

Reliable Testing of AUTOSAR ECUs

Webinar 2011-11-15

> Introduction

Solution Overview

Testing AUTOSAR SWCs

Testing AUTOSAR ECUs

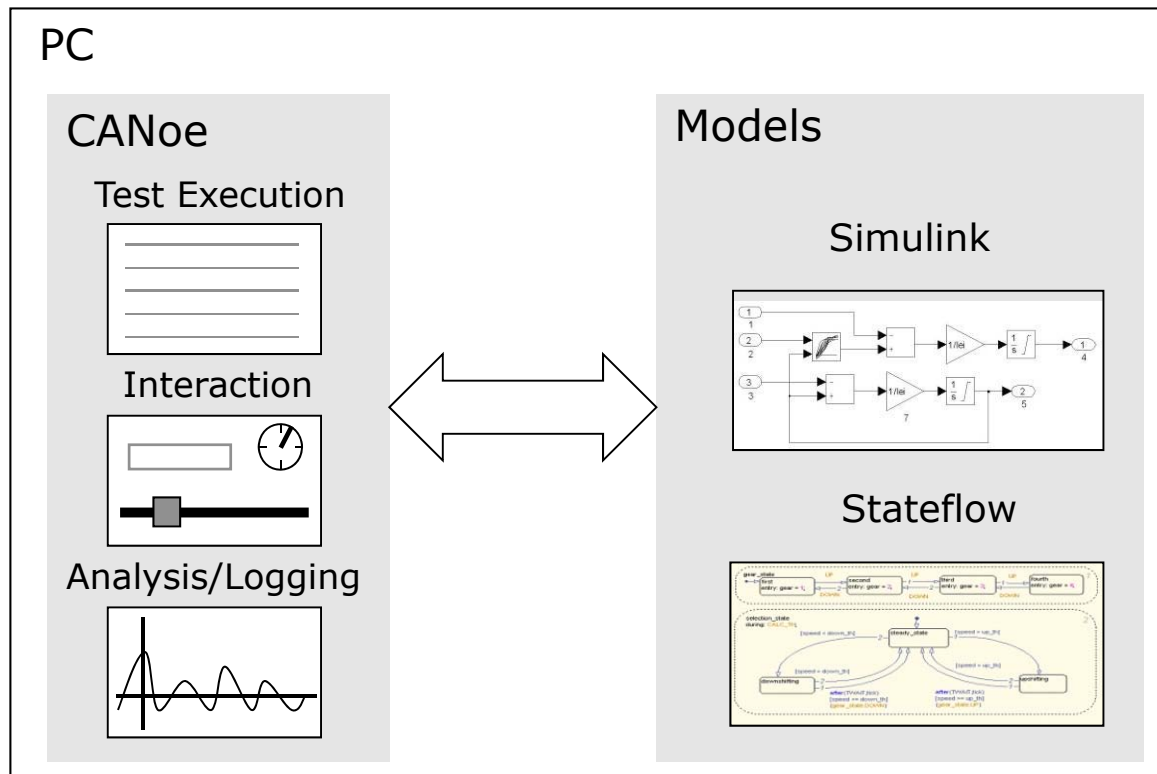
Summary

Webinar Series

Introduction

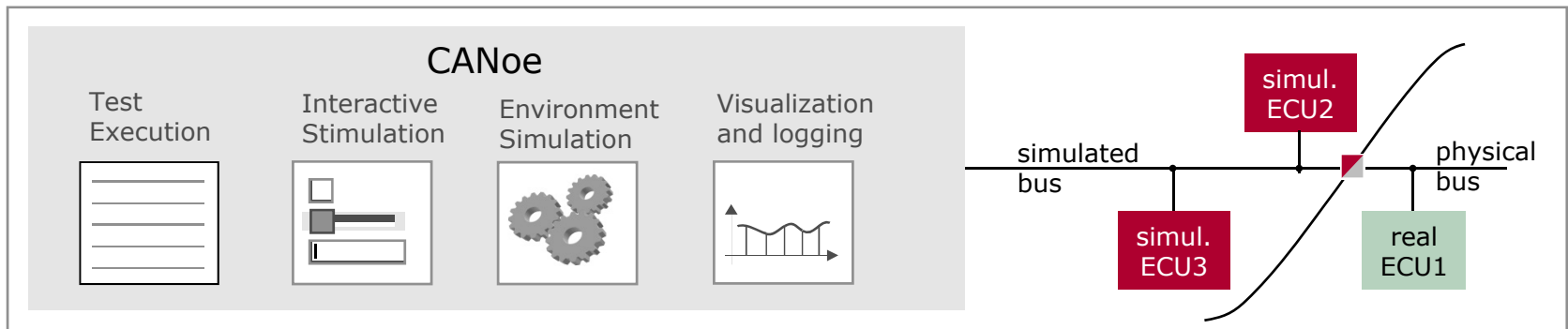
ECU Testing pre AUTOSAR

- ▶ Phase 1: SWC Model Simulation on PC
 - ▶ Model in the loop
 - ▶ System simulation with Simulink / Stateflow
 - ▶ CANoe used for test execution, interaction, analysis and logging



- ▶ Wait...
 - ▶ for availability ECU with fully integrated SW

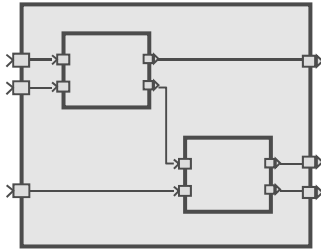
- ▶ Phase 2: Test the integrated ECU
 - ▶ Test on ECU Level with remaining bus simulation
 - ▶ ECU access via communication bus, ECU IO, debug and calibration interfaces (XCP/CCP, JTAG, VX1000,...)



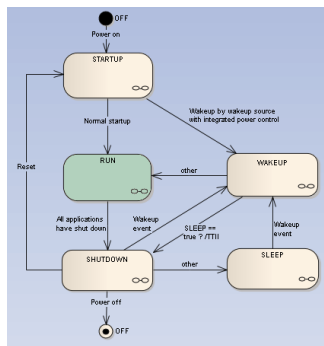
Introduction

New Situation when using AUTOSAR

Modularized applications,
independent of ECU
hardware



Basic SW is
standardized



How can we benefit during
ECU test?

