VECTOR

MICROSAR 4

Product Information

Version 1.04.08

Authors	Hannes Haas
Status	Released

Contents

1 Introduction 5

2 General Information 6
2.1 Compiler Warnings 6
2.2 64-bit Microcontroller Support 6
2.3 Usage of Float Data Types 6
2.4 Delivery Conditions 6
2.5 Delivery of Beta Modules/Features 7
2.6 Delivery of Stub Modules 7
2.7 Delivery of Code Templates 7
2.8 Delivery of Example Code 8
2.9 DaVinci Configurator Product Variants 8
2.10 Usage of Features that are not Licensed 8
2.11 Fulfillment and Traceability 9
2.12 Test Report 9
2.13 MISRA 9
2.14 Test of External Hardware 9
2.15 Test of Third Party Modules 9
2.16 Issue Reporting 9
2.16.1 Issue Reporting for Third Party Software 10
2.16.2 Issue Reporting for DaVinci Tools 10
2.17 Warranty 10
2.17.1 Third Party Modules 10

3 Definition of SLP/HLP/SIP, Maintenance and Release Types 11
3.1 License Parameters 11
3.2 Software License Package (SLP) 11
3.3 Hardware License Package (HLP) 12
3.4 Software Integration Package (SIP) 12
3.5 Mini SIP 13
3.6 SIP Types 13
3.6.1 Beta 13
3.6.2 Production 13
3.6.3 Update 13
3.6.4 Prototype 13
3.6.5 Overview SIP Types 13
3.6.6 Usage of Non-Production Deliveries 14
3.7 Maintenance 14
3.7.1 SLP Maintenance 15

3.7.2 HLP Maintenance 15
3.7.3 SIP Maintenance 15
3.7.4 Differentiation of Warranty Versus Maintenance 16
3.8 Release Types 16
3.8.1 Development 16
3.8.2 Pre Production 17
3.8.3 Production 17
3.8.4 Production (Safety-ready) 17
3.8.5 Production (Safe) 17
3.9 QuickFix Deliveries 17

4 Customer Development Environment 18
4.1 Lead Time 18
4.1.1 Provided Development Environment 18
4.1.2 Joint Workshop 18
4.2 Project Organization 19

5 Modules and Options 20
5.1 vCanCcCdm 20
5.2 vCanCcGbt 20
5.3 CanTrcv (generic) 20
5.4 Cpl Crypto 20
5.5 Crypto (vHsm) 20
5.6 Crypto (Sw) 21
5.7 vDbg (Xcp-based) 23
5.8 Dlt (Xcp-based) 23
5.9 Dlt (AUTOSAR) 23
5.10 Drivers (Can, Lin, Fr, Eth, MCAL) 23
5.11 vEap 23
5.12 Eth (ext) 24
5.12.1 Requirements of Eth (ext) Implementation 24
5.12.2 Requirements of the Hardware Implementation 25
5.13 EthSwt (ext) 26
5.13.1 Requirements of EthSwt (ext) Implementation 26
5.13.2 Requirements of the Hardware Implementation 27
5.14 EthTrcv (generic) 28
5.15 vEtm 28
5.16 Fee 28
5.17 FrTrcv (generic) 29
5.18 IDM - Variant Handling / Post-Build Selectable 29
5.19 LinIf (Slave) 30

5.20 LinTrcv (generic) 30
5.21 MCAL Integration Package 30
5.22 Mixed ASIL Partitioning & BSW Split 31
5.23 PBL - Post-Build Loadable 31
5.24 Rte 33
5.25 vRtm 33
5.26 vScc 33
5.27 vSecMod (Vector) 33
5.28 TcpIp Add-On vIpSec 33
5.29 vTls 34
5.30 WdgM 34
5.31 Xcp (on Can, Fr, Eth) 34

6 Services 35
6.1 MICROSAR Getting Started 35
6.2 MICROSAR Getting Started CHARGE 35

7 SIP Extensions 37
7.1 Customer Hardware 37
7.2 Start Application 37
7.3 vVIRTUALtarget (VTT) 38

8 Contact 41

1 Introduction

All modules and options are described in the general MICROSAR Product Information (refer to http://vector.com/vi_microsar_en.html).

