AUTOSAR Software Development and Testing Solutions

Joe Fairchild
dSPACE Tech Day
April 16th, 2015
Topics Covered in this Workshop

- AUTOSAR system architecture modeling with SystemDesk
- AUTOSAR internal behavior design and code generation with TargetLink
- Round-Trip development with Container Manager
- Virtual Validation of AUTOSAR code with VEOS
- Rapid prototyping of AUTOSAR code with MicroAutoBox II
dSPACE Tools for AUTOSAR Software Development

**Design phase**

- Specification of software requirements
- Software architecture design
- Software component design and implementation
- BSW generation and ECU software integration
- Third Party (e.g. Elektrobit, Vector)

**Testing phase**

- Verification of software requirements
- Software system integration and testing
- Software component testing

*dSPACE HIL Systems*
Library of software components (C code functions with XML describing interfaces)

System (interconnection of software components)

ECU
Configure OS, and basic software; generate code for RTE; compile and link
int16 speed;
int8 temperature;

Runnable 1
Runnable 2
Runnable 3

RTE Event
Interrunnable Variable

Atomic Software Component
Port
System Architecture
Interface
Internal Behavior
Runnable
Data Access
Top-Down Workflow

Architecture Tool (SystemDesk)

- Import into TargetLink DD .arxml

Frame Model

- Function modeling
- Copy/Paste from existing models
- CustomCode block for existing C code

Complete Model

- Import .arxml

- AUTOSAR-compliant C code
- Extended AUTOSAR .arxml file

- Generate
AUTOSAR: Iterative Engineering Round-Trips

Architecture Tool (SystemDesk)

Diff & Merge

TargetLink Data Dictionary

Model Update

TargetLink Model
Challenge:
How to handle all files involved in project in effective way?
How to see what has been modified by the developers or the software architect?
How to avoid introducing unwanted changes?

Solution: Exchange of SWC containers between SystemDesk and TargetLink
AUTOSAR Software Development Demo
AUTOSAR Software Development Process

Design System Architecture

Develop Control Functions

Configure and Build Basic SW

ECU
Virtual ECU Generation Process

- **Design System Architecture**
- **Develop Control Functions**
- **Generate V-ECU**

**Virtual ECU:**
Production-intent application code with simulation-capable BSW stack
Re-use V-ECU across dSPACE Platforms

Virtual ECU

MicroAutoBox II
Run production-intent code on a rapid prototyping platform

VEOS
Simulate realistic ECU on desktop environment without hardware

SCALEXIO
Connect Virtual ECU to physical signals for Hardware-In-Loop testing
Virtual Validation Demo
Adaptive Cruise Control Demo

Device Under Test

Adaptive Cruise Control V-ECU

Environment Models

ASM Vehicle Dynamics / Traffic

SIL Simulation Platform:

VEOS

ControlDesk / MotionDesk
Thank you! Any questions?

© Copyright 2014, dSPACE Inc.

All rights reserved. Written permission is required for reproduction of all or parts of this publication. The source must be stated in any such reproduction.

This publication and the contents hereof are subject to change without notice. Brand names or product names are trademarks or registered trademarks of their respective companies or organizations.