VECTOR

MICROSAR 4

Product Information

Version 1.03.11

Authors	Hannes Haas
Status	Released

Contents

1 Introduction

All modules and options are described in the general Product Information (please refer to the link “Product Descriptions”).

In addition to the general Product Information, this document provides additional details to the offered items.

2 General Information

Note
This chapter provides important information that applies for using MICROSAR basic software.

2.1 Compiler Warnings

Due to the use of standard software modules for a huge number of different hardware platforms and different compilers it is not possible to avoid compiler warnings completely. Vector tries to keep its software free of warnings, but in some cases it is not possible or it may decrease performance.

Vector provides a list of known compiler warnings with each delivery of a Production SIP.

2.2 64-bit Microcontroller Support

Due to limitations given by underlying specifications (ASAM, ISO, AUTOSAR…) not all address based services can be used on 64-bit controllers without limitations. A typical limitation can be that only the lower half of the 64-bit address range can be accessed.

2.3 Usage of Float Data Types

Float data types are used by the implementation and must be supported by the compiler. Usage of a “nofloat” (or similar) compiler option is not possible.

2.4 Delivery of Beta Modules/Features

Deliveries may contain Beta Modules and/or Beta Features. Beta Modules and Beta Features are basically operable, but not sufficiently tested, verified and/or qualified for use in series production and/or in vehicles operating on public or non-public roads. In particular, without limitation, the Beta Modules and Beta Features may cause unpredictable ECU behavior, may not provide all functions necessary for use in series production and/or may not comply with quality requirements which are necessary according to the state of the art. Beta Modules and Beta Features must not be used in series production.

Beta Modules and Beta Features are listed in chapter 2.4 of the issue report which is provided with the delivery. Moreover, the issue report contains information on how to deactivate the Beta Feature for series production projects.

2.5 Delivery of Stub Modules

The delivery (SIP) may include implementations for BSW modules which are only sample code intended for illustrating an example of a possible BSW implementation (hereinafter “Stub Modules”). Stub Modules are not part of the ordered BSW modules and have been added by Vector voluntarily and free of charge solely to enable the customer to perform a preliminary assembly and test of the BSW modules.

The Stub Modules have not passed any quality control measures and may be incomplete. The Stub Modules are neither intended nor qualified for use in series production without applying suitable quality measures. The Stub Modules as well as any of their modifications and/or implementations must be tested with diligent care and must comply with all quality requirements which are necessary according to the state of the art before their use.

Stub Modules can be identified by the comment section “SAMPLE CODE ONLY” in the implementation files. The implementation of Stub Modules can be found in BSW\<Msn>_Stub. Generated files are located in the generator “Source” folder using the naming convention <Msn>_Stub.c/h.

2.6 Delivery of Code Templates

The delivery may contain files that must be adapted during BSW integration. The Technical References of BSW modules lists the files that require to be adapted. This can be dedicated Template Areas or Complete Template Files (hereinafter collectively "Code Template").

Code Templates are incomplete and only intended for providing a signature and an empty implementation. Code Templates are neither intended nor qualified for use in series production without applying suitable quality measures.

Each Code Template must be completed as described in the Technical Reference and/or in the respective Code Template file. The completed implementation must be tested with diligent care and must comply with all quality requirements which are necessary according to the state of the art before its use.

2.7 Delivery of Example Code

The delivery (SIP) may include a Start Application and/or a Demo (hereinafter collectively “Example Code”). The Example Code is only intended for illustrating an example of a possible BSW integration and BSW configuration.

The Example Code has not passed any quality control measures and may be incomplete. The Example Code is neither intended nor qualified for use in series production.

The Example Code as well as any of its modifications and/or implementations must be tested with diligent care and must comply with all quality requirements which are necessary according to the state of the art before their use.

The Start Application is located in the folder .\StartApplication of the SIP.

The Demo is located in the folder .\Demo of the SIP.

2.8 Usage of Features that are not Licensed

BSW features that are not explicitly listed in the order confirmation are not licensed. If the BSW configuration enables a feature that has not been licensed a warning message will be shown or the build or generation process will abort with an error. (e.g. issued at code generation time).

In case of a warning message the feature remains fully functional. The license violation will be communicated in the format “License Violation” and an additional description that describes the kind of violation.

If the build or generation process yields such a “License Violation” the feature may be used for evaluation purposes but must not be used for serial production as:

- the feature is not licensed for serial production purposes
- the feature may include code that is incomplete and/or only partly tested
- there will be no issue reporting, no remedy of defects and no liability for the feature

To avoid the “License Violation” for serial production either:

- Contact Vector and request an update which includes the license for the feature.
- Modify the configuration in such a way that does not violate the license.

2.9 Fulfillment and Traceability

According to the requirements of ISO 26262 we define full traceability to design and test artefacts starting with the “Component Requirements” (CREQs) for all components developed with ASIL A, B, C or D. Thus, we fulfill the requirements according to the concept of “Safety Element out of Context” (SEooC). For QM components these traces are partly defined.

Additionally, we provide information about the fulfillment with respect to the AUTOSAR SWS. Information about the fulfillment with respect to other requirements (such as OEM requirements) is provided only partly.

2.10 Test Report

Each delivery of a Production SIP includes a report about executed tests and their results.

Test specifications are not part of a delivery, but they can be reviewed at Vector upon request.

2.11 MISRA

The offered basic software modules will be checked for compliance with the MISRA rules (MISRA-C:2004, QAC 7.0) based on typical configurations. On request we will send you the list of exceptions.

A test report based on the customer’s configuration has to be ordered explicitly.

2.12 Test of External Hardware

Delivery tests are able to incorporate testing of external hardware such as EEPROM, watchdog or transceiver devices, which are connected to the micro controller e.g. through SPI or DIO pins.

As DIO driver and SPI driver controlling the external devices are part of the MCAL, it is a prerequisite to order the SIP position “MCAL Integration” when external devices shall be tested during delivery tests.

2.13 Test of Third Party Modules

As part of the integration third party modules are configured, compiled and linked with the test and/or demo setup. Runtime integration tests focus on Vector's modules. Therefore, the functionality of the third party modules is only tested implicitly, as far as needed for Vector's modules to run properly.

2.14 Issue Handling

Issue reporting and remedy of defects will be provided only for the last delivered version. Therefore, all earlier delivered software packages and versions will not be traced for issue reporting. Additional issue reporting for already delivered software packages can be offered as a separate line item.

