

MICROSAR CRY FORD

Security Access - Technical Reference

Version 1.0

Authors	Markus Schneider, Tobias Finke
Status	Released

Document Information

History

Author	Date	Version	Remarks
Markus Schneider	2015-04-14	1.00.00	Creation
Markus Schneider	2015-08-12	1.00.01	Minor corrections
Tobias Finke	2015-09-30	1.00.02	Fixed description of CryFord_Cfg.h

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_CryptoServiceManager.pdf	1.2.0
[2]	AUTOSAR	AUTOSAR_SWS_DevelopmentErrorTracer.pdf	3.2.0
[3]	AUTOSAR	AUTOSAR_SWS_DiagnosticEventManager.pdf	4.2.0
[4]	AUTOSAR	AUTOSAR_TR_BSWModuleList.pdf	1.6.0
[5]	AUTOSAR	AUTOSAR_SWS_RTE.pdf	3.2.0
[6]	FORD	EESE DIAGNOSTIC APPLICATION SECURITY ALGORITHM	001

Contents

1	Component History	6
2	Introduction.....	7
2.1	Architecture Overview	8
3	Functional Description	10
3.1	Features	10
3.2	Initialization	10
3.3	Main Functions	10
3.4	Error Handling.....	10
3.4.1	Development Error Reporting.....	10
3.4.2	Production Code Error Reporting	10
4	Integration.....	11
4.1	Scope of Delivery	11
4.1.1	Static Files	11
4.1.2	Dynamic Files	11
4.2	Include Structure.....	12
4.3	Compiler Abstraction and Memory Mapping.....	12
4.4	Critical Sections	13
5	API Description.....	14
5.1	Interfaces Overview	14
5.2	Structures	14
5.2.1	CryFord_MacSecAccessVerifyConfigType	14
5.2.2	CryFord_MacSecAccessWorkSpaceType	14
5.3	Services provided by CRYFORD	15
5.3.1	CryFord_MacSecAccessInit.....	15
5.3.2	CryFord_MacSecAccessVerifyStart	15
5.3.3	CryFord_MacSecAccessVerifyUpdate	16
5.3.4	CryFord_MacSecAccessVerifyFinish	17
5.3.5	CryFord_MacSecAccessVerifyMainFunction.....	18
5.4	Services used by CRYFORD	18
6	Configuration.....	19
6.1	Configuration Variants.....	19
6.2	Manual Configuration	19
6.2.1	CryFord_Cfg.h	19
6.2.1.1	Common Properties	19

- 6.2.1.2 Service Properties 19
 - 6.2.2 CryFord_Cfg.c..... 19
- 7 AUTOSAR Standard Compliance..... 20**
 - 7.1 Deviations 20
 - 7.2 Additions/ Extensions..... 20
 - 7.3 Limitations..... 20
 - 7.3.1 Tool supported configuration 20
- 8 Glossary and Abbreviations 21**
 - 8.1 Glossary 21
 - 8.2 Abbreviations 21
- 9 Contact..... 22**

Illustrations

Figure 2-1	AUTOSAR 4.x Architecture Overview	8
Figure 2-2	Interfaces to adjacent modules of the CRYFORD	9
Figure 4-1	Include structure	12

Tables

Table 1-1	Component history.....	6
Table 4-1	Static files	11
Table 4-2	Generated files	11
Table 4-3	Compiler abstraction and memory mapping.....	13
Table 5-1	CryFord_MacSecAccessVerifyConfigType.....	14
Table 5-2	CryFord_MacSecAccessWorkSpaceType	14
Table 5-3	CryFord_MacSecAccessInit.....	15
Table 5-4	CryFord_MacSecAccessVerifyStart	15
Table 5-5	CryFord_MacSecAccessVerifyUpdate	16
Table 5-6	CryFord_MacSecAccessVerifyFinish	17
Table 5-7	CryFord_MacSecAccessVerifyMainFunction	18
Table 5-8	Services used by the CRYFORD	18
Table 8-1	Glossary	21
Table 8-2	Abbreviations.....	21

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	Initial version

Table 1-1 Component history

2 Introduction

This document describes the functionality, API and configuration of the MICROSAR module CRYFORD as a CRY service specified in [1].

