VECTOR

MICROSAR Ethernet State Manager

Technical Reference

AUTOSAR 4.1.1

Version 4.0.0

Authors	Alex Lunkenheimer, Marc Weber, Mark Harsch, Samridhi Langer
Status	Released

Document Information

History

Author	Date	Version	Remarks
Alex Lunkenheimer	2009-03-27	1.0.0	Creation of document
Alex Lunkenheimer	2009-10-05	2.0.0	Tool based configuration
Alex Lunkenheimer	2009-11-05	2.1.0	Tool based configuration
Alex Lunkenheimer	2012-02-22	2.1.1	Tool based configuration
Marc Weber	2012-04-16	2.2.0	Socket Adaptor call-back
Alex Lunkenheimer	2012-05-07	2.2.1	Inhibit Com Mode Request, Manual configuration removed
Alex Lunkenheimer	2012-05-08	2.2.2	Inhibit Com Mode Request Integration hints added
Mark Harsch	2013-04-18	2.3.0	Auto Mode Change Retry description added
Mark Harsch	2013-06-17	2.3.1	Topic adaption
Mark Harsch	2013-08-29	3.0.0	EthSM according to AUTOSAR 4.1.1
Mark Harsch	2014-07-15	3.1.0	- Updated AUTOSAR architecture figure - BSW changes due to support of AUTOSAR CONC_600_SwitchConfiguration
Mark Harsch	2015-07-20	3.2.0	ESCAN00080851: Rename Technical Reference according to SPEC-63403
Mark Harsch	2015-12-21	3.2.1	Review finding integration
Samridhi Langer	2018-07-20	4.0.0	Removed dummy mode, updated the SID and error ID’s

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_EthernetStateManager.pdf	2.0.0
[2]	AUTOSAR	AUTOSAR_SWS_DET.pdf	3.2.0
[3]	AUTOSAR	AUTOSAR_SWS_DEM.pdf	4.2.0
[4]	AUTOSAR	AUTOSAR_BasicSoftwareModules.pdf	1.0.0
[5]	AUTOSAR	AUTOSAR_SWS_EthernetInterface.pdf	2.0.0
[6]	AUTOSAR	AUTOSAR_SWS_ComManager.pdf	4.0.0
[7]	AUTOSAR	AUTOSAR_SWS_BSWModeManager.pdf	1.2.0

Scope of the Document

This technical reference describes the general use of the EthSM basis software. Please refer to your Release Notes to get a detailed description of the platform (host, compiler) your Vector Ethernet Bundle has been configured for.

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

Contents

Illustrations

Tables

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.00.xx	Created
2.00.xx	Vector Coding Rules Applied
2.01.xx	Inhibit Com Mode Request
2.02.xx	Socket Adaptor call-back
2.03.xx	Auto Mode Change Retry
3.00.xx	EthSM BSW according to AUTOSAR 4.1.1
3.01.xx	Support of AUTOSAR CONC_600_SwitchConfiguration
5.00.xx	Refactoring of the internal state machine

Table 1-1 Component history

2 Introduction

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

Supported AUTOSAR Release*:	4.1.1	
Supported Configuration Variants:	pre-compile	
Vendor ID:	ETHSM_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	ETHSM_MODULE_ID	143 decimal (according to ref. [4])

* For the precise AUTOSAR Release 4.1.1 please see the release specific documentation.

The EthSM realizes a software layer between the Communication Manager (ComM) and the Ethernet Interface (EthIf). The EthSM handles the start-up and shutdown of the communication of an Ethernet network. The EthSM maps the modes of its networks to the modes of the ComM networks and causes the necessary actions to change the EthSM modes to those requested by the ComM.

2.1 Architecture Overview

The following figure shows where the EthSM is located in the AUTOSAR architecture.

The EthSM uses and provides multiple Application Programming Interfaces (API) to other BSW Modules. Figure 2-1 shows the adjacent modules of EthSM. These interfaces are described in chapter 5.

