3.4.1 Status Column Description	9
3.4.2 Ethernet Requirements Deviations	10
3.5 Driver Limitations	10
3.6 Driver usage and configuration tips	11
3.7 Runtime errors	11
3.7.1 Standardized Extended Production Errors	11
3.7.2 Vendor-Specific Extended Production Errors	11
3.8 Symbolic Names Disclaimer	11
4 Tresos Configuration Plug-in	12
4.1 Module Eth	15
4.2 Container EthGeneral	15
4.3 Parameter EthDevErrorDetect	15
4.4 Parameter EthGetDropCountApi	16
4.5 Parameter EthGetEtherStatsApi	16
4.6 Parameter EthGetCounterValuesApi	17
4.7 Parameter EthGetRxStatsApi	17
4.8 Parameter EthGetTxStatsApi	18
4.9 Parameter EthGetTxErrorCounterValuesApi	18
4.10 Parameter EthGlobalTimeSupport	19
4.11 Parameter EthVersionInfoApi	19
4.12 Parameter EthIndex	20
4.13 Parameter EthMainFunctionPeriod	20
4.14 Parameter EthMaxCtrlsSupported	21
4.15 Reference EthEcucPartitionRef	21
4.16 Container EthCtrlOffloading	22
4.17 Parameter EthCtrlEnableOffloadChecksumICMP	22

4.18 Parameter EthCtrlEnableOffloadChecksumIPv4	22
4.19 Parameter EthCtrlEnableOffloadChecksumTCP	23
4.20 Parameter EthCtrlEnableOffloadChecksumUDP	23
4.21 Container EthGeneralVendorSpecific	24
4.22 Parameter EthDisableDemEventDetect	24
4.23 Parameter EthEnableUserModeSupport	25
4.24 Parameter EthMulticoreSupport	25
4.25 Parameter EthUpdatePhysAddrFilterApi	26
4.26 Parameter EthSwtManagementSupportApi	26
4.27 Parameter EthTimeoutMethod	27
4.28 Parameter EthTimeoutDuration	27
4.29 Container EthConfigSet	28
4.30 Container EthCtrlConfig	28
4.31 Parameter EthCtrlEnableMii	29
4.32 Parameter EthCtrlEnableMmd	29
4.33 Parameter EthCtrlEnableRxInterrupt	30
4.34 Parameter EthCtrlEnableTxInterrupt	30
4.35 Parameter EthCtrlIdx	31
4.36 Parameter EthCtrlMacLayerType	31
4.37 Parameter EthCtrlMacLayerSubType	32
4.38 Parameter EthCtrlMacLayerSpeed	32
4.39 Parameter EthCtrlPhyAddress	33
4.40 Reference EthCtrlEcucPartitionRef	33
4.41 Container EthCtrlVendorSpecific	34
4.42 Container EthCtrlConfigGeneral	34
4.43 Parameter EthCtrlEnableAtInit	34
4.44 Parameter EthCtrlAllocateTxDataBuffers	35
4.45 Parameter EthCtrlAllocateRxDataBuffers	35
4.46 Parameter EthEthTrcvDriverVendorIdAndApiInfix	36
4.47 Parameter EthEthSwtDriverVendorIdAndApiInfix	36
4.48 Parameter EthDuplexMode	37
4.49 Parameter EthMaxFrameLength	37
4.50 Parameter EthTimeStampRequiredAccuracy	38
4.51 Parameter EthTimeStampCorrectionCounter	39
4.52 Reference EthModuleReferenceClock	40
4.53 Reference EthTimeStampReferenceClock	40
4.54 Container EthCtrlConfigInterrupts	41
4.55 Parameter EthCtrlConfigEventCallback	41
4.56 Parameter EthCtrlConfigTimeStampCallback	41
4.57 Parameter EthCtrlEnableErrorInterrupt	42

4.58 Parameter EthCtrlEnableWakeUpInterrupt	42
4.59 Parameter EthCtrlEnableTimerInterrupt	43
4.60 Reference ETH_E_ERR	43
4.61 Container EthCtrlConfigReceptionOptions	44
4.62 Parameter ENET_RX_CONFIG_ENABLE_PAYLOAD_LEN_CHECK	44
4.63 Parameter ENET_RX_CONFIG_STRIP_CRC_FIELD	45
4.64 Parameter ENET_RX_CONFIG_FORWARD_PAUSE_FRAMES	45
4.65 Parameter ENET_RX_CONFIG_REMOVE_PADDING	46
4.66 Parameter ENET_RX_CONFIG_ENABLE_FLOW_CONTROL	46
4.67 Parameter ENET_RX_CONFIG_REJECT_BROADCAST_FRAMES	47
4.68 Parameter ENET_RX_CONFIG_ENABLE_PROMISCUOUS_MODE	47
4.69 Parameter ENET_RX_CONFIG_ENABLE_MII_LOOPBACK	48
4.70 Container EthCtrlConfigTransmissionOptions	48
4.71 Parameter ENET_TX_CONFIG_DISABLE_CRC_APPEND	49
4.72 Parameter ENET_TX_CONFIG_ENABLE_MAC_ADDR_INSERTION	49
4.73 Container EthCtrlConfigRxAcceleratorOptions	49
4.74 Parameter ENET_RX_ACCEL_REMOVE_PAD	50
4.75 Parameter ENET_RX_ACCEL_ENABLE_IP_CHECK	50
4.76 Parameter ENET_RX_ACCEL_ENABLE_PROTO_CHECK	51
4.77 Parameter ENET_RX_ACCEL_ENABLE_MAC_CHECK	51
4.78 Parameter ENET_RX_ACCEL_ENABLE_SHIFT16	52
4.79 Container EthCtrlConfigTxAcceleratorOptions	52
4.80 Parameter ENET_TX_ACCEL_ENABLE_SHIFT16	53
4.81 Parameter ENET_TX_ACCEL_INSERT_IP_CHECKSUM	53
4.82 Parameter ENET_TX_ACCEL_INSERT_PROTO_CHECKSUM	54
4.83 Container EthCtrlConfigEgress	54
4.84 Reference EthCtrlConfigEgressLastSchedulerRef	55
4.85 Container EthCtrlConfigEgressFifo	55
4.86 Parameter EthCtrlConfigEgressFifoBufLenByte	56
4.87 Parameter EthCtrlConfigEgressFifoBufTotal	56
4.88 Parameter EthCtrlConfigEgressFifoIdx	57
4.89 Parameter EthCtrlConfigEgressFifoCallback	57
4.90 Parameter EthCtrlConfigEgressFifoPriorityAssignment	57
4.91 Container EthCtrlConfigScheduler	58
4.92 Container EthCtrlConfigSchedulerPredecessor	58
4.93 Parameter EthCtrlConfigSchedulerPredecessorOrder	59
4.94 Reference EthCtrlConfigSchedulerPredecessorRef	59
4.95 Container EthCtrlConfigShaper	60
4.96 Parameter EthCtrlConfigShaperIdleSlope	60
4.97 Reference EthCtrlConfigShaperPredecessorFifoRef	61

4.98 Container EthCtrlConfigIngress	61
4.99 Container EthCtrlConfigIngressFifo	62
4.100 Parameter EthCtrlConfigIngressFifoBufLenByte	62
4.101 Parameter EthCtrlConfigIngressFifoBufTotal	63
4.102 Parameter EthCtrlConfigIngressFifoIdx	63
4.103 Parameter EthCtrlConfigIngressFifoCallback	64
4.104 Parameter EthCtrlConfigIngressFifoPriorityAssignment	64
4.105 Container EthDemEventParameterRefs	65
4.106 Reference ETH_E_ACCESS	65
4.107 Reference ETH_E_RX_FRAMES_LOST	66
4.108 Reference ETH_E_CRC	66
4.109 Reference ETH_E_UNDERSIZEFRAME	67
4.110 Reference ETH_E_OVERSIZEFRAME	67
4.111 Reference ETH_E_ALIGNMENT	68
4.112 Reference ETH_E_SINGLECOLLISION	68
4.113 Reference ETH_E_MULTIPLECOLLISION	69
4.114 Reference ETH_E_LATECOLLISION	69
4.115 Container CommonPublishedInformation	70
4.116 Parameter ModuleId	70
4.117 Parameter VendorId	70
4.118 Parameter VendorApiInfix	71
4.119 Parameter ArReleaseMajorVersion	72
4.120 Parameter ArReleaseMinorVersion	72
4.121 Parameter ArReleaseRevisionVersion	73
4.122 Parameter SwMajorVersion	73
4.123 Parameter SwMinorVersion	74
4.124 Parameter SwPatchVersion	74
5 Module Index	75
5.1 Software Specification	75
6 Data Structure Index	76
6.1 Data Structures	76
7 Module Documentation	77
7.1 ETH Driver	77
7.1.1 Detailed Description	77
7.1.2 Function Reference	78
7.2 ENET Driver	84
7.2.1 Detailed Description	84
7.2.2 Data Structure Documentation	91

7.2.3 Types Reference	107
7.2.4 Enum Reference	108
7.2.5 Function Reference	113
8 Data Structure Documentation	130
8.1 Enet_Ip_ManagementInfo Struct Reference	130
8.1.1 Detailed Description	130



Chapter 1

Revision History

Revision	Date	Author	Description
1.0	24.02.2022	NXP RTD Team	Prepared for release RTD S32K1 Version 1.0.1

Chapter 2

Introduction

- [Supported Derivatives](#)
- [Overview](#)
- [About This Manual](#)
- [Acronyms and Definitions](#)
- [Reference List](#)

This User Manual describes NXP Semiconductors' AUTOSAR Ethernet Driver for S32K1.

AUTOSAR Ethernet Driver configuration parameters description can be found in the Tresos Configuration Plugin section. Deviations from the specification are described in the [Deviations from Requirements](#) section.

AUTOSAR Ethernet driver requirements and APIs are described in the Ethernet Driver Software Specification Document (version 4.4.0).

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48

- s32k142w_lqfp64
- s32k144_lqfp48
- s32k144_lqfp64
- s32k144_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100
- s32k146_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176

All of the above microcontroller devices are collectively named as S32K1.

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR:

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
DEM	Diagnostic Event Manager
DET	Default Error Tracer
ETH	Ethernet
ETHIF	Ethernet Interface
ETHTRCV	Ethernet Transceiver
ETHSWT	Ethernet Switch
MCU	Micro controller Unit
MII	Media Independent Interface
N/A	Not Available
RMII	Reduced Media Independent Interface
RAM	Random Access Memory

- The term "Ethernet Controller" is related to the hardware module providing the Ethernet functionality.
- The term "Ethernet Driver" is related to the software handling the Ethernet Controller.
- The term "Application" is used for the software utilizing the Ethernet Driver.

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
DEM	Diagnostic Event Manager
DET	Default Error Tracer
ETH	Ethernet
ETHIF	Ethernet Interface
ETHTRCV	Ethernet Transceiver
ETHSWT	Ethernet Switch
MCU	Microcontroller Unit
MII	Media Independent Interface
N/A	Not Available
RMII	Reduced Media Independent Interface
RAM	Random Access Memory

- The term "Ethernet Controller" is related to the hardware module providing the Ethernet functionality.
- The term "Ethernet Driver" is related to the software handling the Ethernet Controller.
- The term "Application" is used for the software utilizing the Ethernet Driver.

2.5 Reference List

#	Title	Version
1	Specification of Ethernet Driver	AUTOSAR Release 4.4.0
2	S32K1xx Series Reference Manual	S32K1xx Series Reference Manual, Rev. 14, 09/2021
3	S32K148_0N20V Errata	Rev. 22/OCT/2021
4	S32K1xx Data Sheet	S32K1xx Data Sheet, Rev. 14, 08/2021

Chapter 3

Driver

- [Requirements](#)
- [Driver Design Summary](#)
- [Hardware Resources](#)
- [Deviations from Requirements](#)
- [Driver Limitations](#)
- [Driver usage and configuration tips](#)
- [Runtime errors](#)
- [Symbolic Names Disclaimer](#)

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.4.0 Ethernet Driver Software Specification document (See Table [Reference List](#))

3.2 Driver Design Summary

The Ethernet Driver controls the Ethernet Media Access Controller (ENET) module of the S32K1 device. It provides the following features:

- Configuration and initialization of the Ethernet Controller
- Switching the Ethernet Controller on and off
- Reception and transmission of Ethernet frames
- Access to the Ethernet Controller's RMON counters (RxStats, TxStats and TxErrorCounterValues)
- Access to the Ethernet Transceiver device registers through MDIO
- Handling of the Ethernet Controller's interrupt requests
- Half and full duplex operation support
- Various PHY Interfaces (10/100 Mbps on MII and RMII)
- Time Synchronization over Ethernet (The gPTP stack has to be provided by the upper layers)
- Checksum hardware offloading for IPv4/IPv6 with TCP, UDP or ICMP

3.3 Hardware Resources

For the Ethernet driver, there are just S32K148 supported.

The hardware module configured by the Ethernet driver is ENET (Ethernet Media Access Controller). It has a single hardware instance (ENET_0).

The Ethernet controller to microcontroller pin mapping can be done using the file **S32K148_IO_Signal_↔ Description_Input_Multiplexing.xlsx** attached to the Reference Manual.

3.4 Deviations from Requirements

The driver deviates from the AUTOSAR Ethernet Driver Software Specification in some places.

The table [Status Column Description](#) identifies the AUTOSAR requirements that are not fully implemented, not implemented, or out of scope for the Ethernet Driver.

The table [Ethernet Requirements Deviations](#) provides the "Status" column description.

Term	Definition
N/S	Not In Scope
N/F	Not Fully Implemented
N/I	Not Implemented

3.4.1 Status Column Description

Requirement	Status	Description	Notes
ECUC_Eth_00035	N/S	EthGetDropCountApi configuration	This requirement just exist in AS↔ R4.2
ECUC_Eth_00036	N/S	EthGetEtherStatsApi configuration	This requirement just exist in AS↔ R4.2
CPR_RTD_00350.eth	N/S	Ethernet Bridge feature with dual channels shall be supported by the Ethernet driver.	This requirement removed.
ECUC_Eth_00057	N/S	Container Name - EthCtrlConfig↔ Shaper - Represents a Shaper an the egress side.	S32K1XX just has only one queue so this feature didn't support.
ECUC_Eth_00058	N/S	Name - EthCtrlConfigShaperIdle↔ Slope - Defines the increase of credit in bits per second for the AVB shaper.	S32K1XX just has only one queue so this feature didn't support.
ECUC_Eth_00059	N/S	Name - EthCtrlConfigShaper↔ PredecessorFifoRef - Reference to the fifo which is the predecessor for this shaper.	S32K1XX just has only one queue so this feature didn't support.

Requirement	Status	Description	Notes
SWS_Eth_00257	N/S	The Ethernet Driver module shall reject configurations with partition mappings which are not supported by the implementation.	S32K1XX doesn't support multiple core.
SWS_Eth_00259	N/S	The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.	S32K1XX doesn't support multiple core.
ECUC_Eth_00065	N/S	Name - EthCtrlEcucPartitionRef - Maps the Ethernet controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet driver is mapped to	S32K1XX doesn't support multiple core.
ECUC_Eth_00064	N/S	Name - EthEcucPartitionRef - Maps the Ethernet driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Ethernet driver will operate as an independent instance in each of the partitions.	S32K1XX doesn't support multiple core.
SWS_Eth_00258	N/S	If the driver manages several Ethernet controllers and if a subset of these controllers share peripheral resources or are somehow coupled (E.g. Communication control can only be done globally for all controllers), Ethernet driver shall emulate independent controllers to the upper layers. The coordination (E.g. Communication control) has to be done by the upper layer modules.	S32K1XX doesn't support multiple core.