As long as the SIP is under maintenance, Vector provides an active issue reporting for the delivered BSW modules and the RTE including their code generators. The issue report contains all open issues and is sent periodically to the provided email contacts.

2.14.1 Issue Reporting for Third Party Software

Issues of third party software integrated by Vector are not part of the issue reports send by Vector.

Vector is not able to report third party issues due to several reasons such as:

- NDAs between Vector and the third party software vendor
- Differing release and update cycles of the third party software
- Third party software can be updated by the customer without involving Vector

If third party software is included in the SIP, a documentation of the issue reporting process of the third party software is included in the SIP folder .\Doc\DeliveryInformation\IssueHandling_<Name>.pdf.

2.14.2 Issue Reporting for DaVinci Tools

Issues of the DaVinci tools are not part of the active issue reporting. The issue lists can be downloaded from our internet portal:

DaVinci Developer Link

DaVinci Configurator Link

2.15 Warranty

Vector provides warranty for all modules, which have been licensed by Vector. These are the modules developed by Vector. For modules developed by third party vendors reduced warranty is provided. Please see below for details.

2.15.1 Third Party Modules

Vector does not provide any warranty, liability or issue reporting for third party modules. These topics have to be stipulated with the third party supplier directly.

If the third party modules are configured using DaVinci Configurator, warranty and issue reporting will be given only for the correct integration into the tool, i.e. correct mapping of GUI item to ECUC item and correct control of the generation process.

3 Modules and Options

Note
This chapter defines important constraints and limitations of MICROSAR BSW modules and options that may have been offered or delivered.

3.1 CanTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.6 Delivery of Code Templates).The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

3.2 Crypto (vHsm)

This module allows accessing the hardware secure module (HSM) from a MICROSAR BSW stack using Csm. It requires the MICROSAR HSM firmware (vHsm) to be available on the HSM core.

vHsm is not included and offered separately. Crypto (vHsm) and MICROSAR vHsm must be compatible with each other. If an existing vHsm delivery shall be used, please contact Vector in advance so that it can be determined if both software versions are compatible with each other. It may be required that vHsm need to be updated to a compatible version.

3.3 Crypto (Sw)

Crypto (Sw) supports the following crypto algorithms:

Algorithm / Function	Standard	Crypto (Sw)	CSM 4.3 Service	CSM 4.3 Interface
SHA-1	FIPS 180-4	x	Hash Functions	Hash Interface
SHA-2 (SHA-256, SHA-512)	FIPS 180-4	x		
Pseudo random number generation based on AES	FIPS-186-2	x	Random Numbers	Random Interface
CTR-DRBG based on AES-128 with DF and without DF	NIST SP 800-90A	x		
HMAC based on SHA-1, HMAC based on SHA-256	FIPS 198-1	x	Message Authentication Code (MAC) Generation and Verification	MAC Interface
CMAC based on AES	IETF RFC4493	x		
SipHash		x		
AES-128 in the modes ECB/CBC	FIPS-197	x	Symmetric Encryption and Decryption	Cipher Interface
AEAD: AES-GCM	NIST 800-38D	x		
Asymmetric encryption and decryption based on RSA with key length of 512-4096 bit	PKCS #1 V1.5	x[1]	Asymmetric Encryption and Decryption	Cipher Interface

Algorithm / Function	Standard	Crypto (Sw)	CSM 4.3 Service	CSM 4.3 Interface
KDF in Counter Mode, KDF in Counter Mode with Appendix, concatenation KDF (NIST 800-56A)	NIST SP 800-56A	x	Key Handling	Key Derivation Interface
Key exchange using the Elliptic Curve Diffie-Hellman protocol EC-DHE with ANSIp256r1, SECp256r1 and X25519	ANSI X9.63	x[1]		Key Exchange Interface
Digital Signatures based on RSA: RSA PKCS #1V1.5 Prehashing variants: SHA-1, SHA-256	PKCS #1 V1.5	x[1]	Signature Generation and Verification	Signature Interface
Digital Signatures based on RSA: RSA CRT DSA Verification Prehashing variants: SHA-1, SHA-256	PKCS #1 V1.5	x		
Digital Signatures based on RSA: PSS Prehashing variants: SHA-1, SHA-256	PKCS #1 V2.2	x		
Digital signatures based on the Elliptic Curves: ECDSA Ed25519 PreHashing: None, SHA-1	ANSI X9.62-2005	x[1]		
Digital signatures based on the Elliptic Curves: ECDSA ANSI P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x		
Digital signatures based on the Elliptic Curves: ECDSA NIST P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x[1]		
Digital signatures based on the Elliptic Curves: ECDSA SEC P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x		
Symmetric Key Update Protocol	SHE1.1	x		

[1] Crypto (Sw) Add-On asym. Algorithms required

Table 3-1 Supported crypto algorithms Crypto (Sw)

3.4 vDbg (Xcp-based)

MICROSAR vDbg (Xcp-based) provides access to selected internal state variables of BSW modules. As communication protocol Xcp is used (unlike specified by AUTOSAR). This allows usage of existing Xcp standard tools such as CANoe.AMD or CANape.

3.5 Dlt (Xcp-based)

MICROSAR Dlt (Xcp-based) uses Xcp as communication protocol (unlike specified by AUTOSAR). This results in a more efficient implementation and allows usage of existing Xcp standard tools such as CANoe.AMD or CANape. MICROSAR Dlt (Xcp-based) does not support the Dlt protocol, in particular the module is subject to the following restrictions:

> Only the payload of Dlt messages (without Dlt header) is accessible through Xcp.

> Dlt control messages are not supported.

If these Dlt features are required, MICROSAR Dlt (AUTOSAR) has to be used.

3.6 Dlt (AUTOSAR)

MICROSAR Dlt (AUTOSAR) implements the AUTOSAR Dlt protocol. The Dlt communication interface supports PDU transmission and reception through SoAd only. Please note that a test client is needed which supports the AUTOSAR Dlt protocol.

3.7 Drivers (Can, Lin, Fr, Eth, MCAL)

> Support of Different Modes
On those microcontrollers, which support different modes like user mode and privileged mode or supervisor mode, the driver's implementation is designed to run in privileged or supervisor mode. Rationale: Full access to all registers assigned to the corresponding hardware unit controller is required.

> Support of Fr Controllers
Some platform derivatives include 2 FlexRay controllers. MICROSAR will explicitly support only one of the available FlexRay controllers, with up to 2 FlexRay channels.