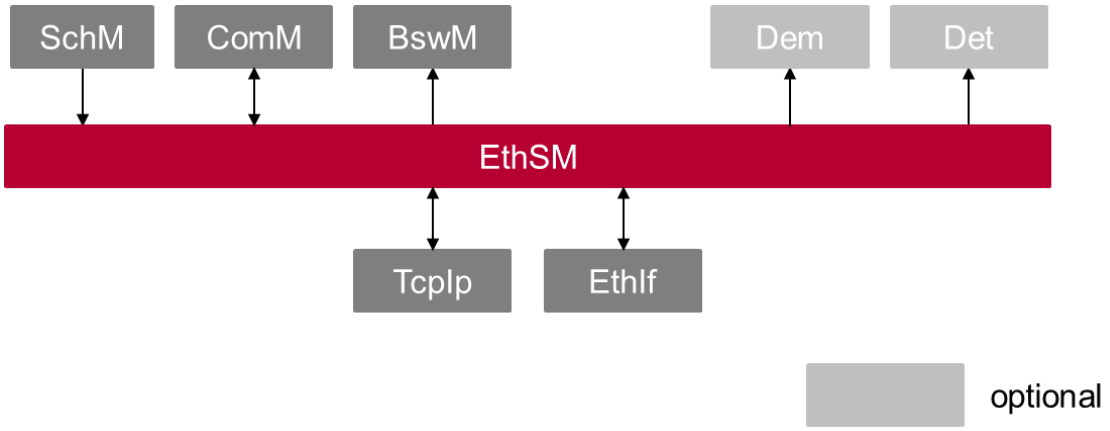

Figure 2-1 Interfaces to adjacent modules of the EthSM

Applications do not access the services of the BSW modules directly. They use the service ports provided by the ComM via the RTE to influence the communication mode of an Ethernet network.

3 Functional Description

3.1 Features

The features listed in the following tables cover the complete functionality specified for the EthSM.

The AUTOSAR standard functionality is specified in [1], the corresponding features are listed in the tables

> Table 3-1 Supported AUTOSAR standard conform features

> Table 3-2 Not supported AUTOSAR standard conform features

For further information of not supported features see section 3.1.3.

Vector Informatik provides further EthSM functionality beyond the AUTOSAR standard. The corresponding features are listed in section 3.1.2 in the table

> Table 3-3 Features provided beyond the AUTOSAR standard

The following features specified in [1] are supported:

Supported AUTOSAR Standard Conform Features
Request a communication mode for a specific Ethernet network
Retrieve the communication mode of a specific Ethernet network
Retrieve version info
Report development errors to DET
Report production errors to DEM

Table 3-1 Supported AUTOSAR standard conform features

3.1.1 Deviations

The following features specified in [1] are not supported:

Not Supported AUTOSAR Standard Conform Features
Dummy mode

Table 3-2 Not supported AUTOSAR standard conform features

3.1.2 Additions/ Extensions

The following features are provided beyond the AUTOSAR standard:

Features Provided Beyond the AUTOSAR Standard
EthIf API call timeout error handling
Operation without TcpIp module

Table 3-3 Features provided beyond the AUTOSAR standard

3.1.3 Limitations

No limitations exist.

3.2 Initialization

The EthSM is initialized by calling `EthSM_InitMemory` service followed by `EthSM_Init`.

3.3 State Machine

The EthSM is operational after initialization (call of `EthSM_InitMemory` followed by `EthSM_Init`). The communication mode of each Ethernet network is `NO COMMUNICATION`.

The overall state machine of an Ethernet network is shown in the following figure.

3.4 Main Function

The EthSM provides the main function `EthSM_MainFunction`, which has to be called by the Schedule Manager (SchM) periodically.

It processes all pending transitions triggered by the EthSM API.

3.5 Mode Requests

Mode Requests are triggered by the ComM by calling `EthSM_RequestComMode`. The API must be called with the corresponding ComM network handle and the mode, which is requested for the Ethernet Network.

The transition between the communication modes itself is handled by the `EthSM_MainFunction` through calls to EthIf to influence the mode of the abstract EthIf Controller and the underlying Eth-Controller and –Transceiver, to TcpIp to request the TcpIp mode of the specific network and by processing the information received by indications of these two modules.

3.5.1 Establishing communication on an Ethernet Network

In order to be able to communicate on a specific Ethernet Network, a mode change must be requested. This is done by calling the `EthSM_RequestComMode` service with the corresponding Network Handle for this network and the mode `FULL COMMUNICATION`. The service will trigger the transition into `FULL COMMUNICATION`.

3.5.2 Shutdown communication on an Ethernet Network

If there is no need for communication on a specific Ethernet Network anymore, the network can be deactivated. This is done by calling the `EthSM_RequestComMode` service with the corresponding Network Handle for this network and the mode NO COMMUNICATION. The service will trigger the transition into `NO COMMUNICATION`.

