VECTOR

Open Alliance TC8 / Getting Started with the MICROSAR Etm

Technical Reference

Overview and Guidance

Version 1.0.0

Authors	Johannes Kempf
Status	Released

Contents

1 Overview

1.1 TC8 and IETF

The Open Alliance (OA) provides with Tech Committee 8 (TC8) a comprehensive set of conformance test cases

See also https://www.opensig.org/tech-committees/tc8/

- The OA TC8 test specs ensure compatibility against Internet Engineering Task Force (IETF) Request for Comments (RfC) conformance of a TCP/IP stack.
  - The OA TC8 tests only covers relevant use cases for the automotive industry - Example: RFC 793 - Transmission Control Protocol (TCP)
  - OA TC8 defines many (~ 1000) test cases for TCP/IP
- In addition to TCP/IP OA TC8 also defines conformance test for the SOME/IP protocol (ETS – **E**nhanced **T**estability **S**ervice) and
- Physical layer tests.

The specification is available for public here: OA TC8 Spec v2.0

Figure 1-1 Open Alliance TC8 Specification Releases

1.2 AUTOSAR TCP/IP Conformance Tests

- AUTOSAR defines own tests that are now covered by TC8 …but come along with own numbering.
- AUTOSAR tests can be considered **obsolete** and are not further maintained.
- AUTOSAR specifications:
  - AUTOSAR_ATS_IPv4
  - AUTOSAR_ATS_TCP
  - AUTOSAR_ATS_UDP
  - Specs are available on https://www.autosar.org/standards/acceptance-test/

    E.g. version 1.2.0: https://www.autosar.org/fileadmin/user_upload/standards/tests/1-2/Specifications.zip

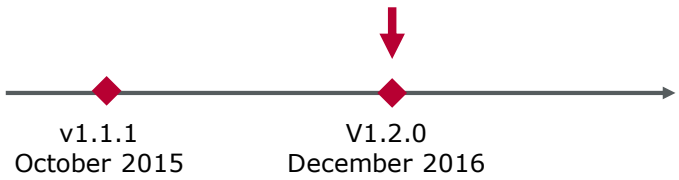


Figure 1-2 AUTOSAR TCP/IP Conf. Test Specification Releases

1.3 Comparison of Test Specification Scope

> AUTOSAR is referring to IETF functionality when it comes to TCP/IP, TLS and in the close future IPsec+IKE

> There is a big overlap

> Furthermore, AUTOSAR defines SOME/IP which is not part of IETF

> The Open Alliance focuses on automotive relevant aspects

> AUTOSAR tests can be assumed to be covered by TC8

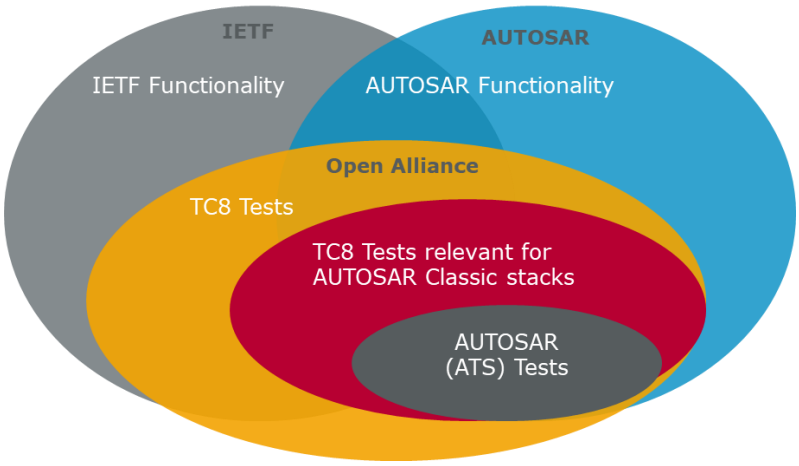


Figure 1-3 Scope of design and test specifications in comparison

1.4 Test Coverage and Customer Test Specifications

There is no 100% test coverage – because of good reasons:

- In a typical ECU project not all features are used e.g. if TCP is not used (also not configured), the TCP test cluster is not relevant
- Some test cases are just not applicable with AUTOSAR classic (they were designed for POSIX / Linux systems)
- MICROSAR has missing features – we just implement what our customers need
- MICROSAR has known issues
- Test systems also have issues
- Etc.

Customers tend to define own test specs

- This is OK. if they use (reference) OA TC8 and extend it to their needs, e.g. TC8 v2.0 does not cover IPv6 and SOME/IP TP
  If the customer uses this functionality, it is OK to add proprietary tests
- If customers define own test specs *from scratch* and do not even reference the standard: we just have a mess: a lot of OEM-specific effort and results we cannot compare – this has to be avoided.

and sometimes customers want to extend the test protocol

- ETM allows that from R22 – but this requires project work

1.5 AUTOSAR Test Protocol (TCP/IP only)

> AUTOSAR defines a test protocol to be used to stimulate tests on the device under test (DUT)

> The specification for the protocol is called **AUTOSAR_PRS_TestabilityProtocolAndServicePrimitives**

> Vector provides a proprietary implementation of the test protocol with the MICROSAR module called **ETM** (please also see What is the module ETM and why do I need it?)

