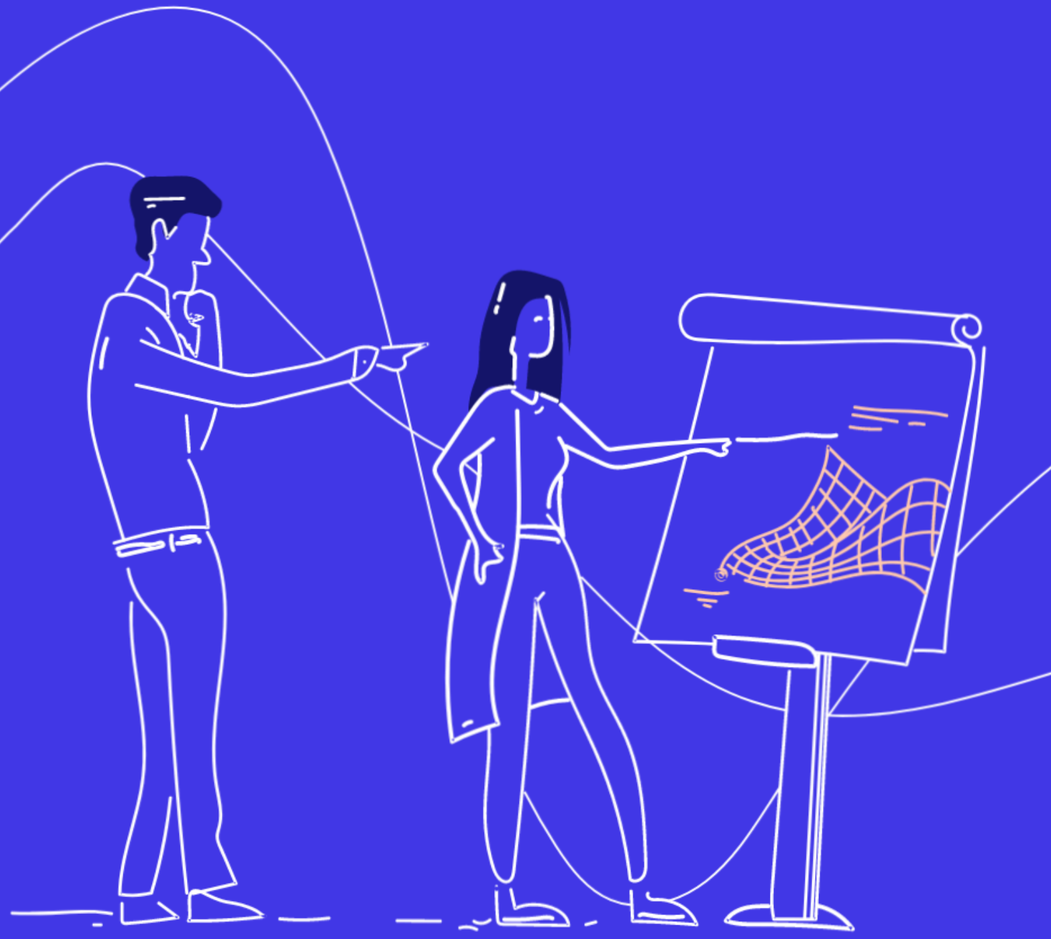
sciengineer

Code Generation for Classic AUTOSAR Software Components



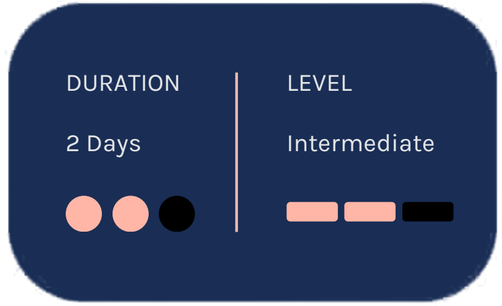
SciEngineer’s training courses are designed to help organizations and individuals close skills gaps, keep up-to-date with the industry-accepted best practices and achieve the greatest value from MathWorks® and COMSOL® Products.