The CRYFORD implements the Ford security access algorithm specified in [6].

Supported AUTOSAR Release*:	4	
Supported Configuration Variants:	pre-compile	
Vendor ID:	CRY_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	CRY_MODULE_ID	255 decimal (according to ref. [4])

* For the precise AUTOSAR Release 4.x please see the release specific documentation.

The Cryptographic library module (CRY) offers cryptographic primitives. The CRY module is used by the Crypto Service Manager (CSM).

2.1 Architecture Overview

The figure shows the interfaces to adjacent modules of the CRYFORD.

These interfaces are described in chapter 5.

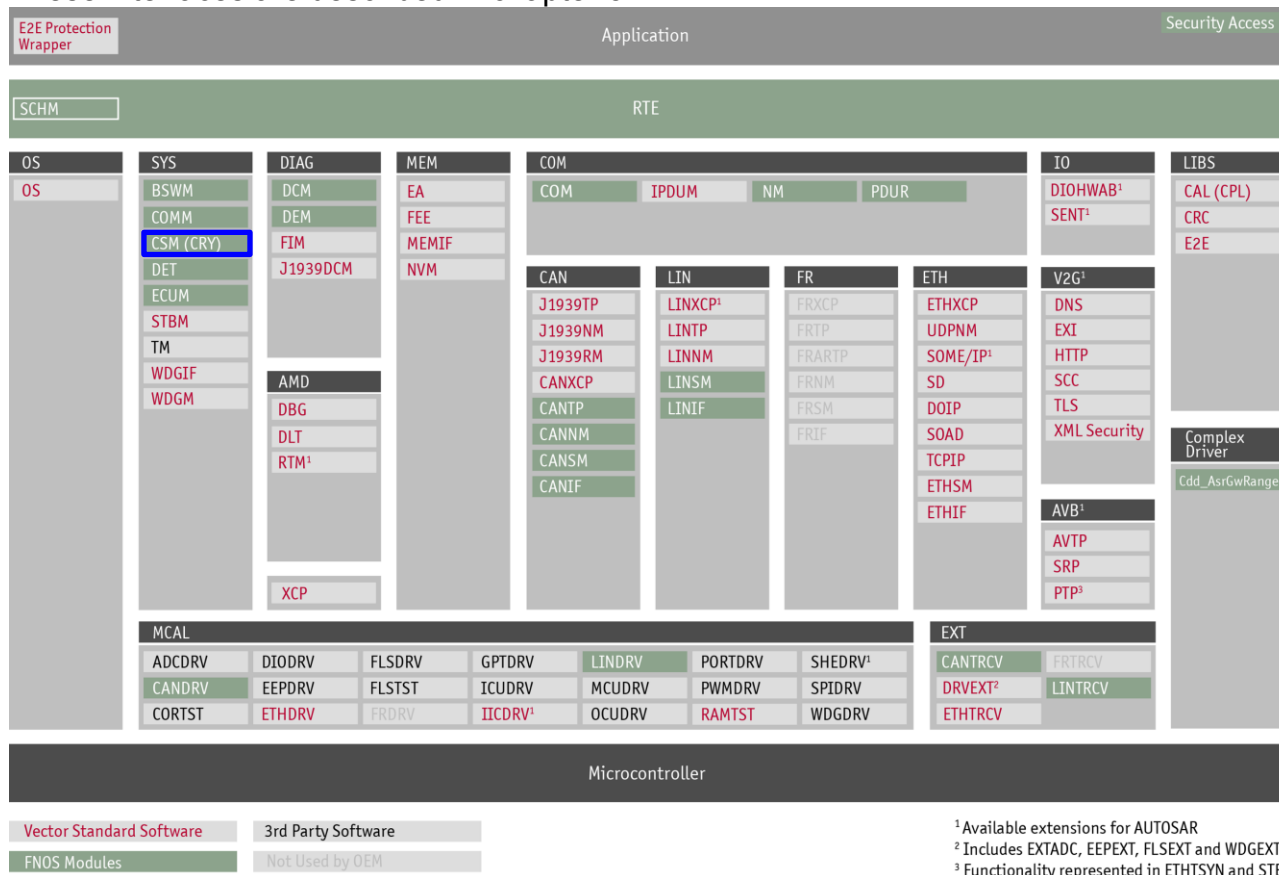


Figure 2-1 AUTOSAR 4.x Architecture Overview

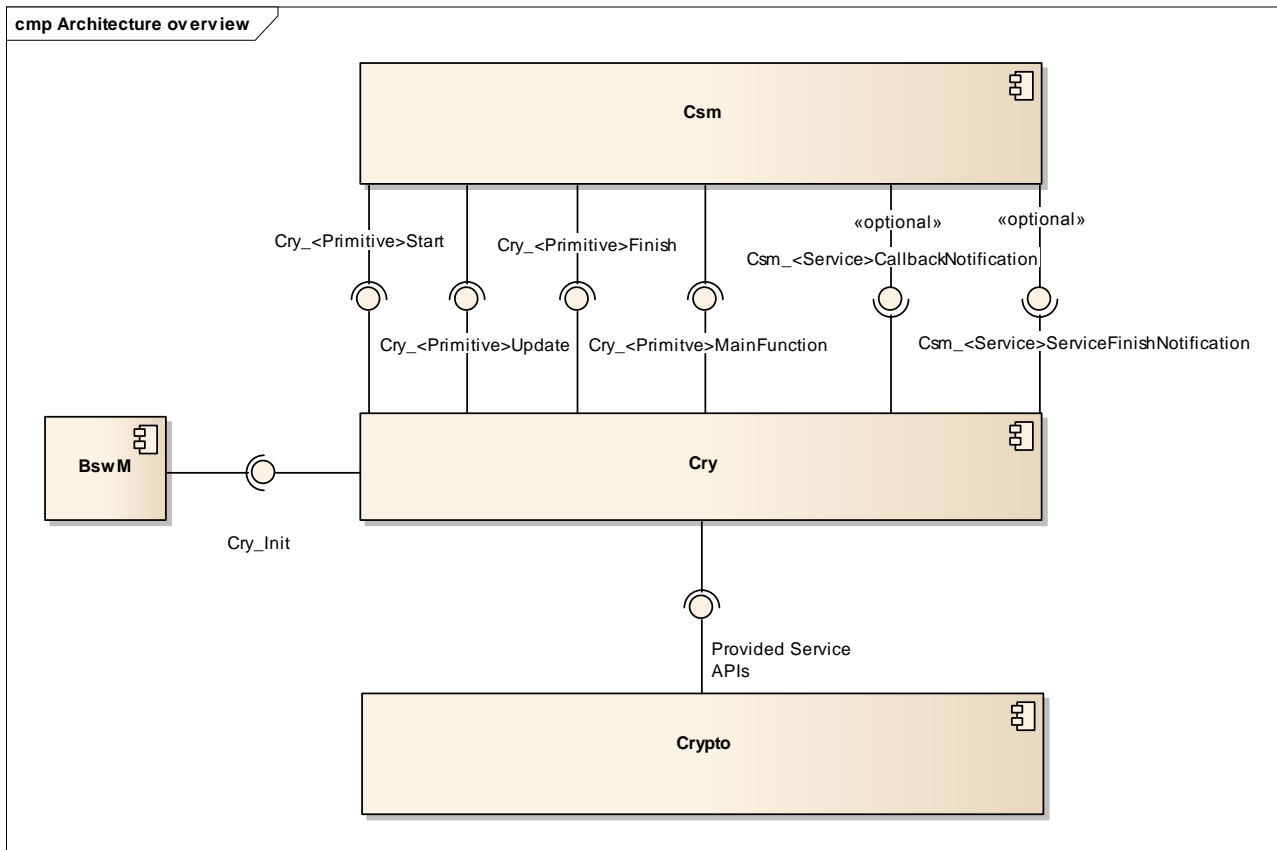


Figure 2-2 Interfaces to adjacent modules of the CRYFORD

3 Functional Description

3.1 Features

The CRYFORD implements the Ford security algorithm, specified in [6], as MAC verification service for the CSM.