> The **ETM** acts as **upper tester** on the DUT and implements the testability protocol in version 1.1.0 and 1.2.0 (from MSR R22)

> Additional Service Primitives in v1.2.0 compared to initial version for: IP(v4), IPv6, ICMP, ICMPv6, DHCP, DHCPv6 + PHY

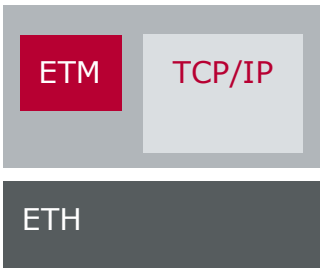


Figure 1-4 Relation ETM and TCP/IP

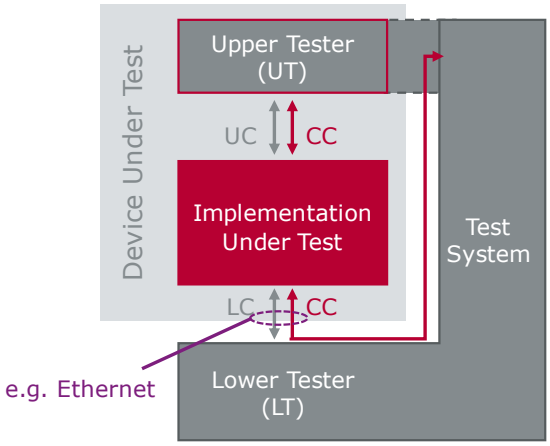


Figure 1-5 Scheme of the test environment (Original from Testability Protocol and Services, AUTOSAR)

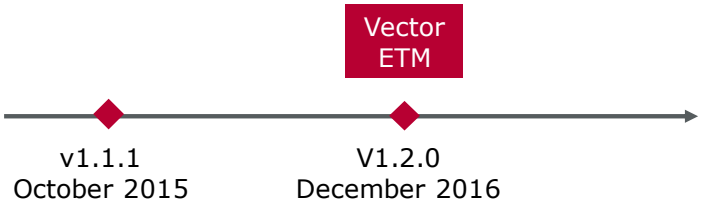


Figure 1-6 Version of Vector ETM

1.6 SOME/IP Conformance Tests with Enhanced Testability Service (ETS)

> In addition to TCP/IP tests within the OA TC8 SOME/IP conformance tests are defined

> Requirements on DUT side (in case of an AUTOSAR Classic ECU):
  > Comprehensive configuration which requires an ARXML ECU Extract of System Description including a SOME/IP service with data types etc.
  > Implementation (in C) of test service behavior
  > Both, extract and implementation are **not provided** by the OA TC8
  > So, the customer or OEM have to provide and maintain ARXML and service implementation

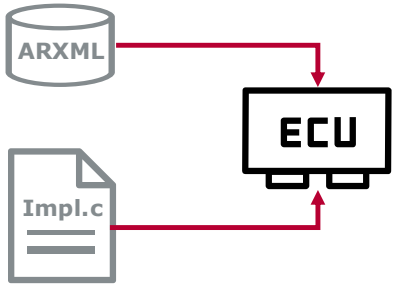


Figure 1-7 Artifacts necessary for SOME/IP tests

1.7 Exemplary Test Setup with Virtual MICROSAR Classic Environment

> Test setup for a TC8 conformance test against a Vector MICROSAR Classic stack may look like this:

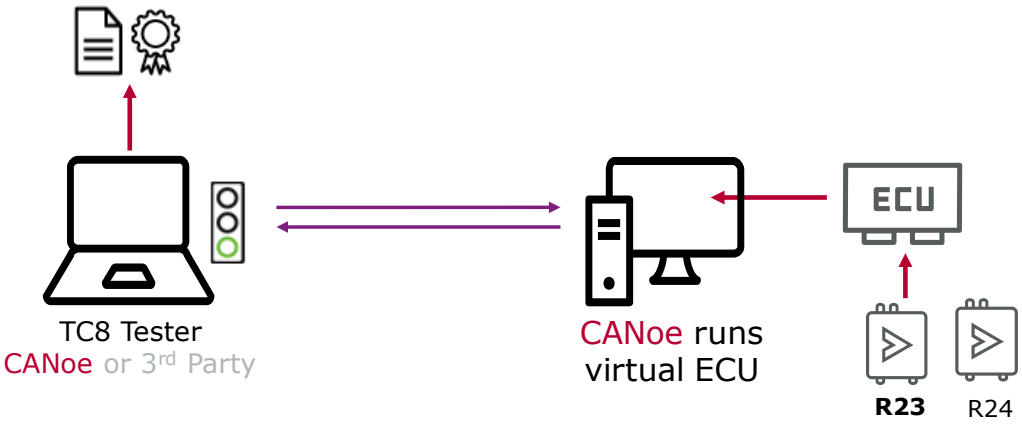


Figure 1-8 Test setup with Virtual MICROSAR Classic Environment

> MICROSAR SIP of a dedicated version runs on real hardware or as a virtual ECU on CANoe.emu using vVTT
> TC8 tester performs tests against virtual ECU on “real” Ethernet