3.8 Eth (ext)

Some of the Eth (ext) drivers have special requirements on the AUTOSAR SPI driver, AUTOSAR BSW modules, and the interconnection between the microcontroller and the external chip that is either related on how the Eth (ext) driver is implemented or derived from hardware implementation (e.g. SPI signal levels, idle levels, etc.).

3.8.1 Requirements of Eth (ext) Implementation

Requirements on the AUTOSAR SPI Driver

Eth (ext) utilizes an AUTOSAR SPI driver to configure and exchange data with the external device. Dependent on the chip and the implementation of the respective Eth (ext) driver the AUTOSAR SPI driver provided for the microcontroller must fulfill specific requirements for interoperability.

Silicon Vendor / Devices / SPI Property	Atmel/Microchip	Qualcomm
	WINC1500 WILC1000	AR7000 AR7005
Synchronous API	recommended (for debug purpose)	recommended (for debug purpose)
Asynchronous API	required	required
Sequence End Notification	required (for async API)	required
Non-DMA-/FIFO-Mode	no need	no need
DMA-Mode	required	required
Data Width [Bit]	8	8
External Buffers	required	required

Silicon Vendor / Devices / SPI Property	Atmel/Microchip	Qualcomm
	WINC1500 WILC1000	AR7000 AR7005
Internal Buffers	-	-
Buffer size [Byte]	[1, 1600] (requires support for any value in range)	[2, 1528] (requires support for any value in range)
Sequence utilization [#]	1	2
Job utilization (per sequence) [#]	1	1
Channel utilization (per job) [#]	1	2

Table 3-2 Eth (ext) requirements on the AUTOSAR SPI driver

Requirements on Other AUTOSAR BSW Modules

Some Eth (ext) drivers must utilize respectively need special configuration of modules others than the AUTOSAR SPI driver.

Silicon Vendor / Devices / Special Functionality	Atmel/Microchip	Qualcomm
	WINC1500 WILC1000	AR7000 AR7005
CRC calculation (Crc)	CRC16	-

Table 3-3 Eth (ext) requirements on other AUTOSAR BSW modules

Requirements on the Interconnection Between External Device and Microcontroller

To be operational some Eth (ext) drivers rely on having specific signals connected between the external Ethernet device and the microcontroller.

Silicon Vendor / Devices / Special Functionality	Atmel/Microchip	Qualcomm
	WINC1500 WILC1000	AR7000 AR7005
Interrupt line of external device connected	no need (required if driver shall operate in interrupt mode)	no need (required if driver shall operate in interrupt mode)

Table 3-4 Eth (ext) requirements on interconnection between external device and microcontroller

3.8.2 Requirements of the Hardware Implementation

Caution
In addition to this information (that has no intention to be outright) it is highly recommended to contact the silicon vendor of the external device used for proper integration into the system design and take the information provided by the respective manuals provided by the silicon vendor into account.

Requirements on the AUTOSAR SPI Driver

The hardware implementation has requirements on how the AUTOSAR SPI driver drives the SPI signals defining the communication protocol on SPI level.

Silicon Vendor / Devices / SPI Property	Atmel/Microchip	Qualcomm
	WINC1500 WILC1000	AR7000 AR7005
Chip-Select Polarity	LOW	LOW
No Chip-Select Idle Enforcement (CS doesn't return to idle state after each data word)	recommended	required
Clock-Idle-Level	HIGH	HIGH
Data Shifting Clock-Edge	FALLING EDGE	FALLING EDGE
Baudrate [Hz]	[312 kHz, 48 MHz] (312 kHz was minimum tested)	[x Hz, 10 MHz] (minimum not known)
Parity Selection	NO PARITY	NO PARITY

Table 3-5 Hardware requirements on AUTOSAR SPI driver

Requirements on Other AUTOSAR BSW Modules

For some special functionality, the hardware implementation needs AUTOSAR BSW modules to be able to interact with the external device in a specific way.

Silicon Vendor / Devices / Special Functionality	Atmel/Microchip	Qualcomm
	WINC1500 WILC1000	AR7000 AR7005
External interrupt detection (ICU, OS, Port)	FALLING EDGE (when Vector driver is configured to operate in interrupt mode)	FALLING EDGE (when Vector driver is configured to operate in interrupt mode)

Table 3-6 Hardware requirements on other AUTOSAR BSW modules

3.9 EthSwt (ext)

Some of the EthSwt (ext) drivers have special requirements on the AUTOSAR SPI driver, AUTOSAR BSW modules and the interconnection between the microcontroller and the external chip that is either related on how the Vector driver is implemented or derived from hardware implementation (e.g. SPI signal levels, idle levels, etc.).

3.9.1 Requirements of EthSwt (ext) Implementation

Requirements on the AUTOSAR SPI Driver

EthSwt (ext) utilizes an AUTOSAR SPI driver to configure and exchange data with the external device. Dependent on the chip and the implementation of the respective EthSwt (ext) driver the AUTOSAR SPI driver provided for the Microcontroller must fulfill specific requirements for interoperability.

Silicon Vendor	**Broadcom**		**NXP**	
Devices	BCM89200 BCM8950x	BCM89230 BCM8953x BCM8954x BCM8955x	SJA1105 SJA1105T	SJA1105P SJA1105Q SJA1105R SJA1105S
SPI Property				
Synchronous API	either Async SPI or Sync SPI	required	required	required
Asynchronous API		-	-	-
Sequence End Notification	-	-	-	-
Non-DMA-/FIFO-Mode	either DMA Mode or Non-DMA Mode			
DMA-Mode				
Data Width [Bit]	8	8	8	8
External Buffers	required	required	required	required
Internal Buffers	required	-	-	-
Buffer size [Byte]	[1, 4] (requires support for any value in range)	[8, 20] (requires support for any value in range)	[8, 256] (requires support for any value in range)	[8, 256] (requires support for any value in range)
Sequence utilization [#]	1	1	1	1
Job utilization (per sequence) [#]	1	1	1	1
Channel utilization (per job) [#]	3	1	1	1

Table 3-7 EthSwt (ext) requirements on the AUTOSAR SPI driver

3.9.2 Requirements of the Hardware Implementation

Caution
In addition to this information (that has no intention to be outright) it is highly recommended to contact the silicon vendor of the external device used for proper integration into the system design and take the information provided by the respective manuals provided by the silicon vendor into account.

Requirements on the AUTOSAR SPI Driver

The hardware implementation has requirements on how the AUTOSAR SPI driver drives the SPI signals defining the communication protocol on SPI level.

