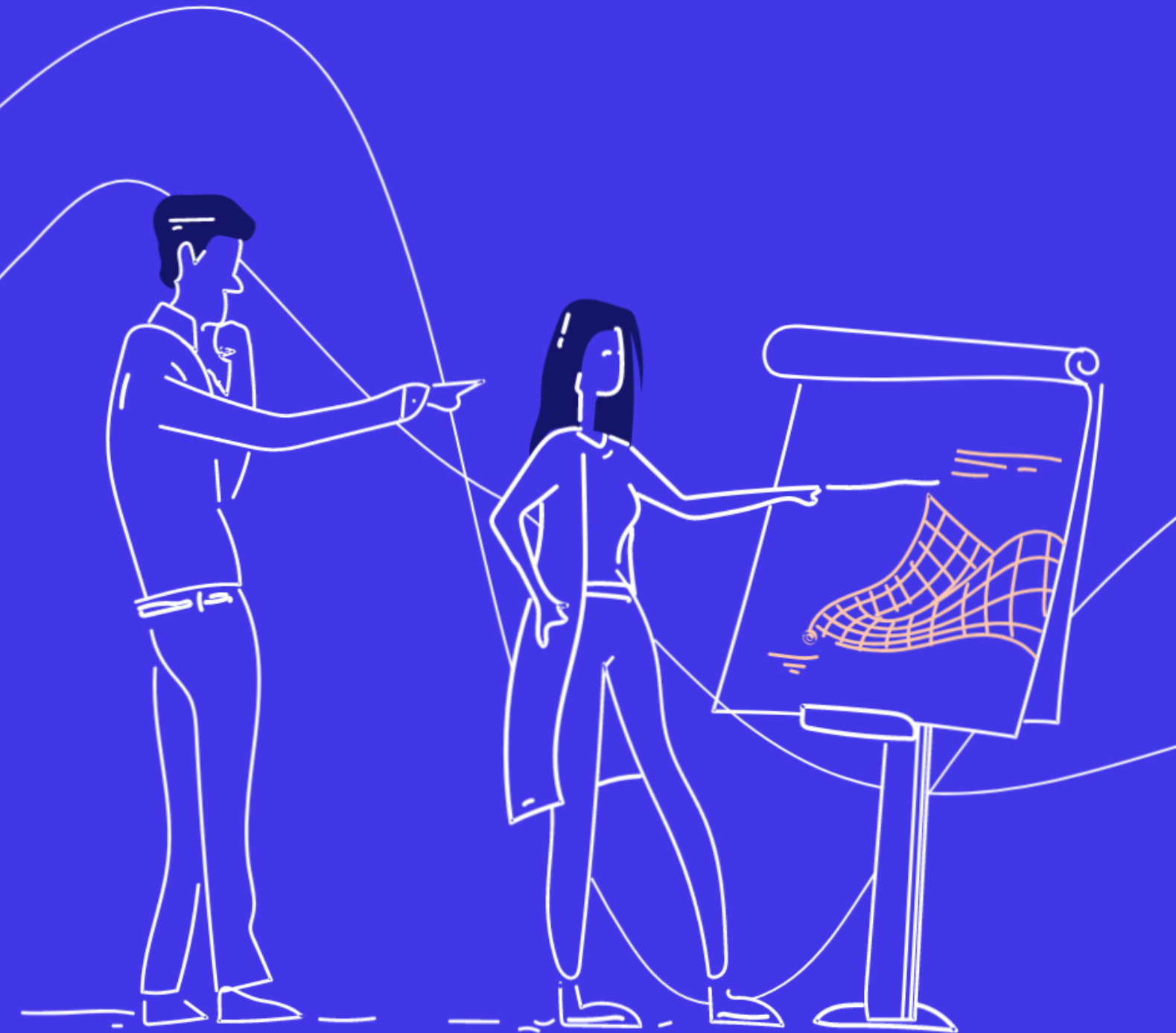




# Code generation for AUTOSAR Adaptive applications



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 AUTOSAR Adaptive applications

This two-day course focuses on AUTOSAR Adaptive-compliant modeling and code generation using AUTOSAR Blockset. Workflows for top-down and bottom-up software development are discussed in the context of Model-Based Design. This course is aimed at software developers in the automotive and vehicle industry as well as system engineers who use Embedded Coder for the automatic generation of C/C++ code.