1.8 Conformance Tests and the MICROSAR Stack

- TCP/IP conformance tests
  - ETM is used to stimulate the TCP/IP stack
  - ETM therefore implements the testability protocol
  - TCP/IP interfaces ETM (works with Vector TCP/IP only)
- SOME/IP conformance tests
  - Covers control path including SD and …
  - Data path including SOMEIPXF
  - Needs ARXML file for stack configuration
  - Needs service implementation for test stimulation

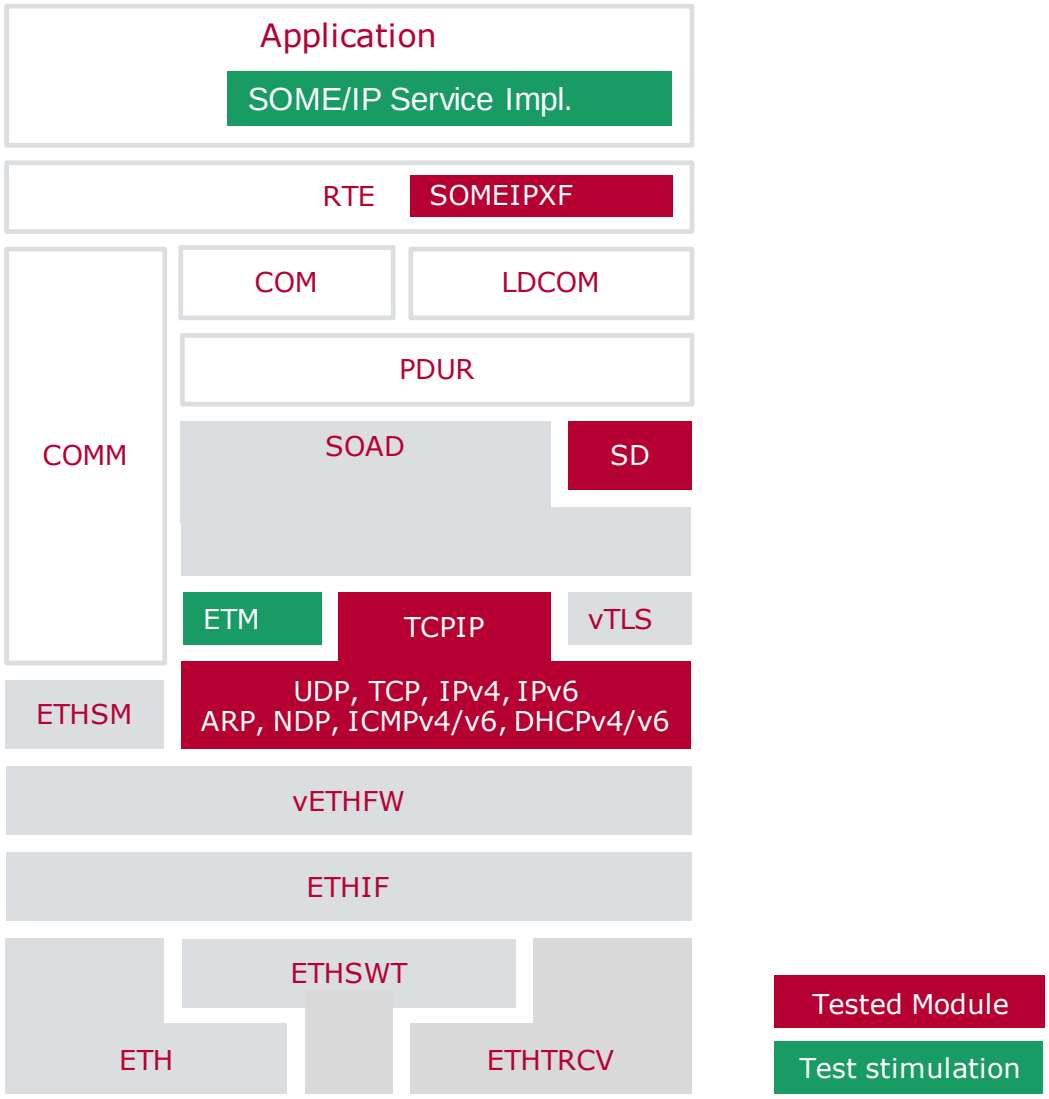


Figure 1-9 Overview MICROSAR Stack and TC8 conformance testing

1.9 Status

Current Status:

- ETM supports AUTOSAR_PRS_TestabilityProtocolAndServicePrimitives v1.1.0 and v1.2.0 (from MSR4 R22)
  - This allows to perform TC8 TCP/IP conformance test with CANoe and 3rd party tooling
  - Configuration that has been used for internal tests can be provided
- Conformance tests with CANoe are now executed automatically at Vector. Result (reports/traces and EcuC) can be provided on request under NDA
- SOME/IP tests are performed on component level only. No conformance tests yet at Vector in-house
  - Vector's TCP/IP stack and SOME/IP implementation are mature and already widely used in many productive projects.