Introduction

> Solution Overview

Testing AUTOSAR SWCs

Testing AUTOSAR ECUs

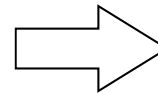
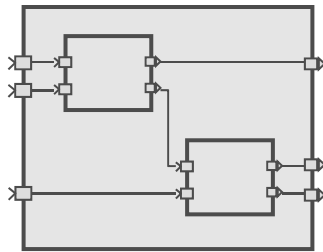
Summary

Webinar Series

Solution Overview

Solution for Testing AUTOSAR ECUs

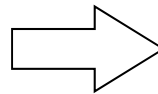
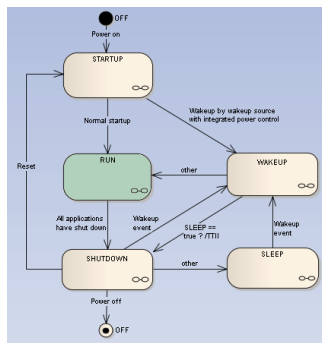
Modularized applications,
independent of ECU
hardware



SWCs can be tested on the
PC (unit tests, integration
tests)

DaVinci
Component Tester

Basic SW is
standardized



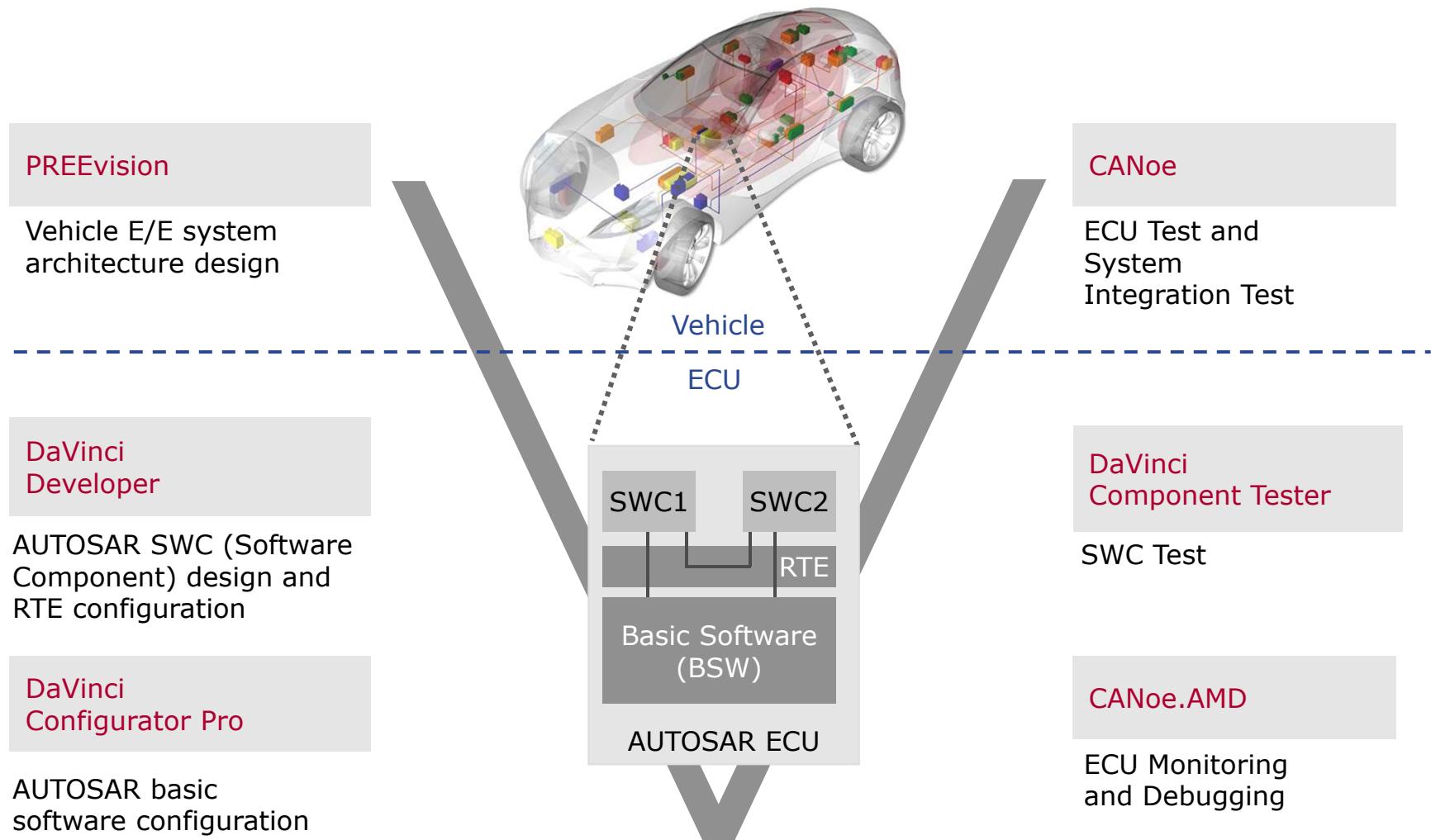
Use AUTOSAR semantics
during measurement and
debugging of ECU

CANoe.AMD

(AUTOSAR Monitoring and
Debugging)

Solution Overview

Vector tools in the AUTOSAR development process



Introduction

Solution Overview

> Testing AUTOSAR SWCs

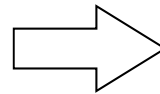
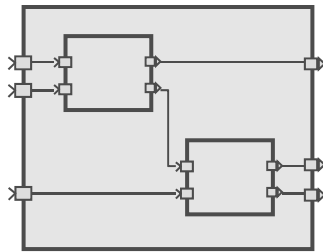
Testing AUTOSAR ECUs

Summary

Webinar Series

Testing AUTOSAR SWCs

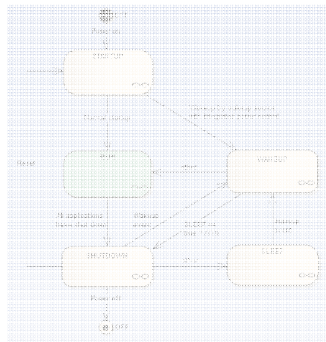
Modularized applications,
independent of ECU
hardware



SWCs can be tested on the
PC (unit tests, integration
tests)

DaVinci
Component Tester

Basic SW is
standardized

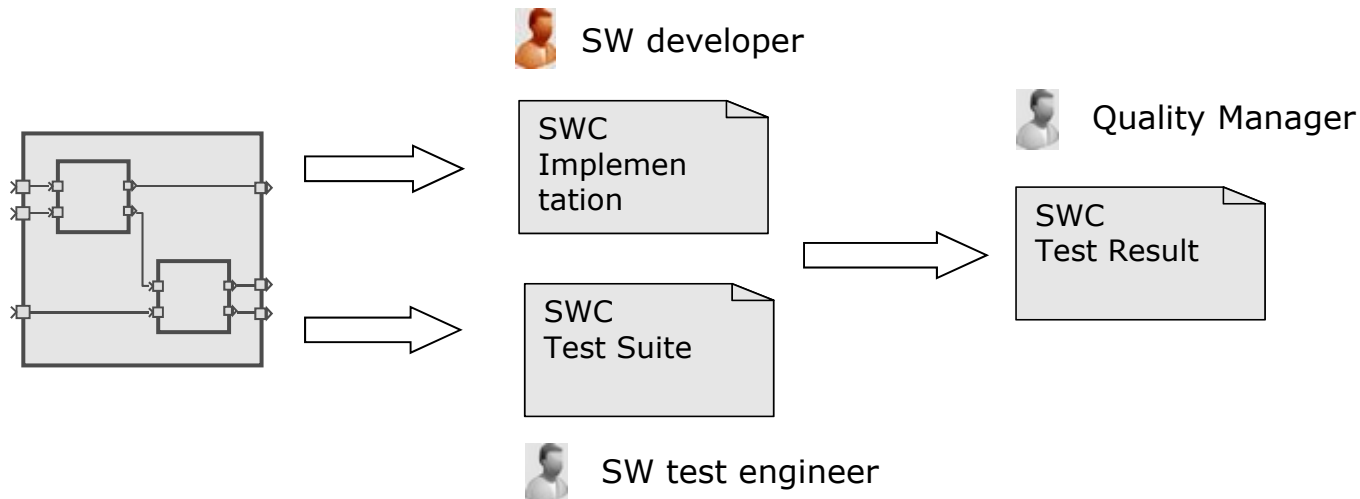


Use AUTOSAR semantics
during measurement and
debugging of ECU

CANoe.AMD

(AUTOSAR Monitoring and
Debugging)