3.2 Initialization

Before calling the CRYFORD module the initialization function `CryFord_MacSecAccessInit()` has to be called. The initialization call shall take place before initializing the CSM.

For API details refer to chapter 5.3.1 'CryFord_MacSecAccessInit'.

3.3 Main Functions

The CRYFORD module implementation provides a main function. When the usage of sync job processing is disabled, this main function has to be called by the CSM whenever a service is active.

For API details refer to chapter 5.3.5 'CryFord_MacSecAccessVerifyMainFunction'.

3.4 Error Handling

3.4.1 Development Error Reporting

The current implementation of the CRYFORD module does not report any development errors.

3.4.2 Production Code Error Reporting

The current implementation of the CRYFORD module does not report any production errors.

4 Integration

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

4.1 Scope of Delivery

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

4.1.1 Static Files

File Name	Source Code Delivery	Library Delivery	Description
CryFord_MacSecAccess.c		■	Implementation of the Ford Security Access algorithm
CryFord_MacSecAccess.h	■		Header file of the module
SecModLib.lib ¹		■	Library file of the cryptographic primitives

Table 4-1 Static files

4.1.2 Dynamic Files

The dynamic files must be adapted manually. Refer to chapter 6.2 'Manual Configuration' for more details.

File Name	Description
CryFord_Cfg.c	This is the configuration source file.
CryFord_Cfg.h	This is the configuration header file.

Table 4-2 Generated files

¹ The name of the underlying cryptographic primitive library may differ.

4.2 Include Structure

Figure 4-1 shows the include structure of the Cry. Some includes are optional and depend on the configuration. `CryFord_<Primitive>.h` stands for every used cryptographic primitive.

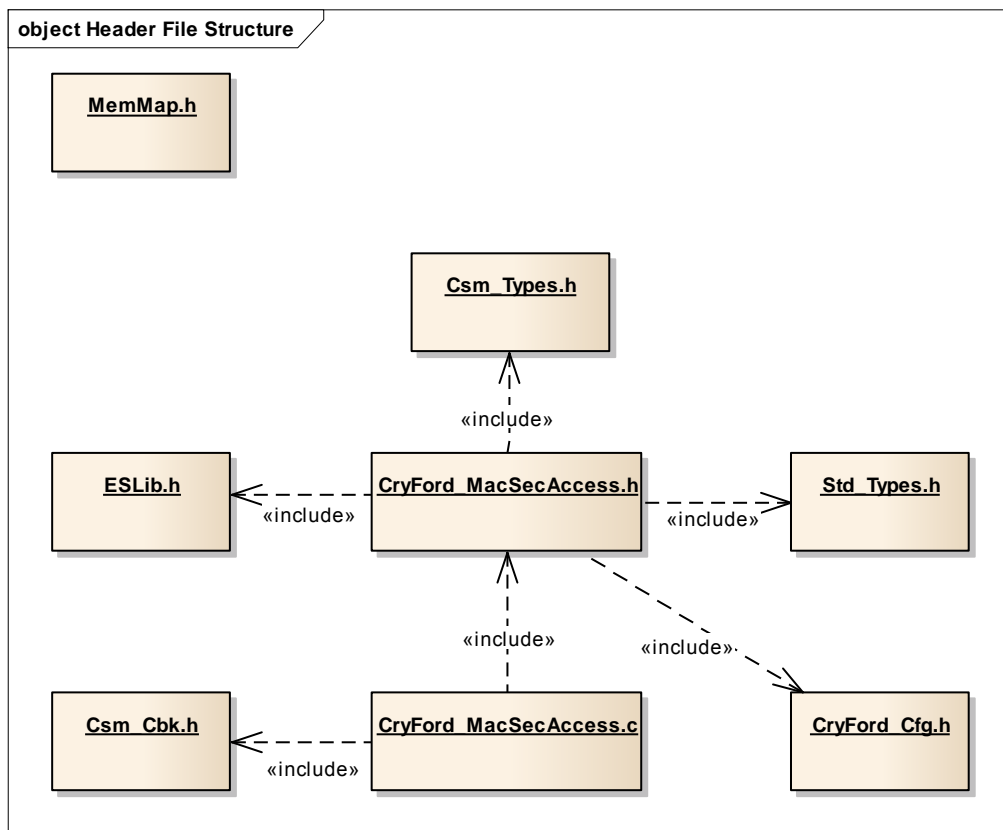


Figure 4-1 Include structure

4.3 Compiler Abstraction and Memory Mapping

The objects (e.g. variables, functions, constants) are declared by compiler independent definitions – the compiler abstraction definitions. Each compiler abstraction definition is assigned to a memory section.

The following table (Table 4-3) contains the memory section names and the compiler abstraction definitions of the CRYFORD and illustrates their assignment among each other.

Memory Mapping Sections	Compiler Abstraction Definitions		
	CRYFORD_CODE	CRYFORD_VAR_NOINIT	CRYFORD_APPL_VAR
CRYFORD_START_SEC_CODE CRYFORD_STOP_SEC_CODE	■		■
CRYFORD_START_SEC_VAR_NOINIT_8BIT CRYFORD_STOP_SEC_VAR_NOINIT_8BIT		■	
CRYFORD_START_SEC_VAR_NOINIT_UNSPECIFIED CRYFORD_STOP_SEC_VAR_NOINIT_UNSPECIFIED		■	

Table 4-3 Compiler abstraction and memory mapping

4.4 Critical Sections

The current implementation of the CRYFORD module does not have any critical section.

5 API Description

5.1 Interfaces Overview

For an interfaces overview please see Figure 2-2.

5.2 Structures

5.2.1 CryFord_MacSecAccessVerifyConfigType

This structure represents the configuration for the MAC security access service.

Struct Element Name	C-Type	Description	Value Range
buffer	CryFord_MacSecAccessWorkSpaceType*	Pointer to a provided buffer which will be used as workspace for the primitives	
useSyncJobProcessing	Boolean	Switch to enable and disable synchronous job processing. True: synchronous job processing enabled False: synchronous job processing disabled	TRUE, FALSE

Table 5-1 CryFord_MacSecAccessVerifyConfigType

5.2.2 CryFord_MacSecAccessWorkSpaceType

This structure represents the work space for the MAC security access service.

Struct Element Name	C-Type	Description	Value Range
workspace	uint8	Work space type for the MAC security access algorithm	

Table 5-2 CryFord_MacSecAccessWorkSpaceType

5.3 Services provided by CRYFORD

5.3.1 CryFord_MacSecAccessInit

Prototype	
<code>void CryFord_MacSecAccessInit (void)</code>	
Parameter	
-	
Return code	
-	
Functional Description	
This interface shall be used to initialize the MAC verification service of the CSM module.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. > This function has to be called during start-up. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-3 CryFord_MacSecAccessInit

5.3.2 CryFord_MacSecAccessVerifyStart

Prototype	
<code>Csm_ReturnType CryFord_MacSecAccessVerifyStart (Const void *cfgPtr, const Csm_SymKeyType *keyPtr)</code>	
Parameter	
cfgPtr	Holds the identifier of the CSM module configuration.
keyPtr	Holds a pointer to the key which has to be used.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is busy.
Functional Description	
This interface shall be used to initialize the HMAC SHA1 verification.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-4 CryFord_MacSecAccessVerifyStart

5.3.3 CryFord_MacSecAccessVerifyUpdate

Prototype	
Csm_ReturnType CryFord_MacSecAccessVerifyUpdate (Const void *cfgPtr, const uint8 *dataPtr, uint32 dataLength)	
Parameter	
cfgPtr	Holds the identifier of the CSM module configuration.
dataPtr	Holds a pointer to the data for which a MAC shall be computed.
dataLength	Contains the number of bytes for which the MAC shall be computed.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is busy
Functional Description	
This interface shall be used to feed the MAC verification.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-5 CryFord_MacSecAccessVerifyUpdate

5.3.4 CryFord_MacSecAccessVerifyFinish

Prototype	
Csm_ReturnType CryFord_MacSecAccessVerifyFinish (Const void *cfgPtr, uint8 *MacPtr, uint32 *MacLength)	
Parameter	
cfgPtr	Holds the identifier of the CSM module configuration.
MacPtr	Holds a pointer to the memory location which will hold the MAC to verify.
MacLength	Holds the length of the MAC to be verified.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is busy.
Functional Description	
This interface shall be used to finish the MAC verification.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-6 CryFord_MacSecAccessVerifyFinish

5.3.5 CryFord_MacSecAccessVerifyMainFunction


Prototype	
void CryFord_MacSecAccessVerifyMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is not reentrant. > This function has to be called by CSM. > This function must not be called by the application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-7 CryFord_MacSecAccessVerifyMainFunction

5.4 Services used by CRYFORD

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

Component	API
CSM	Csm_MacVerifyCallbackNotification Csm_MacVerifyServiceFinishNotification
SecMod ²	Provided Service APIs

Table 5-8 Services used by the CRYFORD

² Name of the module may differ

6 Configuration

The current implementation of the CRYFORD does not have a tool supported configuration. The configuration must be edited manually.

6.1 Configuration Variants

The CRYFORD supports only the configuration variant `VARIANT-PRE-COMPILE`.

6.2 Manual Configuration

6.2.1 CryFord_Cfg.h

The file `CryFord_Cfg.h` contains all necessary defines. There is no need for the Integrator to adapt these defines.

6.2.1.1 Common Properties

Attribute Name	Values <small>Default value is typed bold</small>	Description
CRY_USE_DUMMY_STATEMENT	STD_ON STD_OFF	If enabled, dummy statements are inserted for not used parameters

6.2.1.2 Service Properties

The following attributes enabled or disables the supported services.

Attribute Name	Values	Description
CRYFORD_MACSECACCESS_ENABLE D	STD_ON STD_OFF	Enables or disables the MAC

6.2.2 CryFord_Cfg.c

The template `_CryFord_Cfg.c` contains the configurations as well as the workspace buffers for the provided services. Each available service has a sample configuration. Please refer '5.2 Structures' for a description of the structure elements.

7 AUTOSAR Standard Compliance

7.1 Deviations

The current implementation does not have any deviations.

7.2 Additions/ Extensions

The current implementation does not have any extensions.

7.3 Limitations

7.3.1 Tool supported configuration

Currently, a tool supported configuration is not implemented. Therefore, the CRYFORD module must be configured manually by editing the configuration files.

8 Glossary and Abbreviations

8.1 Glossary

Term	Description
Cryptographic Primitive	An underlying cryptographic module or library

Table 8-1 Glossary

8.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CRY	Cryptographic library module
CSM	Crypto Service Manager
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
RTE	Runtime Environment
SchM	Schedule Manager
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 8-2 Abbreviations

9 Contact

Visit our website for more information on

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

www.vector.com