3.4.2 Ethernet Requirements Deviations Files **Eth_<VariantName>_PBcfg.c** and **Eth_<VariantName>_PBcfg.h** will contain the definitions for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB).

Files **Eth_Cfg.c** and **Eth_Cfg.h** will contain the definitions for all parameters that are not variant aware.

3.5 Driver Limitations

The Ethernet Driver has the following limitations:

- The API functions *Eth_ReadMii* and *Eth_WriteMii* (and their counterparts *Eth_ReadMmd* and *Eth_WriteMmd*) are synchronous instead of asynchronous due to a variety of reasons further detailed in [this](#) AUTOSAR ticket.
- The length of a single received frame (including the 14-bytes Ethernet frame header and the 4-bytes FCS) must be less than or equal to `EthCtrlConfigIngressFifoBufLenByte`.

3.6 Driver usage and configuration tips

None

3.7 Runtime errors

The Ethernet driver generates the following DEM extended production errors at runtime:

3.7.1 Standardized Extended Production Errors

Function	Error code	Condition triggering the error
Eth_MainFunction	ETH_E_ACCESS	Controller access failed
Eth_MainFunction	ETH_E_RX_FRAMES_LOST	Rx frame lost detected
Eth_MainFunction	ETH_E_CRC	CRC failure detected
Eth_MainFunction	ETH_E_UNDERSIZEFRAME	Undersized frame detected
Eth_MainFunction	ETH_E_OVERSIZEFRAME	Oversized frame detected
Eth_MainFunction	ETH_E_ALIGNMENT	Alignment error detected
Eth_MainFunction	ETH_E_SINGLECOLLISION	Single collision detected
Eth_MainFunction	ETH_E_MULTIPLECOLLISION	Multiple collision detected
Eth_MainFunction	ETH_E_LATECOLLISION	Late collision detected

Function	Error code	Condition triggering the error
Eth_Ipw_EventIrqCallback	ETH_E_ERR	Payload receive error, Collision retry limit reached, Late collision detected,

3.7.2 Vendor-Specific Extended Production Errors | AXI Bus Error detected, Babbling transmit error, Babbling receive error, Transmit FIFO underrun.

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

```
#define <Mip>Conf_<Container_ShortName>_<Container_ID>
```

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations or to use names that may trigger other compile issues (e.g. match existing `#ifdefs` arguments).

Chapter 4

Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the driver. All the parameters are described below.

- Module [Eth](#)
 - Container [EthGeneral](#)
 - * Parameter [EthDevErrorDetect](#)
 - * Parameter [EthGetDropCountApi](#)
 - * Parameter [EthGetEtherStatsApi](#)
 - * Parameter [EthGetCounterValuesApi](#)
 - * Parameter [EthGetRxStatsApi](#)
 - * Parameter [EthGetTxStatsApi](#)
 - * Parameter [EthGetTxErrorCounterValuesApi](#)
 - * Parameter [EthGlobalTimeSupport](#)
 - * Parameter [EthVersionInfoApi](#)
 - * Parameter [EthIndex](#)
 - * Parameter [EthMainFunctionPeriod](#)
 - * Parameter [EthMaxCtrlsSupported](#)
 - * Reference [EthEcucPartitionRef](#)
 - * Container [EthCtrlOffloading](#)
 - Parameter [EthCtrlEnableOffloadChecksumICMP](#)
 - Parameter [EthCtrlEnableOffloadChecksumIPv4](#)
 - Parameter [EthCtrlEnableOffloadChecksumTCP](#)
 - Parameter [EthCtrlEnableOffloadChecksumUDP](#)
 - * Container [EthGeneralVendorSpecific](#)
 - Parameter [EthDisableDemEventDetect](#)
 - Parameter [EthEnableUserModeSupport](#)
 - Parameter [EthMulticoreSupport](#)
 - Parameter [EthUpdatePhysAddrFilterApi](#)
 - Parameter [EthSwtManagementSupportApi](#)
 - Parameter [EthTimeoutMethod](#)
 - Parameter [EthTimeoutDuration](#)
 - Container [EthConfigSet](#)
 - * Container [EthCtrlConfig](#)

- Parameter [EthCtrlEnableMii](#)
- Parameter [EthCtrlEnableMmd](#)
- Parameter [EthCtrlEnableRxInterrupt](#)
- Parameter [EthCtrlEnableTxInterrupt](#)
- Parameter [EthCtrlIdx](#)
- Parameter [EthCtrlMacLayerType](#)
- Parameter [EthCtrlMacLayerSubType](#)
- Parameter [EthCtrlMacLayerSpeed](#)
- Parameter [EthCtrlPhyAddress](#)
- Reference [EthCtrlEcucPartitionRef](#)
- Container [EthCtrlVendorSpecific](#)
- Container [EthCtrlConfigGeneral](#)
- Parameter [EthCtrlEnableAtInit](#)
- Parameter [EthCtrlAllocateTxDataBuffers](#)
- Parameter [EthCtrlAllocateRxDataBuffers](#)
- Parameter [EthEthTrcvDriverVendorIdAndApiInfix](#)
- Parameter [EthEthSwtDriverVendorIdAndApiInfix](#)
- Parameter [EthDuplexMode](#)
- Parameter [EthMaxFrameLength](#)
- Parameter [EthTimeStampRequiredAccuracy](#)
- Parameter [EthTimeStampCorrectionCounter](#)
- Reference [EthModuleReferenceClock](#)
- Reference [EthTimeStampReferenceClock](#)
- Container [EthCtrlConfigInterrupts](#)
- Parameter [EthCtrlConfigEventCallback](#)
- Parameter [EthCtrlConfigTimeStampCallback](#)
- Parameter [EthCtrlEnableErrorInterrupt](#)
- Parameter [EthCtrlEnableWakeUpInterrupt](#)
- Parameter [EthCtrlEnableTimerInterrupt](#)
- Reference [ETH_E_ERR](#)
- Container [EthCtrlConfigReceptionOptions](#)
- Parameter [ENET_RX_CONFIG_ENABLE_PAYLOAD_LEN_CHECK](#)
- Parameter [ENET_RX_CONFIG_STRIP_CRC_FIELD](#)
- Parameter [ENET_RX_CONFIG_FORWARD_PAUSE_FRAMES](#)
- Parameter [ENET_RX_CONFIG_REMOVE_PADDING](#)
- Parameter [ENET_RX_CONFIG_ENABLE_FLOW_CONTROL](#)
- Parameter [ENET_RX_CONFIG_REJECT_BROADCAST_FRAMES](#)
- Parameter [ENET_RX_CONFIG_ENABLE_PROMISCUOUS_MODE](#)
- Parameter [ENET_RX_CONFIG_ENABLE_MII_LOOPBACK](#)
- Container [EthCtrlConfigTransmissionOptions](#)
- Parameter [ENET_TX_CONFIG_DISABLE_CRC_APPEND](#)
- Parameter [ENET_TX_CONFIG_ENABLE_MAC_ADDR_INSERTION](#)
- Container [EthCtrlConfigRxAcceleratorOptions](#)
- Parameter [ENET_RX_ACCEL_REMOVE_PAD](#)
- Parameter [ENET_RX_ACCEL_ENABLE_IP_CHECK](#)
- Parameter [ENET_RX_ACCEL_ENABLE_PROTO_CHECK](#)
- Parameter [ENET_RX_ACCEL_ENABLE_MAC_CHECK](#)

- Parameter [ENET_RX_ACCEL_ENABLE_SHIFT16](#)
- Container [EthCtrlConfigTxAcceleratorOptions](#)
- Parameter [ENET_TX_ACCEL_ENABLE_SHIFT16](#)
- Parameter [ENET_TX_ACCEL_INSERT_IP_CHECKSUM](#)
- Parameter [ENET_TX_ACCEL_INSERT_PROTO_CHECKSUM](#)
- Container [EthCtrlConfigEgress](#)
- Reference [EthCtrlConfigEgressLastSchedulerRef](#)
- Container [EthCtrlConfigEgressFifo](#)
- Parameter [EthCtrlConfigEgressFifoBufLenByte](#)
- Parameter [EthCtrlConfigEgressFifoBufTotal](#)
- Parameter [EthCtrlConfigEgressFifoIdx](#)
- Parameter [EthCtrlConfigEgressFifoCallback](#)
- Parameter [EthCtrlConfigEgressFifoPriorityAssignment](#)
- Container [EthCtrlConfigScheduler](#)
- Container [EthCtrlConfigSchedulerPredecessor](#)
- Parameter [EthCtrlConfigSchedulerPredecessorOrder](#)
- Reference [EthCtrlConfigSchedulerPredecessorRef](#)
- Container [EthCtrlConfigShaper](#)
- Parameter [EthCtrlConfigShaperIdleSlope](#)
- Reference [EthCtrlConfigShaperPredecessorFifoRef](#)
- Container [EthCtrlConfigIngress](#)
- Container [EthCtrlConfigIngressFifo](#)
- Parameter [EthCtrlConfigIngressFifoBufLenByte](#)
- Parameter [EthCtrlConfigIngressFifoBufTotal](#)
- Parameter [EthCtrlConfigIngressFifoIdx](#)
- Parameter [EthCtrlConfigIngressFifoCallback](#)
- Parameter [EthCtrlConfigIngressFifoPriorityAssignment](#)
- Container [EthDemEventParameterRefs](#)
- Reference [ETH_E_ACCESS](#)
- Reference [ETH_E_RX_FRAMES_LOST](#)
- Reference [ETH_E_CRC](#)
- Reference [ETH_E_UNDERSIZEFRAME](#)
- Reference [ETH_E_OVERSIZEFRAME](#)
- Reference [ETH_E_ALIGNMENT](#)
- Reference [ETH_E_SINGLECOLLISION](#)
- Reference [ETH_E_MULTIPLECOLLISION](#)
- Reference [ETH_E_LATECOLLISION](#)
- Container [CommonPublishedInformation](#)
 - * Parameter [ModuleId](#)
 - * Parameter [VendorId](#)
 - * Parameter [VendorApiInfix](#)
 - * Parameter [ArReleaseMajorVersion](#)
 - * Parameter [ArReleaseMinorVersion](#)
 - * Parameter [ArReleaseRevisionVersion](#)
 - * Parameter [SwMajorVersion](#)
 - * Parameter [SwMinorVersion](#)
 - * Parameter [SwPatchVersion](#)

4.1 Module Eth

Configuration of the Eth (Ethernet Driver) module.

Included containers:

- [EthGeneral](#)
- [EthConfigSet](#)
- [CommonPublishedInformation](#)

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD

4.2 Container EthGeneral

General configuration of Ethernet Driver module.

Included subcontainers:

- [EthCtrlOffloading](#)
- [EthGeneralVendorSpecific](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Parameter EthDevErrorDetect

Enables / Disables development error detection.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.4 Parameter EthGetDropCountApi

Obsolete and kept only for backwards compatibility. The Eth_GetDropCount API has been replaced with Eth_GetCounterVal since R4.3.1. Use parameter EthGetCounterValuesApi instead.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.5 Parameter EthGetEtherStatsApi

Obsolete and kept only for backwards compatibility. The Eth_GetEtherStats API has been replaced with the Eth_GetTxStats and Eth_GetRxStats APIs since R4.3.1.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.6 Parameter EthGetCounterValuesApi

Enables / Disables Eth_GetCounterValues API.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.7 Parameter EthGetRxStatsApi

Enables / Disables Eth_GetRxStats API.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	false

4.8 Parameter EthGetTxStatsApi

Enables / Disables Eth_GetTxStats API.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	false

4.9 Parameter EthGetTxErrorCounterValuesApi

Enables / Disables Eth_GetTxErrorCounterValues API.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.10 Parameter EthGlobalTimeSupport

Enables/Disables the GlobalTime APIs used amongst others by Global Time Synchronization over Ethernet.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.11 Parameter EthVersionInfoApi

Enables / Disables Eth_GetVersionInfo API.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.12 Parameter EthIndex

Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	255
min	0

4.13 Parameter EthMainFunctionPeriod

The period between successive calls to the main function in seconds.

Ethernet driver does not require this information but the BSW scheduler.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0.001
max	65.535
min	0.001

4.14 Parameter EthMaxCtrlsSupported

Limits the total number of supported controllers.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	1
max	1
min	1

4.15 Reference EthEcucPartitionRef

Maps the Ethernet driver to zero or multiple ECUC partitions to make the module's API available in this partition.

The Ethernet driver will operate as an independent instance in each of the partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.16 Container EthCtrlOffloading

Configuration of hardware offloading features.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.17 Parameter EthCtrlEnableOffloadChecksumICMP

Enables / Disables hardware offloading for ICMP checksums.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.18 Parameter EthCtrlEnableOffloadChecksumIPv4

Enables / Disables hardware offloading for IPv4 checksums.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.19 Parameter EthCtrlEnableOffloadChecksumTCP

Enables / Disables hardware offloading for TCP checksums.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.20 Parameter EthCtrlEnableOffloadChecksumUDP

Enables / Disables hardware offloading for UDP checksums.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.21 Container EthGeneralVendorSpecific

General vendor-specific configuration of Ethernet Driver.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.22 Parameter EthDisableDemEventDetect

When this option is enabled, no DEM events are reported.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
	VARIANT-LINK-TIME: PRE-COMPILE
valueConfigClasses	

Property	Value
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	false

4.23 Parameter EthEnableUserModeSupport

When this parameter is enabled, the Eth module will adapt to run from User Mode, with the following measures:

(if applicable) a) configuring REG_PROT for the Eth Controllers so that the registers under protection eth be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1.

(if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode.

(if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	false

4.24 Parameter EthMulticoreSupport

Switches multicore support on or off:

False: For all variants, no EcucPartition shall be referenced in EthEcucPartitionRef.

True: For all variants, at least one EcucPartition needs to be referenced in EthEcucPartitionRef.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.25 Parameter EthUpdatePhysAddrFilterApi

Enables/Disables Eth_UpdatePhysAddrFilter API.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.26 Parameter EthSwtManagementSupportApi

Enables / Disables the Switch Management APIs.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.27 Parameter EthTimeoutMethod

Configures the timeout method.

Based on this selection a certain timeout method from OsIf will be used in the driver.