Code Generation for Classic AUTOSAR Software Components

This two-day course discusses Classic AUTOSAR-compliant modeling and code generation using AUTOSAR Blockset. Workflows for top-down and bottom-up software development approaches are discussed in the context of Model-Based Design. This course is intended for automotive industry software developers and systems engineers who use Embedded Coder for automatic C/C++ code generation.

Prerequisites

- Simulink Fundamentals (or Simulink Fundamentals for Automotive Applications or Simulink Fundamentals for Aerospace Applications)
- Embedded Coder for Production Code Generation
- Knowledge of C programming language and the AUTOSAR standard



TOPICS

Day 1

- Importing AUTOSAR Software Components
- Exporting AUTOSAR Software Components
- Modeling AUTOSAR Ports and Interfaces
- Modeling AUTOSAR Events
- Modeling Calibration Parameters and Per-Instance Memory

Day 2

- Modeling AUTOSAR Variants
- Modeling AUTOSAR Compositions and Software Architecture
- Modeling Basic Software Services

Importing AUTOSAR Software Components

OBJECTIVE: Import existing ARXML files to automatically generate Simulink models with correct architecture and interfaces.

- AUTOSAR XML (ARXML) description
- Importing AUTOSAR software components
- Array and enumeration data types
- Adding functionality to generated Simulink models
- Simulating with a harness model
- AUTOSAR code generation
- Structure and fixed-point data types
- Managing imported data types in Simulink
- Importing modified ARXML files

Exporting AUTOSAR Software Components

OBJECTIVE: Configure Simulink models as AUTOSAR software components and define AUTOSAR ports and interfaces for ARXML generation.

- Creating AUTOSAR Components
- Mapping Simulink model objects to AUTOSAR software components
- Generating code and exporting ARXML files

Modeling AUTOSAR Ports and Interfaces

OBJECTIVE: Configure Simulink Inport and Outport blocks as various types AUTOSAR ports.

- AUTOSAR Communication overview (ports and interfaces)
- Modeling sender-receiver communication
- Modeling AUTOSAR modes
- Modeling client-server communication

Modeling AUTOSAR Events

OBJECTIVE: Model periodic and non-periodic AUTOSAR events in Simulink models.

- AUTOSAR events overview
- Simulating events in Simulink models
- Timing (periodic) events
- Nonperiodic events overview
- Modeling data received events
- Modeling mode switch events
- Triggering initialization
- Simulating nonperiodic events

Modeling Calibration Parameters and Per-Instance Memory

OBJECTIVE: Specify AUTOSAR calibration parameters and per-instance memory in a Simulink model using data objects and code mappings.

- Modeling calibration parameters provided by parameter SWCs
- Modeling internal calibration parameters: shared by all instances
- Modeling internal calibration parameters: separate copy for each instance
- Modeling per-instance memory

Modeling AUTOSAR Variants

OBJECTIVE: Configure a variant-based design within an AUTOSAR software component, and map Simulink variants to AUTOSAR variation points.

- Creating variant subsystems in rate-based models
- Generating AUTOSAR variants

Modeling AUTOSAR Compositions and Software Architecture

OBJECTIVE: Create an AUTOSAR software architecture design, and populate it with working AUTOSAR software components and compositions.

- Creating AUTOSAR software architectures and compositions
- Generating AUTOSAR software components
- Exporting AUTOSAR compositions and software components
- Simulating AUTOSAR software architecture models
- Importing AUTOSAR compositions
- Generating code for reusable software components

Modeling Basic Software Services

OBJECTIVE: Configure AUTOSAR software components to read and write diagnostic statuses and manage non-volatile RAMs.

- Basic software services overview
- Setting diagnostic statuses
- Monitoring diagnostic statuses
- Reading and writing NVRAM
- Monitoring NVRAM

sciengineer

Expand your knowledge