1.10 Setup of Conformance Tests is Time Consuming!

- The TC8 test spec defines test ideas but not exactly how tests have to be implemented
  - This leads to test tool vendor specific test implementations, i.e. a test which is passed with Ixia may fail with Spirent
- TC8 defines a lot of test cases and it makes sense to discuss with the customer upfront which tests are mandatory, optional or can be skipped completely
  - This task is quite time consuming
- If a test fails, typically you need the ECU configuration, the Tester configuration and the trace to understand the failure
  - Debugging is a tough job and therefore time consuming too
- Setting up a test environment is not included with the ETM module!
  - This is a project and may take weeks with many iterations
  - Setting up a conformance test via support is not an option (see also Support / Coaching)
- To clarify test scope, test setup and test tool vendor upfront helps a lot

2 How to Get Started?

2.1 Getting an Overview

Before starting the practical part, we recommend you to get an overview of the whole topic by reading the Overview part in this article and making yourself familiar with the relevant documents.

You can freely download the OPEN Alliance documents from the homepage (www.OPENSIG.org) or via the members portal (https://members.opensig.org). The AUTOSAR documents are also freely available at (https://www.autosar.org). You cannot obtain them via Vector. To successfully implement and perform the TC8 tests, you must understand the contents of the following documents:

> OPEN Alliance Automotive Ethernet ECU Test Specification TC8 ECU Test, version 2.0 (or 3.x, as soon it's final)

> AUTOSAR Testability protocol and service primitives, version 1.2.0

Please also have a look into the section Relevant documents / links.

2.2 What is the Module ETM and why do I Need it?

The primary objective of the Etm module is to validate a remote TCP stack's adherence to standards as specified in AUTOSAR standard specifications for TCP. In order to trigger a TCP stack to send/receive TCP packets (which can be validated for compliance). So, to summarize, Etm module is an agent who resides on the TCP stack and can communicate with a remote tester over a command channel. The primary job of the Etm module is to get specific command from the test tool, interpret them and using socket API instruct the local TCP stack under test to take TCP actions to generate corresponding TCP packets.

Please also refer to AUTOSAR Test Protocol (TCP/IP only) and also have a deeper look into the corresponding technical reference (TechnicalReference_Etm.pdf) which gets delivered with your SIP, in case you have ordered the module. This document gives you more specific information on the module and fits to the MICROSAR release you are using.

2.3 Integrating the ETM Module into the Stack

In this section we want to roughly describe, which steps are necessary in order to integrate the ETM module into your stack.

Prerequisites
As a prerequisite please make sure that your configured ethernet stack works flawlessly on your hardware (or your virtual target) and you can see a proper ethernet communication in CANoe.

Level of detail
As the details of the integration steps might change from time to time (or from MSR release to MSR release), these instructions should not be understood as an exact Step-By-Step tutorial. We want to give a rough overview and as good as possible independent from your actual MSR release.

> **Add the module Etm** to your Configurator 5 configuration.

- After adding the module there should appear a bunch of solving actions in your validation tab. Try to **follow those validation messages and use the solving actions**. (you can also use “Solve all default actions”)

  E.g.:

  - Create Etm Socket Owner into TcpIp
  - Set default values for Etm SocketOwner
  - Enable/Add the LocalAddresses for IPv4 and/or IPv6 to be used in Etm (IPv4 or IPv6 unicast address in mandatory)
  - Create all missing Etm Tcp parameters
  - Create TcpIp SocketBuffers
  - etc.
- As soon as you do not see any errors anymore in Etm and TcpIp, you should be done with the first part.
- Make sure to enable the **initialization of the module** with the BSW Management and Initialization comfort editor in order to have the Etm_InitMemory() and and Etm_Init() properly called.
- Perform the **mapping of the Etm_MainFunction()** to a corresponding task. Please make sure that the Etm_MainFunction() cannot be interrupted and also doesn’t interrupt other Mainfunctions (e.g. Etm_MainFunction() should not interrupt the TcpIp or EthIf main function or vice versa)
- If these steps are properly done, the configuration for the **basic functionality** of the module should already be given.
- Now you should **generate your configuration** again, add the needed files in your build system and **compile the project**.

Go to the next chapter to find out how you can verify that the basic functionality of the module is working properly.

2.4 Confirming the Basic Runtime Functionality of the ETM

With your output of the section before (binary file or .dll in case of a virtual target) you can now let your ECU-software run on your hardware or on your virtual target. As first steps you should verify, that the Init-Functions and the main function of the ETM module gets called without calling any DET-Errors and the ECU runs stable with ethernet communication.

Next you can use the **TestabilityProtocolTester** CANoe configuration (there is a zip folder available in this KnowledgeBase article) for verifying the basic runtime functionality of your ETM integration. This small CANoe configuration offers you a very basic test infrastructure which allows you to send a valid test message / service primitive (`START_TEST`, PID 0x02, test group GENERAL) to the DUT which should get successfully answered by your DUT.

Service	START_TEST
Group	GENERAL
PID	0x02
Definition	The purpose of this SP is to be a defined entry tag in trace at the point in time the test case was started. This SP does not have any request parameters.