Note: If SystemTimer or CustomTimer are selected make sure the corresponding timer is enabled in OsIf General configuration.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_SYSTEM', 'OSIF_COUNTER_CUSTOM', 'OSIF_COUNTER_DUMMY']

4.28 Parameter EthTimeoutDuration

The unit of measurement is given in number of microseconds.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1000
max	65535
min	1

4.29 Container EthConfigSet

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

Included subcontainers:

- [EthCtrlConfig](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.30 Container EthCtrlConfig

Configuration of the individual controller

Included subcontainers:

- [EthCtrlVendorSpecific](#)
- [EthCtrlConfigEgress](#)
- [EthCtrlConfigIngress](#)
- [EthDemEventParameterRefs](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	255
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.31 Parameter EthCtrlEnableMii

Enables / Disables Media Independent Interface (MII) for transceiver access.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	false

4.32 Parameter EthCtrlEnableMmd

Enables / Disables Clause 45 Media Independent Interface (MII) for transceiver access.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.33 Parameter EthCtrlEnableRxInterrupt

Enables / Disables receive interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.34 Parameter EthCtrlEnableTxInterrupt

Enables / Disables transmit interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.35 Parameter EthCtrlIdx

Specifies the instance ID of the configured controller.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	0
max	0
min	0

4.36 Parameter EthCtrlMacLayerType

Defines the MAC layer type of the ethernet controller.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	ETH_MAC_LAYER_TYPE_XMII
literals	['ETH_MAC_LAYER_TYPE_XMII', 'ETH_MAC_LAYER_TYPE_XGMII', 'ETH_MAC_LAYER_TYPE_XXGMII']

4.37 Parameter EthCtrlMacLayerSubType

Defines the MAC layer subtype of the ethernet controller.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	STANDARD
literals	['REDUCED', 'SERIAL', 'STANDARD']

4.38 Parameter EthCtrlMacLayerSpeed

Defines the baud rate of the MAC layer.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	ETH_MAC_LAYER_SPEED_10M
literals	['ETH_MAC_LAYER_SPEED_10M', 'ETH_MAC_LAYER_SPEED_100M', 'ETH_MAC_LAYER_SPEED_1G', 'ETH_MAC_LAYER_SPEED_10G']

4.39 Parameter EthCtrlPhyAddress

Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	66:55:44:33:22:11

4.40 Reference EthCtrlEcucPartitionRef

Maps the Ethernet controller to zero or one ECUC partition.

The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.41 Container EthCtrlVendorSpecific

Vendor specific configuration of the individual controller

Included subcontainers:

- [EthCtrlConfigGeneral](#)
- [EthCtrlConfigInterrupts](#)
- [EthCtrlConfigReceptionOptions](#)
- [EthCtrlConfigTransmissionOptions](#)
- [EthCtrlConfigRxAcceleratorOptions](#)
- [EthCtrlConfigTxAcceleratorOptions](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.42 Container EthCtrlConfigGeneral

General Controller Configuration.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.43 Parameter EthCtrlEnableAtInit

Enabled - The controller is enabled at initialization.

Disabled - The controller is left disabled at initialization and must be later enabled by calling `Enet_Ip_EnableController`

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.44 Parameter EthCtrlAllocateTxDataBuffers

Enabled - The driver will allocate both Tx Buffer Descriptors and Tx Data Buffers.

Disabled - The driver will only allocate Tx Buffer Descriptors. The Tx Data Buffers will be allocated by the application.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.45 Parameter EthCtrlAllocateRxDataBuffers

Enabled - The driver will allocate both Rx Buffer Descriptors and Rx Data Buffers.

Disabled - The driver will only allocate Rx Buffer Descriptors. The Rx Data Buffers will be allocated by the application.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.46 Parameter EthEthTrcvDriverVendorIdAndApiInfix

The vendor id and vendor api infix for the EthTrcv driver, if any.

This is used to include the correct header file and to point to the EthTrcv callbacks.

Example: 43_PHYGROUPB for a 43_PHYGROUPB driver or leave it blank to include EthTrcv.h

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	

4.47 Parameter EthEthSwtDriverVendorIdAndApiInfix

The vendor id and vendor api infix for the EthSwt driver, if any.

This is used to include the correct header file and to point to the EthSwt APIs.

Example: 43_SJA1110 for a 43_SJA1110 driver or leave it blank to include EthSwt.h

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	

4.48 Parameter EthDuplexMode

Defines the controller's duplex mode.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	ETH_FULL_DUPLEX
literals	['ETH_FULL_DUPLEX', 'ETH_HALF_DUPLEX']

4.49 Parameter EthMaxFrameLength

Maximum frame length starting at DA and includes the CRC at the end of the frame.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1518
max	1518
min	64

4.50 Parameter EthTimeStampRequiredAccuracy

Value given in nanoseconds.

Represents the minimum required accuracy for this gPTP device.

To ensure there's enough headroom for clock drifts from both the GM and Slave, a maximum drift percentage per host must be assumed.

That is, $\text{TotalDriftPercentage} = \text{MaximumGmDriftPercentage} + \text{MaximumSlaveDriftPercentage}$. Then the following inequality must be true:

$\text{EthTimeStampReferenceClock} * (1 + \text{TotalDriftPercentage} / 100) > (1 / \text{EthTimeStampRequiredAccuracy}) * 10^9$.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	40
max	4294967295
min	1

4.51 Parameter EthTimeStampCorrectionCounter

Value given in clock cycles.

Defines after how many timer clock cycles the correction counter should be reset and trigger a correction increment on the timer.

A value of 0 disables the correction counter and no corrections occur.

If remote frequency > local frequency, local timer need to slow down. So Correction increment must be smaller normal increment.

If remote frequency < local frequency, local timer need to speed up. So Correction increment must be bigger normal increment.

Calculation expression for Correction increment: $(\text{Origin} - (\text{EthTimeStampCorrectionCounter} * \text{normal Increment})) / ((\text{Local} / \text{normal increment}) - \text{EthTimeStampCorrectionCounter})$

With Origin and Local get value from type Eth_RateRatioType. Normal increment = $(1 / \text{local frequency}) * 10^9$.

Note: + Need to select value of EthTimeStampCorrectionCounter such that :

- Value Correction Increment is interger after calculation.
- This value also need to configure smaller than $(\text{Local} / \text{normal increment})$.
- Satisfied $\text{Origin} > \text{EthTimeStampCorrectionCounter} * \text{increment}$.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	0
max	1073741823
min	0

4.52 Reference EthModuleReferenceClock

Reference to the module clock for the ENET.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcuDefs/Mcu/McuModuleConfiguration/McuClockSetting↔ Config/McuClockReferencePoint

4.53 Reference EthTimeStampReferenceClock

Reference to the Timestamp source clock for the ENET.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcuDefs/Mcu/McuModuleConfiguration/McuClockSetting↔ Config/McuClockReferencePoint

4.54 Container EthCtrlConfigInterrupts

Interrupt sources Implements

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.55 Parameter EthCtrlConfigEventCallback

Callback function invoked when one of the events: wakeup, parse, error is encountered.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	Eth_Ipw_EventIrqCallback

4.56 Parameter EthCtrlConfigTimeStampCallback

Callback to be invoked when a timer event occurs

Property	Value
type	ECUC-FUNCTION-NAME-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	Eth_Ipw_TimeStampCallback

4.57 Parameter EthCtrlEnableErrorInterrupt

Enable/disable error interrupts

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
default Value	false

4.58 Parameter EthCtrlEnableWakeUpInterrupt

Enable/disable wakeup interrupts

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.59 Parameter EthCtrlEnableTimerInterrupt

Enable/disable interrupts related to the timer

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.60 Reference ETH_E_ERR

Reference to the DemEventParameter which shall be issued when any payload receive/
collision retry limit reached/ late collision/ AXI Bus Error/ Babbling transmit/
Babbling receive/ Transmit FIFO underrun errors has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.61 Container EthCtrlConfigReceptionOptions

Reception special options Configuration.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.62 Parameter ENET_RX_CONFIG_ENABLE_PAYLOAD_LEN_CHECK

The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLC] field

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.63 Parameter ENET_RX_CONFIG_STRIP_CRC_FIELD

Specifies whether the CRC field of received frames is transmitted or stripped. NOTE: If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.64 Parameter ENET_RX_CONFIG_FORWARD_PAUSE_FRAMES

If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.65 Parameter ENET_RX_CONFIG_REMOVE_PADDING

Padding is removed from received frames.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.66 Parameter ENET_RX_CONFIG_ENABLE_FLOW_CONTROL

If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.67 Parameter

ENET_RX_CONFIG_REJECT_BROADCAST_FRAMES

If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.68 Parameter

ENET_RX_CONFIG_ENABLE_PROMISCUOUS_MODE

All frames are accepted regardless of address matching.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.69 Parameter

ENET_RX_CONFIG_ENABLE_MII_LOOPBACK

Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.70 Container EthCtrlConfigTransmissionOptions

Transmission special options Configuration.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.71 Parameter ENET_TX_CONFIG_DISABLE_CRC_APPEND

The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
default Value	false

4.72 Parameter ENET_TX_CONFIG_ENABLE_MAC_ADDR_INSERTION

The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
default Value	false

4.73 Container EthCtrlConfigRxAcceleratorOptions

Enable/Disable available reception acceleration options.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.74 Parameter ENET_RX_ACCEL_REMOVE_PAD

Any bytes following the IP payload section of the frame are removed from the frame.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.75 Parameter ENET_RX_ACCEL_ENABLE_IP_CHECK

If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO is cleared). operates in store and forward mode (RSFL).

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
default Value	false

4.76 Parameter ENET_RX_ACCEL_ENABLE_PROTO_CHECK

If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
default Value	false

4.77 Parameter ENET_RX_ACCEL_ENABLE_MAC_CHECK

Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.78 Parameter ENET_RX_ACCEL_ENABLE_SHIFT16

When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary.

NOTE: This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.79 Container EthCtrlConfigTxAcceleratorOptions

Enable/Disable available transmit acceleration options.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.80 Parameter ENET_TX_ACCEL_ENABLE_SHIFT16

Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.81 Parameter ENET_TX_ACCEL_INSERT_IP_CHECKSUM

If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.82 Parameter

ENET_TX_ACCEL_INSERT_PROTO_CHECKSUM

If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.83 Container EthCtrlConfigEgress

Controller Configuration Egress: Configuration of one Ethernet controller egress behavior.

Included subcontainers:

- [EthCtrlConfigEgressFifo](#)
- [EthCtrlConfigScheduler](#)
- [EthCtrlConfigShaper](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.84 Reference EthCtrlConfigEgressLastSchedulerRef

Reference to the scheduler which is the last in the Egress structure.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfig↵ Egress/EthCtrlConfigScheduler

4.85 Container EthCtrlConfigEgressFifo

Controller Configuration Egress: Represents a FIFO at the egress side.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	255
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.86 Parameter EthCtrlConfigEgressFifoBufLenByte

Length of FIFO elements in bytes.

Note that this represents the length of a single packet buffer (including the 14-bytes Ethernet frame header and the 4-bytes FCS).

Therefore, the payload length will be equal to EthCtrlConfigEgressFifoBufLenByte - 18.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	64
max	1536
min	64

4.87 Parameter EthCtrlConfigEgressFifoBufTotal

FIFO Buffer Count (defines the number of packet buffers of length EthCtrlConfigEgressFifoBufLenByte).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	16
max	32
min	1

4.88 Parameter EthCtrlConfigEgressFifoIdx

Egress FIFO index.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	0
max	0
min	0

4.89 Parameter EthCtrlConfigEgressFifoCallback

Callback function invoked when an Egress (Tx) FIFO event is encountered.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	Eth_TxIrqCallback

4.90 Parameter EthCtrlConfigEgressFifoPriorityAssignment

Message egress priority assignment.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	7
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	7
min	0

4.91 Container EthCtrlConfigScheduler

Controller Configuration Scheduler: Represents a Scheduler on the Egress side.

Included subcontainers:

- [EthCtrlConfigSchedulerPredecessor](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	255
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.92 Container EthCtrlConfigSchedulerPredecessor

Controller Configuration Scheduler Predecessors: Defines an ordered list of predecessors for this scheduler.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.93 Parameter EthCtrlConfigSchedulerPredecessorOrder

Defines the order of the scheduler predecessors.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	255
min	0

4.94 Reference EthCtrlConfigSchedulerPredecessorRef

Choice reference to the scheduler predecessor.

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destinations	['/AUTOSAR/EcuDefs/Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfig↵ Egress/EthCtrlConfigEgressFifo', '/AUTOSAR/EcuDefs/Eth/EthConfigSet/↵ EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler', '/AUTOSA↵ R/EcuDefs/Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrl↵ ConfigShaper']

4.95 Container EthCtrlConfigShaper

Controller Configuration Shaper : Represents a Shaper at the Egress side.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.96 Parameter EthCtrlConfigShaperIdleSlope

Defines the increase of credit in bits per second for the AVB shaper.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	4294967295
min	0

4.97 Reference EthCtrlConfigShaperPredecessorFifoRef

Reference to the FIFO which is the predecessor for this Shaper.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfig← Egress/EthCtrlConfigEgressFifo

4.98 Container EthCtrlConfigIngress

Controller Configuration Ingress: Configuration of one Ethernet controller ingress behavior.

Included subcontainers:

- [EthCtrlConfigIngressFifo](#)

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.99 Container EthCtrlConfigIngressFifo

Controller Configuration Ingress : Represents a FIFO at the ingress side.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	255
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.100 Parameter EthCtrlConfigIngressFifoBufLenByte

Length of FIFO elements in bytes.

Note that this represents the length of a single packet buffer (including the 14-bytes Ethernet frame header and the 4-bytes FCS).

Therefore, the payload length will be equal to EthCtrlConfigIngressFifoBufLenByte - 18.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	64
max	1536
min	64

4.101 Parameter EthCtrlConfigIngressFifoBufTotal

FIFO Buffer Count (defines the number of packet buffers of length EthCtrlConfigIngressFifoBufLenByte).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	16
max	32
min	1

4.102 Parameter EthCtrlConfigIngressFifoIdx

Ingress FIFO index.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	0
max	0
min	0

4.103 Parameter EthCtrlConfigIngressFifoCallback

Callback function invoked when an Ingress (Rx) FIFO event is encountered.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	Eth_RxIrqCallback

4.104 Parameter EthCtrlConfigIngressFifoPriorityAssignment

Message ingress priority assignment.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	7
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	0
max	7
min	0

4.105 Container EthDemEventParameterRefs

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus in case the corresponding error occurs.

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.

Container is editable only when EthGeneral/EthGeneralVendorSpecific/EthDisableDemEventDetect = FALSE.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.106 Reference ETH_E_ACCESS

Reference to the DemEventParameter which shall be issued when the error 'Controller access failed' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.107 Reference ETH_E_RX_FRAMES_LOST

Reference to the DemEventParameter which shall be issued when the error 'Rx Frame Lost' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.108 Reference ETH_E_CRC

Reference to the DemEventParameter which shall be issued when the error 'CRC Failure' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.109 Reference ETH_E_UNDERSIZEFRAME

Reference to the DemEventParameter which shall be issued when the error 'Undersized Frame' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.110 Reference ETH_E_OVERSIZEFRAME

Reference to the DemEventParameter which shall be issued when the error 'Oversized Frame' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.111 Reference ETH_E_ALIGNMENT

Reference to the DemEventParameter which shall be issued when the error 'Alignment Error' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.112 Reference ETH_E_SINGLECOLLISION

Reference to the DemEventParameter which shall be issued when the error 'Single Collision' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.113 Reference ETH_E_MULTIPLECOLLISION

Reference to the DemEventParameter which shall be issued when the error 'Multiple Collisions' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.114 Reference ETH_E_LATECOLLISION

Reference to the DemEventParameter which shall be issued when the error 'Late Collisions' has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.115 Container CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.116 Parameter ModuleId

Module ID of this module from the BSW Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	88
max	88
min	88

4.117 Parameter VendorId

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

4.118 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU,

BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name.

In total, the implementation specific name is generated as follows:

[ModuleName]_[VendorId]_[VendorApiInfix][API name from SWS].