3.6 Transceiver Link Indication

The EthSM reacts on a Transceiver Link indication (`EthSM_TrcvLinkStateChg`) if the corresponding network is in an internal state, which allows accepting Transceiver Link indications.

The resulting transitions are processed in the EthSM Main Function.

3.7 TcpIp Mode Indication

The EthSM reacts on TcpIp Mode indications (`EthSM_TcpIpModeIndications`) if the corresponding network is in an internal state, which allows accepting TcpIp Mode indications.

The resulting transitions are processed in the EthSM Main Function.

3.8 Additional Features

The EthSM provides additional features, which can be enabled in the configuration tool.

3.8.1 Ethernet Interface (EthIf) API Timeout

The EthIf API Timeout feature extends the EthSM by the ability to handle errors occurring during transitions between internal states that rely on successful calls to the EthIf API.

It allows the EthSM to interrupt the transitions, which otherwise would endlessly retry to call the EthIf API in each task cycle until it returns successfully. This interruption is realized by introducing a configurable amount of retries to the EthSM. After the retries are exceeded, EthSM reports the error to the DET and/or DEM depending on if they are enabled and continues retrying. If all the retries are used again then the DET and/or DEM are not reported again. To avoid a high calling rate to the EthIf API a time gap between the calls is configurable.

3.9 Error Handling

3.9.1 Development Error Reporting

To report development errors to the DET using the service `Det_ReportError()` as specified in [2], development error reporting must be enabled (i.e. configuration parameter /MICROSAR/EthSM/EthSMGeneral/EthSMDevErrorDetect).

If another module is used for development error reporting, the function prototype for reporting the error can be configured by the integrator, but must have the same signature as the service `Det_ReportError()`.

The reported EthSM ID is 143.

The reported service IDs identify the services which are described in 5.2. The following table presents the service IDs and the related services:

Service ID	Service
0x01	ETHSM_SID_MAIN_FUNCTION
0x02	ETHSM_SID_GET_VERSION_INFO
0x03	ETHSM_SID_GET_CURRENT_INTERNAL_MODE
0x04	ETHSM_SID_GET_CURRENT_COM_MODE
0x05	ETHSM_SID_REQUEST_COM_MODE
0x06	ETHSMSID_TRCV_LINK_STATE_CHG
0x07	ETHSM_SID_INIT
0x08	ETHSM_SID_TCPIP_MODE_INDICATION

Service ID	Service
0x09	ETHSM_SID_CTRL_MODE_INDICATION
0x10	ETHSM_SID_TRCV_MODE_INDICATION

Table 3-4 Service IDs

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

Error Code	Description
0x01	ETHSM_E_INVALID_NETWORK_MODE
0x02	ETHSM_E_UNINIT
0x03	ETHSM_E_PARAM_POINTER
0x04	ETHSM_E_INVALID_NETOWRK_HANDLE
0x05	ETHSM_E_INVALID_TcpIpMode
0x06	ETHSM_E_INVALID_TRCV_LINK_STATE
0x07	ETHSM_E _PARAM_CONTROLLER
0x08	ETHSM_E_PARAM_TRANSCEIVER
0x09	ETHSM_E_ETHIF_TIMEOUT
0x0A	ETHSM_E_STACKTYPE_WRONG
0x0B	ETHSM_E_ALREADY_INITIALIZED
0x0C	ETHSM_E_PARAM_CONFIG

Table 3-5 Errors reported to DET

3.9.2 Production Code Error Reporting

Production Error Reporting is enabled by configuring the corresponding Development Error Manager Events (DEM Events) for each Ethernet network. If events are configured they are reported by calling the service `Dem_ReportErrorStatus()` with the corresponding DEM Event as specified in [3].

If another module is used for production code error reporting, the function prototype for reporting the error can be configured by the integrator, but must have the same signature as the service `Dem_ReportErrorStatus()`.

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

Error Code	Description
ETHSM_E_LINK_DOWN_<X>	Reported on transitions triggered by a Transceiver Link Change indication. (for each Ethernet network the DEM Event is configured for)
ETHSM_E_TIMEOUT_<X>	Reported, if timeouts and retries for EthIf API calls exceed. (for each Ethernet network the DEM Event is configured for)

Table 3-6 Errors reported to DEM

4 Integration

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

4.1 Scope of Delivery

The delivery of the EthSM contains the files which are described in the chapters 4.1.1 and 4.1.2.

4.1.1 Static Files

The source code delivery contains the files listed in Table 4-1.

File Name	Description
`EthSM.c`	Implementation
`EthSM.h`	API declaration
`EthSM_Types.h`	Data types declaration

Table 4-1 Static files (Source Code Delivery)

Do not edit manually
The static files of a Delivery must not be modified.

4.1.2 Dynamic Files

The dynamic files are generated by the configuration tool.

File Name	Description
`EthSM_Cfg.h`	This file contains the generated define switches and data prototypes of EthSM.
`EthSM_Cfg.c`	This file contains the generated data definition and source code of EthSM.

Table 4-2 Generated files

Caution
You should avoid modifying the dynamic files generated by the configuration tool manually.

Instead change the values of parameters or more complex configuration structures within the tool and trigger the generation of the files.

5 API Description

For an interfaces overview please see Figure 2-1.

5.1 Type Definitions

The types defined by the EthSM are described in this chapter.

Type Name	C-Type	Description	Value Range
EthSM_ConfigType	void	Defines the EthSM configuration type	NULL_PTR Pre-compile or link-time configuration
			Pointer Pointer to Post-build configuration
EthSM_NetworkModeStateType	uint8	Defines all possible EthSM states	ETHSM_STATE_OFFLINE EthSM in mode NO COMMUNICATION
			ETHSM_STATE_WAIT_TRCVLINK EthSM in mode NO COMMUNICATION
			ETHSM_STATE_WAIT_ONLINE EthSM in mode NO COMMUNICATION
			ETHSM_STATE_ONLINE EthSM in mode FULL COMMUNICATION
			ETHSM_STATE_ONHOLD EthSM in mode FULL COMMUNICATION
			ETHSM_STATE_WAIT_OFFLINE EthSM in mode FULL COMMUNICATION

Table 5-1 Type definitions

5.2 Services provided by EthSM

5.2.1 EthSM_InitMemory

Prototype	
void **EthSM_InitMemory** (void)	
Parameter	
void	none
Return Code	
void	none
Functional Description	
Initializes global variables.	
Particularities and Limitations	
AUTOSAR extension.	

	**Caution** Has to be called before any other calls to the module.
Pre-Conditions	
none	
Call Context	
Task level	

Table 5-2 EthSM_InitMemory

5.2.2 EthSM_Init

Prototype	
`void EthSM_Init (const EthSM_ConfigType* ConfigPtr)`	
Parameter	
ConfigPtr	Pointer to the Post-build configuration (not used)
Return Code	
void	None
Functional Description	
Initializes the data needed by EthSM for proper operation.	
Particularities and Limitations	
EthSM supports only the PRECOMPILE configuration variant. Therefore, the passed pointer parameter is neither used nor checked by parameter checks.	
	**Caution** Has to be called before usage of the module.
Pre-Conditions	
none	
Call Context	
Task level	

Table 5-3 EthSM_Init

5.2.3 EthSM_MainFunction

Prototype	
`void EthSM_MainFunction (void)`	
Parameter	
void	none
Return Code	
void	none

Functional Description
Main function of the Ethernet State Manager.
Particularities and Limitations
The main function processes the state transitions.
Pre-Conditions
none
Call Context
Task level

Table 5-4 EthSM_MainFunction

5.2.4 EthSM_RequestComMode

Prototype	
`Std_ReturnType` **`EthSM_RequestComMode`** `(NetworkHandleType NetworkHandle, ComM_ModeType Mode)`	
Parameter	
NetworkHandle	Network Handle of the Ethernet network.
Mode	Requested Communication Mode. COMM_NO_COMMUNICATION : Request no communication. COMM_FULL_COMMUNICATION : Request full communication.
Return Code	
Std_ReturnType	> E_OK : Request was called with correct parameters. > E_NOT_OK : Request was called with erroneous parameters.
Functional Description	
Requests a communication mode change for an Ethernet network.	
Particularities and Limitations	
Dependent on the current state of the Ethernet Network and the requested communication mode transitions to other states are triggered and processed in next main function cycle.	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-5 EthSM_RequestComMode

5.2.5 EthSM_GetCurrentComMode

Prototype	
`Std_ReturnType EthSM_GetCurrentComMode (NetworkHandleType NetworkHandle, ETHSM_P2VARAPPLDATA(ComM_ModeType) ModePtr)`	
Parameter	
NetworkHandle	Network Handle of the Ethernet network.
ModePtr	Pointer to store the communication mode value.
Return Code	
Std_ReturnType	> E_OK : Communication mode of network could be retrieved. > E_NOT_OK : Communication mode of network couldn't be retrieved.
Functional Description	
Get the current communication mode for an Ethernet network.	
Particularities and Limitations	
none	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-6 EthSM_GetCurrentComMode

5.2.6 EthSM_GetCurrentInternalMode

Prototype	
`Std_ReturnType EthSM_GetCurrentInternalMode (NetworkHandleType NetworkHandle, ETHSM_P2VARAPPLDATA(EthSM_NetworkModeStateType) ModePtr)`	
Parameter	
NetworkHandle	Network Handle of the Ethernet network.
ModePtr	Pointer to store the internal state value.
Return Code	
Std_ReturnType	> E_OK : Internal state of network could be retrieved. > E_NOT_OK : Internal state of network couldn't be retrieved.
Functional Description	
Get the current internal EthSM state for an Ethernet network.	
Particularities and Limitations	
none	
Pre-Conditions	
none	

Call Context
Task and Interrupt level

Table 5-7 EthSM_GetCurrentInternalMode

5.2.7 EthSM_TrcvLinkStateChg

Prototype	
`Std_ReturnType` **`EthSM_TrcvLinkStateChg`** `(NetworkHandleType NetworkHandle, EthTrcv_LinkStateType TransceiverLinkState)`	
Parameter	
NetworkHandle	Network Handle of the Ethernet network.
TransceiverLinkState	Reported Link State.
Return Code	
Std_ReturnType	> E_OK : Link State change called with correct parameters. > E_NOT_OK : Link State change called with erroneous parameters.
Functional Description	
Called by EthIf to report a Link State change for an Ethernet network.	
Particularities and Limitations	
Dependent on the current state of the Ethernet Network and the reported Link State transitions to other states are triggered and processed in next main function cycle.	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-8 EthSM_TrcvLinkStateChg

5.2.8 EthSM_VTrcvLinkStateChg

Prototype	
`void` **`EthSM_VTrcvLinkStateChg`** `(uint8 CtrlIdx, EthTrcv_LinkStateType TrcvLinkState)`	
Parameter	
CtrlIdx	Index of the EthIf Controller.
TrcvLinkState	Reported Link State.
Return Code	
void	> E_OK : Link State change called with correct parameters. > E_NOT_OK : Link State change called with erroneous parameters.

Functional Description
Called by EthIf to report a Link State change for an EthIf Controller.
Particularities and Limitations
This API wraps the AUTOSAR API EthSM_TrcvLinkStateChg to fit the EthSM according to AUTOSAR 4.1.1 into the Vector IP-Stack. If the EthSM isn't used in a configuration containing the Vector EthIf the API will not be available.
Pre-Conditions
none
Call Context
Task and Interrupt level

Table 5-9 EthSM_VTrcvLinkStateChg

5.2.9 EthSM_TcpIpModeIndication

Prototype	
`Std_ReturnType EthSM_TcpIpModeIndication (NetworkHandleType NetworkHandle, TcpIp_StateType TcpIpState)`	
Parameter	
NetworkHandle	Network Handle of the Ethernet network.
TcpIpState	Reported TcpIp Mode.
Return Code	
Std_ReturnType	> E_OK : TcpIp mode indication called with correct parameters. > E_NOT_OK : TcpIp mode indication called with erroneous parameters.
Functional Description	
Called by TcpIp to report a TcpIp mode change for an Ethernet network.	
Particularities and Limitations	
Dependent on the current state of the Ethernet Network and the reported TcpIp mode transitions to other states are triggered and processed in next main function cycle. If no TcpIp is contained in the configuration the API will not be available. Use Cases for this configuration variant are e.g. AVB use cases.	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-10 EthSM_TcpIpModeIndication

5.2.10 EthSM_VTcpIpModeIndication

Prototype	
`void EthSM_VTcpIpModeIndication (uint8 CtrlIdx, boolean Assigned, ETHSM_P2CONSTAPPLDATA(IpBase_SockAddrType) SockAddrPtr)`	
Parameter	
CtrlIdx	Index of the EthIf Controller.
Assigned	Reported IP assignment.
SockAddrPtr	Pointer to Socket address information.
Return Code	
void	> E_OK : TcpIp mode indication called with correct parameters. > E_NOT_OK : TcpIp mode indication called with erroneous parameters.
Functional Description	
Called by TcpIp to report a TcpIp mode change for an Ethernet network.	
Particularities and Limitations	
This API wraps the AUTOSAR API EthSM_TcpIpModeIndication to fit the EthSM according to ASR 4.1.1 into the Vector IP-Stack. If the EthSM isn't used in a configuration containing the Vector TcpIp the API will not be available.	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-11 EthSM_VTcpIpModeIndication

5.2.11 EthSM_CtrlModeIndication

Prototype	
`void EthSM_CtrlModeIndication (uint8 CtrlIdx, Eth_ModeType CtrlMode)`	
Parameter	
CtrlIdx	Controller Index
CtrlMode	Controller Mode
Return Code	
void	none
Functional Description	
Purpose of this API not specified by AUTOSAR SWS.	

Particularities and Limitations
The API is an empty implementation and not available by default. To enable the API use a User Config File and add #define ETHSM_ENABLE_CTRL_TRCV_IND_API == STD_ON
Pre-Conditions
none
Call Context
Task and Interrupt level

Table 5-12 EthSM_CtrlModeIndication

5.2.12 EthSM_TrcvModeIndication

Prototype	
`void` **`EthSM_TrcvModeIndication`** `(uint8 TrcvIdx, EthTrcv_ModeType TrcvMode)`	
Parameter	
TrcvIdx	Transceiver Index
TrcvMode	Transceiver Mode
Return Code	
void	none
Functional Description	
Purpose of this API not specified by AUTOSAR SWS.	
Particularities and Limitations	
The API is an empty implementation and not available by default. To enable the API use a User Config File and add #define ETHSM_ENABLE_CTRL_TRCV_IND_API == STD_ON	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-13 EthSM_TrcvModeIndication

5.2.13 EthSM_GetVersionInfo

Prototype
`void` **`EthSM_GetVersionInfo`** `(ETHSM_P2VARAPPLDATA(Std_VersionInfoType) VersionInfoPtr)`

Parameter	
VersionInfoPtr	Pointer to store the version info.
Return Code	
void	none
Functional Description	
Get Ethernet State Manager version information.	
Particularities and Limitations	
API is only available when ETHSM_VERSION_INFO_API is switched ON.	
Pre-Conditions	
none	
Call Context	
Task and Interrupt level	

Table 5-14 EthSM_GetVersionInfo

5.3 Services used by EthSM

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

Component	API
ComM	`ComM_BusSM_ModeIndication`
EthIf	`EthIf_ControllerInit`
EthIf	`EthIf_SetControllerMode`
TcpIp	`TcpIp_RequestComMode`
BswM	`BswM_EthSM_CurrentState`
Det (optional)	`Det_ReportError`
Dem (optional)	`Dem_ReportErrorStatus`

Table 5-15 Services used by the EthSM

6 Configuration

The EthSM can be configured with the Vector configuration tool DaVinci Configurator Pro.

6.1 Configuration Variants

The EthSM supports the configuration variants

> VARIANT-PRE-COMPILE

The configuration classes of the EthSM parameters depend on the supported configuration variants. For their definitions please see the EthSM_bswmd.arxml file.

6.2 Configuration Parameters not covered by Generation Tool

The listed Macros allow configuring features, which can't be configured in the configuration tool. A macro must be added to the User Config file, which must be selected in the configuration tool, if one desires to use the related feature.

Macro Name	Value-Type	Description	Value Range
ETHSM_ENABLE_CTRL_TRCV_IND_API	Boolean	Enables/Disables the API > EthSM_CtrlModeIndication > EthSM_TrcvModeIndication By turning on the feature the listed APIs are provided, to avoid compiler errors. The services itself have no effect on the operation of the EthSM but only return E_OK to the caller.	STD_OFF APIs don't exist STD_ON APIs exist

Table 6-1 Macros for manual configuration

7 Glossary and Abbreviations

7.1 Glossary

Term	Description
DaVinci Configurator Pro	Configuration tool to configure MICROSAR components

Table 7-1 Glossary

7.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
BswM	Basic Software Mode Manager
ComM	Communication Manager
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
EthIf	Ethernet Interface
HIS	Hersteller Initiative Software
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
RTE	Runtime Environment
TcpIp	TcpIp Basic Software
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 7-2 Abbreviations

8 Contact

Visit our website for more information on

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

www.vector.com