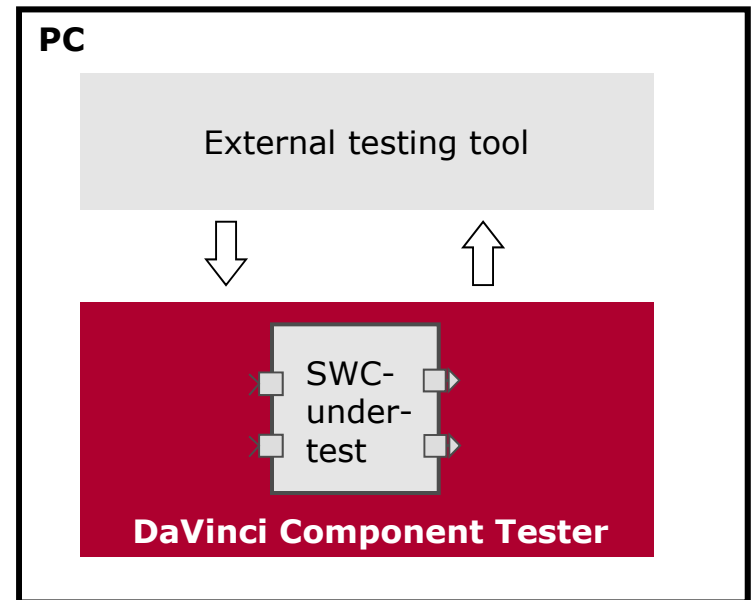
- ▶ Why testing SWCs?
 - ▶ “Front-loading” of tests to avoid late issues during ECU integration
 - ▶ Enhanced quality of application SW by **unit tests**
 - ▶ Ensure the robustness of the SWC in various usage scenarios (error injection)
 - ▶ Ensure the AUTOSAR API compliance
 - ▶ Validate the functional behavior
 - ▶ Introduce test-driven development
 - ▶ Setup of application SW libraries for ECU development (commercial off-the-shelf)



Testing AUTOSAR SWCs

Solution overview

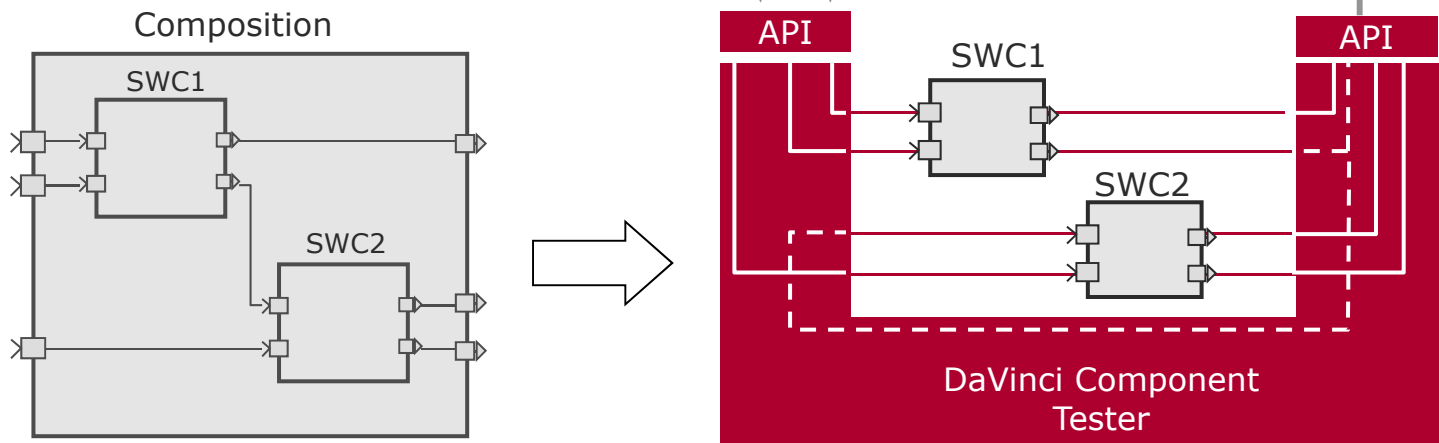
- ▶ DaVinci Component Tester as backend emulator
 - ▶ Executes SWCs directly on the PC by emulating the VFB (Virtual Function Bus) - no RTE or AUTOSAR basic SW required
 - ▶ Atomic SWCs and Compositions supported
 - ▶ Gives the testing tool access to the APIs of the SWC
- ▶ External testing tool as front-end
 - ▶ Definition and organization of test cases
 - ▶ Test execution
 - ▶ Creation of test reports



Testing AUTOSAR SWCs

Project Setup

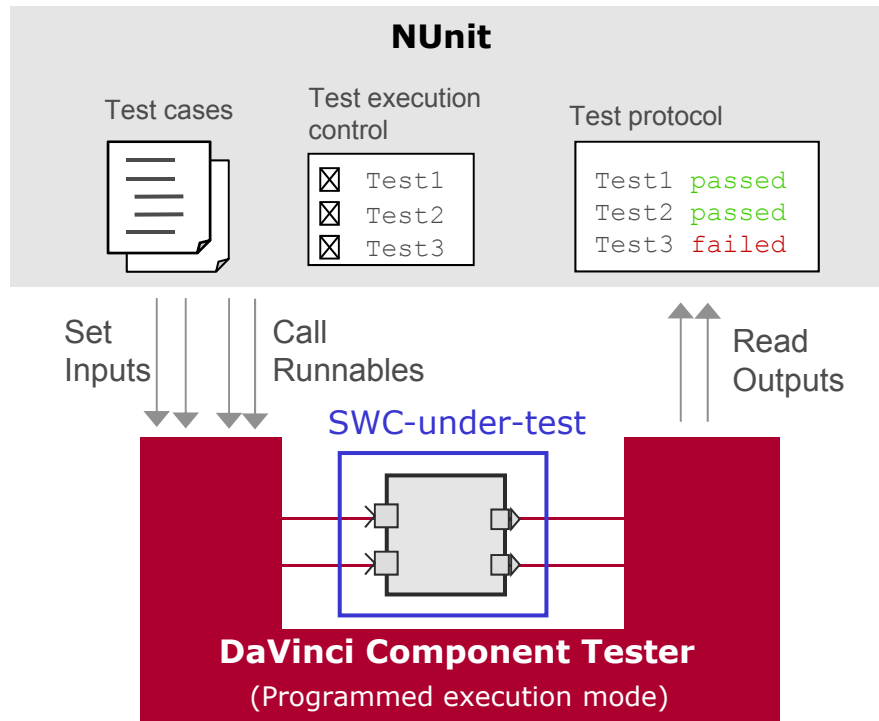
- ▶ Emulator is automatically configured via SWC AUTOSAR XML
 - ▶ ECU configuration is not required
 - ▶ Short turnaround time
- ▶ API provided for the testing tool
 - ▶ Access of SWC ports
 - ▶ Invoke SWC runnables