E.g. Assuming that the VendorId of the implementor is 123 and the implementer chose a

VendorApiInfix of 'v11r456' an API named Can_Write defined in the SWS will translate to Can_123_v11r456Write.

This parameter is mandatory for all modules with upper multiplicity > 1.

It shall not be used for modules with upper multiplicity = 1.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION

4.119 Parameter ArReleaseMajorVersion

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.120 Parameter ArReleaseMinorVersion

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.121 Parameter ArReleaseRevisionVersion

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.122 Parameter SwMajorVersion

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1

4.123 Parameter SwMinorVersion

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.124 Parameter SwPatchVersion

Patch version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1



Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

ETH Driver	77
ENET Driver	84



Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

Enet_Ip_ManagementInfo	
Management Frame information	130

Chapter 7

Module Documentation

7.1 ETH Driver

7.1.1 Detailed Description

7.1.1.1 Functions Expectation called for transmission in polling mode:

- `Eth_ProvideTxBuffer`
- `Eth_Transmit`
- `Eth_TxConfirmation`

Function Reference

- void `Eth_Init` (const `Eth_ConfigType` *`CfgPtr`)
Initializes the Ethernet Driver.
- Std_ReturnType `Eth_SetControllerMode` (uint8 `CtrlIdx`, `Eth_ModeType` `CtrlMode`)
Enables or disables the given controller.
- Std_ReturnType `Eth_GetControllerMode` (uint8 `CtrlIdx`, `Eth_ModeType` *`CtrlModePtr`)
Obtains the mode of the given controller.
- Std_ReturnType `Eth_TxTimeAwareShaperInit` (uint8 `CtrlIdx`)
Control transmit frame in duration time.
- void `Eth_GetPhysAddr` (uint8 `CtrlIdx`, uint8 *`PhysAddrPtr`)
Obtains the physical source address used by the indexed controller (the node MAC address).
- void `Eth_SetPhysAddr` (uint8 `CtrlIdx`, const uint8 *`PhysAddrPtr`)
Set or change physical address to the defined controller.
- BufReq_ReturnType `Eth_ProvideTxBuffer` (uint8 `CtrlIdx`, uint8 `Priority`, `Eth_BufIdxType` *`BufIdxPtr`, uint8 **`BufPtr`, uint16 *`LenBytePtr`)
Provides access to a transmit buffer of the specified controller.
- Std_ReturnType `Eth_Transmit` (uint8 `CtrlIdx`, `Eth_BufIdxType` `BufIdx`, `Eth_FrameType` `FrameType`, boolean `TxConfirmation`, uint16 `LenByte`, const uint8 *`PhysAddrPtr`)
Triggers transmission of a previously granted and then filled transmit buffer.
- void `Eth_Receive` (uint8 `CtrlIdx`, uint8 `FifoIdx`, `Eth_RxStatusType` *`RxStatusPtr`)
Triggers frames reception notifications.
- void `Eth_TxConfirmation` (uint8 `CtrlIdx`)
Triggers frame transmission confirmations.

Module Documentation

7.1.2 Function Reference

7.1.2.1 Eth_Init()

```
void Eth_Init (
    const Eth_ConfigType * CfgPtr )
```

Initializes the Ethernet Driver.

The configuration pointer is internally stored and the driver is initialized. The Ethernet controller is also reset.

Note

Function should be called only once.

Warning

Second call can cause undefined behavior. Call the [Eth_SetControllerMode\(\)](#) and pass ETH_MODE_DOWN to the CtrlMode argument before the second Eth_Init call to avoid problems.

Parameters

in	<i>CfgPtr</i>	Points to the implementation specific structure containing the Eth driver configuration Compiler_Warning: this warning due to behavior of compiler depend on configs. implements Eth_Init_Activity
----	---------------	---

7.1.2.2 Eth_SetControllerMode()

```
Std_ReturnType Eth_SetControllerMode (
    uint8 CtrlIdx,
    Eth_ModeType CtrlMode )
```

Enables or disables the given controller.

Warning

Disabling the controller clears all receive and transmit buffers. The application should ensure that no data is lost.

Parameters

in	<i>CtrlIdx</i>	Index of the controller to be enabled or disabled. The index is valid within the context of the Ethernet Driver only.
in	<i>CtrlMode</i>	Mode which shall be entered <ul style="list-style-type: none">ETH_MODE_DOWN: disable the controller S32K1 ETH Driver
78		<ul style="list-style-type: none">ETH_MODE_ACTIVE: enable the controller <div>NXP Semiconductors</div>

Returns

Error status

Return values

<i>E_OK</i>	No error was detected during the function execution.
<i>E_NOT_OK</i>	Development error was detected and the function failed. implements Eth_SetControllerMode_Activity

7.1.2.3 Eth_GetControllerMode()

```
Std_ReturnType Eth_GetControllerMode (
    uint8 CtrlIdx,
    Eth_ModeType * CtrlModePtr )
```

Obtains the mode of the given controller.

Parameters

in	<i>CtrlIdx</i>	Index of the controller which state shall be read. The index is valid within the context of the Ethernet Driver only.
out	<i>CtrlModePtr</i>	Pointer where to store the current controller mode.

Returns

Error status

Return values

<i>E_OK</i>	No error was detected during the function execution.
<i>E_NOT_OK</i>	Development error was detected and the function failed. implements Eth_GetControllerMode_Activity

7.1.2.4 Eth_TxTimeAwareShaperInit()

```
Std_ReturnType Eth_TxTimeAwareShaperInit (
    uint8 CtrlIdx )
```

Control transmit frame in duration time.

Parameters

in	<i>CtrlIdx</i>	Index of the controller which state shall be read. The index is valid within the context of the Ethernet Driver only.
----	----------------	---

Returns

Std_ReturnType

7.1.2.5 Eth_GetPhysAddr()

```
void Eth_GetPhysAddr (
    uint8 CtrlIdx,
    uint8 * PhysAddrPtr )
```

Obtains the physical source address used by the indexed controller (the node MAC address).

Parameters

in	<i>CtrlIdx</i>	Index of the controller whose MAC address should be read. The index is valid within the context of the Ethernet Driver only.
out	<i>PhysAddrPtr</i>	Pointer where to store the physical source address (MAC address). The address in network byte order is stored into 6 bytes at the given memory address. implements Eth_GetPhysAddr_Activity

7.1.2.6 Eth_SetPhysAddr()

```
void Eth_SetPhysAddr (
    uint8 CtrlIdx,
    const uint8 * PhysAddrPtr )
```

Set or change physical address to the defined controller.

Parameters

in	<i>CtrlIdx</i>	Index of the controller whose MAC address should be changed. The index is valid within the context of the Ethernet Driver only.
in	<i>PhysAddrPtr</i>	Pointer to the physical source address which should be set to the controller. The address is stored in 6 bytes of memory in network byte order.

Warning

This function may be called only when the controller is down. Calling function `Eth_Init` will change the controller's MAC address to the default value! implements `Eth_SetPhysAddr_Activity`

7.1.2.7 Eth_ProvideTxBuffer()

```
BufReq_ReturnType Eth_ProvideTxBuffer (
    uint8 CtrlIdx,
    uint8 Priority,
    Eth_BufIdxType * BufIdxPtr,
    uint8 ** BufPtr,
    uint16 * LenBytePtr )
```

Provides access to a transmit buffer of the specified controller.

Warning

The application should handle possible difference between the requested and granted buffer lengths. It is not necessary to use whole granted buffer i.e. some space at the end may not be written.

Parameters

in	<i>CtrlIdx</i>	Index of the controller which buffer shall be provided. The index is valid within the context of the Ethernet Driver only.
in	<i>Priority</i>	Frame priority for transmit buffer FIFO selection
out	<i>BufIdxPtr</i>	Index to the granted transmit buffer resource. It uniquely identifies the buffer in all subsequent calls of functions Eth_Transmit() and Eth_TxConfirmation() .
out	<i>BufPtr</i>	Pointer to the granted buffer. This is the space where the data to be transmitted shall be stored.
in, out	<i>LenBytePtr</i>	Buffer payload length <ul style="list-style-type: none"> • In: desired length in bytes • Out: granted length in bytes

Returns

Error and buffer status

Return values

<i>BUFREQ_OK</i>	Buffer was successfully granted and no error has occurred.
<i>BUFREQ_E_NOT_OK</i>	A development error was detected and no buffer was granted.
<i>BUFREQ_E_BUSY</i>	All available buffers in use therefore no buffer was granted. No error has been detected. implements <code>Eth_ProvideTxBuffer_Activity</code>

7.1.2.8 Eth_Transmit()

```
Std_ReturnType Eth_Transmit (
    uint8 CtrlIdx,
    Eth_BufIdxType BufIdx,
    Eth_FrameType FrameType,
    boolean TxConfirmation,
    uint16 LenByte,
    const uint8 * PhysAddrPtr )
```

Triggers transmission of a previously granted and then filled transmit buffer.

Parameters

in	<i>CtrlIdx</i>	Index of the controller which buffer shall be transmitted. The index is valid within the context of the Ethernet Driver only.
in	<i>BufIdx</i>	Index of the buffer resource to be transmitted.
in	<i>FrameType</i>	Desired value of the Ethernet frame type in the frame header.
in	<i>TxConfirmation</i>	Activates transmission confirmation.
in	<i>LenByte</i>	Buffer data length in bytes (payload length).
in	<i>PhysAddrPtr</i>	Physical target address (MAC address) in network byte order.

Returns

Error status

Return values

<i>E_OK</i>	No error was detected during the function execution.
<i>E_NOT_OK</i>	Development error was detected and the function failed. implements Eth_Transmit_Activity

7.1.2.9 Eth_Receive()

```
void Eth_Receive (
    uint8 CtrlIdx,
    uint8 FifoIdx,
    Eth_RxStatusType * RxStatusPtr )
```

Triggers frames reception notifications.

All receive buffers are checked and the first received frame is passed to the EthIf module. The caller is notified whether any frame was received and whether more frames are available in the receive queue.

Parameters

in	<i>CtrlIdx</i>	Index of the controller which shall be checked whether any new frames were received. The index is valid within the context of the Ethernet Driver only.
in	<i>FifoIdx</i>	Specifies the related fifo
out	<i>RxStatusPtr</i>	Informes the caller whether a frame was received (ETH_RECEIVED or ETH_NOT_RECEIVED) and whether more frames are available in the queue (ETH_RECEIVED or ETH_RECEIVED_MORE_DATA_AVAILABLE). implements Eth_Receive_Activity

7.1.2.10 Eth_TxConfirmation()

```
void Eth_TxConfirmation (
    uint8 CtrlIdx )
```

Triggers frame transmission confirmations.

Parameters

in	<i>CtrlIdx</i>	Index of the controller which shall be checked whether any frame transmission has finished. The index is valid within the context of the Ethernet Driver only.
----	----------------	--

All transmit buffers are checked and upper layers are informed about successfully transmitted frames. Buffers containing transmitted frames are unlocked after the confirmation. implements Eth_TxConfirmation_Activity

7.2 ENET Driver

7.2.1 Detailed Description

7.2.1.1 Platform-specific prerequisites

None

7.2.1.2 How to use the ENET driver in your application

The ENET driver does not handle clock setup or any kind of pin configuration. This is handled by the **Clock Driver** and **Pin Driver** modules, respectively. The driver assumes that the correct clock and pins configurations have been made, so it is the user's responsibility to set up clocking and pin configurations correctly.

In order to use the ENET driver in your application, the [Enet_Ip_Init\(\)](#) function should be called prior to using the rest of the API. The parameters of this function specify:

- The ENET instance to be initialized
- A structure which will hold the internal state of the driver
- A structure specifying the configuration of the ENET module
- Two arrays of structures specifying the configuration of the buffers (one structure for each rings - an array for Rx and one for Tx)
- The MAC address to be configured for the module

The configuration of the module is specified through the [Enet_Ip_ConfigType](#) structure and contains:

- MII-related configurations (mode, speed, duplex)
- Configuration options for the receive and transmit path
- The maximum frame length
- The MAC interrupt sources which should be enabled
- A callback function to be invoked on events
- The number of Rx and Tx rings used by the driver
- MAC address filtering options

The buffers configuration is specified through the [Enet_Ip_BufferConfigType](#) structure and contains:

- The size of the ring
- The size of each buffer in ring
- Pointers to the beginning of the buffer descriptor rings
- Pointers to the beginning of the memory area where the buffer data resides
- Channel interrupt sources
- A callback function to be invoked on per channel events

In order to de-initialize the driver, the [Enet_Ip_Deinit\(\)](#) function shall be used. This function will disable the ENET interrupts and the module, so calling other ENET driver functions after de-initializing the driver will have undefined behavior. In order to use the driver again, [Enet_Ip_Init\(\)](#) should be called.

7.2.1.3 How to enable zero-copy optimization

The buffers allocation is controlled by the following configuration parameters:

- EthCtrlAllocateTxDataBuffers
- EthCtrlAllocateRxDataBuffers

Setting them both to FALSE will prevent the driver from using internal buffers.

However, it is important to keep in mind that the programming sequences will also be slightly affected:

- EthAllocateTxDataBuffers = TRUE
 - Call sequence for each frame: Enet_Ip_GetTxBuff -> Enet_Ip_SendFrame
 - Enet_Ip_SendFrame must be called with the same buffer returned by Enet_Ip_GetTxBuff
- EthAllocateTxDataBuffers = FALSE
 - Call sequence for each frame: Enet_Ip_SendFrame
 - Enet_Ip_SendFrame can be called with any buffer provided by the application
- EthCtrlAllocateRxDataBuffers = TRUE
 - Call sequence for each frame: Enet_Ip_ReadFrame -> Enet_Ip_ProvideRxBuff
 - Enet_Ip_ProvideRxBuff must be called with the same buffer returned by Enet_Ip_ReadFrame
- EthCtrlAllocateRxDataBuffers = FALSE
 1. Application has an RX buffer pool
 - After controller initialization: Call Enet_Ip_ProvideRxBuff for each buffer
 - Call sequence for each frame: Enet_Ip_ReadFrame -> Enet_Ip_ProvideRxBuff
 - Enet_Ip_ProvideRxBuff can be called with either:
 - * The same buffer returned by Enet_Ip_ReadFrame if the application wants to reuse the buffer
 - * A buffer different than the one returned by Enet_Ip_ReadFrame if the application doesn't want to reuse the buffer
 2. Application doesn't have an RX buffer pool and wants to provide RX buffers on demand
 - After controller initialization: Nothing to be done
 - Call sequence for each frame: Enet_Ip_ProvideRxBuff -> Enet_Ip_ReadFrame

7.2.1.4 How to calculate size of ethernet buffer areas

Ethernet buffer areas is placed in noncache memory for each queue in each controller with order:

- Rx descriptor buffer areas: number of buffers configured(by `EthCtrlConfigIngressFifoBufTotal`) * size of each buffer ([Enet_Ip_BufferDescriptorType](#)).
- Rx data buffer areas: number of buffers configured(by `EthCtrlConfigIngressFifoBufTotal`) * size of each buffer (`EthCtrlConfigIngressFifoBufLenByte`).
- Tx descriptor buffer areas: number of buffers configured(by `EthCtrlConfigEgressFifoBufTotal`) * size of each buffer ([Enet_Ip_BufferDescriptorType](#)).
- Tx data buffer areas: number of buffers configured(by `EthCtrlConfigEgressFifoBufTotal`) * size of each buffer (`EthCtrlConfigEgressFifoBufLenByte`).