Silicon Vendor / Devices / SPI Property	Broadcom		NXP	
	BCM89200 BCM8950x	BCM89230 BCM8953x BCM8954x BCM8955x	SJA1105 SJA1105T	SJA1105P SJA1105Q SJA1105R SJA1105S
Chip-Select Polarity	LOW	LOW	LOW	LOW
No Chip-Select Idle Enforcement (CS doesn't return to idle state after each data word)	required	required	required	required
Clock-Idle-Level	HIGH	LOW	LOW	LOW
Data Shifting Clock-Edge	FALLING EDGE	FALLING EDGE	RISING EDGE	RISING EDGE
Baudrate [Hz]	[x Hz, 25 MHz] (minimum not known)	[x Hz, 62.5 MHz] (minimum not known)	[x Hz, 25 MHz] (minimum not known)	[x Hz, 25 MHz] (minimum not known)
Parity Selection	NO PARITY	NO PARITY	NO PARITY	NO PARITY

Table 3-8 Hardware requirements on AUTOSAR SPI-Driver

3.10 EthTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.6 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

3.11 vEtm

vEtm is intended for ECU test purposes and is therefore basically operable, but not sufficiently tested, verified and/or qualified for use in series production and/or in vehicles

operating on public or non-public roads. In particular, without limitation, the usage of vEtm may cause unpredictable ECU behavior and/or may not comply with quality requirements which are necessary according to the state of the art. For serial production, the vEtm must either be removed from the ECU or be deactivated as described in TechnicalReference_Etm.pdf by using the provided vEtm APIs.

3.12 Fee

Fee (Standard and SmallSector) do support a single Fls only. If multiple Fls devices need to be accessed by the MICROSAR Fee, please contact Vector for a detailed analyzis.

Caused by physical restrictions of flash memory devices, MICROSAR Fee offers two alternative implementation concepts. Each has distinct advantages and drawbacks. At configuration time one has to be chosen per memory block:

- Single Sector Usage
  This concept ensures robustness against a high number of reset events but requires a higher number of erase and write cycles on the flash memory device shortening its life time.
- Parallel Sector Usage
  This concept reduces the number of erase and write cycles but leaves a low risk of data loss.

3.13 FrTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.6 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

3.14 vHsm

vHsm is executed on the hardware security module (HSM) of the microcontroller and offered as a separate project. It will be delivered as a dedicated HSM SIP.

To access vHsm from the host controller the MICROSAR SIP and/or the Vector Flash Bootloader must be extended with respective options:

- MICROSAR 4: Crypto (vHsm)
- Vector Flash Bootloader: Secure Boot (HW) Vector vHSM Integration

The vHsm implementation must be compatible with the above host controller software implementations. If existing deliveries shall be reused, compatibility must be clarified with Vector in advance. Updates of the Flash Bootloader and MICROSAR stack may be required.

The following crypto algorithms are supported:

Algorithm / Function	Standard	vHsm	Csm 4.3 Service	Csm 4.3 Interface
SHA-1	FIPS 180-4	x	Hash Functions	Hash Interface
SHA-2 (SHA-256, SHA-512)	FIPS 180-4	x		
Pseudo random number generation based on AES	FIPS-186-2	x	Random Numbers	Random Interface
CTR-DRBG based on AES-128 with DF and without DF	NIST SP 800-90A	x		
HMAC based on SHA-1, HMAC based on SHA-256	FIPS 198-1	x	Message Authentication Code (MAC) Generation and Verification	MAC Interface
CMAC based on AES	IETF RFC4493	x		
SipHash		x		
AES-128 in the modes ECB/CBC	FIPS-197	x	Symmetric Encryption and Decryption	Cipher Interface
AEAD: AES-GCM	NIST 800-38D	x		
Asymmetric encryption and decryption based on RSA with key length of 512-4096 bit	PKCS #1 V1.5	x[1]	Asymmetric Encryption and Decryption	Cipher Interface
KDF in Counter Mode, KDF in Counter Mode with Appendix, concatenation KDF (NIST 800-56A)	NIST SP 800-56A	x	Key Handling	Key Derivation Interface
Key exchange using the Elliptic Curve Diffie-Hellman protocol EC-DHE with ANSIp256r1, SECp256r1 and X25519	ANSI X9.63	x[1]		Key Exchange Interface
Digital Signatures based on RSA: RSA PKCS #1V1.5 Prehashing variants: SHA-1, SHA-256	PKCS #1 V1.5	x[1]	Signature Generation and Verification	Signature Interface
Digital Signatures based on RSA: RSA CRT DSA Verification Prehashing variants: SHA-1, SHA-256	PKCS #1 V1.5	x[1]		
Digital Signatures based on RSA: PSS Prehashing variants: SHA-1, SHA-256	PKCS #1 V2.2	x[1]		
Digital signatures based on the Elliptic Curves: ECDSA Ed25519 PreHashing: None, SHA-1	ANSI X9.62-2005	x[1]		
Digital signatures based on the Elliptic Curves: ECDSA ANSI P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x[1]		
Digital signatures based on the Elliptic Curves: ECDSA NIST P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x[1]		
Digital signatures based on the Elliptic Curves: ECDSA SEC P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x[1]		

Algorithm / Function	Standard	vHsm	Csm 4.3 Service	Csm 4.3 Interface
Secure Boot Protocol	SHE1.1	x		
Symmetric Key Update Protocol	SHE1.1	x		

[1] vHsm Add-On Asymmetric Crypto required

Table 3-9 Supported crypto algorithms vHsm

For some hardware platforms HSM MCAL modules are not developed by Vector and instead need to be licensed by the customer from the hardware vendor. The customer needs to provide these modules to Vector for integration at the latest 12 weeks before the planned delivery date. The following table lists the affected hardware platforms:

HW vendor	HW platform	Required HSM MCAL Modules
Cypress	Traveo II	PROT, IPC, CRYPTO, FLS with AUTOSAR interfaces

Table 3-10 Hardware platforms of required HSM MCAL modules

3.15 IDM - Variant Handling / Post-Build Selectable

By default, MICROSAR supports the configuration variant Pre-Compile without variant handling. The option MICROSAR Identity Manager (IDM) implements post-build selectable which allows choosing between several ECU behaviors at startup time. Variants are defined pre-compile time.

If variant handling is requested the option IDM lists the clusters that will support variant handling. If not defined otherwise, variant handling is available for modules that support variant handling (POST-BUILD) according to AUTOSAR.