Testing AUTOSAR SWCs

Unit Tests with DaVinci Component Tester and NUnit

- ▶ Test cases are programmed in any .NET language (e.g. C# or VB)
- ▶ Test control, tracing and protocol generation done by NUnit*
- ▶ Allows easy debugging of SWC implementation on the PC

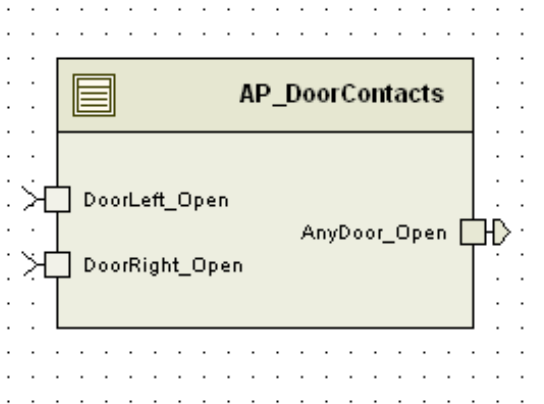


*Open source tool, see www.nunit.org

Testing AUTOSAR SWCs

Unit Test Example

SWC-under-test



Pattern

- ▶ Set input data elements
- ▶ Invoke runnable
- ▶ Get output data elements
- ▶ Assert the expected values

Test Suite

```
[TestFixture]
public class InputValuePermutation
{
    [TestFixtureSetUp]
    public void Init()
    {
        FileInfo sutFile = null;
        try
        {
            sutFile = new FileInfo(@"..\SUT\AP_DoorContacts.dll");
        }
        catch (Exception e)
        {
            throw new ApplicationException(string.Format("error accessing directory <0>", sutFile.FullName));
        }
        AP_DoorContacts.API.Init(sutFile.FullName, true);
    }

    [Test]
    public void Test_DoorRight_Open()
    {
        AP_DoorContacts.RPORT.DoorRight_Open.DEP.DoorOpen.API.Value = true;
        AP_DoorContacts.RPORT.DoorRight_Open.DEP.DoorOpen.API.ReturnVal = RTEErrors.RTE_E_OK;
        AP_DoorContacts.RPORT.DoorLeft_Open.DEP.DoorOpen.API.Value = false;
        AP_DoorContacts.RPORT.DoorLeft_Open.DEP.DoorOpen.API.ReturnVal = RTEErrors.RTE_E_OK;

        AP_DoorContacts.RUNNABLE.AP_DoorContacts.API.Trigger();

        Assert.That(true == AP_DoorContacts.PPORT.AnyDoor_Open.DEP.DoorOpen.API.Value);
    }

    [Test]
    public void Test_DoorLeft_Open()
    {
        AP_DoorContacts.RPORT.DoorRight_Open.DEP.DoorOpen.API.Value = false;
        AP_DoorContacts.RPORT.DoorLeft_Open.DEP.DoorOpen.API.Value = true;

        AP_DoorContacts.RUNNABLE.AP_DoorContacts.API.Trigger();

        Assert.That(true == AP_DoorContacts.PPORT.AnyDoor_Open.DEP.DoorOpen.API.Value);
    }
}
```

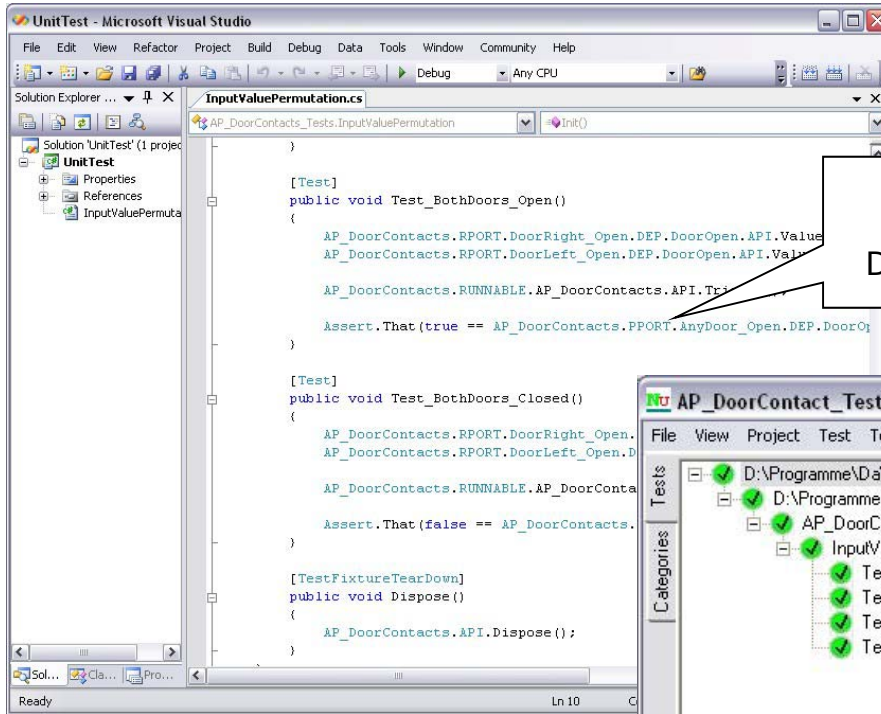
Load the SWC executable DLL

Test case

Testing AUTOSAR SWCs

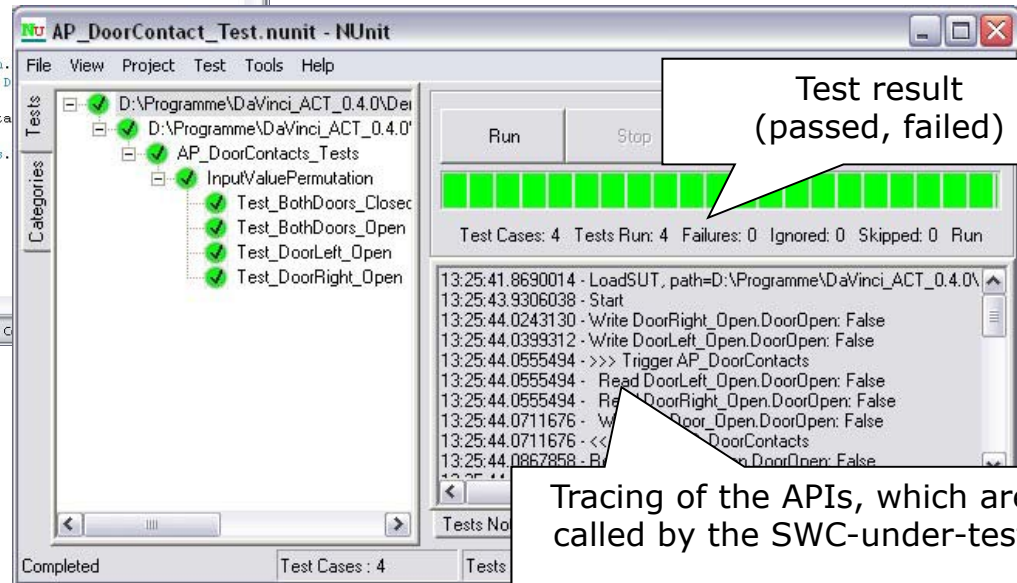
Unit Test Example

MS Visual Studio as test case authoring tool



Intelli-sense authoring support provided by DaVinci Component Tester

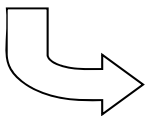
NUnit as testing tool



Test result (passed, failed)

