Simulink를 이용한 AUTOSAR SW 개발
From Architecture to Design to Testing
Demo: AUTOSAR ASW Architecture Design
Agenda

▪ AUTOSAR Blockset Introduction
  – Adaptive Platform
  – Classic Platform

▪ AUTOSAR ASW Development
  – AUTOSAR ASW architecture design
  – Testing in AUTOSAR Composition Editor

▪ System Composer with AUTOSAR Blockset
Introduction to AUTOSAR Blockset

- Model and simulate AUTOSAR software in Simulink
  - Model AUTOSAR Classic and Adaptive software
  - Simulate AUTOSAR compositions and ECUs
  - Import and export AUTOSAR descriptions (ARXML files)
  - Create AUTOSAR software architecture

- Supports C/C++ production code generation and AUTOSAR ARXML export (with Embedded Coder)
  - Blocks for AUTOSAR library routines
  - Qualified for ISO 26262 standard (with IEC Cert Kit).

AUTOSAR Blockset Library

- Blocks to model and simulate Basic Software and AUTOSAR library routines
  - Blocks for both Classic and Adaptive Platforms
  - Out-of-the-box AUTOSAR Basic Software emulation blocks
  - Generate code that calls into Basic Software services using Basic Software caller blocks
  - Reliably generate IFX and IFL AUTOSAR library routines during code generation
AUTOSAR Component Quick Start

- **Motivation**
  - Make users more comfortable with the *bottom-up workflow*
  - AUTOSAR wizard is infrequently used by customers
  - Use same layout as Embedded Coder Quick Start

- **Design**
  - Step-by-step workflow
  - Help panel on the right of the UI
  - Additional configuration options
    - Specify Component type
    - Import properties
  - Supports Adaptive AUTOSAR
AUTOSAR Software Architecture Modeling

**AUTOSAR Composition Editor**

→ Create architecture models

- Create an AUTOSAR architecture model in a canvas for developing AUTOSAR composition and component models for the Classic Platform.
- In the architecture model:
  - Add and connect AUTOSAR compositions and components.
  - Link components to requirements (requires Simulink Requirements™).
  - Define component behavior by creating or linking Simulink models.
AUTOSAR Today

Non- AUTOSAR

Software

Hardware

OVER THE AIR UPDATE

Classic AUTOSAR (C Code)

Application Software

RTE

Basic Software

Hardware

Adaptive AUTOSAR (C++ Code)

Adaptive Application Software

ARA

Services

Basis

High Performance Hardware

N  Non - AUTOSAR

C  Classic - AUTOSAR

A  Adaptive - AUTOSAR
Support for AUTOSAR Adaptive Platform

AUTOSAR Run-time for Adaptive (ARA)

- API: OS
- API: Execution
- API: Communication
- Service: S/W CM
- Service: Diagnostics

Adaptive AUTOSAR Foundation

- Adaptive AUTOSAR Services

High Performance Hardware/Virtual Machine
Generate Production AUTOSAR Adaptive C++ Code

Out-of-box AUTOSAR support
1. Configure Model
   ✓ Target: autosar_adaptive.tlc
   ✓ AUTOSAR Dictionary
2. Generate C++ code
Demo: Model Adaptive Software Components
Agenda

- AUTOSAR Blockset Introduction
  - Adaptive Platform
  - Classic Platform

- AUTOSAR ASW Development
  - AUTOSAR ASW architecture design
  - Testing in AUTOSAR Composition Editor

- System Composer with AUTOSAR Blockset
AUTOSAR Blockset and Embedded Coder for Classic Platform

Software Architecture Definition

Application Layer

Run Time Environment (RTE)

Basic Software

NVRAM Manager

Diagnostics Event Manager

ECU Abstraction Layer

Microcontroller Abstraction Layer

Complex Device Drivers

ECU Hardware

Modeling and Simulation

BSW Configuration & RTE Generation

Modeling & C Code Generation
Supported AUTOSAR Design Workflows

1. **Bottom-up workflow**
   - ARXML
   - Export
   - Simulink
   - Embedded Coder
   - C Code

2. **Top-down workflow**
   - ARXML
   - Import

3. **Round trip workflow**
   - AUTOSAR authoring tool
   - Integrate/Update
   - Export
   - Import

**MathWorks AUTOMOTIVE CONFERENCE 2019**
Remind S/W Component Workflows

Design of software components

Export software component description (arxml) and integrate C/C++ code

Integrate/Update

AUTOSAR authoring tool

Export

Import/ update external software component description (arxml)
AUTOSAR Software Architecture
Success so far

Integrate/Update

Export

AUTOSAR authoring tool

Integration test using composition

Export software component description (arxml) and integrate C/C++ code

Import external software composition/ component description (arxml)

Design of software components
AUTOSAR Software Architecture
Success so far?