All of above memory areas need to align with start address by `FEATURE_ENET_BUFFDESCR_ALIGNMENT_BYTES`.

This configuration will be placed in loop with multiple queues and multiple controllers.

7.2.1.4.1 In code: - For Reception buffer descriptors for Rx Ring `["num:i($RingIdx)"]` :

```
VAR_ALIGN(Enet\_Ip\_BufferDescriptorType ENET_["num:i($CtrlIdx)"]_RxRing_["num:i($RingIdx)"]_DescBuffer[ENET_
FEATURE_ENET_BUFFDESCR_ALIGNMENT_BYTES])
```

- For Reception data buffers for Rx Ring `["num:i($RingIdx)"]`

```
VAR_ALIGN(uint8 ENET_["num:i($CtrlIdx)"]_RxRing_["num:i($RingIdx)"]_DataBuffer[ENET_["num:i($CtrlIdx)"]_MA
* ENET_["num:i($CtrlIdx)"]_MAX_RXBUFFLEN_SUPPORTED], FEATURE_ENET_BUFFDATA_ALIGNMENT_BYTES)
```

- For Transmission buffer descriptors for Tx Ring `["num:i($RingIdx)"]`

```
VAR_ALIGN(Enet\_Ip\_BufferDescriptorType ENET_["num:i($CtrlIdx)"]_TxRing_["num:i($RingIdx)"]_DescBuffer[ENET_
FEATURE_ENET_BUFFDESCR_ALIGNMENT_BYTES])
```

- For Transmission data buffers for Tx Ring `["num:i($RingIdx)"]`

```
VAR_ALIGN(uint8 ENET_["num:i($CtrlIdx)"]_TxRing_["num:i($RingIdx)"]_DataBuffer[ENET_["num:i($CtrlIdx)"]_MA
* ENET_["num:i($CtrlIdx)"]_MAX_TXBUFFLEN_SUPPORTED], FEATURE_ENET_BUFFDATA_ALIGNMENT_BYTES)
```

7.2.1.5 How to set up MAC address filters

The Address Filtering Module of the MAC checks the source address and destination address fields of each incoming packet. By default, the module only receives the frames having the destination address equal to the MAC address with which the ENET module has been initialized or the broadcast address.

The module has 1 additional 48-bit perfect address filter with masks for each byte, which can be configured for either source address or destination address filtering. This filter can be set-up by calling the `ENET_Ip_SetAddrPerfectFilter()` function.

Multicast and unicast destination addresses can be added to the hash filter by calling the [Enet_Ip_AddDstAddrToHashFilter\(\)](#) function and removed by calling the [Enet_Ip_RemoveDstAddrFromHashFilter\(\)](#) function.

For multicast and broadcast frames, the filters can be bypassed by calling the [Enet_Ip_SetBroadcastForwardAll\(\)](#) and [Enet_Ip_SetMulticastForwardAll\(\)](#) functions.

7.2.1.6 Examples:

7.2.1.6.1 Initializing the module: `Enet_Ip_Init(INST_ENET_0, &ENET_0_ConfigPB_<functionalGroupName>);`

```
/* ... */
```

```
Enet_Ip_Deinit(INST_ENET_0);
```

7.2.1.6.2 Sending a frame: `Enet_Ip_BufferType` buffer;

```
Enet_Ip_TxOptionsType options = {FALSE, FALSE, ENET_CHECKSUM_INSERTION_DISABLE};
```

```
buffer.length = 8;
```

```
if (ENET_STATUS_SUCCESS == Enet_Ip_GetTxBuff(INST_ENET_0, 0, &buffer, NULL_PTR))
```

```
{
```

```
    for (uint32 i = 0; i < 8; ++i)
```

```
    {
```

```
        buffer.data[i] = i;
```

```
    }
```

```
    Enet_Ip_SendFrame(INST_ENET_0, 0, &buffer, &options);
```

```
}
```

7.2.1.6.3 Receiving a frame - polling method (EthCtrlEnableRxInterrupt = FALSE): `Enet_Ip_BufferType` buffer;

```
Enet_Ip_RxInfoType info;
```

```
for (;;) 
```

```
{
```

```
    if (status == Enet_Ip_ReadFrame(INST_ENET_0, 0, &buffer, &info))
```

```
    {
```

```
        /* Process buffer */
```

```
        /* buffer is no longer needed, provide it to the driver in order to be used by the reception mechanism */
```

```
        Enet_Ip_ProvideRxBuff(INST_ENET_0, 0, &buffer);
```

```
    }
```

```
}
```

7.2.1.6.4 Receiving a frame - interrupt method (EthCtrlEnableRxInterrupt = TRUE): void rx_callback(uint8 instance, uint8 channel)

```

{
    Enet_Ip_BufferType buffer;

    if (status == Enet_Ip_ReadFrame(INST_ENET_0, 0, &buffer, NULL_PTR))
    {
        /* Process buffer */

        /* buffer is no longer needed, provide it to the driver in order to be used by the reception mechanism */
        Enet_Ip_ProvideRxBuff(INST_ENET_0, 0, &buffer);
    }
}

int main(void)
{
    /* ... */

    Enet_Ip_Init(INST_ENET_0, &ENET_0_ConfigPB_<functionalGroupName>);

    /* ... */
}

```

Data Structures

- struct [Enet_Ip_BufferType](#)
Send/Receive buffer information for the user. [More...](#)
- struct [Enet_Ip_BufferDescriptorType](#)
Send/Receive internal buffer descriptor. [More...](#)
- struct [Enet_Ip_BufferConfigType](#)
Specialization of ring configuration for Tx Rings. [More...](#)
- struct [Enet_Ip_ConfigType](#)
Defines the ENET module configuration structure. [More...](#)
- struct [Enet_Ip_StateType](#)
Internal driver state structure. [More...](#)
- struct [Enet_Ip_TimeStampType](#)
Defines the TimeStamp. [More...](#)
- struct [Enet_Ip_RxInfoType](#)
Enhanced information related to a receive buffer descriptor. [More...](#)
- struct [Enet_Ip_TxInfoType](#)
Enhanced information related to a transmit buffer descriptor. [More...](#)
- struct [Enet_Ip_TimerConfigType](#)
Configuration of the adjustable timer. [More...](#)
- struct [Enet_Ip_TimerChannelConfigType](#)
Configuration structure of the timer channels. [More...](#)
- struct [Enet_Ip_TxOptionsType](#)
Transmit options for a particular frame. [More...](#)

Types Reference

- typedef void(* [Enet_Ip_CallbackType](#)) (uint8 Instance, [Enet_Ip_EventType](#) Event)
Callback function invoked when a general event is encountered.
- typedef void(* [Enet_Ip_ChCallbackType](#)) (uint8 Instance, uint8 Ring)
Callback function invoked when a channel event is encountered.
- typedef void(* [Enet_Ip_TimerCallbackType](#)) (uint8 Instance)
Callback function invoked when a timer event is encountered. The channels parameter represents a mask of the channels on which an event occurred.

Enum Reference

- enum [Enet_Ip_StatusType](#)
Return values.
- enum [Enet_Ip_EventType](#)
Event specifier for the callback function.
- enum [Enet_Ip_TxSpecialConfigType](#)
Special transmit control configurations.
- enum [Enet_Ip_TxAcceleratorType](#)
Transmit accelerator configurations.
- enum [Enet_Ip_RxAcceleratorType](#)
Receive accelerator configurations.
- enum [Enet_Ip_CounterType](#)
Statistics counters enumeration.
- enum [Enet_Ip_MiiSpeedType](#)
Media Independent Interface speed selection.
- enum [Enet_Ip_PowerStateType](#)
Controller power state indication.
- enum [Enet_Ip_SysTimeCorrOffsetType](#)
System time correction offset.
- enum [Enet_Ip_TimerChannelModeType](#)
Timer channels configuration modes.
- enum [Enet_Ip_ChecksumInsControlType](#)
Checksum insertion control.
- enum [Enet_Ip_MmfrOpType](#)
Management Frame operation type.
- enum [Enet_Ip_FifoThresholdType](#)
Available FIFO threshold levels.

Initialization and De-initialization

- void [Enet_Ip_Init](#) (uint8 Instance, const [Enet_CtrlConfigType](#) *Config)
Initializes the ENET module.
- [Enet_Ip_PowerStateType](#) [Enet_Ip_GetPowerState](#) (uint8 Instance)
Gets the current power state of the ENET module.
- void [Enet_Ip_Deinit](#) (uint8 Instance)
Deinitializes the ENET module.
- void [Enet_Ip_EnableController](#) (uint8 Instance)
Enables all configured transmit and receive buffers and then enables the controller.
- void [Enet_Ip_RecoverFilter](#) (uint8 Instance)
Recover the hash table filtering caused by add add/remove open/close to hash table.
- void [Enet_Ip_DisableController](#) (uint8 Instance)
Disables the controller and resets all the configured transmit and receive buffers.
- void [Enet_Ip_SetSpeed](#) (uint8 Instance, [Enet_Ip_MiiSpeedType](#) Speed)
Sets the speed of the MII interface.
- [Enet_Ip_StatusType](#) [Enet_Ip_GetTxBuff](#) (uint8 Instance, uint8 Ring, [Enet_Ip_BufferType](#) *Buff, uint16 *BuffId)
Provides a transmit buffer to be used by the application for transmission.
- [Enet_Ip_StatusType](#) [Enet_Ip_SendFrame](#) (uint8 Instance, uint8 Ring, const [Enet_Ip_BufferType](#) *Buff, const [Enet_Ip_TxOptionsType](#) *Options)
Sends an Ethernet frame.
- [Enet_Ip_StatusType](#) [Enet_Ip_ReadFrame](#) (uint8 Instance, uint8 Ring, [Enet_Ip_BufferType](#) *Buff, [Enet_Ip_RxInfoType](#) *Info)
Reads a received Ethernet frame.
- void [Enet_Ip_ProvideRxBuff](#) (uint8 Instance, uint8 Ring, const [Enet_Ip_BufferType](#) *Buff)
Provides a receive buffer to be used by the driver for reception.
- boolean [Enet_Ip_IsFrameAvailable](#) (uint8 Instance, uint8 Ring)
Checks if there are more frames available in the given queue.
- [Enet_Ip_StatusType](#) [Enet_Ip_GetTransmitStatus](#) (uint8 Instance, uint8 Ring, const [Enet_Ip_BufferType](#) *Buff, [Enet_Ip_TxInfoType](#) *Info)
Checks if the transmission of a buffer is complete.
- uint32 [Enet_Ip_GetCounter](#) (uint8 Instance, [Enet_Ip_CounterType](#) Counter)
Gets statistics from the specified counter.
- void [Enet_Ip_ConfigCounters](#) (uint8 Instance, boolean Enable)
Enables/Disables the MIB counters.
- void [Enet_Ip_EnableMDIO](#) (uint8 Instance, boolean MiiPreambleDisabled, uint32 ModuleClk)
Enables the MDIO interface.
- [Enet_Ip_StatusType](#) [Enet_Ip_MDIORead](#) (uint8 Instance, uint8 PhyAddr, uint8 PhyReg, uint16 *Data, uint32 TimeoutMs)
Reads the selected register of the PHY.
- [Enet_Ip_StatusType](#) [Enet_Ip_MDIOWrite](#) (uint8 Instance, uint8 PhyAddr, uint8 PhyReg, uint16 Data, uint32 TimeoutMs)
Writes the selected register of the PHY.
- [Enet_Ip_StatusType](#) [Enet_Ip_MDIOReadMMD](#) (uint8 Instance, uint8 PhyAddr, uint8 Mmd, uint16 PhyReg, uint16 *Data, uint32 TimeoutMs)
Reads a register of the specified MMD in a PHY device.

- [Enet_Ip_StatusType Enet_Ip_MDIOWriteMMD](#) (uint8 Instance, uint8 PhyAddr, uint8 Mmd, uint16 Phy↔Reg, uint16 Data, uint32 TimeoutMs)
Writes a register of the specified MMD in a PHY device.
- void [Enet_Ip_SetMacAddr](#) (uint8 Instance, const uint8 *MacAddr)
Configures the physical address of the MAC.
- void [Enet_Ip_GetMacAddr](#) (uint8 Instance, uint8 *MacAddr)
Gets the physical address of the MAC.
- void [Enet_Ip_SetBroadcastForwardAll](#) (uint8 Instance, boolean Enable)
Enables/Disables forwarding of the broadcast traffic.
- void [Enet_Ip_SetMulticastForwardAll](#) (uint8 Instance, boolean Enable)
Enables/Disables forwarding of the multicast traffic, irrespective of the destination MAC address.
- void [Enet_Ip_AddDstAddrToHashFilter](#) (uint8 Instance, const uint8 *MacAddr)
Adds a hardware address to the hash filter. The destination address of an incoming packet is passed through CRC logic and then compared to the entries in the hash table.
- void [Enet_Ip_RemoveDstAddrFromHashFilter](#) (uint8 Instance, const uint8 *MacAddr)
Removes a hardware address from the hash filter. The destination address of an incoming packet is passed through CRC logic and then compared to the entries in the hash table.
- void [Enet_Ip_SetSysTimeCorr](#) (uint8 Instance, [Enet_Ip_SysTimeCorrOffsetType](#) Offset, [Enet_Ip_TimeStampType](#) DiffTimeStamp)
Set system time correction.
- void [Enet_Ip_TimerSetCorrection](#) (uint8 Instance, uint32 CorrectValue, uint32 CorrectPeriod)
Sets the timer correction period and correction increment value.
- void [Enet_Ip_TimerStop](#) (uint8 Instance)
The timer stop at the current value.
- [Enet_Ip_StatusType Enet_Ip_TimerEnableChannel](#) (uint8 Instance, uint8 Channel, const [Enet_Ip_TimerChannelConfig](#) *Config)
Configures and enables a specific timer channel.
- void [Enet_Ip_TimerSetCompare](#) (uint8 Instance, uint8 Channel, uint32 Value)
Sets the compare value for a channel configured as output compare.
- void [Enet_Ip_TimerStart](#) (uint8 Instance)
The timer starts incrementing.
- void [Enet_Ip_TimerInit](#) (uint8 Instance, const [Enet_Ip_TimerConfigType](#) *TimerConfig)
Initializes the adjustable timer using the given configuration and enables it.
- void [Enet_Ip_TimerSet](#) (uint8 Instance, [Enet_Ip_TimeStampType](#) TimerValue)
Sets the value of the timer.
- void [Enet_Ip_TimerGet](#) (uint8 Instance, [Enet_Ip_TimeStampType](#) *TimeStamp)
Gets the value of the timer.
- void [Enet_Ip_EnableTxStoreAndForward](#) (uint8 Instance)
Enable store and forward .
- void [Enet_Ip_SetFIFOThreshold](#) (uint8 Instance, [Enet_Ip_FifoThresholdType](#) ThresholdType, uint8 ThresholdValue)
Sets FIFO threshold levels.
- void [Enet_Ip_SetTxFIFOWatermark](#) (uint8 Instance, uint16 Watermark)
Sets the transmit FIFO watermark.
- #define **ETH_STOP_SEC_CODE**

7.2.2 Data Structure Documentation

7.2.2.1 struct Enet_Ip_BufferType

Send/Receive buffer information for the user.

Definition at line 345 of file Enet_Ip_Types.h.