Tracing of the APIs, which are called by the SWC-under-test

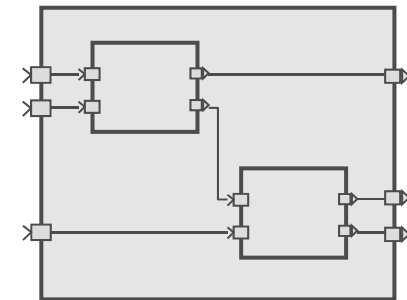
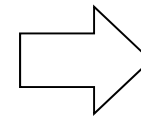
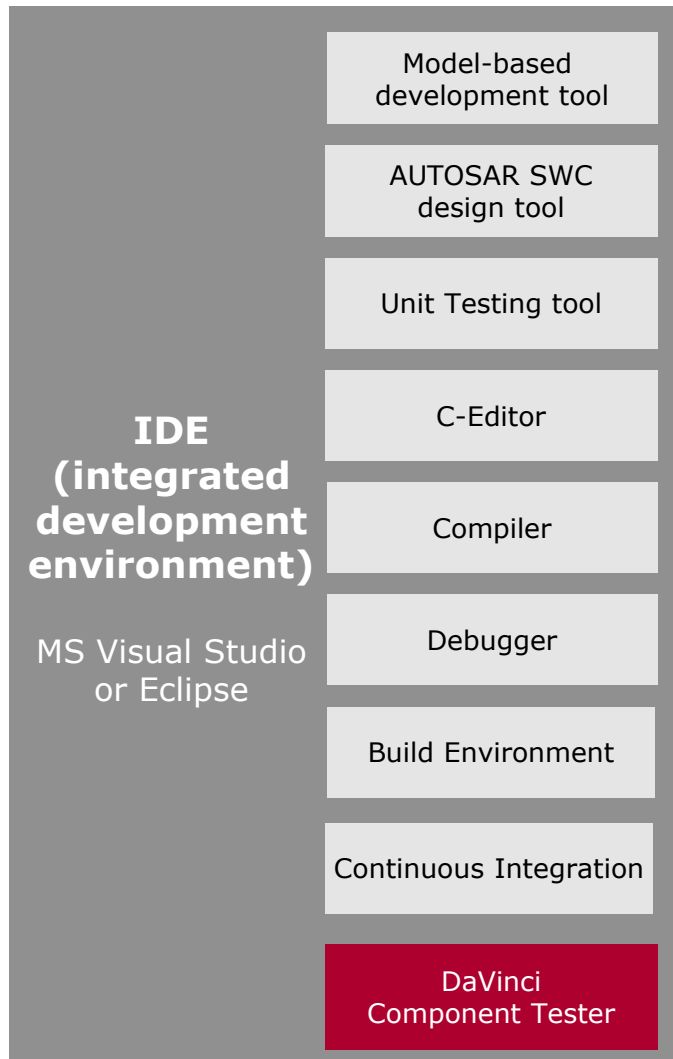
Compile the test case and execute it in NUnit



DaVinci Component Tester runs in background during the test

Testing AUTOSAR SWCs

IDE Integration



DaVinci Component Tester
as part of an IDE for
developing AUTOSAR SWCs

Introduction

Solution Overview

Testing AUTOSAR SWCs

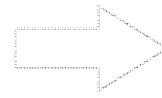
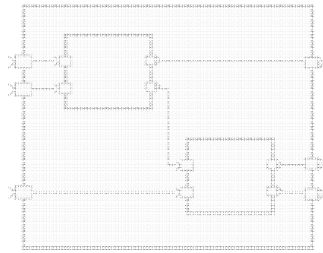
> Testing AUTOSAR ECUs

Summary

Webinar Series

Testing AUTOSAR ECUs

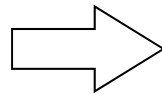
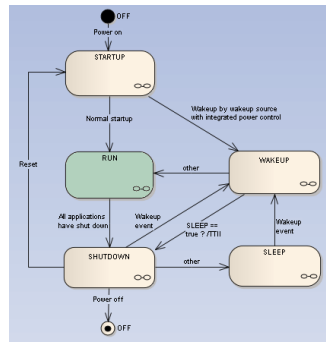
Modularized applications,
independent of ECU
hardware



SWCs can be tested on the
PC (unit tests, integration
tests)

DaVinci
Component Tester

Basic SW is
standardized



Use AUTOSAR semantics
during measurement and
debugging of ECU

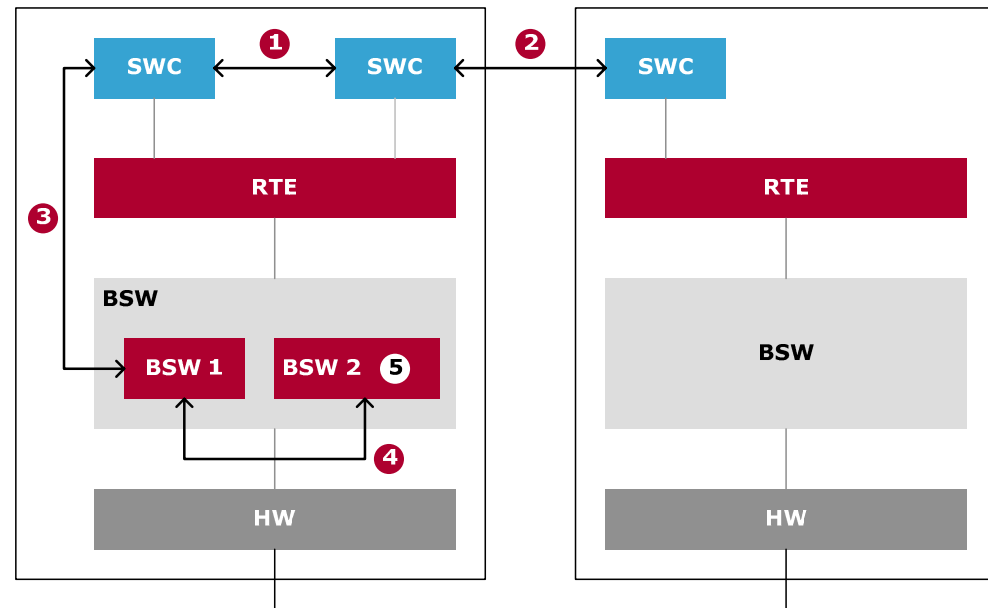
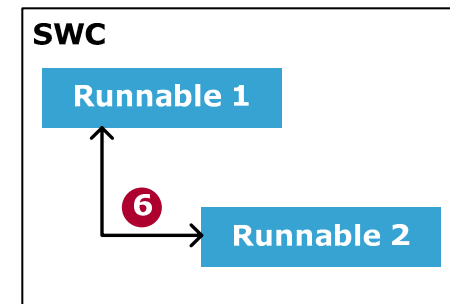
CANoe.AMD

(AUTOSAR Monitoring and
Debugging)