In addition to the general MICROSAR 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 Conditions

The delivery dates stated in the quotation are subject to the provision of the required development environment and tools, as well as required Third Party Software, to Vector free of charge and in time.

This includes:

- compiler and linker
- evaluation board
- debugger and emulator
- if applicable: Third Party Software such as MCAL, BSP, startup code, CRY
- if applicable: POSIX operating system
- if applicable: hardware specific board support package

A detailed list of required items is provided after order.

SPI connected devices and related MCAL drivers can only be tested during SIP Integration if Customer Hardware (refer to chapter 7.1 Customer Hardware) has been ordered.

It is assumed that the development hardware can be operated without difficulties and is fully documented. Quotations are based on the assumption that the effort to install the hardware does not exceed one day. If the installation requires more time we will invoice the additional efforts, after having consulted you.

It is the customer's responsibility to select an appropriate device/derivative which provides enough resources (RAM, Flash, Throughput) to accommodate the quoted software.

Please note that the derivative and compiler version need to be defined at the latest with the purchase order.

2.5 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.

2.6 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.

2.7 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.

2.8 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 and - optionally - in the folder .\Application of the SIP.

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

2.9 DaVinci Configurator Product Variants

The variant DaVinci Configurator Ltd can be used alternatively to DaVinci Configurator Pro. The usage is only possible in combination with a MICROSAR SIP that enables the usage of DaVinci Configurator Ltd. You find according information in the Customer Report of your MICROSAR SIP.

2.10 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.

2.11 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.

2.12 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.

2.13 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.

2.14 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.

2.15 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.

2.16 Issue Reporting

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.

2.16.1 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.

2.16.2 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

2.17 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.

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

3.1 License Parameters

The license parameters mentioned in your quote are defined as follows:

License Parameter	Description
ECU supplier	The license is restricted to one ECU supplier (Tier 1). There is no restriction to a local site. The license is valid worldwide.
ECU product	The product parameter comprises a restriction to one ECU product. An ECU product is defined as an ECU that appears once in a vehicle system (e.g. Airbag ECU). There are no restrictions to variants of the ECU product (e.g. low-end, high-end). ECUs with multiple instances in a vehicle system (e.g. Door Module or Seat Module) are considered as one ECU product.
OEM	The OEM parameter comprises a restriction to one OEM. All brands that belong to the OEM at the time of the specific quotation are covered. Those brands are listed in the quote.
Hardware and Compiler	The hardware and the compiler parameter comprises a restriction to one microcontroller platform (reference derivative) and one compiler vendor. Derivatives within one microcontroller platform (according to the definition of the semiconductor vendor) can require different implementations of hardware dependent modules. Since Vector has no influence on the evolution of these microcontroller platforms the license is restricted to one known relevant hardware technology as reference (= reference derivative). Depending on the compatibility of hardware dependent modules with the reference derivative, an existing license covers a derivative or not. This will be decided and explained by Vector case by case. There are no restrictions to the compiler version and options.

3.2 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 3.4).

3.3 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 3.4).

3.4 Software Integration Package (SIP)

The SIP includes integration, test, release and delivery of the modules that are licensed as SLP and/or 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…)

The requested derivative is specified in the customer report. For reasons of availability, we reserve the right to perform the test on a compatible derivative which is not identical with the requested derivative.

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).

3.5 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.

3.6 SIP Types

3.6.1 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.

3.6.2 Production

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

3.6.3 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 2.16 Issue Reporting for details.

3.6.4 Prototype

Prototype SIPs are intended for evaluation of existing product functionality prior a following production project. The Prototype SIP software is only preliminary integrated and tested. For a Prototype SIP License packages (HLP, SLP) are not required.

The usage of a Prototype SIP for serial production is prohibited. Vector provides no warranty and/or liability to the extent permitted by law or statute.