Table 2-1 **Start Test** Service as given by AUTOSAR_PRS_TestabilityProtocolAndServicePrimitives.pdf

One important step after opening the configuration is to adapt the default parameters to your project. Please adapt the IP-Address of your DUT which shall be used for the test (e.g. the IPv4/IPv6 Unicast address of the DUT) and also the UDP port you have configured in the ETM-module (`/MICROSAR/Etm/EtmConfig/EtmStubListenPort`):

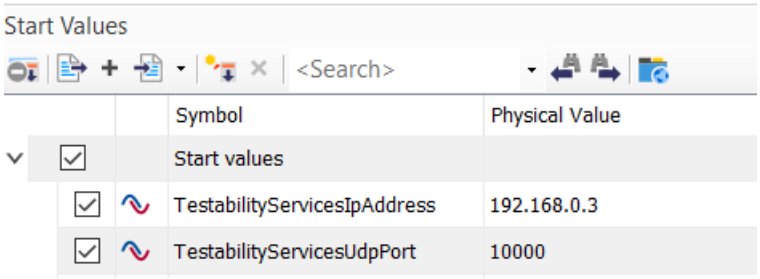


Figure 2-1 Input parameters for TestabilityProtocolTester

Please also make sure that the tester's IP address is part of the same subnet as your DUT (if you want to use IPv6, you need to enable the IPv6 functionality and add an IPv6 address in the same dialogue):

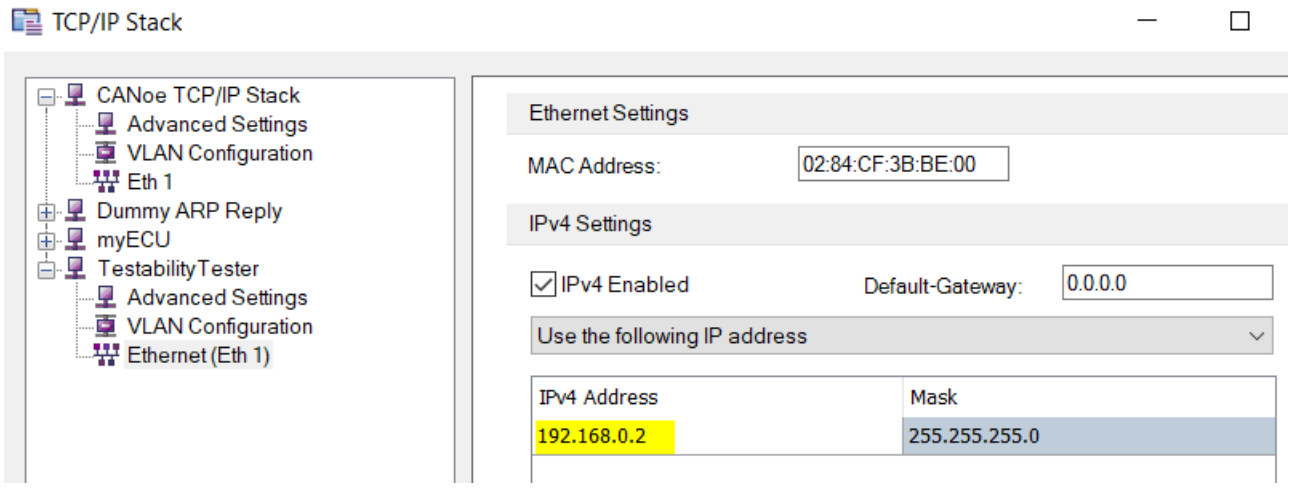


Figure 2-2 TCP/IP configuration in CANoe

And of course, your ECU needs to be connected to the CANoe Ethernet Channel (either as a virtual target / VTT binary or as a real hardware).

Then you can start the CANoe measurement / simulation and let the test run:

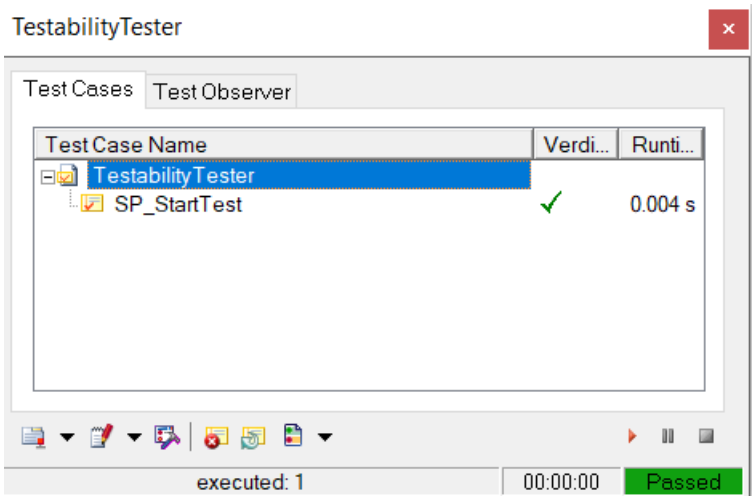


Figure 2-3 Test run in CANoe

In the trace window you can check the (simple) communication which was done during this test.