Testing AUTOSAR ECUs

Measured AUTOSAR semantics

- ▶ 1 Communication among SWC – same ECU
- ▶ 2 Communication among SWC – different ECU
- ▶ 3 Communication SWC and BSW
- ▶ 4 Communication among BSW
- ▶ 5 Internal States of a BSW
- ▶ 6 Inter-Runnable Communication



- ▶ XCP in conjunction with high speed interfaces offers many possibilities to gather information from within an ECU
- ▶ This information can be not only used for simple measurement but also for:
 - ▶ Debugging
 - ▶ Logging
 - ▶ Tracing of internal states (debugging of state machines)
 - ▶ Profiling of software (runtime, lock time, stack usage, ...)
 - ▶ Modification of data (stimulation, calibration, ...)
- ▶ The impact on the application shall be as low as possible
 - ▶ Only minimum overhead thanks to XCP Events and dynamic DAQ Lists

Testing AUTOSAR ECUs

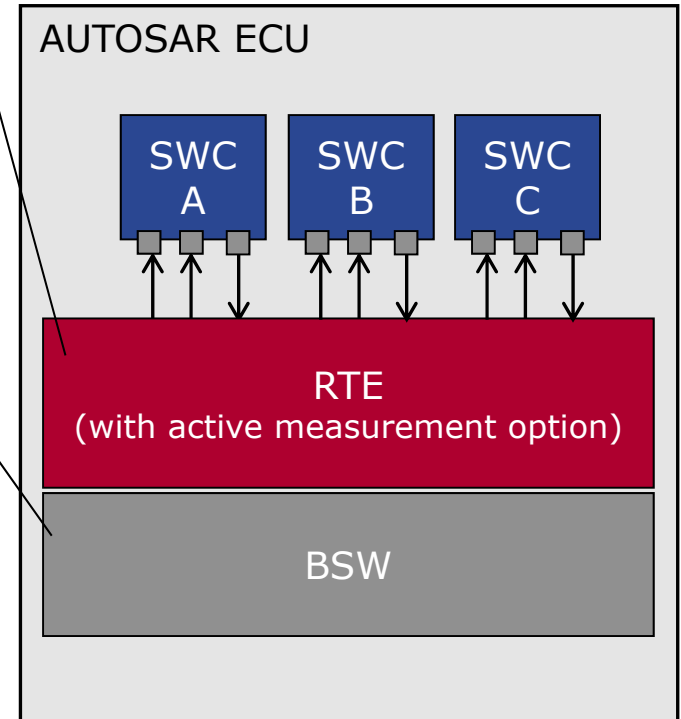
MICROSAR Measurement Support

MICROSAR RTE

- ▶ Sender/Receiver Ports, Mode Ports
- ▶ Inter-Runnable Variables
- ▶ Per Instance Memory
- ▶ Communication SWC <-> BSW

MICROSAR BSW

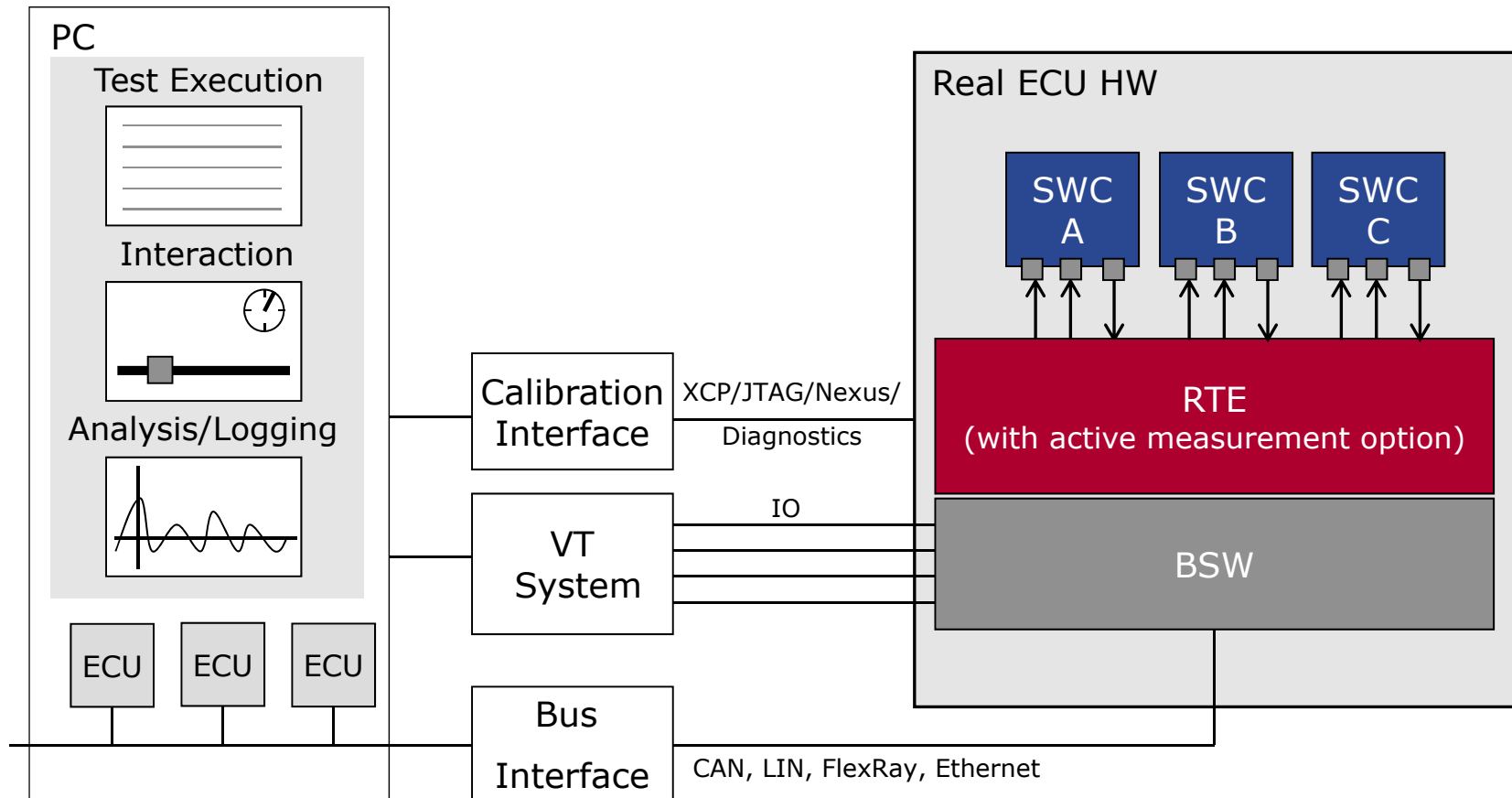
- ▶ Internal variables e.g. NM States, Modes
- ▶ Access to COM signals (CAN, LIN, FlexRay and Ethernet)
- ▶ I/O signals
- ▶ OS runtime statistics (max. execution time or blocking time of tasks/ISRs)



