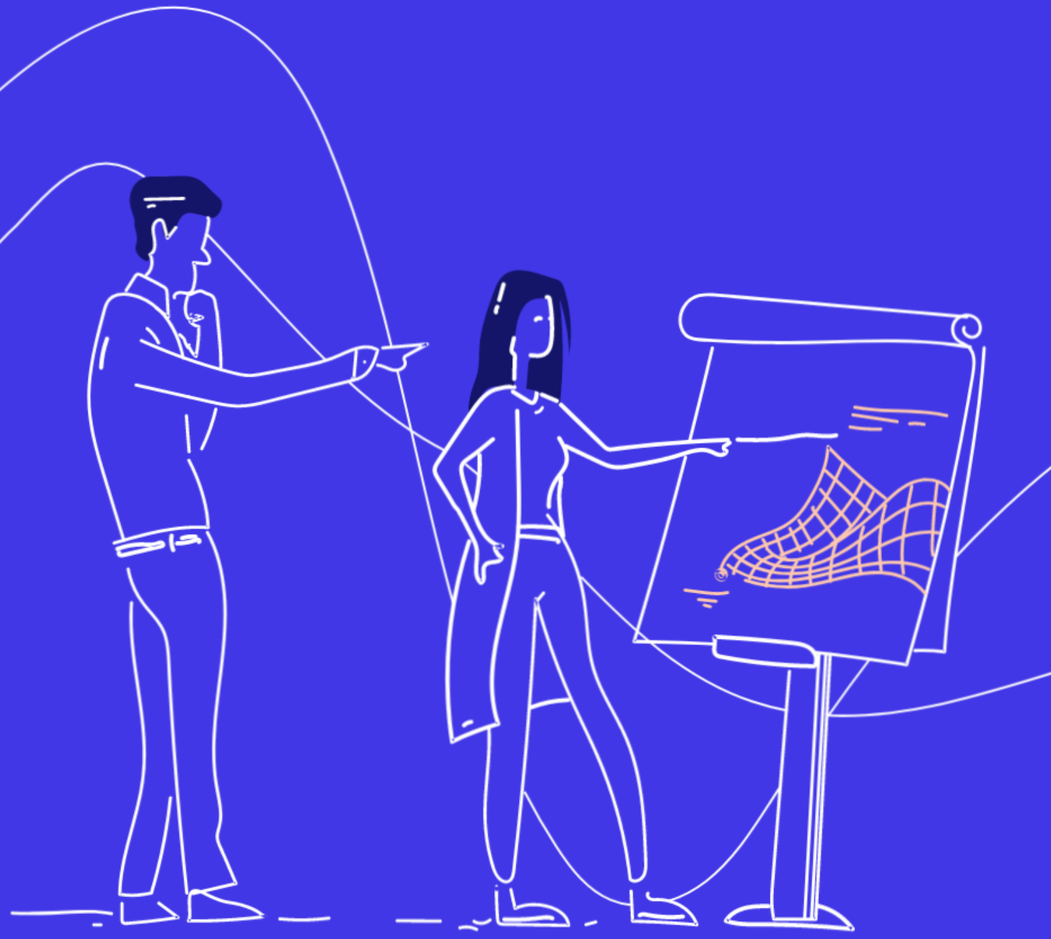
sciengineer

Automotive Software Certification Curriculum



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.

Automotive Software Certification Curriculum

SciEngineer's Training Services

Foundational

Stateflow for Automotive Applications
(2 Days)

System architecture

System Composer for Architecture Modeling
(1 Day)

System architecture

Modeling Physical Systems with Simscape
(1 Day)

Modeling Multibody Mechanical Systems with Simscape
(1 Day)

Real-Time Testing with Simulink Real-Time and Speedgoat Hardware
(2 Days)

Algorithm Design

Design Verification with Simulink
(1 Day)

Polyspace for C/C++ Code Verification
(1 Day)

Verification & Validation Implementation

Code Generation for Classic AUTOSAR Software Components
(2 Days)

Polyspace for C/C++ Code Verification
(1 Day)

Stateflow for Automotive Applications

This two-day course shows how to model and simulate decision logic using Stateflow. The course focuses on how to employ flow charts, state machines, truth tables, state transition tables and component-based modeling in Simulink designs. The course offers hands-on automotive examples and exercises that apply basic techniques to realistic problems in the automotive industry.

Prerequisites

MATLAB Fundamentals for Automotive Applications and Simulink Fundamentals for Automotive Applications.

Detailed course outline »

TOPICS

Day 1

- Modeling Flow Charts
- Modeling State Machines
- Hierarchical State Diagrams
- Parallel State Diagrams

Day 2

- Using Events in State Diagrams
- Calling Functions from Stateflow
- Truth Tables and State Transition Tables
- Component-Based Modeling in Stateflow

System Composer™ for Architecture Modeling

This one-day course focuses on developing and analyzing model-based architectures with System Composer™ and Requirements Toolbox™.