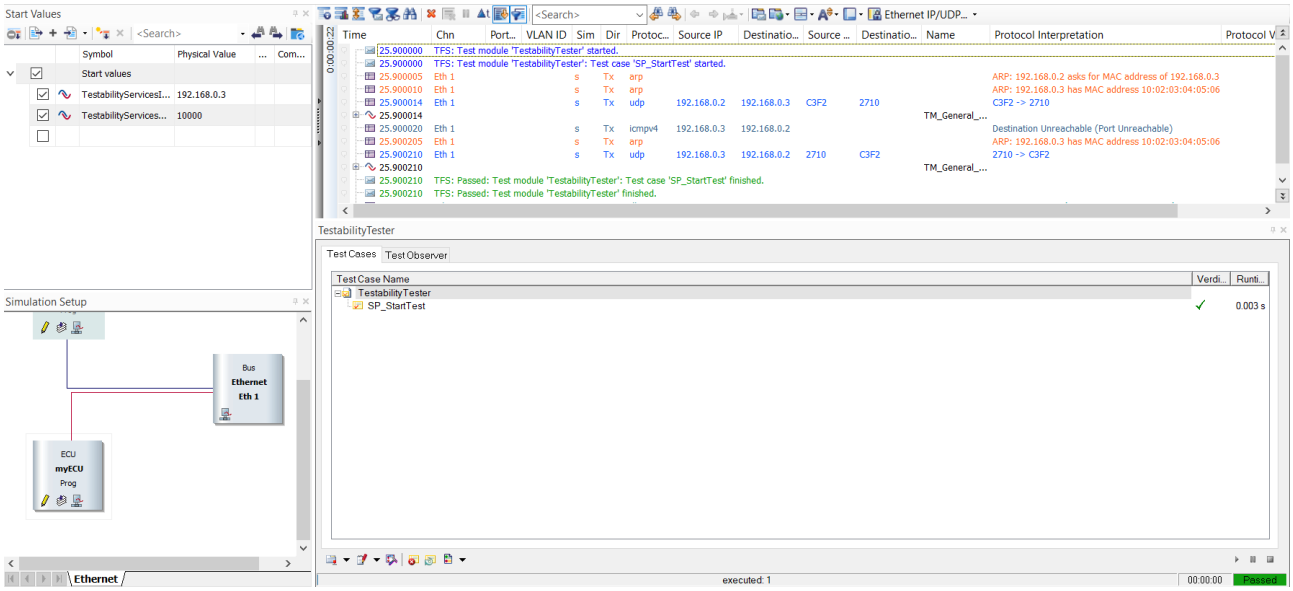
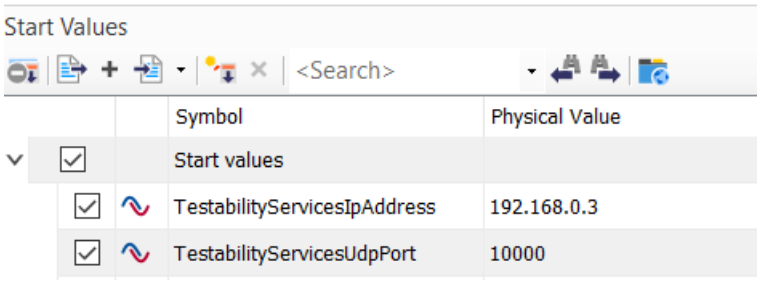


Figure 2-4 Test communication visible in CANoe trace

If your test has passed, this means that your integration of the ETM module was successful. If it was not successful you can go to the Troubleshooting section, otherwise you can skip it.

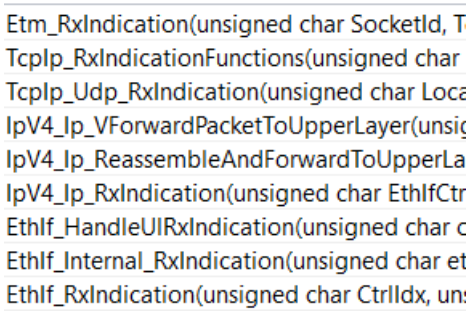
2.5 Troubleshooting

1. Make sure that the DET is activated for all modules and you do not DET Errors during runtime
2. Make sure that the Etm_InitMemory() amd Etm_Init() gets called when starting the stack
3. Make sure the Etm_Mainfunction is mapped and gets called cyclically
4. Make sure the configured IP-address and UDP port fit to your configuration:



In this case your referenced local address (in the ETM module) of the IPv4 should match the IP address 192.168.0.3 and the configured UDP port:

5. When starting the test case (“SP_StartTest”) of the TestabilityProtocolTester you can verify if the Etm_RxIndication gets reached. It should get reached with a call tree similar to this one:



If you could reach the point that all of those steps seem to be fine and you cannot find out why you still do not get an answer, you can contact our Support, see Support / Coaching.