Testing AUTOSAR ECUs

Tool ECU Connections

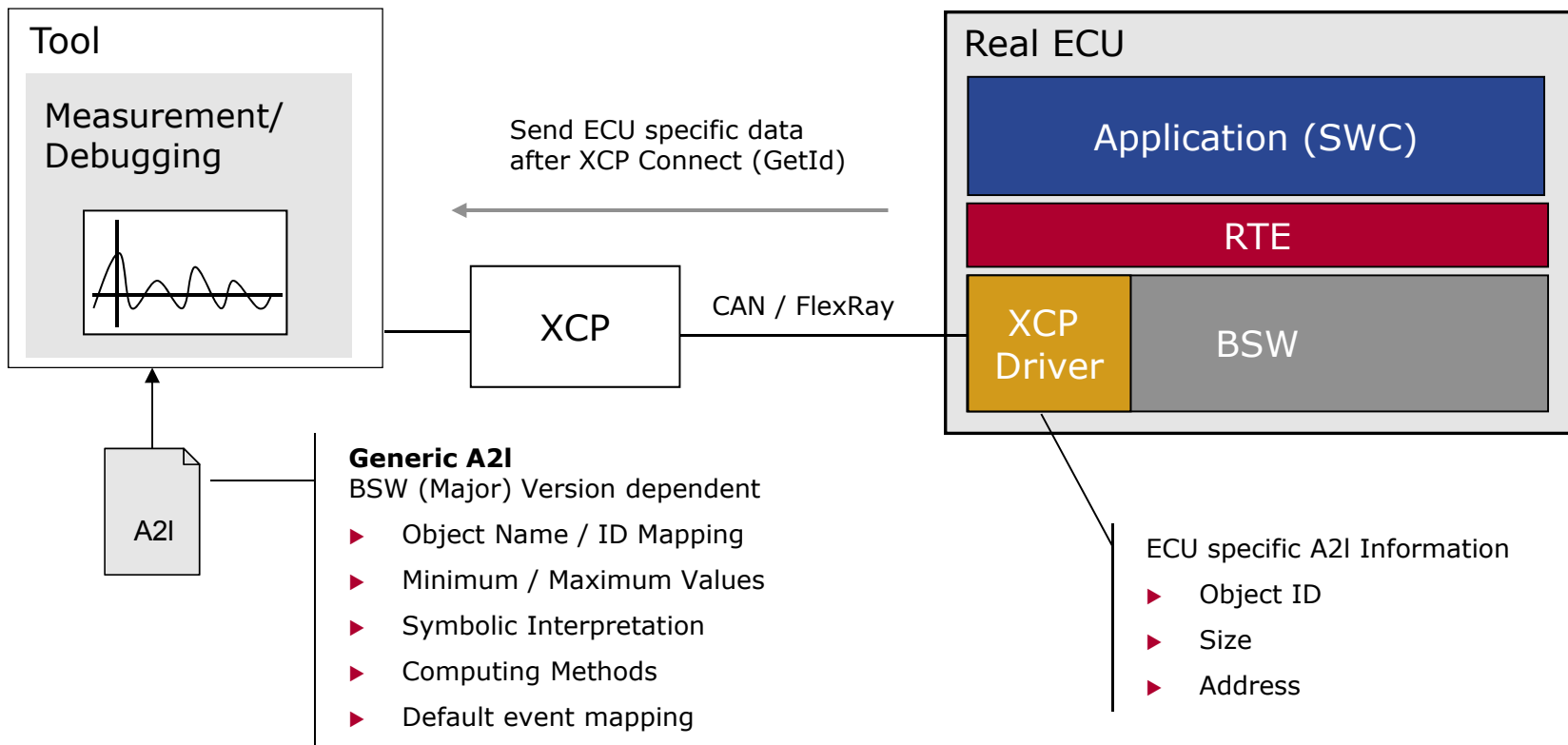
- ▶ Access to ECU via IO, bus or calibration interface



Testing AUTOSAR ECUs

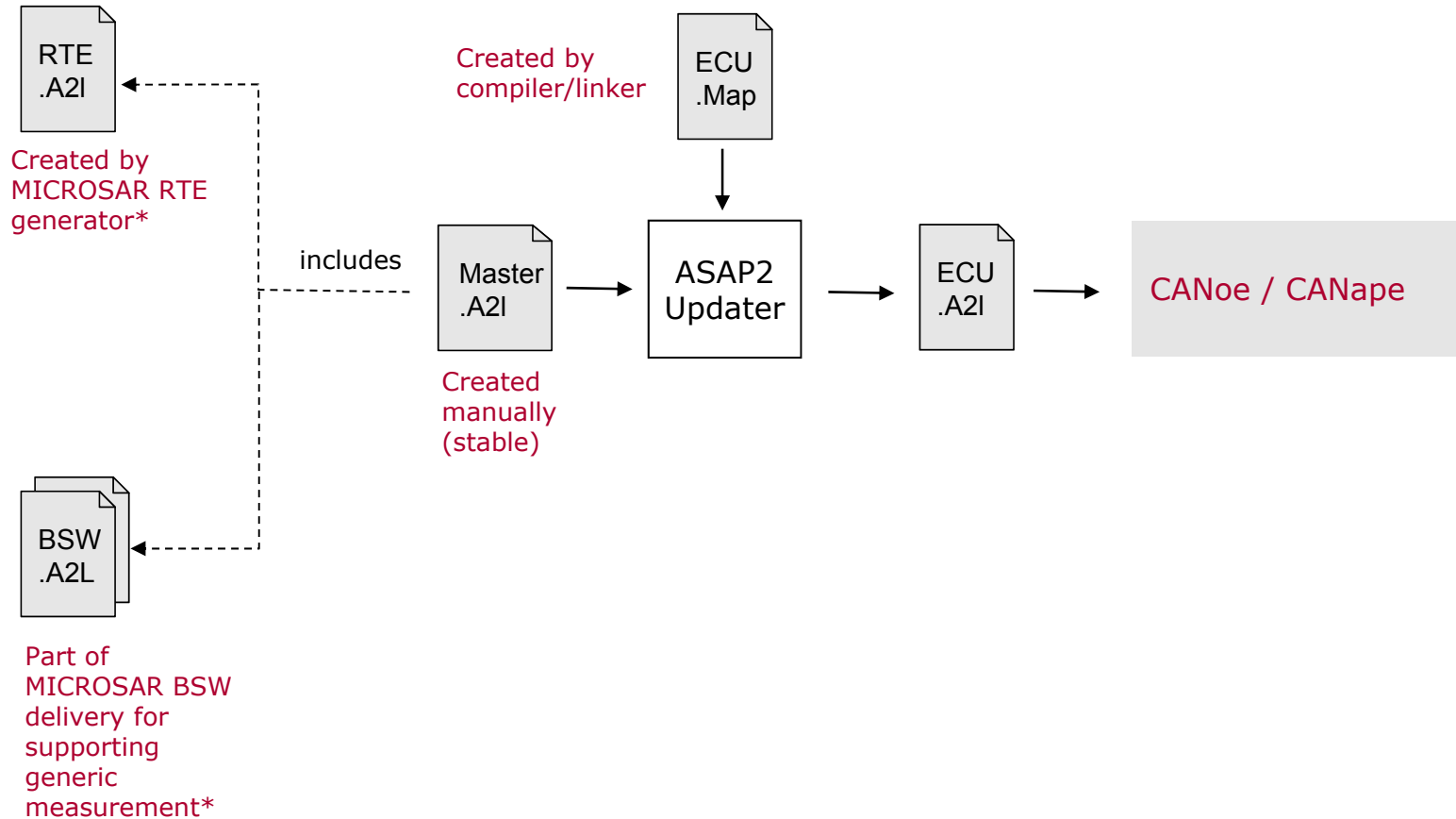
Advanced Features - Generic Measurement

- ▶ One generic measurement description (A2L) for all AUTOSAR ECUs
 - ▶ No need to update the A2L after each ECU SW build
 - ▶ Measure with full flexibility of XCP protocol (dynamic DAQ, Polling, ...)