7.2.2.2 struct Enet_Ip_BufferDescriptorType

Send/Receive internal buffer descriptor.

Definition at line 355 of file Enet_Ip_Types.h.

Data Fields

- uint16 [Length](#)
- uint16 [Control](#)
- uint8 * [Buffer](#)
- uint32 [Enh1](#)
- uint32 [Enh2](#)
- uint32 [Enh3](#)
- uint32 [TimeStamp](#)
- uint32 [Reserved1](#)
- uint32 [Reserved2](#)

7.2.2.2.1 Field Documentation

7.2.2.2.1.1 Length uint16 Length

Buffer descriptor data length.

Definition at line 357 of file Enet_Ip_Types.h.

7.2.2.2.1.2 Control uint16 Control

Buffer descriptor control and status.

Definition at line 358 of file Enet_Ip_Types.h.

7.2.2.2.1.3 Buffer uint8* Buffer

Data buffer pointer.

Definition at line 359 of file Enet_Ip_Types.h.

7.2.2.2.1.4 Enh1 uint32 Enh1

First word of the extended part of the buffer descriptor.

Definition at line 360 of file Enet_Ip_Types.h.

7.2.2.2.1.5 Enh2 uint32 Enh2

Second word of the extended part of the buffer descriptor.

Definition at line 361 of file Enet_Ip_Types.h.

7.2.2.2.1.6 Enh3 uint32 Enh3

Third word of the extended part of the buffer descriptor.

Definition at line 362 of file Enet_Ip_Types.h.

7.2.2.2.1.7 TimeStamp uint32 TimeStamp

TimeStamp of the frame.

Definition at line 363 of file Enet_Ip_Types.h.

7.2.2.2.1.8 Reserved1 uint32 Reserved1

Reserved.

Definition at line 364 of file Enet_Ip_Types.h.

7.2.2.2.1.9 Reserved2 uint32 Reserved2

Reserved.

Definition at line 365 of file Enet_Ip_Types.h.

7.2.2.3 struct Enet_Ip_BufferConfigType

Specialization of ring configuration for Tx Rings.

Definition at line 372 of file Enet_Ip_Types.h.

Data Fields

- [Enet_Ip_BufferDescriptorType](#) * [RingDesc](#)
- [Enet_Ip_ChCallbackType](#) [Callback](#)
- uint8 * [Buffer](#)
- uint32 [Interrupts](#)
- uint16 [BufferLen](#)
- uint16 [RingSize](#)

7.2.2.3.1 Field Documentation

7.2.2.3.1.1 RingDesc [Enet_Ip_BufferDescriptorType](#)* [RingDesc](#)

Buffer descriptor ring start address.

Definition at line 374 of file Enet_Ip_Types.h.

7.2.2.3.1.2 Callback [Enet_Ip_ChCallbackType](#) [Callback](#)

Callback function for current channel.

Definition at line 375 of file Enet_Ip_Types.h.

7.2.2.3.1.3 Buffer [uint8*](#) [Buffer](#)

Buffer data pool start address

Definition at line 376 of file Enet_Ip_Types.h.

7.2.2.3.1.4 Interrupts [uint32](#) [Interrupts](#)

Channel interrupt sources. A logical OR of "Enet_Ip_InterruptType".

Definition at line 377 of file Enet_Ip_Types.h.

7.2.2.3.1.5 BufferLen `uint16 BufferLen`

Length of each individual buffer in a pool

Definition at line 378 of file `Enet_Ip_Types.h`.

7.2.2.3.1.6 RingSize `uint16 RingSize`

Buffer descriptors number.

Definition at line 379 of file `Enet_Ip_Types.h`.

7.2.2.4 struct Enet_Ip_ConfigType

Defines the ENET module configuration structure.

Definition at line 386 of file `Enet_Ip_Types.h`.

Data Fields

- `uint8 RxRingCount`
- `uint8 TxRingCount`
- `Enet_Ip_TimerCallbackType TimerCallback`
- `uint32 Interrupts`
- `Enet_Ip_CallbackType Callback`
- `uint16 MaxFrameLen`
- `Enet_Ip_MiiModeType MiiMode`
- `Enet_Ip_MiiSpeedType MiiSpeed`
- `Enet_Ip_DuplexType MiiDuplex`
- `uint32 RxConfig`
- `uint32 TxConfig`
- `uint8 RxAccelerConfig`
- `uint8 TxAccelerConfig`
- `boolean EnableCtrl`

7.2.2.4.1 Field Documentation**7.2.2.4.1.1 RxRingCount** `uint8 RxRingCount`

The number of Receive rings to be used by the driver.

Definition at line 388 of file `Enet_Ip_Types.h`.

7.2.2.4.1.2 TxRingCount `uint8 TxRingCount`

The number of Transmit rings to be used by the driver.

Definition at line 389 of file `Enet_Ip_Types.h`.

7.2.2.4.1.3 TimerCallback `Enet_Ip_TimerCallbackType TimerCallback`

Callback to be invoked when a timer event occurs.

Definition at line 390 of file `Enet_Ip_Types.h`.

7.2.2.4.1.4 Interrupts `uint32 Interrupts`

MAC interrupt source. A logical OR of "Enet_Ip_InterruptType".

Definition at line 391 of file `Enet_Ip_Types.h`.

7.2.2.4.1.5 Callback `Enet_Ip_CallbackType Callback`

Callback function

Definition at line 392 of file `Enet_Ip_Types.h`.

7.2.2.4.1.6 MaxFameLen `uint16 MaxFameLen`

Maximum frame length.

Definition at line 393 of file `Enet_Ip_Types.h`.

7.2.2.4.1.7 MiiMode `Enet_Ip_MiiModeType MiiMode`

MII mode.

Definition at line 394 of file `Enet_Ip_Types.h`.

7.2.2.4.1.8 MiiSpeed `Enet_Ip_MiiSpeedType` `MiiSpeed`

MII Speed.

Definition at line 395 of file `Enet_Ip_Types.h`.

7.2.2.4.1.9 MiiDuplex `Enet_Ip_DuplexType` `MiiDuplex`

MII duplex.

Definition at line 396 of file `Enet_Ip_Types.h`.

7.2.2.4.1.10 RxConfig `uint32` `RxConfig`

MAC receive special configuration. A logical OR of "Enet_Ip_RxSpecialConfigType".

Definition at line 397 of file `Enet_Ip_Types.h`.

7.2.2.4.1.11 TxConfig `uint32` `TxConfig`

MAC transmit special configuration. A logical OR of "Enet_Ip_TxSpecialConfigType".

Definition at line 398 of file `Enet_Ip_Types.h`.

7.2.2.4.1.12 RxAccelerConfig `uint8` `RxAccelerConfig`

Receive accelerator, A logical OR of "Enet_Ip_RxAcceleratorType".

Definition at line 399 of file `Enet_Ip_Types.h`.

7.2.2.4.1.13 TxAccelerConfig `uint8` `TxAccelerConfig`

Transmit accelerator, A logical OR of "Enet_Ip_TxAcceleratorType".

Definition at line 400 of file `Enet_Ip_Types.h`.

7.2.2.4.1.14 EnableCtrl `boolean EnableCtrl`

Configures whether the controller is enabled at initialization.

Definition at line 401 of file `Enet_Ip_Types.h`.

7.2.2.5 struct Enet_Ip_StateType

Internal driver state structure.

Definition at line 408 of file `Enet_Ip_Types.h`.

Data Fields

- [Enet_Ip_CallbackType Callback](#)
- [Enet_Ip_ChCallbackType RxChCallback](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_ChCallbackType TxChCallback](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_TimerCallbackType TimerCallback](#)
- [Enet_Ip_BufferDescriptorType * RxBdBase](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_BufferDescriptorType * RxCurrentDesc](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_BufferDescriptorType * RxBdAlloc](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_BufferDescriptorType * TxBdBase](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_BufferDescriptorType * TxCurrentDesc](#) [FEATURE_ENET_NUM_RINGS]
- [Enet_Ip_BufferDescriptorType * TxTransmitDesc](#) [FEATURE_ENET_NUM_RINGS]
- `uint8 * TxDataBuffer` [FEATURE_ENET_NUM_RINGS]
- `uint8 * RxDataBuffer` [FEATURE_ENET_NUM_RINGS]
- `uint16 RxBuffNums` [FEATURE_ENET_NUM_RINGS]
- `uint16 TxBuffNums` [FEATURE_ENET_NUM_RINGS]
- `uint16 RxBuffLen` [FEATURE_ENET_NUM_RINGS]
- `uint16 TxBuffLen` [FEATURE_ENET_NUM_RINGS]
- `uint8 RxRingCount`
- `uint8 TxRingCount`

7.2.2.5.1 Field Documentation

7.2.2.5.1.1 Callback `Enet_Ip_CallbackType Callback`

Callback function for general interrupts.

Definition at line 410 of file `Enet_Ip_Types.h`.

7.2.2.5.1.2 RxChCallback `Enet_Ip_ChCallbackType RxChCallback` [FEATURE_ENET_NUM_RINGS]

Callback function for receive channels.

Definition at line 411 of file `Enet_Ip_Types.h`.

7.2.2.5.1.3 TxChCallback `Enet_Ip_ChCallbackType TxChCallback[FEATURE_ENET_NUM_RINGS]`

Callback function for transmit channels.

Definition at line 412 of file `Enet_Ip_Types.h`.

7.2.2.5.1.4 TimerCallback `Enet_Ip_TimerCallbackType TimerCallback`

Timer callback function.

Definition at line 413 of file `Enet_Ip_Types.h`.

7.2.2.5.1.5 RxBdBase `Enet_Ip_BufferDescriptorType* RxBdBase[FEATURE_ENET_NUM_RINGS]`

Receive buffer descriptor base address pointer array.

Definition at line 414 of file `Enet_Ip_Types.h`.

7.2.2.5.1.6 RxCurrentDesc `Enet_Ip_BufferDescriptorType* RxCurrentDesc[FEATURE_ENET_NUM_RINGS]`

The current available receive buffer descriptor pointer array.

Definition at line 415 of file `Enet_Ip_Types.h`.

7.2.2.5.1.7 RxBdAlloc `Enet_Ip_BufferDescriptorType* RxBdAlloc[FEATURE_ENET_NUM_RINGS]`

Pointer array used for enqueueing Rx buffers provided using `ENET_DRV_ProvideRxBuff`.

Definition at line 416 of file `Enet_Ip_Types.h`.

7.2.2.5.1.8 TxBdBase `Enet_Ip_BufferDescriptorType* TxBdBase[FEATURE_ENET_NUM_RINGS]`

Transmit buffer descriptor base address pointer array.

Definition at line 417 of file `Enet_Ip_Types.h`.

7.2.2.5.1.9 TxCurrentDesc `Enet_Ip_BufferDescriptorType*` TxCurrentDesc[FEATURE_ENET_NUM_RINGS]

The current available provided buffer descriptor pointer array.

Definition at line 418 of file Enet_Ip_Types.h.

7.2.2.5.1.10 TxTransmitDesc `Enet_Ip_BufferDescriptorType*` TxTransmitDesc[FEATURE_ENET_NUM_RINGS]

The current available transmit buffer descriptor pointer array.

Definition at line 419 of file Enet_Ip_Types.h.

7.2.2.5.1.11 TxDataBuffer `uint8*` TxDataBuffer[FEATURE_ENET_NUM_RINGS]

Transmit data buffer base address pointer array.

Definition at line 420 of file Enet_Ip_Types.h.

7.2.2.5.1.12 RxDataBuffer `uint8*` RxDataBuffer[FEATURE_ENET_NUM_RINGS]

Receive data buffer base address pointer array.

Definition at line 421 of file Enet_Ip_Types.h.

7.2.2.5.1.13 RxBuffNums `uint16` RxBuffNums[FEATURE_ENET_NUM_RINGS]

Number of Rx buffers in ring

Definition at line 422 of file Enet_Ip_Types.h.

7.2.2.5.1.14 TxBuffNums `uint16` TxBuffNums[FEATURE_ENET_NUM_RINGS]

Number of Tx buffers in ring

Definition at line 423 of file Enet_Ip_Types.h.

7.2.2.5.1.15 RxBuffLen `uint16 RxBuffLen[FEATURE_ENET_NUM_RINGS]`

Rx Length of buffer

Definition at line 424 of file `Enet_Ip_Types.h`.

7.2.2.5.1.16 TxBuffLen `uint16 TxBuffLen[FEATURE_ENET_NUM_RINGS]`

Tx Length of buffer

Definition at line 425 of file `Enet_Ip_Types.h`.

7.2.2.5.1.17 RxRingCount `uint8 RxRingCount`

The number of rings used by the driver.

Definition at line 426 of file `Enet_Ip_Types.h`.

7.2.2.5.1.18 TxRingCount `uint8 TxRingCount`

The number of rings used by the driver.

Definition at line 427 of file `Enet_Ip_Types.h`.

7.2.2.6 struct Enet_Ip_TimeStampType

Defines the TimeStamp.

Definition at line 444 of file `Enet_Ip_Types.h`.

Data Fields

- `uint32` [nanoseconds](#)
- `uint32` [seconds](#)
- `uint16` [secondsHi](#)

7.2.2.6.1 Field Documentation

7.2.2.6.1.1 nanoseconds `uint32 nanoseconds`

Nanoseconds part of the time.

Definition at line 446 of file `Enet_Ip_Types.h`.

7.2.2.6.1.2 seconds `uint32 seconds`

The 32 least significant bits of the 48 bits seconds part of the time.

Definition at line 447 of file `Enet_Ip_Types.h`.

7.2.2.6.1.3 secondsHi `uint16 secondsHi`

The 16 most significant bits of the 48 bit seconds part of the time.

Definition at line 448 of file `Enet_Ip_Types.h`.

7.2.2.7 struct `Enet_Ip_RxInfoType`

Enhanced information related to a receive buffer descriptor.

Definition at line 455 of file `Enet_Ip_Types.h`.

Data Fields

- `uint32 ErrMask`
- `uint16 Checksum`
- `uint8 VlanPrio`
- `uint8 HeaderLen`
- `uint16 PktLen`
- `uint8 Proto`
- `boolean Unicast`
- `boolean Ipv6`
- `boolean Ipv4Frag`
- `Enet_Ip_TimeStampType TimeStamp`

7.2.2.7.1 Field Documentation

7.2.2.7.1.1 ErrMask `uint32 ErrMask`

Mask of ENET_RX_ENH_ERR_*, representing the errors that occurred.

Definition at line 456 of file Enet_Ip_Types.h.

7.2.2.7.1.2 Checksum `uint16 Checksum`

One's complement sum of the payload section of the IP frame.

Definition at line 457 of file Enet_Ip_Types.h.

7.2.2.7.1.3 VlanPrio `uint8 VlanPrio`

Frame priority level. Valid values are from 0 (best effort) to 7 (highest).

Definition at line 458 of file Enet_Ip_Types.h.

7.2.2.7.1.4 HeaderLen `uint8 HeaderLen`

Number of 32-bit words found within the IP and its following protocol headers.

Definition at line 459 of file Enet_Ip_Types.h.

7.2.2.7.1.5 PktLen `uint16 PktLen`

Byte length of the received packet that was transferred to system memory (including CRC).

Definition at line 460 of file Enet_Ip_Types.h.

7.2.2.7.1.6 Proto `uint8 Proto`

The 8-bit protocol field found within the IP header of the frame.