3.6.5 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]	Development[5]	■		■	■
	Pre Production[5]		■	■	
	Production		■	■	
	Production (Safety-ready)		■	■	
	Production (Safe)		■	■	
	Quickfix[5]	■			

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 3.6.6

Table 3-1 Overview SIP Types

3.6.6 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.

3.7 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.

3.7.1 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.

3.7.2 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.

3.7.3 SIP Maintenance

The SIP Standard Maintenance includes:

- reports on 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 3.8).
> 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 Production Maintenance.

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

3.7.4 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 3-2 Scope of services

3.8 Release Types

The release type defines the scope of function and quality of the delivered software. It therefore defines the possible application areas of the delivered software (e.g. can the software be used for serial production or not). The release type is defined in the SIP’s Delivery Description. The Delivery Description file is located in the SIP folder: .\Doc\DeliveryInformation\DeliveryDescription_<CBD number>.html.

3.8.1 Development

The software may not be feature complete and the development process has not yet been completed. The usage of the software (or parts of it) is intended for development phase usage or prototyping only.

3.8.2 Pre Production

The software is operational but verification measures have not been completed. The usage of the software (or parts of it) is intended for development and preproduction phase usage only.

3.8.3 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.

Deviations to this are documented as defined in chapter 2 General Information.

3.8.4 Production (Safety-ready)

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

Deviations to this are documented as defined in chapter 2 General Information.

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).

3.8.5 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.

3.9 QuickFix Deliveries

A QuickFix delivery is an immediate reaction on customer requests. Its content is individually agreed with the customer.

Software that is shipped as part of a QuickFix delivery must not be used for production.

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 Modules and Options

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

5.1 vCanCcCdm

This module implements the CHAdeMO charging protocol Version 1.1.0 of the CHAdeMO Association. The component provides backward compatibility for wall-mounted charging stations that communicate with version 0.9.1 or 1.0.1.

5.2 vCanCcGbt

This module implements the Chinese Standard GB/T 27930-2015. The Appendix B ("Charger and BMS Fault Diagnosis Messages") is not supported.

5.3 CanTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.7 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

5.4 Cpl Crypto

The library of the crypto algorithms is listed as item 5D002 in the Addendum to the EU Dual-Use Regulation (EU) no. 428/2009 in the version of the delegated regulation (EU) no. 1382/2014 by the Commission of 10-22-2014.

Fulfillment of the contract by Vector is subject to the provision that contract fulfillment does not violate any legal regulations and/or government directives related to foreign trade law.

5.5 Crypto (vHsm)

This module allows accessing the hardware secure module (HSM) from a MICROSAR BSW stack using Csm. It requires Vectors HSM firmware (vHsm) to be available on the HSM core. See the HW specific product information for details on vHsm.

Crypto (vHsm) and 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.

5.6 Crypto (Sw)

Crypto (Sw) supports the following crypto algorithms:

Algorithm / Function	Standard	Crypto (Sw)	CSM 4.3 Service	CSM 4.3 Interface
SHA-2 (SHA-256)	FIPS 180-4	x	Hash Functions	Hash Interface
SHA-2 (SHA-384)	FIPS 180-4	x		
SHA-2 (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		
AES-CMAC (based on AES 128)	IETF RFC4493	x	Message Authentication Code (MAC) Generation and Verification	MAC Interface
AES-CMAC-96 (based on AES 128) via manual trunction	RFC4494	x		
AES-GMAC (based on AES 256)	IETF RFC4543	x		
Poly1305	IETF RFC7539	x		
SipHash		x		
AES-128,256 in the modes ECB/CBC	FIPS-197	x	Symmetric Encryption and Decryption	Cipher Interface (Symmetric)
AEAD: AES 128, 256 in GCM mode	NIST 800-38D	x		
ChaCha20	IETF RFC7539	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 (Asymmetric)
Asymmetric encryption and decryption based on RSA with Optimal Asymmetric Encryption Padding (RSA-OAEP)	IETF RFC8017	x		
		x x x x	Key Handling	Key Setting Interface Key Extraction Interface Key Copying Interface Key Generation Interface
KDF in Counter Mode, KDF in Counter Mode with Appendix, concatenation KDF (NIST 800-56A)	NIST SP 800-56A	x		Key Derivation Interface
Key exchange using the Elliptic Curve Burmester-Desmedt (EC-BD) with X25519		x		Key Exchange Interface
Key exchange using the Elliptic Curve Diffie-Hellman protocol EC-DHE with ANSIp256r1, SECp256r1, SECp384r1 and X25519	ANSI X9.63	x[1]		