3 Further Steps

3.1 Figure out what are the Relevant Test Cases

As every OEM has different requirements with respect to the ethernet architecture, the OEM should be able to give you a list of the TC8 test cases which are required to be passed. It is not necessary and probably not possible to pass all the TC8 test cases as it's highly usecase dependent which test cases make sense for your configuration. Of course, you can only test features that are part of your configuration. E.g. if you're not using any DHCP functionality with your ECU, then of course you do not need to pass the DHCP correlated test cases.

3.2 More Specific Configuration of your Ethernet Stack

Please also go through the TechnicalReference_Etm.pdf (which is part of your SIP in case you have ordered the Etm module) in order learn about other topics concerning the module: e.g. features, deviations, extensions, handling of user defined, unsupported or unknown Service Primitives or configuration of ETH/PHY Service Primitives etc.

The testability protocol defined in AUTOSAR might not be complete, hence the ETM provides a generic mechanism to extend the testability protocol according the specific needs of the test setup/tester. For this usecase we deliver the "_Appl_Etm.c" within your SIP as a template.

In case you have a more recent version of the ETM (8.00.05 and later) the _Appl_Etm.c already contains an exemplary implementation which could be used together with our "EthernetTC8Test" CANoe Sample Configuration, see also Sample Configuration: EthernetTC8Test. If you're not using our EthernetTC8Test (instead another tester / test house is involved) you can still use the provided _Appl_Etm.c as a reference to see how it could be implemented in general.

In case your MICROSAR SIP contains an "older" version of the ETM (8.00.04 and older) this file just contains an empty stub of the function ApplEtm_UndefinedServicePrimitive(), which needs to be filled with your needed code in order to fit to the needs of the tester. On request (e.g. email to our support address) we can provide you also with the _Appl_Etm.c with the exemplary implementation, which fits to our "EthernetTC8Test" CANoe Sample Configuration.

3.3 Performing TC8 Test Bench

Given that you have clarified, which TC8 test cases need to be passed, you can let the ECU be tested with all valid test cases.

Before giving your DUT to a specialized test house (see also the document TC8_RequirementsOnTestHouses.pdf provided by OA TC8) you can let your DUT run against our EthernetTC8Test CANoe configuration. With that approach you can already get a good feeling in advance in which areas you might need to work on in order to pass more test cases.

3.4 Analysis of the Failed Test Cases

If a test case is failing, we suggest you to check the following questions/hints:

1. Verify that this specific failed test case is **required by your OEM**.
2. Make sure that this test case is also **relevant for your ECU project**. (Is the tested feature/functionality of this test case part of your configuration at all?)

3. Make sure the **configured protocol version** in the ETM module (/MICROSAR/Etm/EtmConfig/EtmProtocolVersion) fits to the protocol version which is used by the tester.
4. Please make sure that you **fully understand** what **the test case** is all about. This can be achieved with the help of the test specification and also the RFC what is mentioned in the test specification can help you to understand the technical details.
5. Compare the **ethernet trace** of this specific test case with the **test specification** and find out at which step there is a deviation between your trace and the test procedure / pass criteria.
    a. Is the deviation on the tester side you might need to go into contact with the responsible test house or owner of this test system in order to clarify what's going wrong on tester side. No need to go on in this list.
    b. Is the deviation on DUT side, you can go on in this list.
6. Find out if this behavior occurs **permanently or only sporadic**. How often / when does it occur?
7. Is the handling of all **necessary ServicePrimitives** implemented? Is your ECU running into **AppIEtm_UndefinedServicePrimitive()** when performing the failing test case? Please make sure, that you handle the requested ServicePrimitive at all and also with the correct implementation.
8. Try to find out if a **specific parameter (or missing elements)** in your configuration might be responsible for the deviation to the expected test result.

If you have checked all the listed questions / hints, tried to solve them and still the test case is failing you can also contact Vector in order to get help. Please refer to the chapter Support / Coaching for further information. Please make sure to provide enough information, so that we can help you in an efficient way.

3.5 Reference Test Reports, Traces and BSW Configurations etc.

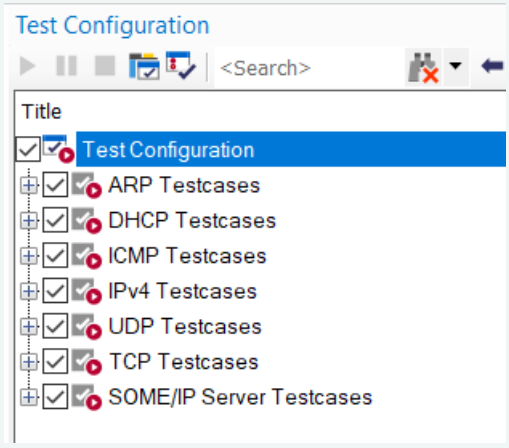
Upon request we can provide you with our reference configuration for the Etm and TcpIp modules (EcuC-Files/ARXML), with which our internal TC8 testing is performed.