Definition at line 461 of file Enet_Ip_Types.h.

7.2.2.7.1.7 **Unicast** `boolean Unicast`

Indicates that the frame is an unicast frame.

Definition at line 462 of file `Enet_Ip_Types.h`.

7.2.2.7.1.8 **Ipv6** `boolean Ipv6`

Indicates that the frame is an IPv6 frame.

Definition at line 463 of file `Enet_Ip_Types.h`.

7.2.2.7.1.9 **Ipv4Frag** `boolean Ipv4Frag`

Indicates that the frame is an IPv4 fragment frame.

Definition at line 464 of file `Enet_Ip_Types.h`.

7.2.2.7.1.10 **TimeStamp** `Enet_Ip_TimeStampType TimeStamp`

TimeStamp of the received frame.

Definition at line 465 of file `Enet_Ip_Types.h`.

7.2.2.8 **struct Enet_Ip_TxInfoType**

Enhanced information related to a transmit buffer descriptor.

Definition at line 472 of file `Enet_Ip_Types.h`.

Data Fields

- `uint32 ErrMask`
- `Enet_Ip_TimeStampType TimeStamp`

7.2.2.8.1 Field Documentation

7.2.2.8.1.1 ErrMask `uint32 ErrMask`

Mask of ENET_TX_ENH_ERR_*, representing the errors that occurred.

Definition at line 473 of file Enet_Ip_Types.h.

7.2.2.8.1.2 TimeStamp `Enet_Ip_TimeStampType TimeStamp`

TimeStamp of the transmitted frame.

Definition at line 474 of file Enet_Ip_Types.h.

7.2.2.9 struct Enet_Ip_TimerConfigType

Configuration of the adjustable timer.

Definition at line 481 of file Enet_Ip_Types.h.

Data Fields

- `uint32` [TimerPeriod](#)
- `uint8` [TimerInc](#)
- `uint32` [CorrectionPeriod](#)
- `uint8` [CorrectionInc](#)

7.2.2.9.1 Field Documentation**7.2.2.9.1.1 TimerPeriod** `uint32 TimerPeriod`

When the timer reaches this value, the period event occurs and the timer restarts.

Definition at line 482 of file Enet_Ip_Types.h.

7.2.2.9.1.2 TimerInc `uint8 TimerInc`

The timer increments by this amount each clock cycle.

Definition at line 483 of file Enet_Ip_Types.h.

7.2.2.9.1.3 CorrectionPeriod `uint32 CorrectionPeriod`

Defines after how many timer clock cycles the correction counter should be reset and triggers a correction increment on the timer.

Definition at line 484 of file `Enet_Ip_Types.h`.

7.2.2.9.1.4 CorrectionInc `uint8 CorrectionInc`

This value is added every time the correction timer expires.

Definition at line 486 of file `Enet_Ip_Types.h`.

7.2.2.10 struct `Enet_Ip_TimerChannelConfigType`

Configuration structure of the timer channels.

Definition at line 517 of file `Enet_Ip_Types.h`.

Data Fields

- [Enet_Ip_TimerChannelModeType Mode](#)
- boolean [IrqEnable](#)
- boolean [DmaEnable](#)

7.2.2.10.1 Field Documentation

7.2.2.10.1.1 Mode [Enet_Ip_TimerChannelModeType Mode](#)

The mode in which the timer channel shall be configured.

Definition at line 518 of file `Enet_Ip_Types.h`.

7.2.2.10.1.2 IrqEnable `boolean IrqEnable`

Enable timer interrupt for this channel. Note: When enabling a channel interrupt, the timer interrupt must also be enabled.

Definition at line 519 of file `Enet_Ip_Types.h`.

7.2.2.10.1.3 DmaEnable `boolean DmaEnable`

Enable DMA request for this channel.

Definition at line 521 of file `Enet_Ip_Types.h`.

7.2.2.11 struct Enet_Ip_TxOptionsType

Transmit options for a particular frame.

Definition at line 552 of file `Enet_Ip_Types.h`.

Data Fields

- `boolean` [NoCRC](#)
- `boolean` [NoInt](#)
- [Enet_Ip_ChecksumInsControlType](#) `ChecksumIns`

7.2.2.11.1 Field Documentation**7.2.2.11.1.1 NoCRC** `boolean NoCRC`

Do not append CRC. It will be provided by the application.

Definition at line 553 of file `Enet_Ip_Types.h`.

7.2.2.11.1.2 NoInt `boolean NoInt`

Do not generate a transmit interrupt.

Definition at line 554 of file `Enet_Ip_Types.h`.

7.2.2.11.1.3 ChecksumIns [Enet_Ip_ChecksumInsControlType](#) `ChecksumIns`

Controls the IP checksum insertion for Tx packet.

Definition at line 555 of file `Enet_Ip_Types.h`.

7.2.3 Types Reference

7.2.3.1 Enet_Ip_CallbackType

```
typedef void(* Enet_Ip_CallbackType) (uint8 Instance, Enet_Ip_EventType Event)
```

Callback function invoked when a general event is encountered.

Definition at line 217 of file Enet_Ip_Types.h.

7.2.3.2 Enet_Ip_ChCallbackType

```
typedef void(* Enet_Ip_ChCallbackType) (uint8 Instance, uint8 Ring)
```

Callback function invoked when a channel event is encountered.

Definition at line 222 of file Enet_Ip_Types.h.

7.2.3.3 Enet_Ip_TimerCallbackType

```
typedef void(* Enet_Ip_TimerCallbackType) (uint8 Instance)
```

Callback function invoked when a timer event is encountered. The channels parameter represents a mask of the channels on which an event occurred.

Implements : Enet_Ip_TimerCallbackType_class

Definition at line 230 of file Enet_Ip_Types.h.

7.2.4 Enum Reference

7.2.4.1 Enet_Ip_StatusType

```
enum Enet_Ip_StatusType
```

Return values.

Enumerator

ENET_STATUS_SUCCESS	Generic operation success status
ENET_STATUS_ERROR	Generic operation failure status
ENET_STATUS_BUSY	Generic operation busy status
ENET_STATUS_TIMEOUT	Generic operation timeout status
ENET_STATUS_UNSUPPORTED	Generic operation unsupported status
ENET_STATUS_RX_QUEUE_EMPTY	There is no available frame in the receive queue
ENET STATUS TX QUEUE FULL	There is no available space for the frame in the transmit queue
ENET_STATUS_BUFF_NOT_FOUND	The specified buffer was not found in the queue
ENET STATUS TX_BUFF_BUSY	All transmit TX buffers are busy

Definition at line 108 of file Enet_Ip_Types.h.

7.2.4.2 Enet_Ip_EventType

enum `Enet_Ip_EventType`

Event specifier for the callback function.

Definition at line 156 of file Enet_Ip_Types.h.

7.2.4.3 Enet_Ip_TxSpecialConfigType

enum `Enet_Ip_TxSpecialConfigType`

Special transmit control configurations.

Definition at line 184 of file Enet_Ip_Types.h.

7.2.4.4 Enet_Ip_TxAcceleratorType

enum `Enet_Ip_TxAcceleratorType`

Transmit accelerator configurations.

Definition at line 194 of file Enet_Ip_Types.h.

7.2.4.5 Enet_Ip_RxAcceleratorType

enum `Enet_Ip_RxAcceleratorType`

Receive accelerator configurations.

Definition at line 205 of file Enet_Ip_Types.h.

7.2.4.6 Enet_Ip_CounterType

enum `Enet_Ip_CounterType`

Statistics counters enumeration.

Definition at line 238 of file Enet_Ip_Types.h.

7.2.4.7 Enet_Ip_MiiSpeedType

enum `Enet_Ip_MiiSpeedType`

Media Independent Interface speed selection.

Enumerator

ENET_MII_SPEED_10M	Speed 10 Mbps.
ENET_MII_SPEED_100M	Speed 100 Mbps.

Definition at line 301 of file Enet_Ip_Types.h.

7.2.4.8 Enet_Ip_PowerStateType

enum `Enet_Ip_PowerStateType`

Controller power state indication.

Enumerator

ENET_PSTATE_INACTIVE	Controller is powered on and Transmitter/Receiver are disabled
ENET_PSTATE_ACTIVE	Controller is powered on and Transmitter/Receiver are enabled
ENET_PSTATE_SLEEP	Controller is sleeping (WoL or LPI) and waiting for wake-up packet
ENET_PSTATE_RESET	Controller is resetting its internal logic and registers

Definition at line 333 of file Enet_Ip_Types.h.

7.2.4.9 Enet_Ip_SysTimeCorrOffsetType

enum `Enet_Ip_SysTimeCorrOffsetType`

System time correction offset.

Enumerator

ENET_SYS_TIME_CORR_POSITIVE	Correction value is added to the system time.
ENET_SYS_TIME_CORR_NEGATIVE	Correction value is subtracted from the system time.

Definition at line 434 of file Enet_Ip_Types.h.

7.2.4.10 Enet_Ip_TimerChannelModeType

enum `Enet_Ip_TimerChannelModeType`

Timer channels configuration modes.

Enumerator

ENET_TIMER_OC_SOFTWARE	Timer Channel is configured for Output Compare - software only.
------------------------	---

Definition at line 493 of file Enet_Ip_Types.h.

7.2.4.11 Enet_Ip_ChecksumInsControlType

enum [Enet_Ip_ChecksumInsControlType](#)

Checksum insertion control.

Enumerator

ENET_CHECKSUM_INSERTION_DISABLE	Checksum insertion is disabled.
ENET_CHECKSUM_INSERTION_IP	IP header checksum calculation and insertion are enabled.
ENET_CHECKSUM_INSERTION_PROTO	Insert protocol specific checksum.
ENET_CHECKSUM_INSERTION_IP_PROTO	Both Ip header and protocol are enabled

Definition at line 529 of file Enet_Ip_Types.h.

7.2.4.12 Enet_Ip_MmfrOpType

enum [Enet_Ip_MmfrOpType](#)

Management Frame operation type.

Definition at line 540 of file Enet_Ip_Types.h.

7.2.4.13 Enet_Ip_FifoThresholdType

enum [Enet_Ip_FifoThresholdType](#)

Available FIFO threshold levels.

Enumerator

ENET_RX_SECTION_FULL	Value, in 64-bit words, of the receive FIFO section full threshold. When the FIFO level reaches the value in this field, data is available in the Receive FIFO.
ENET_RX_SECTION_EMPTY	Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued.
ENET_RX_ALMOST_EMPTY	Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches this level and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read. A minimum value of 4 should be set.
ENET_RX_ALMOST_FULL	Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. A minimum value of 4 should be set.
ENET_TX_SECTION_EMPTY	Value, in 64-bit words, of the transmit FIFO section empty threshold. When the FIFO level reaches this value, a MAC status signal indicates that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow.
ENET_TX_ALMOST_EMPTY	Value, in 64-bit words, of the transmit FIFO almost empty threshold. When the FIFO level reaches this level and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication.
ENET_TX_ALMOST_FULL	Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required. When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication.

Definition at line 562 of file Enet_Ip_Types.h.

7.2.5 Function Reference

7.2.5.1 Enet_Ip_Init()

```
void Enet_Ip_Init (
    uint8 Instance,
    const Enet_CtrlConfigType * Config )
```

Initializes the ENET module.

This function initializes and enables the ENET module, configuring receive and transmit control settings, the receive and transmit descriptors rings, and the MAC physical address.

Note: All untagged packets will be received by ring 0 (zero). VLAN tagged packets are routed depending on the VLAN Tag Priority field according to the provided configuration.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Config</i>	The module configuration structure

7.2.5.2 Enet_Ip_GetPowerState()

```
Enet_Ip_PowerStateType Enet_Ip_GetPowerState (  
    uint8 Instance )
```

Gets the current power state of the ENET module.

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

Returns

Enet_Ip_PowerStateType The power state of the controller

7.2.5.3 Enet_Ip_Deinit()

```
void Enet_Ip_Deinit (  
    uint8 Instance )
```

Deinitializes the ENET module.

This function disables the interrupts and then disables the ENET module.

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.4 Enet_Ip_EnableController()

```
void Enet_Ip_EnableController (  
    uint8 Instance )
```

Enables all configured transmit and receive buffers and then enables the controller.

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.5 Enet_Ip_RecoverFilter()

```
void Enet_Ip_RecoverFilter (
    uint8 Instance )
```

Recover the hash table filtering caused by add add/remove open/close to hash table.

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.6 Enet_Ip_DisableController()

```
void Enet_Ip_DisableController (
    uint8 Instance )
```

Disables the controller and resets all the configured transmit and receive buffers.

Warning: This function ignores all pending transmission and reception requests

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.7 Enet_Ip_SetSpeed()

```
void Enet_Ip_SetSpeed (
    uint8 Instance,
    Enet_Ip_MiiSpeedType Speed )
```

Sets the speed of the MII interface.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Speed</i>	MII speed

7.2.5.8 Enet_Ip_GetTxBuff()

```
Enet_Ip_StatusType Enet_Ip_GetTxBuff (
    uint8 Instance,
    uint8 Ring,
    Enet_Ip_BufferType * Buff,
    uint16 * BuffId )
```

Provides a transmit buffer to be used by the application for transmission.

This function provides an internal buffer which can further be used by the application to store the transmit data.

Note: The buffer will be marked as locked and won't be released until after a call to Enet_Ip_GetTransmitStatus for the same buffer returns ENET_STATUS_SUCCESS.

Important: The driver does not ensure synchronization between different threads trying to get a buffer at the same time. This synchronization shall be implemented by the application.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Ring</i>	Ring number
in, out	<i>Buff</i>	In: Buffer containing the desired length Out: Buffer containing the granted length or available length in case of overflow.
out	<i>BuffId</i>	Index of the buffer (descriptor) within the ring. If this information is not needed, this parameter should be NULL_PTR.

Return values

<i>ENET_STATUS_SUCCESS</i>	The buffer has been successfully locked.
<i>ENET_STATUS_TX_BUFF_BUSY</i>	All buffers are currently in use.
<i>ENET_STATUS_TX_BUFF_OVERFLOW</i>	The requested buffer length cannot be granted.

7.2.5.9 Enet_Ip_SendFrame()

```
Enet_Ip_StatusType Enet_Ip_SendFrame (
    uint8 Instance,
    uint8 Ring,
    const Enet_Ip_BufferType * Buff,
    const Enet_Ip_TxOptionsType * Options )
```

Sends an Ethernet frame.

This function sends an Ethernet frame, contained in the buffer received as parameter.

Note: Since the transmission of the frame is not complete when this function returns, the application must not change/alter/re-use the provided buffer until after a call to `Enet_Ip_GetTransmitStatus` for the same buffer returns `ENET_STATUS_SUCCESS`.

Important: The driver does not ensure synchronization between different threads trying to send a frame at the same time. This synchronization shall be implemented by the application.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Ring</i>	The ring number
in	<i>Buff</i>	The buffer containing the frame
in	<i>Options</i>	Configuration options applicable to this buffer's transmission only. Can be <code>NULL_PTR</code> , if no special option is required.

Return values

<i>ENET_STATUS_SUCCESS</i>	The frame was successfully enqueued for transmission.
<i>ENET_STATUS_TX_QUEUE_FULL</i>	There is no available space for the frame in the queue.