## 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

DURATION	LEVEL
2 Days	Advanced
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>

### TOPICS

## Day 1

- Overview of the AUTOSAR Adaptive Platform
- Overview of the AUTOSAR Adaptive Platform
- Modeling AUTOSAR Adaptive Events
- Modeling AUTOSAR Events
- Modeling AUTOSAR Adaptive Multirate Applications

## Day 2

- Modeling AUTOSAR Adaptive Methods
- Importing AUTOSAR Adaptive Applications
- Modeling the AUTOSAR Adaptive Software Architecture
- Modeling AUTOSAR Adaptive Runtime Services

## Overview of the AUTOSAR Adaptive Platform

OBJECTIVE: Understand the different AUTOSAR platforms and how Simulink and the AUTOSAR Blockset simplify AUTOSAR Adaptive-compliant software development.

- Overview of AUTOSAR
- AUTOSAR Adaptive Platform
- Service-Oriented Architecture
- Using Simulink to develop AUTOSAR Adaptive-compliant applications

## Understanding the C++ code and artifacts in AUTOSAR Adaptive

OBJECTIVE: Configure a Simulink model for AUTOSAR Adaptive-compliant code generation and simulation. Navigate between the generated C++ code and the ARXML files.

- Generating generic C and C++ code
- Configuring Models for AUTOSAR Adaptive-Conformant Code Generation
- Generating AUTOSAR Adaptive C++ code and ARXML files
- Simulating AUTOSAR Adaptive Models

## Modeling AUTOSAR Adaptive Events

OBJECTIVE: Create and model AUTOSAR Adaptive ports and interfaces for event-based communication.

- Configuring AUTOSAR Adaptive Ports and Interfaces
- Using Simulink buses in AUTOSAR Adaptive applications

# Modeling AUTOSAR Adaptive Multirate Applications

OBJECTIVE: Planning the timing control with periodic and aperiodic events in an AUTOSAR Adaptive application.

- AUTOSAR Adaptive modeling domains
- Modeling Event-based AUTOSAR Adaptive Applications

## Modeling AUTOSAR Adaptive Methods

OBJECTIVE: Model client and server applications that communicate using AUTOSAR Adaptive methods.

- Configuring service interfaces between AUTOSAR Adaptive applications
- Modeling blocking method calls
- Modeling non-blocking method calls

## Importing AUTOSAR Adaptive Applications

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

- Description of AUTOSAR Adaptive XML (ARXML)
- Importing AUTOSAR Adaptive Applications
- array and enumeration data types
- Adding functionality to the generated Simulink models
- Simulating with a Harness Model
- AUTOSAR Adaptive Code Generation
- structure and fixed-point data types
- Managing imported data types in Simulink
- Importing modified ARXML files

## Modeling the AUTOSAR Adaptive Software Architecture

OBJECTIVE: Create an AUTOSAR Adaptive software architecture design and add functional AUTOSAR Adaptive applications and compositions.

- Creating AUTOSAR Adaptive software architectures and compositions
- Generating AUTOSAR Adaptive Applications
- Exporting AUTOSAR Adaptive compositions and applications
- Simulating AUTOSAR Adaptive software architecture models

# Modeling AUTOSAR Adaptive Runtime Services

OBJECTIVE: Configure AUTOSAR Adaptive applications to read and write persistent data and use the standard logging program.

- Modeling Key-Value Persistence
- Output in standard program for logging





# Expand your knowledge