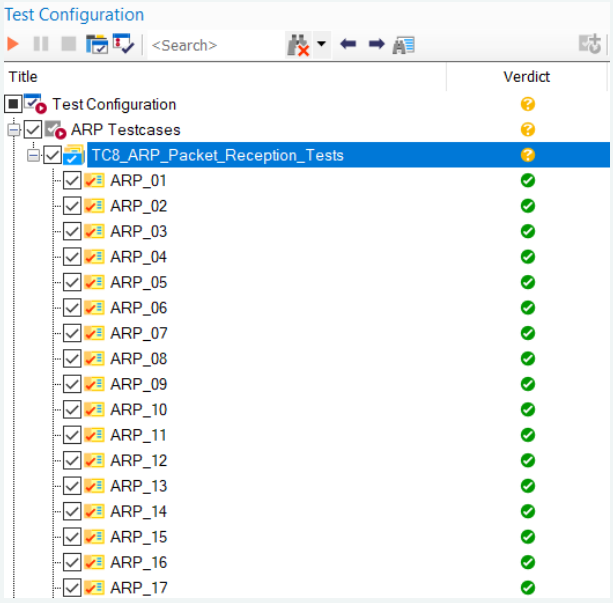
Our internal test reports for the TC8 tests can only be provided on request with signed NDA.

3.6 Sample Configuration: EthernetTC8Test

Note

There is an extensive CANoe sample configuration available in the Vector Knowledge-Base.
The sample configuration is (when writing this article) based on version 2 of the TC8 specification. The configuration contains many of the TC8 test cases, which can be executed with CANoe.





The test sequences and parameters have been defined using vTESTstudio. The configuration provides the tester part and also the DUT part. You could either use it as a reference (e.g. to analyze the communication/trace between tester and simulated DUT) or you could also use the tester in combination with your ECU. For further details please download the Sample Configuration at the referenced link and have a look in the documentation included.

4 Support / Coaching

Basic Functionality of the Etm module → Support

In case you integrated the Etm module (as illustrated in Integrating the ETM module into the stack) and you tried to confirm the functionality (as illustrated in Confirming the basic runtime functionality of the ETM) but you encountered problems with it, then please go through the section Troubleshooting to make sure that you have checked the typical issues one can face when bringing the Etm into service.

If you're still having problems to confirm the basic functionality (e.g. you do not receive a response when sending the test request) you can also contact our support (support@vector.com) and describe your problem as detailed as possible. As usual when using this support email address make sure to also add your CBD-Number and delivery number. Additionally, you should add some keywords like MICROSAR, Etm and TC8 Conformance testing, so that the support can quickly push your request into the right direction.

Such a troubleshooting request we can usually handle as regular support (which is already included in your maintenance).

Testing your ECU against TC8 test bench → Coaching

Confirming the basic functionality/integration of the ETM module is one thing. The other thing is testing the ECU with the actual TC8 test cases. As failing or passing of specific TC8 test cases very much depends on your configuration and your use case you cannot expect to get completely green test results at the very beginning (or at all). Moreover, you should get the requirements from your OEM which test cases you need to pass and which are irrelevant.

However, we can offer you a coaching in which a MICROSAR Coach can analyze together with you which test cases should pass with your use case, why are they failing and how to modify the configuration in order to make them pass.

In case you want to use such a service please write to support@vector.com (if you're already in contact with a MICROSAR Coach or a project manager for your ECU project you can of course trigger such a Coaching directly over this contact). Please provide as much details as possible:

- CBD-Number
- Delivery number
- OEM list of required test cases
- Support Request Package (SRP) of your configuration (Configurator 5 → Help → Create Support Request Package)
- Information on which test case is failing
- Information on your test setup and tester / test house
- complete traces of the ethernet bus (please provide trace for each test case which is failing and tell us the lines / times / packages where would expect a different behavior)
- failing testreport

This way you do not lose much time with email communications, only to request further inputs from your side.

5 Abbreviations

5.1 Abbreviations

Abbreviation	Description
OA	Open Alliance
TC8	Tech Committee 8
ETM	Ethernet Testability Module
VTT	vVIRTUALTarget
ETS	Enhanced Testability Service
DUT	Device Under Test
SRP	Support Request Package

6 Relevant documents / links

Topic	Link
Open Alliance TC8	https://www.opensig.org/tech-committees/tc8/
TC8 Spec v2.0	http://www.opensig.org/download/document/215/OA_Automotive_Ethernet_ECU_TestSpecification_v2.0_final_11_17.pdf
TC8 General Definitions	TC8_GeneralDefinitions.pdf
TC8 Requirements on TestHouses	TC8_RequirementsOnTestHouses.pdf
TC8 Test Process	http://www.opensig.org/download/document/198/2_Test_Process_ECU+and+network+test.pdf
AUTOSAR Acceptance Tests	https://www.autosar.org/standards/acceptance-test/
TestabilityProtocolAndServicePrimitives	AUTOSAR_PRS_TestabilityProtocolAndServicePrimitives.pdf
Vector Testing Symposium – TC8	https://assets.vector.com/cms/content/events/2019/vTES19/50_Lectures/10_Slides_Dr_Heiner_HILD_Vector_Testing_Symposium_2019_EN.pdf
Vector provided ECU tests as a sample configuration based on version 2 of the TC8 specification	https://kb.vector.com/entry/1531/

7 Contact

Visit our website for more information on

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

www.vector.com