Algorithm / Function	Standard	Crypto (Sw)	CSM 4.3 Service	CSM 4.3 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 RFC8032	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 NIST P384 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x		
Digital signatures based on the Elliptic Curves: ECDSA SEC P256 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x		
Digital signatures based on the Elliptic Curves: ECDSA SEC P384 R1 PreHashing: None, SHA-1, SHA-256	ANSI X9.62-2005	x		
Certificate installation and update according to ISO15118	ISO15118	x		
Symmetric Key Update Protocol	SHE1.1	x		

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

Table 5-1 Supported crypto algorithms Crypto (Sw)

Caution

The library of the crypto algorithms for the Crypto (Sw) is listed as item 5D002 in the Addendum to the EU Dual-Use Regulation (EU) no. 428/2009 in the version of the delegated regulation (EU) no. 1382/2014 by the Commission of 10-22-2014.

Fulfillment of the contract by Vector is subject to the provision that contract fulfillment does not violate any legal regulations and/or government directives related to foreign trade law.

5.7 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.

5.8 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.

5.9 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.

5.10 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.

5.11 vEap

The vEap module implements the authentication according to IEEE 802.1X.

The Integrity Check Value is tested with the Csm hash functions

- HMAC-SHA1,
- HMAC-SHA256.

vEap requires a MICROSAR.ETH stack and Csm from MICROSAR.CRYPTO.

Note
From R22 this module is available as Beta Module (refer to chapter 2.5 Delivery of Beta Modules/Features).

5.12 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.).

5.12.1 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
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 5-2 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 5-3 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 5-4 Eth (ext) requirements on interconnection between external device and microcontroller

5.12.2 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 5-5 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 5-6 Hardware requirements on other AUTOSAR BSW modules

5.13 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.).

5.13.1 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 5-7 EthSwt (ext) requirements on the AUTOSAR SPI driver

5.13.2 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	Broadcom		NXP	
Devices	BCM89200 BCM8950x	BCM89230 BCM8953x BCM8954x BCM8955x	SJA1105 SJA1105T	SJA1105P SJA1105Q SJA1105R SJA1105S
SPI Property				
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 5-8 Hardware requirements on AUTOSAR SPI-Driver

5.14 EthTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.7 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

5.15 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 be removed from the ECU.

5.16 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 analyzes.

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.

5.17 FrTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.7 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

5.18 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 (LIN): Lin Slave is not supported.

- 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.

5.19 LinIf (Slave)

LinIf (Slave) does not support LIN Configuration Services and Unconditional Frames. Please inform Vector if this functionality is required.

5.20 LinTrcv (generic)

The module allows customer specific implementation. Therefore, a code template is delivered that provides an implementation framework (please refer to chapter 2.7 Delivery of Code Templates). The transceiver template implementation design requires that the transceiver changes the internal state synchronous to the state change request.

5.21 MCAL Integration Package

The MCAL Integration package covers the integration of the following third party MCAL modules to the MICROSAR stack and DaVinci Configurator Pro: Adc, Dio, Eep, Fls, Gpt, Icu, Mcu, Ocu, Port, Pwm, Spi, Wdg.

The integration of a third party module is only done if

- the module is part of the provided 3rd party MCAL package,
- the module is AUTOSAR compliant and derived from the AUTOSAR Standard Definition.

It is the sole responsibility of the customer to test the freedom from defects and the functionality of the third party modules. Testing these modules is not part of the MCAL Integration Package and will not be carried by Vector. AUTOSAR compliant BSW description files need to be provided by the customer as a prerequisite for the integration into the DaVinci Configurator Pro.