Please note the following constraints when using the option (IDM):

- IDM (COM, DIAG): Transformers are excluded
- IDM (DIAG): DCM supports the communication side only. Variance of diagnostic services is modelled as a superset in the input files. Using DCM APIs it is possible to enabled/disable services at runtime.
- IDM (DIAG): Features are limited to selected features:
  - Variant specific enabling of DTCs and definition of OBD relevance
  - Variance of selected DTC properties with respect to Priority, Aging, Healing and Confirmation
  - Variance of Events and selected pre-debouncing properties using the same type
  - Variance of enable- and storage conditions
  - Variance of DID IDs and snapshot assignment
  - Variant specific existence of DCM diagnostic services, sub-services, DIDs, RIDs, PIDs, MIDs, TIDs, VIDs and memory-ranges

- Variant specific in ECU composition of OBD “supported DIDs, RIDs, PIDs, MIDs, TIDs and VIDs” response bit mask value
- Variance in the DCM communication interface

- IDM (CAN, FR, SYS): <Bus>TSyn modules and StbM are excluded
- IDM (RTE): Limited to data mapping variance
- It is assumed that all BSW modules are used in all variants. It is not possible to not use a module in one or more variants.

To ensure that the required use-cases are supported, please provide your use-cases to Vector.

3.16 LinTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.6 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

3.17 MCAL Integration Package

The MCAL Integration Package covers the integration of a third party module provided by the customer into the Vector MICROSAR basic software and the DaVinci Configurator Pro. It is the sole responsibility of the customer to test the freedom from defects and the functionality of third party modules and such testing will not be carried out by Vector. Prerequisite for the integration into the DaVinci Configurator Pro is that the description files of these modules provided by the customer are compliant to AUTOSAR.

The MCAL Integration Package does not include the license to use the third party MCAL. It is the sole responsibility of the customer to obtain the license necessary for the use of the third party MCAL intended by the customer.

An MCAL integration is only possible if

- the MCAL version is communicated to Vector as part of the answer to the Questionnaire Step II at the latest 6 weeks before the planned delivery date.
- the specified MCAL is available in general.

On most HW platforms the MCAL integration will result in the delivery of two artefacts as part of the SIP:

- MCAL Integration Helper Tool: This tool allows placing a third party MCAL into the SIP as self-service as well as MCAL specific maintenance activities such as derivative selection and supports the user in up-grading or down-grading the MCAL version.
- Included third party MCAL: The Questionnaire Step II will contain an upload link where the customer can send MCAL files to Vector to proof ownership. If the customer sends the MCAL (which has to match the specified MCAL version from Questionnaire Step II and which must encompass all artefacts including implementation, tooling and license files) not later than 6 weeks before the planned delivery date, Vector will use these files during testing and also include them in the delivery. If the MCAL is not sent in

time, the SIP will not contain an MCAL and the customer will have to include the MCAL using the MCAL Integration Helper Tool.

3.18 PBL - Post-Build Loadable

By default, MICROSAR supports the configuration variant Pre-Compile, without the possibility to update the ECU configuration at post-build time. The option Post-Build Loadable (PBL) provides support for a post-build time update.

If post-build loadable is requested the option PBL lists the clusters that support a post-build time update of the configuration. Post-build loadable allows changing aspects of the ECU's communication matrix after the software has been downloaded to the ECU. The post-build time update process does not require the usage of a compiler. Only post-build configuration data (e.g. a dedicated flash page) are downloaded to the ECU.

Deliveries supporting post-build loadable include a special license for the DaVinci Configurator that will allow the OEM to perform an update at post-build time without the need of own tool licenses.

If post-build loadable is requested the option PBL lists the supported clusters. If not defined otherwise, post-build loadable is available for modules that POST-BUILD according to AUTOSAR.

Please note the following constraints when using the option PBL:

- A check if the chosen compiler and compiler options are supported by the post-build time update process is performed as part of the Software Integration Package (SIP)
- PBL (COM, DIAG): Transformers are excluded
- PBL (COM): Deleting signals at post-build time is only possible for signals that have been added during post-build phase.
- PBL (ETH): Only selected features of Sd and SoAd support PBL
- PBL (DIAG): Features are limited to selected features such as:
  - Disabling of DTCs as well as removing the OBD relevance at post-build time
  - Adding new DTCs for existing events
  - Modification of selected DTC properties with respect to Priority, Aging, Healing and Confirmation
  - Disable existing events
  - Modification of selected event properties such as pre-debouncing properties using the same debouncing type
  - Modification of the event enable- and storage conditions using existing conditions
  - Modification of DID IDs and snapshot assignment
  - Modification of the DCM communication interface
- PBL (CAN, FR, SYS): <Bus>TSyn modules and StbM are excluded
- PBL (SYS): WDG stack is excluded

- The post-build loadable update of an ECU is typically performed by the OEM. As a prerequisite, the Tier1 needs to forward the MICROSAR SIP and his DaVinci ECU configuration project to the OEM. The forwarded SIP must not include any c-files which have to be deleted by the Tier1 in advance.
- With respect to the System Description the following limitations apply:
  - ComIPdu
    - Pdu.shortName has to unique
    - The Pdu is referenced by only one PduToFrameMapping in a frame with only one triggering for each direction
  - ComSignal, ComSignalGroup, ComGroupSignal:
    - ISignalToIPduMapping.shortName has to be unique
    - Only one triggering for each direction
  - Usage of PBL in Safety related projects is not recommended. Details are given in ProductInformation_2_MSR4-MICROSARSafe.pdf chapter 2.5.2 Process.

To ensure that the required use-cases are supported, please provide your use-cases to Vector.

3.19 Rte

The MICROSAR Rte uses the AUTOSAR 4.3 APIs Interrupt Source and Hardware Peripheral Access to allow hardware specific components to safely access peripherals. The MICROSAR OS provides this APIs. If a non MICROSAR OS is used, these APIs need to be provided either by the OS or by integrator code.

3.20 vRtm

To make use of this module, CANoe.AMD is required with version 8.1 or later.

3.21 vScc

RFC 6961 (“Multiple Certificate Status Request Extension“) required by ISO 15118 with requirement V2G2-070 is not supported by the vTls.