Integrate/Update

Export

• Does not look like an AUTOSAR architecture model
  - Too much noise
  - Missing relevant information
• No purpose-built AUTOSAR composition editor in Simulink
• Can’t export Composition to ARXML
• Workflow requires using different tools
Now, You can start from AUTOSAR architecture design!
AUTOSAR Software Architecture
Today’s Focus - Make it easier

Step 1
Design of Software Composition(s)

Step 2
Design of Software Components & AUTOSAR configuration

Step 3
Integrate software description (arxml) and C/C++ code

INTEGRATE INTO PRODUCTION
Example: Automotive Throttle Body
Create AUTOSAR ASW Composition
Create Models from AUTOSAR ASW Component
Detailed Design using Simulink
Design and Configure for AUTOSAR SW
AUTOSAR Software Architecture
Bottom-up Workflow

Step 1: Design of Software Composition(s)

Step 2: Design of Software Components & AUTOSAR configuration

Step 3: Integrate software description (arxml) and C/C++ code

INTEGRATE INTO PRODUCTION
Launch Code Perspective for AUTOSAR Configuration
AUTOSAR Component Quick Start
Simulink to AUTOSAR Configuration

AUTOSAR Dictionary

Simulink-AUTOSAR Mapping Editor
Assemble AUTOSAR SW-C Models into a Composition
AUTOSAR Software Architecture

Import ARXML

Step 1
Design of Software Composition(s)

Step 2
Design of Software Components & AUTOSAR configuration

Step 3
Integrate software description (arxml) and C/C++ code

INTEGRATE INTO PRODUCTION

INTEGRATE INTO PRODUCTION
AUTOSAR Architecture from ARXML
Agenda

- AUTOSAR Blockset Introduction
  - Adaptive Platform
  - Classic Platform

- AUTOSAR ASW Development
  - AUTOSAR ASW architecture design
  - Testing in AUTOSAR Composition Editor

- System Composer with AUTOSAR Blockset
Simulate with BSW Service Blocks and Schedule Editor
Generate AUTOSAR Code and Export ARXML
Agenda

▪ AUTOSAR Blockset Introduction
  – Adaptive Platform
  – Classic Platform

▪ AUTOSAR ASW Development
  – AUTOSAR ASW architecture design
  – Testing in AUTOSAR Composition Editor

▪ System Composer with AUTOSAR Blockset
What is Systems Engineering?

- Interdisciplinary approach and means to enable the realization of successful systems

- Systems engineers:
  - Ensure requirements of customers, users and other stakeholders are met
  - Design optimized system architectures
  - Validate system architecture meets requirements
  - Evaluate system level behaviors

Source: SEBoK v. 1.9.1, released 16 October 2018
What is System Architecture?

- A conceptual model that defines the structure, behavior, and other views of a system, organized in a way that supports reasoning about the system.
Consideration in System Architecture

- Simulation in a system level using Simulink

AUTOSAR Composition

Throttle body control system
System Architecture with System Composer

- System architect considers physical and logical architecture

- Mechanical & Electrical hardware
- Basic software
- Application software
- Connection
  - Hard-wires
  - CAN
  - LIN
  - Etc.
Demo: Simulation in System Composer
System Engineering with System Composer

Intuitively design system and software architectures

Add stereotypes and trade study to optimize architecture

Link requirements, architectures, design, code and test

Simulation with multi-domain environment
요구사항부터 아키텍처 설계와 시뮬레이션까지 시스템 엔지니어링을 위한 방안

MathWorks Korea
Key Takeaways

- Design and develop AUTOSAR software
- Create Simulink component models
- Author AUTOSAR composition architecture
- Support for both Classic and Adaptive Platforms
- AUTOSAR perspective to view/edit mapping
- Blocks to model and simulate Basic Software and AUTOSAR library routines
- Optimized AUTOSAR C/C++ code generation and ARXML export
System Composer enables the definition, analysis, and specification of architectures and compositions for model-based systems engineering and software design. With System Composer, you allocate requirements while refining an architecture model that can then be designed and simulated in Simulink.

System Composer lets you create or import architecture models that describe a system in terms of components and interfaces. You can also populate an architecture model from the architectural elements of Simulink designs or C/C++ code. You can create custom live views of the model to study specific design or analysis concerns. With these architecture models you can analyze requirements, capture properties via stereotyping, perform trade studies, and produce specifications and ICDs.

AUTOSAR 블록셋
AUTOSAR 소프트웨어 설계 및 시뮬레이션

AUTOSAR 블록셋은 Simulink® 모델을 사용하여 AUTOSAR Dictionary 및 블록셋의 모든 블록을 구성할 수 있습니다. AUTOSAR 소프트웨어 패키지의 블록셋과 데이터를 구성하는 데 사용할 수 있습니다. AUTOSAR 블록셋은 Simulink 모델을 사용하여 컴포넌트 간의 인터페이스를 정의하고, 이를 AUTOSAR 레이어 구성을 구성하여 기존 Simulink 모델에 매핑할 수 있습니다. 또한, AUTOSAR 블록셋은 AUTOSAR XML 파일에서 소프트웨어 컴포넌트와 컴포넌트의 인터페이스를 가져오며 AUTOSAR의 새로운 Simulink 모델을 사용으로 만들어질 수 있는 것을 프로그램으로 제공할 수 있습니다.

AUTOSAR 블록셋은 Nv/RAM 및 Diagnostics를 포함한 AUTOSAR 레이어의 특성과 BSW(기본 소프트웨어) 세션의 블록셋 및 구성을 제공합니다. 자동 프로그램 소프트웨어 모델과 함께 BSW 서비스를 사용하여 Simulink를 만들어서 AUTOSAR ECU 소프트웨어를 생성할 수 있습니다.

AUTOSAR 블록셋은 C 및 C++ 채용 블록 설계 및 AUTOSAR XML 파일 내부에 구현되며 Embedded Coder®의 확장 ISO 26622 표준(EETWG 기반 표준)과 함께 사용할 수 있습니다.

Thank You!