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. The 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. Please ensure that the MCAL to be provided by you is released by the MCAL supplier for your desired compiler version.
- 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 must 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.

5.22 Mixed ASIL Partitioning & BSW Split

The feature BSW Split is limited to MICROSAR.ETH.

Due to the need of cross core synchronization in modules such as PduR an improvement in the CPU load on the main BSW core cannot be guaranteed. The achievable CPU load reduction is highly project specific and the integration complex. Hence it is recommended to use BSW Split only if there is an acute or impending CPU overload situation on the BSW core.

Vector recommends prior consultation before using the option BSW Split.

5.23 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 (LIN): Lin Slave is not supported.
- 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.

Note
The binary/hex files generated by the MICROSAR Post-Build Loadable tool chain are neither intended nor qualified for use in series production without applying suitable quality measures. The resulting ECU configuration 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.

5.24 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.

5.25 vRtm

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

5.26 vScc

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

5.27 vSecMod (Vector)

vSecMod (Vector) implements the Freshness Value Manager (FVM) SWC based on AUTOSAR_SWS_SecureOnboardCommunication (Version 4.4) Annex A: Application hints for the development of SW-C Freshness Value Manager. vSecMod (Vector) supports the FVM based on decoupled counters.

vSecMod (Vector) does not implement OEM specific requirements.

5.28 TcpIp Add-On vIpSec

The TcpIp Add-On vIpSec allows to establish an IPsec communication according to IETF RfC 4301. The functionality is restricted to transport mode and the usage of Authentication Header only according to RfC 4302. The Authentication Header adds data integrity and data authentication to the payload but does not allow confidentially.

The Authentication Header is created and validated using Csm functionality. Supported Csm crypto functions see below. The module includes Internet Key Exchange v2 (IKEv2) which allows to set up a Security Association (SA).

The TcpIp Add-On vIpSec is tested with the Csm crypto functions

- Encryption: AES-CBC,
- Integrity: HMAC-SHA-256-128,
- Diffie-Hellman: ECP 256,

and allows authentication based on RSA X.509v3 certificates. This functionality requires certificate parsing.

The TcpIp Add-On vIpSec requires a MICROSAR.ETH stack and Csm from MICROSAR.CRYPTO.

Note
From R22 this module is available as Beta Module (refer to chapter 2.5 Delivery of Beta Modules/Features).

5.29 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.

5.30 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.

5.31 Xcp (on Can, Fr, Eth)

Xcp (on Can, Fr, Eth) is implemented according to ASAM Specification 1.1, except the feature CAN-FD support, that is implemented according to ASAM Specification 1.2.

6 Services

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

6.1 MICROSAR Getting Started

The following requirements must be prepared and provided by the customer for the MICROSAR Getting Started workshop:

- At least one PC must be available during the workshop including
  - installed MICROSAR SIP,
  - installed and licensed DaVinci Configurator Pro 5.x,
  - installed and licensed third party MCAL including generation framework,
  - installed compiler/linker in the version and with the options which were specified in the Vector Questionnaire Step 2,
  - installed and licensed CANoe 8.5 or higher,
- debugger for the specific hardware including all tools and valid debugger hardware and software licenses,
- working Evaluation Board or ECU for the ordered derivative including power supply, debugger connection, and communication interfaces to connect CANoe,
- Hardware Network Interface for each used networking technology.

Missing artefacts can result in missing the MICROSAR Getting Started objectives or can be the root cause for additional costs.

6.2 MICROSAR Getting Started CHARGE

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

7 SIP Extensions

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

7.1 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.

7.2 Start Application

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 and diagnostics prior to the project start.

The Start Application is included in the initial MICROSAR SIP delivery.

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

7.3 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. VTT MCAL modules do not support implementation variant post-build-loadable.
- **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. Furthermore, generic Ethernet drivers are not supported.
- **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 (Express/Pro) or 2017 (Community/Pro)
- Microsoft Windows 7 (64 bit)

8 Contact

Visit our website for more information on

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

www.vector.com