3.22 vTls

vTls supports the following cipher suites:

- TLS_NULL_WITH_NULL_NULL (only for first handshake)
- TLS_RSA_WITH_NULL_SHA*
- TLS_RSA_WITH_AES_128_CBC_SHA*
- TLS_RSA_WITH_NULL_SHA256*
- TLS_RSA_WITH_AES_128_CBC_SHA256*
- TLS_ECDH_ECDSA_WITH_NULL_SHA

- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256*

* not supported together with Csm

The implementation is done in software. At a later point in time vTls will be adapted to use the Csm stack.

3.23 WdgM

The default implementation of the WdgM supports only “Alive Supervision”. In case “Deadline Supervision” or “Logical Supervision” (control flow monitoring) functionality is required (e.g. in safety projects due to IEC 61508 or ISO26262), it is supported by the SafeWatchdog module.

4 Customer Development Environment

Note
This chapter is only relevant for projects that refer this chapter (“Customer Development Environment”) in the section “Delivery Conditions” of the quote and order confirmation. For all other projects, this chapter does not apply.

Due to the complexity of some hardware devices Vector requires the customer to provide a working development environment to start development. For projects with such hardware additional conditions apply that are described in this chapter.

4.1 Lead Time

The lead time is counted after the purchase order and the handover of a working development environment by the customer to Vector. This handover takes place in a joint workshop.

4.1.1 Provided Development Environment

The customer provides one development environment to Vector free of charge. The handover takes place in the joint workshop. This development environment includes:

- Hardware (customer hardware sample, EVAL board) with the required peripherals and harnesses
- Debugger or Emulator
- Required software and licenses such as
  - Compiler, Linker, IDE
  - Complete build environment (including e.g. Make files, linker command files, scripts…)
  - Firmware and Libraries if required
- Code for basic initialization and startup: startup code, port initialization, PLL, peripheral clock and watchdog
- Hardware setup documentation (description of power up sequence and schematic of power supply, transceiver wiring)

The development environment is returned by Vector after the completion of the project. Please provide the address where Vector shall return the development environment along with your delivery.

4.1.2 Joint Workshop

The need for a workshop is defined by Vector after the purchase order and communicated to the customer. Its location is defined by Vector and is typically a Vector location such as Vector Informatik in Stuttgart.

The duration of the workshop depends on the complexity of the hardware and on the time required to handover a working development environment. The workshop also includes a

training of the involved Vector staff with the specifics of the provided development environment. Typically, a workshop is expected to last for 3-5 days.

The objectives of the workshop are

- It is possible to download a small test program to the hardware and set breakpoints via debugger or emulator.
- The MCU core(s) that shall be used by MICROSAR are operational without the need to utilize dedicated tools such as flash or debugger tools. This is proven by:
  - Entering an endless loop after startup code
  - Toggling a port at 1kHz (1ms)

Workshop related travel expenses are carried by the customer.

4.2 Project Organization

After reception of the purchase order, the Vector project coordinator will contact you within two weeks to clarify the next steps such as the organization of the joint workshop and the provision of the development environment.

5 Services

Note
This chapter defines requirements of MICROSAR services that may have been offered or ordered.

5.1 MICROSAR Getting Started V2G

The following requirements must be prepared and provided by the customer for the getting started session. Missing artefacts can result in missing the getting started objectives or can be the root cause for additional costs.

- Hardware (e.g. evaluation board) with integrated and working BSW stack. Including:
  - Working BSW integration with at least the following modules
    - SPI, NVM stack, ETH stack incl. bus communication
    - OS: Providing an initialization task as well as a 5ms cyclic preemptive task
  - QCA7000\7005 being integrated and fully functional
    - Recommended: QCA interrupt handling via ICU
- At least one PC that can be used during the workshop
  - Two free Ethernet ports. For PnC: three ports plus one switch/hub
  - MS Windows. Version must be compatible with the used tools
    - IPv6 must be enabled and the firewall must be deactivated or configurable
  - Working and mastered development environment for the chosen hardware
    - E.g. Compiler, debugger, build environment, licenses
    - CANoe.Ethernet 8.2 or later and Smart Charging Package 1.5.5 or later
  - DaVinci Configurator Pro 5 installed and license available

6 SIP Extensions

The following extensions are available as product services, carried out by Vector.

6.1 Customer Hardware

Integration and test of the Software Integration Package (SIP) is done on the customer specific hardware instead of an evaluation board. Prerequisite is the provision of the complete development environment:

- ECU or customer hardware
- Additionally, required hardware, e.g. wiring harness, etc.
- IDE/compiler, linker and workbench incl. license (if not available at Vector)
- Debugger (incl. license) & connector (if not available at Vector)
- If applicable: POSIX operating system including the hardware specific boards support package

It is the customer's responsibility to ensure that the hardware is running. This may require that necessary software has to be handed over to Vector. All required drivers to setup the hardware must be available for integration. In case the SIP includes generic drivers (e.g. Transceiver, SBC, etc.) it is the responsibility of the customer to provide the concrete software implementation for such hardware. In case the software drivers are not available in time or do not work properly, Vector will not be able to setup the system and will not be able to perform the required tests.

At project start, the detailed requirements will be clarified with the customer.

After initial operation of the hardware has been achieved, the different software modules are integrated and the integration tests are performed.

Please be aware that the SIP lead time is based on the following milestones:

- to provide the development environment within 2 weeks after purchase order
- to get the development environment running within 1 week.

6.2 Start Application

The initial delivery of a project includes a free of charge Start Application (including the complete build environment). The Start Application is based on the ECU specific input data for communication (e.g. ECU Extract) and diagnostics (e.g. ECUC). It demonstrates basic SIP functionality based on the ordered modules. Some examples are listed below.

- Communication: The SWCs included in the Start Application demonstrate the reception and transmission of one signal on runnable level. The BSW is configured in a way that the system gets initialized and communicates on all communication busses defined in the input data.
- Diagnostics: The SWCs included in the Start Application will be extended by a sample implementation of an RDBI/WDBI service on runnable level. Moreover it offers the possibility to trigger a DTC.

- Memory: The SWCs included in the Start Application will be extended by a sample implementation to store data in a NV-Block.

Preconditions
A Start Application can be provided by Vector if the following preconditions are met:

- The order contains a complete MICROSAR basic software stack (incl. third party MCAL, Os, and Rte)
- The customer provides the databases for communication & diagnostics prior to the project start.

Deliverables
- SIP Modules
- Start Application

Test Reports
- Vector test reports

Note
The delivered configuration and source code is for demonstration purpose only.

6.3 vVIRTUALtarget (VTT)

vVIRTUALtarget (VTT) is a virtual integration platform to run and test a MICROSAR stack on the PC. MICROSAR VTT realizes the embedded aspect of vVIRTUALtarget by providing VTT modules for MCAL and OS that are designed to run in the virtual environment.

Components
MICROSAR VTT provides the following BSW modules:

- VTT OS
- VTT communication drivers for
  - CAN
  - LIN
  - FlexRay
  - Ethernet
  - SPI
- VTT I/O drivers for
  - ICU
  - PWM
  - ADC
  - DIO

- PORT
- VTT memory drivers for
  - EEPROM
  - Flash
- VTT microcontroller drivers for
  - MCU
  - Watchdog
  - GPT
- VTT CRYPTO

The set of VTT modules that are actually delivered depends on the configuration of the customer's Software Integration Package (SIP), e.g., if MICROSAR FR is part of your delivery, then the VTT FlexRay driver will be delivered.

The scalability class and the support for multi-core configurations in the VTT OS depend on the customer's SIP and the OS that is delivered for the hardware.

Limitations
Due to the fact that a virtual ECU in vVIRTUALtarget is executed on the PC it does not provide an accurate simulation of a microcontroller. In particular, the runtime behavior of the virtual ECU may deviate from the behavior on the target hardware.

Please note the following main limitations when using vVIRTUALtarget:

- **Memory Protection (SC3)**
  The protection mechanism is not executed, even if memory protection (SC3) can be configured in the OS (depends on your SIP).
- **Post-Build Loadable**
  Post-build loadable can be selected as implementation variant in VTT modules, however, the post-build time flash process is not supported and will have to be tested on the hardware.
- **Precision Time Protocol**
  As VTT does not provide an accurate timing simulation, time synchronization protocols are not supported.
- **Ethernet Hardware Time Stamping**
  Ethernet hardware time stamping is not supported in VTT.
- **Ethernet Driver Features**
  The Ethernet features Quality of Service (QoS) and Forwarding and Queuing for Time-Sensitive Streams (FQTSS) are not supported in VTT.
- **Limited Transceiver Functionality**
  Transceiver drivers have limited functionality; in particular, VTT does not support simulation of partial networking for selective wake-up of an ECU.

- **SPI Communication**
  VTT does not support SPI transmission functionality, i.e., the VTT SPI driver only serves as a stub.
- **Multiple Instances of Modules**
  VTT modules cannot be instantiated multiple times in a single configuration project.
- **SoAd BSD Socket**
  VTT does not support BSD Socket.

Required Software
To run VTT on a PC the following software must be installed:

- CANoe 8.5 or later is used as runtime environment. A license of all used bus systems is mandated.
- DaVinci Configurator Pro 5 is needed to enable the configuration of VTT modules.
- Licensed vVIRTUALtarget basic is an editor for VTT projects. Basic features are:
  - Configuration of source code files, including paths and static libraries
  - Configuration of simulation parameters
  - Generation of configuration codes for the virtual ECU
  - Generation of a Microsoft Visual Studio solution
- Microsoft Visual Studio (2013, free Express edition is sufficient)
- Microsoft Windows 7 (64 bit)

7 Definition of SLP/HLP/SIP, Maintenance and Release Types

7.1 Software License Package (SLP)

The SLP includes the license and usage rights (see quotation) for a hardware-independent module (e.g. a COM layer) based on a defined specification.

Depending on the type of module, the software will be based on the following technical specifications:

- AUTOSAR, ASAM, OSEK/VDX, ISO, HIS, etc.
- OEM requirements (for OEM-specific deliveries)
- Customer-specific requirements (optional)

The following work products are licensed by purchase of an SLP:

- The module as source code
- Detailed technical documentation
- Generator plug-in to generate ANSI C code
- Configuration files for convenient configuration with Vector configuration tools

The module will be tested using a component test suite on a standard hardware platform (CANoe Emulation). Integration testing on the specific µController and delivery will be performed when a SIP is purchased (see section 7.3).

7.2 Hardware License Package (HLP)

The HLP includes the license and usage rights (see quotation) for a hardware-dependent module (e.g. CAN Driver) and is valid for a combination of compiler (version independent), microcontroller family and relevant hardware (e.g. CAN cell).

Depending on the type of module, the software will be based on the following technical specifications:

- AUTOSAR, ASAM, OSEK, ISO, HIS, etc.
- OEM requirements (for OEM-specific deliveries)
- Customer-specific requirements (optional)
- Specification of microcontroller and compiler

The following work products are licensed by purchase of an HLP:

- The module as source code
- Detailed technical documentation
- Generator plug-in to generate ANSI C code
- Configuration files for convenient configuration with Vector configuration tools.

The module will be tested using a component test suite on a derivative of the microcontroller family which is defined by Vector. Integration testing on the specific µController and delivery will be performed when a SIP is purchased (see section 7.3).

7.3 Software Integration Package (SIP)

The SIP includes integration, test, release and delivery of the modules that are licensed by an SLP and/or an HLP.

To perform integration and closely test the customer's use case, the following must be defined in accordance with the customer:

- Microcontroller derivative
- Compiler, compiler version, and compiler options
- Use case (e.g. number of CAN channels)
- Car manufacturer (regarding communication description, pre-configuration…)

For each SIP, all delivered work products will be added to a configuration management system, thus allowing redeliveries for a minimum of 10 years (excluding the Beta SIPs).

7.4 Mini SIP

The Mini SIP includes the delivery of individual modules. The software is tested on module level. The delivery is not tested on the target hardware and compiler. But Vector provides remedy of defects if they are based on defined processor and compiler mentioned in the quote.

7.5 SIP Types

7.5.1 Beta

A Beta SIP can only be ordered in combination with a Production SIP. Depending on the agreement with the customer, the software may include the complete or reduced functionality. The software is only preliminary integrated and tested. The usage of a Beta SIP for serial production is prohibited. The Beta SIP software may only be used for test purposes. Vector provides no warranty and/or liability to the extent permitted by law or statute.

7.5.2 Production

A Production SIP is a package which includes the release for serial production.

7.5.3 Update

An Update SIP is a package which replaces a previous Production SIP. An Update SIP is necessary if an additional delivery is needed because of modified requirements (compiler version, options, functionality, etc.). The delivery will be performed according to the existing contractual agreement and must be scheduled with Vector.

Note
Update SIPs have an impact on the issue reporting and remedy of defects. Please refer to chapter 2.14 Issue Handling for details.

7.5.4 Prototype

A Prototype SIP is a special package which may only be purchased by a Tier 1. The use is limited to a defined application area, e.g. prototyping. The software is only preliminary integrated and tested. The SIP is only available for platforms which are already supported by Vector. License packages (HLP, SLP) are not necessary. An upgrade to a Production SIP is only possible if identical technical features are used and the Prototype SIP is not older than one year. Vector provides no warranty and/or liability to the extent permitted by law or statute.

7.5.5 Overview SIP Types

The following table lists the scope of services, deliverables and release types for each SIP Type

	SIP Type / Mini SIP	Beta	Production	Update	Prototype
Scope of Services	Integration and test of selected modules	■	■	■	■
	Delivery	■	■	■	■
	Remedy of defects		■	■	
Deliverables	Software (Source or object code dependent on the contractual agreement - NDA)	■	■	■	■
	Documentation	■ 1	■	■	■ 1
	Test report		■ 4	■	
	Demo application	■ 2	■ 2,4		
Release Types[3]	Quickfix[5]	■			
	Development[5]	■		■	■
	Pre Production[5]		■	■	
	Production		■	■	
	Production (Safety-ready)		■	■	
	Production (Safe)		■	■	

1 The documentation is not necessarily complete
2 Depends on the OEM package
3 Release Types are described in detail on page 6
4 Not part of the Mini SIP
5 Non-production delivery, please refer to 7.5.6

Table 7-1 Overview SIP Types

7.5.6 Usage of Non-Production Deliveries

Non-Production Deliveries are basically operable, but not sufficiently tested, verified and/or qualified for use in series production and/or in vehicles operating on public or non-public

roads. In particular, without limitation, the Non-Production Deliveries may cause unpredictable ECU behavior, may not provide all functions necessary for use in series production and/or may not comply with quality requirements which are necessary according to the state of the art. Non-Production Deliveries must not be used in series production.

The Release Type of a delivery is defined in the Delivery Description that is part of the delivery.

7.6 Maintenance

Maintenance of Vector embedded software is available for the SLP, HLP and SIP. Maintenance generally includes bug-fixing.

Note
Only the latest delivered version will be maintained.

7.6.1 SLP Maintenance

The maintenance of the SLP includes the adaptation of the appropriate working products (components, documentation, generators …). The SLP Maintenance includes:

- Adaptations caused by “minor“ or “patch“-version modifications of the related AUTOSAR specification. In case of new functions, the extension of the licensed functionality has to be discussed with Vector.
- Adaptations caused by comparable changes of other specifications based on Vector’s assessment.
- Registered RfCs for the AUTOSAR Specification are included as far as they are required by the OEM.

Deliveries are not included as part of SLP maintenance; they are part of SIP maintenance.

The SLP maintenance period begins with the first delivery.

7.6.2 HLP Maintenance

The maintenance of the HLP includes the adaptation of the appropriate work products (components, documentation, generators …). The HLP maintenance includes:

- Adaptations caused by “minor“ or “patch“-version modifications of the related AUTOSAR specification. In case of new functions, the extension of the licensed functionality has to be discussed with Vector.
- Adaptation caused by comparable changes of other specifications based on Vector’s assessment.
- Adaptation caused by minor hardware changes based on Vector’s assessment.
- The implementation of simple workarounds for hardware issues.

Deliveries are not included as part of HLP maintenance; they are part of the SIP maintenance.

The HLP maintenance period begins with the first delivery.

7.6.3 SIP Maintenance

The SIP Standard Maintenance includes:

> Reporting about known issues (“Active Issue Reporting“).

> One delivery per year (Update SIP or update of a Beta SIP)

Note
This update will be delivered on customer request.

> If a bug-fix delivery is needed on short notice such a delivery will be performed without comprehensive tests. In this case the status of the delivery will be a Beta or Pre Production (see Release Types in chapter 7.7).
A Production Release has to be scheduled with Vector separately.

The SIP maintenance period starts with the first delivery of the product (Beta SIP or Production SIP).

In addition to the SIP Standard Maintenance, Vector offers SIP Extended Maintenance and SIP Production Maintenance.

SIP Extended Maintenance contains the same features as SIP Standard Maintenance. It includes one further delivery per year (Update SIP or Beta SIP).

SIP Production Maintenance only includes reporting about known issues (“Active Issue Reporting“). But Vector provides remedy of defects in case of high or critical issues. SIP Production Maintenance requires Standard or Extended Maintenance for more than 2 years.

7.6.4 Differentiation of Warranty Versus Maintenance

The table shows the scope of services depending on validity of warranty and/or maintenance:

	SIP Maintenance running	**SIP Maintenance expired**
Warranty period running	Bugfixing ✓	Bugfixing ✓
	Issue reporting ✓	Issue reporting ✓
Warranty period expired	Bugfixing ✓	Bugfixing ✗
	Issue reporting ✓	Issue reporting ✗

Table 7-2 Scope of services

7.7 Release Types

The release type gives information about the scope of function, quality and application field restrictions of the delivered software. The release type depends on the delivered Software Integration Package and is defined in the delivery description.

7.7.1 QuickFix

A QuickFix delivery is an immediate reaction on customer requests. The content of a QuickFix delivery is individually agreed with the customer. Only available as supplement to a development release.

7.7.2 Development

The software includes operational standard functions. Feature extensions of the software are not verified finally and the development process is not fully implemented. The usage of the software is intended for development phase or prototyping.

7.7.3 Pre Production

The software is operational but verification measures have not been completed. The usage of the software is intended for development and preproduction phase.

7.7.4 Production

For all features included in the software planned verification measures have been completed and all known issues are documented. The software can be used for series production as QM software if the documented issues are considered.

7.7.5 Production (Safety-ready)

For all features included in the software planned verification measures have been completed and all known issues are documented.

Safety activities for components required as ASIL have been completed. The delivery includes the Safety Manual as well as all required tools (e.g. MICROSAR Safe Silence Verifier and RTE Analyzer) in a released version.

The software can be used for series production as QM software if the documented issues are considered.

Note
The Safety Case has to be ordered separately in order to receive a delivery with Release Type Production (Safe).

7.7.6 Production (Safe)

This delivery comprises the Safety Case documentation. The Safety Case is created based on the delivery with release type Production (Safety-ready).

The components listed in the Safety Case can be used as ASIL software in series production if the documented issues and the Safety Manual are considered.

8 Contact

Visit our website for more information on

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

www.vector.com