7.2.5.10 Enet_Ip_ReadFrame()

```
Enet_Ip_StatusType Enet_Ip_ReadFrame (
    uint8 Instance,
    uint8 Ring,
    Enet_Ip_BufferType * Buff,
    Enet_Ip_RxInfoType * Info )
```

Reads a received Ethernet frame.

This function reads the first received Ethernet frame in the Rx queue. The buffer received as parameter will be updated by the driver and the `.data` field will point to a memory area containing the frame data.

Note: Once the application finished processing the buffer, it could be reused by the driver for further receptions by invoking `Enet_Ip_ProvideRxBuff`.

Important: The driver does not ensure synchronization between different threads trying to read a frame at the same time. This synchronization shall be implemented by the application.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Ring</i>	The ring number
out	<i>Buff</i>	The buffer containing the frame
out	<i>Info</i>	Enhanced information related to the data contained by this receive buffer. If this information is not needed, this parameter should be <code>NULL_PTR</code> .

Return values

<i>ENET_STATUS_SUCCESS</i>	A frame was successfully read.
<i>ENET_STATUS_RX_QUEUE_EMPTY</i>	There is no available frame in the queue.

7.2.5.11 Enet_Ip_ProvideRxBuff()

```
void Enet_Ip_ProvideRxBuff (
    uint8 Instance,
    uint8 Ring,
    const Enet_Ip_BufferType * Buff )
```

Provides a receive buffer to be used by the driver for reception.

This function provides a buffer which can further be used by the reception mechanism in order to store the received data.

Note: The application can either provide a buffer previously obtained in a `Enet_Ip_ReadFrame` call (when it is no longer needed after being fully processed), or allocate a new buffer. The former approach is recommended as it has a simpler usage model and re-uses the same initial memory range for the entire driver lifetime operation. The later approach could provide more flexibility, but since it involves constant memory free/alloc operations it is only recommended with an efficient pool-based memory allocator.

Important: The driver does not ensure synchronization between different threads trying to provide a buffer at the same time. This synchronization shall be implemented by the application.

Important: The application is responsible for providing one Rx buffer for every frame it receives, otherwise the reception ring can fill-up, affecting further reception.

Usage example:

```
stat = Enet_Ip_ReadFrame(INST_ENET0, 0U, &rxBuff);

if (stat == ENET_STATUS_SUCCESS) { process_buffer(&rxBuff); Enet_Ip_ProvideRxBuff(INST_ENET0, 0U,
&rxBuff); }
```

Parameters

in	<i>Instance</i>	Instance number
in	<i>Ring</i>	The ring number
in	<i>Buff</i>	The buffer to be added to the reception ring

7.2.5.12 Enet_Ip_IsFrameAvailable()

```
boolean Enet_Ip_IsFrameAvailable (
    uint8 Instance,
    uint8 Ring )
```

Checks if there are more frames available in the given queue.

This function takes a peek at the given Rx queue to check if there are more Ethernet frames to be received. Its intended usage is to provide this information without also extracting the frame as "Enet_Ip_ReadFrame".

Parameters

in	<i>Instance</i>	Instance number
in	<i>Ring</i>	The ring number

Return values

<i>TRUE</i>	There is an available frame in the queue.
<i>FALSE</i>	There is no available frame in the queue.

7.2.5.13 Enet_Ip_GetTransmitStatus()

```
Enet_Ip_StatusType Enet_Ip_GetTransmitStatus (
    uint8 Instance,
    uint8 Ring,
    const Enet_Ip_BufferType * Buff,
    Enet_Ip_TxInfoType * Info )
```

Checks if the transmission of a buffer is complete.

This function checks if the transmission of the given buffer is complete.

Note: If the buffer is not found in the Tx ring, the function will return ENET_STATUS_BUFF_NOT_FOUND.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Ring</i>	The ring number
in	<i>Buff</i>	The transmit buffer for which the status shall be checked
out	<i>Info</i>	Extended information related to the buffer. If this information is not needed, this parameter should be NULL_PTR.

Return values

<i>ENET_STATUS_BUSY</i>	The frame is still enqueued for transmission.
<i>ENET_STATUS_BUFF_NOT_FOUND</i>	The buffer was not found in the Tx queue.
<i>ENET_STATUS_SUCCESS</i>	Otherwise.

7.2.5.14 Enet_Ip_GetCounter()

```
uint32 Enet_Ip_GetCounter (
    uint8 Instance,
    Enet_Ip_CounterType Counter )
```

Gets statistics from the specified counter.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Counter</i>	The counter to be read

Returns

The value of the requested counter

7.2.5.15 Enet_Ip_ConfigCounters()

```
void Enet_Ip_ConfigCounters (
    uint8 Instance,
    boolean Enable )
```

Enables/Disables the MIB counters.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Enable</i>	The MIB logic is enabled(TRUE) else disabled(FALSE)

Returns

The Value of the requested counter

7.2.5.16 Enet_Ip_EnableMDIO()

```
void Enet_Ip_EnableMDIO (
    uint8 Instance,
    boolean MiiPreambleDisabled,
    uint32 ModuleClk )
```

Enables the MDIO interface.

Parameters

in	<i>Instance</i>	Instance number
in	<i>MiiPreambleDisabled</i>	Enables/disables prepending a preamble to the MII management frame.
in	<i>ModuleClk</i>	Clock frequency provide MDC.

7.2.5.17 Enet_Ip_MDIORead()

```
Enet_Ip_StatusType Enet_Ip_MDIORead (
    uint8 Instance,
    uint8 PhyAddr,
    uint8 PhyReg,
    uint16 * Data,
    uint32 TimeoutMs )
```

Reads the selected register of the PHY.

Parameters

in	<i>Instance</i>	Instance number
in	<i>PhyAddr</i>	PHY device address
in	<i>PhyReg</i>	PHY register address
out	<i>Data</i>	Data read from the PHY
in	<i>TimeoutMs</i>	Timeout for the read operation (in milliseconds)

Return values

<i>ENET_STATUS_SUCCESS</i>	The operation completed successfully.
<i>ENET_STATUS_TIMEOUT</i>	The specified timeout expired before completing the operation.

7.2.5.18 Enet_Ip_MDIOWrite()

```
Enet_Ip_StatusType Enet_Ip_MDIOWrite (
```

```
uint8 Instance,
uint8 PhyAddr,
uint8 PhyReg,
uint16 Data,
uint32 TimeoutMs )
```

Writes the selected register of the PHY.

Parameters

in	<i>Instance</i>	Instance number
in	<i>PhyAddr</i>	PHY device address
in	<i>PhyReg</i>	PHY register address
in	<i>Data</i>	Data to be written in the specified register of the PHY
in	<i>TimeoutMs</i>	Timeout for the write operation (in milliseconds)

Return values

<i>ENET_STATUS_SUCCESS</i>	The operation completed successfully.
<i>ENET_STATUS_TIMEOUT</i>	The specified timeout expired before completing the operation.

7.2.5.19 Enet_Ip_MDIOReadMMD()

```
Enet_Ip_StatusType Enet_Ip_MDIOReadMMD (
    uint8 Instance,
    uint8 PhyAddr,
    uint8 Mmd,
    uint16 PhyReg,
    uint16 * Data,
    uint32 TimeoutMs )
```

Reads a register of the specified MMD in a PHY device.

Parameters

in	<i>Instance</i>	Instance number
in	<i>PhyAddr</i>	PHY device address
in	<i>Mmd</i>	The MMD index of the target register
in	<i>PhyReg</i>	PHY register address
out	<i>Data</i>	Data read from the PHY
in	<i>TimeoutMs</i>	Timeout for the read operation (in milliseconds)

Return values

<i>ENET_STATUS_SUCCESS</i>	The operation completed successfully.
<i>ENET_STATUS_TIMEOUT</i>	The specified timeout expired before completing the operation.

7.2.5.20 Enet_Ip_MDIOWriteMMD()

```
Enet_Ip_StatusType Enet_Ip_MDIOWriteMMD (
    uint8 Instance,
    uint8 PhyAddr,
    uint8 Mmd,
    uint16 PhyReg,
    uint16 Data,
    uint32 TimeoutMs )
```

Writes a register of the specified MMD in a PHY device.

Parameters

in	<i>Instance</i>	Instance number
in	<i>PhyAddr</i>	PHY device address
in	<i>Mmd</i>	The MMD index of the target register
in	<i>PhyReg</i>	PHY register address
in	<i>Data</i>	Data to be written in the specified register of the PHY
in	<i>TimeoutMs</i>	Timeout for the write operation (in milliseconds)

Return values

<i>ENET_STATUS_SUCCESS</i>	The operation completed successfully.
<i>ENET_STATUS_TIMEOUT</i>	The specified timeout expired before completing the operation.

7.2.5.21 Enet_Ip_SetMacAddr()

```
void Enet_Ip_SetMacAddr (
    uint8 Instance,
    const uint8 * MacAddr )
```

Configures the physical address of the MAC.

Parameters

in	<i>Instance</i>	Instance number
in	<i>MacAddr</i>	The MAC address to be configured

7.2.5.22 Enet_Ip_GetMacAddr()

```
void Enet_Ip_GetMacAddr (
    uint8 Instance,
    uint8 * MacAddr )
```

Gets the physical address of the MAC.

Parameters

in	<i>Instance</i>	Instance number
out	<i>MacAddr</i>	The physical address of the MAC

7.2.5.23 Enet_Ip_SetBroadcastForwardAll()

```
void Enet_Ip_SetBroadcastForwardAll (
    uint8 Instance,
    boolean Enable )
```

Enables/Disables forwarding of the broadcast traffic.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Enable</i>	If true, the application will receive all the broadcast traffic; if false, it stops forwarding this kind of traffic.

7.2.5.24 Enet_Ip_SetMulticastForwardAll()

```
void Enet_Ip_SetMulticastForwardAll (
    uint8 Instance,
    boolean Enable )
```

Enables/Disables forwarding of the multicast traffic, irrespective of the destination MAC address.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Enable</i>	If true, the application will receive all the multicast traffic; if false, it stops forwarding this kind of traffic.

7.2.5.25 Enet_Ip_AddDstAddrToHashFilter()

```
void Enet_Ip_AddDstAddrToHashFilter (
    uint8 Instance,
    const uint8 * MacAddr )
```

Adds a hardware address to the hash filter. The destination address of an incoming packet is passed through CRC logic and then compared to the entries in the hash table.

Parameters

in	<i>Instance</i>	Instance number
in	<i>MacAddr</i>	The physical address

7.2.5.26 Enet_Ip_RemoveDstAddrFromHashFilter()

```
void Enet_Ip_RemoveDstAddrFromHashFilter (
    uint8 Instance,
    const uint8 * MacAddr )
```

Removes a hardware address from the hash filter. The destination address of an incoming packet is passed through CRC logic and then compared to the entries in the hash table.

Parameters

in	<i>Instance</i>	Instance number
in	<i>MacAddr</i>	The physical address

7.2.5.27 Enet_Ip_SetSysTimeCorr()

```
void Enet_Ip_SetSysTimeCorr (
    uint8 Instance,
```

```
Enet_Ip_SysTimeCorrOffsetType Offset,
Enet_Ip_TimeStampType DiffTimeStamp )
```

Set system time correction.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Offset</i>	Chooses between negative and positive correction
in	<i>TimeStamp</i>	Offset between time stamp grandmaster and time stamp by local clock.

7.2.5.28 Enet_Ip_TimerSetCorrection()

```
void Enet_Ip_TimerSetCorrection (
    uint8 Instance,
    uint32 CorrectValue,
    uint32 CorrectPeriod )
```

Sets the timer correction period and correction increment value.

Parameters

in	<i>Instance</i>	Instance number
in	<i>CorrectValue</i>	Correction timer added.
in	<i>CorrectPeriod</i>	Number of timer clock cycles

7.2.5.29 Enet_Ip_TimerStop()

```
void Enet_Ip_TimerStop (
    uint8 Instance )
```

The timer stop at the current value.

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.30 Enet_Ip_TimerEnableChannel()

```
Enet_Ip_StatusType Enet_Ip_TimerEnableChannel (
```

```
uint8 Instance,
uint8 Channel,
const Enet_Ip_TimerChannelConfigType * Config )
```

Configures and enables a specific timer channel.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Channel</i>	Timer channel
in	<i>Config</i>	the value(mode, interrupt, enable DMA or not) to set for timer channel.

Return values

<i>ENET_STATUS_SUCCESS</i>	The operation completed successfully.
<i>ENET_STATUS_TIMEOUT</i>	The specified timeout expired before completing the operation.

7.2.5.31 Enet_Ip_TimerSetCompare()

```
void Enet_Ip_TimerSetCompare (
    uint8 Instance,
    uint8 Channel,
    uint32 Value )
```

Sets the compare value for a channel configured as output compare.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Channel</i>	Timer channel
in	<i>Value</i>	the value to compare.

7.2.5.32 Enet_Ip_TimerStart()

```
void Enet_Ip_TimerStart (
    uint8 Instance )
```

The timer starts incrementing.

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.33 Enet_Ip_TimerInit()

```
void Enet_Ip_TimerInit (
    uint8 Instance,
    const Enet_Ip_TimerConfigType * TimerConfig )
```

Initializes the adjustable timer using the given configuration and enables it.

Parameters

in	<i>Instance</i>	Instance number
in	<i>TimerConfig</i>	timer value need to initialize.

7.2.5.34 Enet_Ip_TimerSet()

```
void Enet_Ip_TimerSet (
    uint8 Instance,
    Enet_Ip_TimeStampType TimerValue )
```

Sets the value of the timer.

Parameters

in	<i>Instance</i>	Instance number
in	<i>TimerConfig</i>	timer value set.

7.2.5.35 Enet_Ip_TimerGet()

```
void Enet_Ip_TimerGet (
    uint8 Instance,
    Enet_Ip_TimeStampType * TimeStamp )
```

Gets the value of the timer.

Parameters

in	<i>Instance</i>	Instance number
in	<i>TimerConfig</i>	timer value gotten.

7.2.5.36 Enet_Ip_EnableTxStoreAndForward()

```
void Enet_Ip_EnableTxStoreAndForward (
    uint8 Instance )
```

Enable store and forward .

Parameters

in	<i>Instance</i>	Instance number
----	-----------------	-----------------

7.2.5.37 Enet_Ip_SetFIFOThreshold()

```
void Enet_Ip_SetFIFOThreshold (
    uint8 Instance,
    Enet_Ip_FifoThresholdType ThresholdType,
    uint8 ThresholdValue )
```

Sets FIFO threshold levels.

Parameters

in	<i>Instance</i>	Instance number
in	<i>ThresholdType</i>	FIFO threshold types (defined in Enet_Ip_FifoThresholdType) need to update
in	<i>ThresholdValue</i>	FIFO threshold value

7.2.5.38 Enet_Ip_SetTxFIFOWatermark()

```
void Enet_Ip_SetTxFIFOWatermark (
    uint8 Instance,
    uint16 Watermark )
```

Sets the transmit FIFO watermark.

Parameters

in	<i>Instance</i>	Instance number
in	<i>Watermark</i>	Number of bytes written to the transmit FIFO before transmission.



Chapter 8

Data Structure Documentation

8.1 Enet_Ip_ManagementInfo Struct Reference

Management Frame information.

```
#include <Enet_Ip_Hw_Access.h>
```

8.1.1 Detailed Description

Management Frame information.

Definition at line 140 of file Enet_Ip_Hw_Access.h.

How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2022 NXP B.V.