Testing AUTOSAR ECUs

Workflow A2L Creation

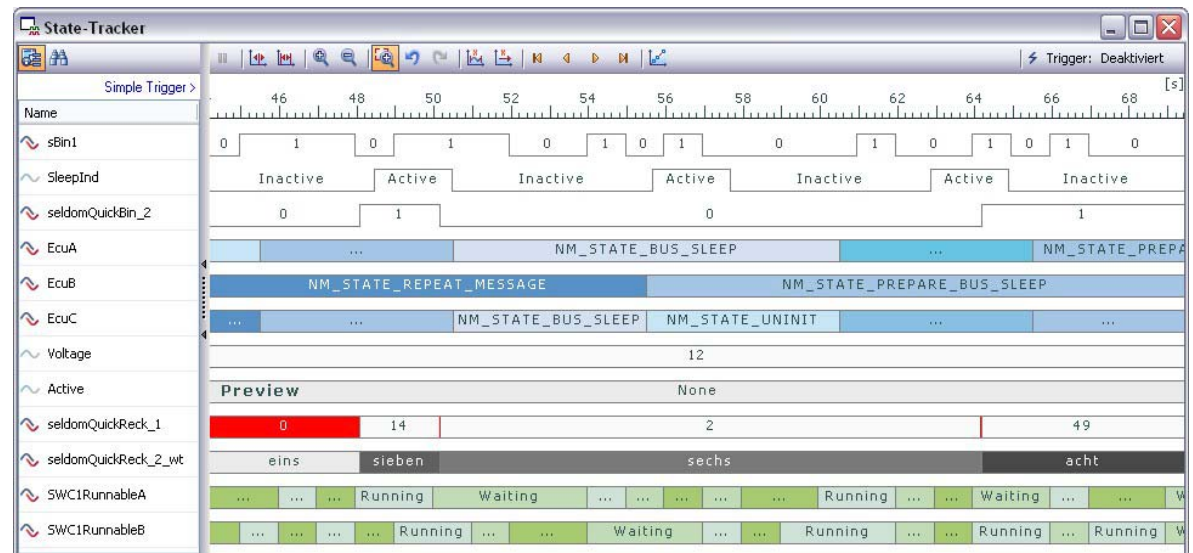


*symbols and interpretation (scaling, value tables, ...)

Testing AUTOSAR ECUs

Usage of State Tracker

- ▶ Monitor ECU internal States of BSW and SWC
- ▶ Monitor Communication bus and protocol (e.g. NM) events
- ▶ Trigger functions to detect predefined scenarios
- ▶ Measure Runtime of
 - ▶ Runnables
 - ▶ Interrupts
 - ▶ Data communication
- ▶ Diagram features
 - ▶ Set bookmarks
 - ▶ Measure events



Testing AUTOSAR ECUs

Analyzing Windows - SWC browser to select SUT internal variables

The screenshot displays the SWC browser interface for the 'SWCsInteriorLightECU'. The main window shows a hierarchical tree of SWC components and their internal variables:

- IoHwAb::IoHwAb**
 - PP_FrontLeftDoor
 - PP_FrontRightDoor
 - PP_FrontInteriorLight
- CP_SA_Door_Front::CT_SA_Door**
 - PP_DoorState_FrontLeft
 - PP_DoorState_FrontRight
- CP_SA_InteriorLight_Front::CT_SA_Interior...**
 - PP_LightState_InteriorFront
- CP_AP_InteriorLightProcessing::CT_AP_Int...**
 - PP_DoorState_RearLeft
 - PP_DoorState_RearRight
 - PP_LightState_RearInteriorLight
 - PP_LightDefect
 - PP_LightOnOffCounter
 - PP_Send_RearInteriorLight
 - PP_Odometer
- StateManagement_DemoAppInstance::St...**
 - StateManagement_Commands
 - StateMngt_Info

The bottom-left window shows a car model with a 'Sleep Request Defect' icon and two data fields: 'Odometer: 0' and 'Light On Off Count: 0'.

The bottom-right window shows a 'SWC Datatrace' table with the following data:

Time	ID	Name	Data
11.783362	SV:	PP_DoorState_FrontRight	1
12.278483	SV:	PP_DoorState_FrontRight	0
15.941599	SV:	PP_DoorState_FrontRight	1
16.141797	SV:	PP_DoorState_FrontRight	0
16.326369	SV:	PP_DoorState_FrontRight	1
17.214074	SV:	PP_DoorState_FrontLeft	1
17.766815	SV:	PP_DoorState_FrontRight	0
18.583238	SV:	PP_DoorState_FrontLeft	0

The bottom-right window also shows a 'Graphics' window with a signal plot for 'PP_Send_FrontInteriorLight' and 'sig_State_RearRightDoor'. The plot shows a signal that is 0 for most of the time but has several pulses reaching 1.

Introduction

Solution Overview

Testing AUTOSAR SWCs

Testing AUTOSAR ECUs

> Summary

Webinar Series

ECU Tests can be improved thanks to AUTOSAR!

- ▶ **CANoe.AMD**
 - ▶ Test execution, stimulation, analysis, measurement and logging
 - ▶ ECU access via calibration interface
- ▶ **CANape**
 - ▶ Calibration, bypassing, measurement, rapid prototyping and logging
 - ▶ ECU access via calibration interface
- ▶ **DaVinci Component Tester**
 - ▶ SWC execution environment
 - ▶ NUnit testing of SWCs
- ▶ **Fleet logger**
 - ▶ Logging via XCP on SWC level
- ▶ **Vx1000**
 - ▶ Measurement and calibration hardware

- ▶ The webinar series about AUTOSAR
 - ▶ 2011-10-06 ECU development with AUTOSAR - An introduction for AUTOSAR beginners
 - ▶ 2011-10-19 Introduction to the AUTOSAR Method for ECU development
 - ▶ 2011-11-02 AUTOSAR Configuration Process - How to handle 1000s of parameters
 - ▶ 2011-11-15 Reliable Testing of AUTOSAR ECUs
 - ▶ 2011-12-06 MICROSAR Safe: AUTOSAR basic software for safety-relevant ECUs (ISO 26262)
 - ▶ Various Webinars on MICROSAR OS – the AUTOSAR operating system from Vector

- ▶ Registration to the upcoming Webinars and the list of recorded Webinars:
http://www.vector.com/vi_webinars_en.html

- ▶ The overview of Vector's training services:
http://www.vector.com/vi_training_en.html

- ▶ We stay online for some more minutes to answer your questions. Please ask your questions in the Q&A window.

- ▶ Contact data for additional questions, product information or presentation :
 - ▶ embedded@de.vector.com

Thank you for your attention.

For detailed information about Vector
and our products please have a look at:

www.vector.com

Author:

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart