VECTOR

MICROSAR COM Based Transformer

Technical Reference

Version 1.8.0

Authors	Cornelius Reuss, Sascha Sommer, Katharina Benkert
Status	Released

Document Information

History

Author	Date	Version	Remarks
Cornelius Reuss	2015-07-07	1.0.0	Initial version
Cornelius Reuss	2016-02-26	1.1.0	New Layout
Sascha Sommer	2016-05-17	1.2.0	Version update only
Sascha Sommer	2016-05-17	1.3.0	Version update only
Cornelius Reuss	2016-06-23	1.4.0	Version update only
Katharina Benkert	2016-11-17	1.5.0	Version update only
Bernd Sigle	2017-03-20	1.6.0	Version update only
Bernd Sigle	2017-06-06	1.7.0	Version update only
Bernd Sigle	2017-08-17	1.8.0	Support for AUTOSAR 4.3.0

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_COMBasedTransformer.pdf	4.3.0
[2]	AUTOSAR	AUTOSAR_TR_BSWModuleList.pdf	4.3.0
[3]	AUTOSAR	AUTOSAR_SWS_COM.pdf	4.3.0

Scope of the Document

This technical reference describes the general use of the COM Based Transformer.

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
1.0.0	Initial Creation
1.2.0	Version update only
1.3.0	Version update only
1.4.0	Version update only
1.5.0	MISRA enhancements
1.6.0	Version update only
1.7.0	Version update only
1.8.0	Support of AUTOSAR 4.3.0

Table 1-1 Component history

2 Introduction

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

Supported AUTOSAR Release*:	4	
Supported Configuration Variants:	pre-compile	
Vendor ID:	COMXF_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	COMXF_MODULE_ID	175 decimal (according to ref. [2])

* For the detailed functional specification please also refer to the corresponding AUTOSAR SWS.

The ComXf module provides the functionality to serialize complex data when the target bus system uses a fixed communication matrix.

2.1 Architecture Overview

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

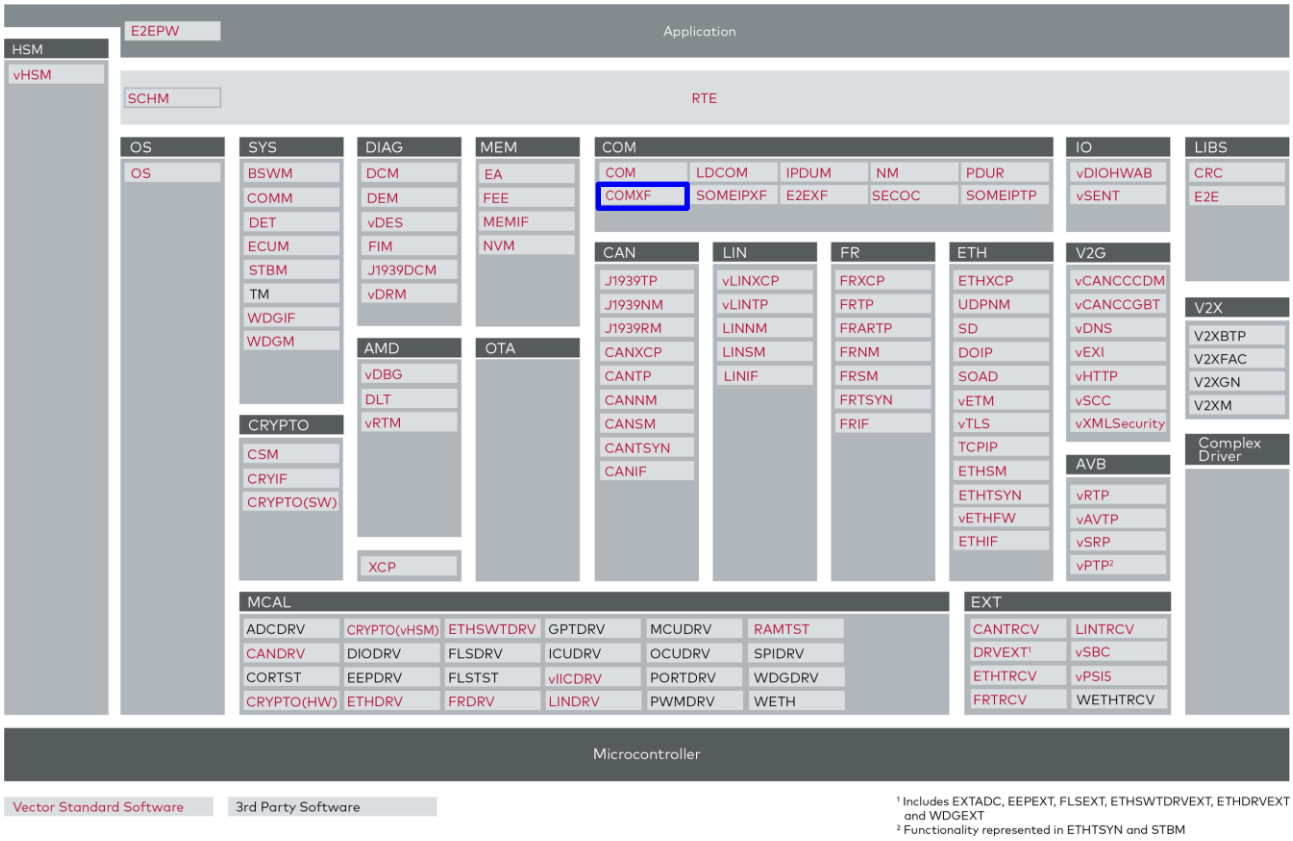

Figure 2-1 AUTOSAR Architecture Overview

3 Functional Description

3.1 Features

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

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

The following features specified in [1] are supported:

Supported AUTOSAR Standard Conform Features
Serialization / Deserialization of complex data for S/R communication.

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
Development error detection.
Postbuild support.

Table 3-2 Not supported AUTOSAR standard conform features

3.2 Initialization

The ComXf does not have to be initialized or deinitialized. Calls to `ComXf_Init()` and `ComXf_DeInit()` can be omitted.

3.3 States

No internal states exist.

3.4 Main Functions

No main function exists because all functionality is performed within the called API.

3.5 Error Handling

3.5.1 Development Error Reporting

No development error reporting is currently supported by ComXf.

3.5.2 Production Code Error Reporting

No production errors are specified for ComXf.

4 Integration

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

4.1 Scope of Delivery

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

4.1.1 Static Files

File Name	Description
-	-

Table 4-1 Static files

4.1.2 Dynamic Files

The dynamic files are generated by the configuration tool DaVinci Configurator.

File Name	Description
ComXf.c	Source file of the ComXf module.
ComXf.h	Main header file which shall be included by modules using the ComXf module.
ComXf_MemMap.h	Template contains ComXf specific part of the memory mapping.
ComXf_Compiler_Cfg.h	Template contains ComXf specific part of the compiler abstraction.
ComXf_rules.mak, ComXf_defs.mak, ComXf_check.mak, ComXf_cfg.mak	Make files according to the AUTOSAR make environment proposal are generated into the mak subdirectory.

Table 4-2 Generated files

5 API Description

5.1 Services provided by ComXf

5.1.1 ComXf_Init

Prototype	
`void` **`ComXf_Init`** `(const ComXf_ConfigType *config)`	
Parameter	
`config`	Pointer to the transformer's configuration data.
Return code	
`void`	none
Functional Description	
Initialization function.	
Particularities and Limitations	
none	
Expected Caller Context	
This function can be called in any context.	

Table 5-1 ComXf_Init

5.1.2 ComXf_DeInit

Prototype	
`void` **`ComXf_DeInit`** `(void)`	
Parameter	
`void`	none
Return code	
`void`	none
Functional Description	
Deinitialization function.	
Particularities and Limitations	
none	
Expected Caller Context	
This function can be called in any context.	

Table 5-2 ComXf_DeInit

5.1.3 ComXf_GetVersionInfo

Prototype	
`void` **`ComXf_GetVersionInfo`** `(Std_VersionInfoType *versionInfo)`	
Parameter	
`versioninfo`	Pointer to where to store the version information of this module.
Return code	
`void`	none
Functional Description	
This API returns version information, vendor ID and AUTOSAR module ID of the called transformer module.	
Particularities and Limitations	
This API is only available if enabled by the configuration parameter XfrmVersionInfoApi.	
Expected Caller Context	
This function can be called in any context.	

Table 5-3 ComXf_GetVersionInfo

5.1.4 ComXf_<transformerId>

Prototype	
`Std_ReturnType` **`ComXf_<transformerId>`** `(uint8 *buffer, uint16 *bufferLength, const <type> *dataElement)`	
Parameter	
`buffer`	Buffer allocated by the RTE, where the transformed data has to be stored by the transformer.
`bufferLength`	Used length of the buffer.
`dataElement`	Data element which shall be transformed.
Return code	
`E_OK`	Serialization successful.
Functional Description	
Serialization of signal group based on COM ECU configuration for S/R communication.	
Particularities and Limitations	
none	
Expected Caller Context	
This function can be called in any context.	

Table 5-4 ComXf_<transformerId>

5.1.5 ComXf_Inv_<transformerId>

Prototype	
`Std_ReturnType` **`ComXf_Inv_<transformerId>`** `(const uint8 *buffer, uint16 bufferLength, <type> *dataElement)`	
Parameter	
`buffer`	Buffer allocated by the RTE, where the serialized data is stored by the Rte.
`bufferLength`	Used length of the buffer.
`dataElement`	Data element which is the result of the transformation and contains the deserialized data element.
Return code	
`E_OK`	Deserialization successful.
Functional Description	
Deserialization of signal group based on COM ECU configuration for S/R communication.	
Particularities and Limitations	
none	
Expected Caller Context	
This function can be called in any context.	

Table 5-5 ComXf_Inv_<transformerId>

6 Configuration

In the ComXf the attributes can be configured with the following tools:

> Configuration in DaVinci Configurator

Currently, only the GetVersionInfo API can be enabled / disabled in the ComXf ECU configuration.
The serialization / deserialization of the signal groups is based on the ECU configuration of the COM module (for details see [3]).

6.1 Configuration Variants

The ComXf supports the configuration variants

> `VARIANT-PRE-COMPILE`

The configuration classes of the ComXf parameters depend on the supported configuration variants. For their definitions please see the `ComXf_bswmd.arxml` file.

6.2 Enabling / Disabling of data transformation

If a signal shall be handled by the ComXf, the parameter “Signal Group Array Access” has to be set in the ECU configuration of the COM module (see Figure 6-1 Enable Data Transformation).

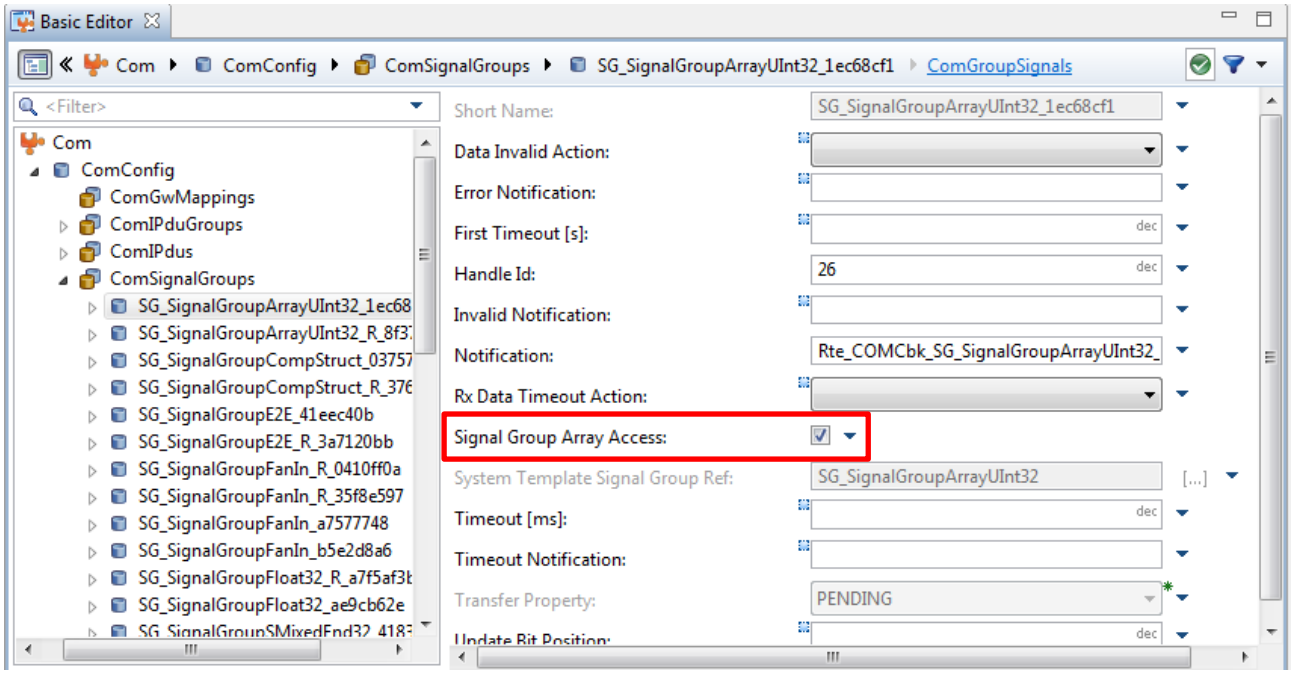

Figure 6-1 Enable Data Transformation

7 Glossary and Abbreviations

7.1 Glossary

Term	Description
DaVinci Configurator	Configuration and generation tool for MICROSAR components

Table 7-1 Glossary

7.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
RTE	Runtime Environment
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