Prerequisites

Simulink Fundamentals or equivalent experience using Simulink is needed. Basic knowledge of Systems Engineering concepts is recommended.

[Detailed course outline »]

TOPICS

Day 1

- Introduction to Model-Based Design
- Working with Requirements
- Logical System Architectures
- Physical System Architectures
- Capturing and Analyzing System Properties
- Enhancing Architecture Traceability

Modeling Physical Systems with Simscape

This one-day course discusses how to model systems in several physical domains such as electrical, mechanical, and hydraulic. This course focuses on interpreting Simscape diagrams, combining them with Simulink models, modeling energy transfer between different physical domains, and creating userdefined Simscape components.

Prerequisites

MATLAB Fundamentals and Simulink Fundamentals

Detailed course outline »

TOPICS

Day 1

- Introduction to Simscape and the Physical Network Approach
- Working with Simscape Components
- Connecting Physical Domains
- Combining Simscape Models and Simulink Models
- Creating Custom Components with the Simscape Language

Modeling Multibody Mechanical Systems with Simscape

This one-day course focuses on how to model rigid-body mechanical systems in the Simulink environment using Simscape Multibody. Topics include: modeling simple multibody systems, creating reusable models of mechanical systems, importing models from CAD software and combining Simulink, Simscape and Simscape Multibody blocks.

Prerequisites

MATLAB Fundamentals, Simulink Fundamentals, and Modeling Physical Systems with Simscape

Detailed course outline »

TOPICS

Day 1

- Introduction to Multibody Simulation
- Refining Components
- Assembling Mechanisms
- Importing CAD Models
- Connecting to Simscape and Simulink

Real-Time Testing with Simulink Real-Time and Speedgoat Hardware

This two-day course focuses on real-time testing workflows using Simulink Real-Time and Speedgoat real-time target computers.

Topics include: Converting desktop-based simulation applications into real-time applications; Conducting rapid control prototyping with physical device under control; Creating interactive interfaces and formal test suites, Using standard communication protocols; Ptimizing real-time applications and hardware-in-the-loop testing.

Prerequisites

- Simulink Fundamentals (or Simulink Fundamentals for Automotive Applications or Simulink Fundamentals for Aerospace Applications)
- Knowledge of Simscape ™ preferred

Detailed course outline »

TOPICS

Day 1

- Workflow Overview
- Developing Real-Time Applications
- Building Interactive Interfaces

Day 2

- Automating Real-Time Tests
- Using Communications Protocols
- Optimizing Plant Models for Real-Time Execution
- Hardware-in-the-Loop Testing

Design Verification with Simulink

This one-day course focuses on using Simulink Design Verifier to ensure that a design is devoid of possible design errors, is fully tested, and satisfies necessary requirements. Themes of detecting design errors, automatically generating tests, property proving and managing model complexity are explored throughout the course.

Prerequisites

MATLAB Fundamentals and Simulink Fundamentals

Detailed course outline »

TOPICS

Day 1

- Understanding the Verification Workflow
- Detecting Design Errors
- Automatically Generating Tests
- Property Proving
- Managing Model Complexity

Polyspace for C/C++ Code Verification

This two-day course discusses the use of Polyspace Code Prover to prove code correctness, improve software quality metrics, and ensure product integrity. This course describes techniques for creating a verification project, reviewing and understanding verification results, emulating target execution environments, handling missing functions and data, managing unproven code, applying MISRA-C rules and reporting analysis results.

Prerequisites

Strong knowledge of C or C++

Detailed course outline »

TOPICS

Day 1

- Polyspace Workflow Overview
- Polyspace Bug Finder Analysis
- Analyzing Polyspace Code Prover Results
- Code Verification Checks

Day 2

- Managing Polyspace Code Prover Verifications and Results
- Adding Precision to Polyspace Code Prover Verifications
- Integration Analysis
- Application Analysis

Day 3

- (optional, available with private training only)
- Hands-On Instruction (Optional)

Code Generation for AUTOSAR Software Components

This one-day course discusses AUTOSAR-compliant modeling and code generation using the Embedded Coder Support Package for AUTOSAR Standard.

Topics included: Importing and exporting AUTOSAR software components; Configuring AUTOSAR communication elements; Modeling AUTOSAR events; Creating calibration parameters.

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
- Knowledge of the AUTOSAR standard

[Detailed course outline »]

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

Polyspace for C/C++ Code Verification

This two-day course discusses the use of Polyspace Code Prover to prove code correctness, improve software quality metrics, and ensure product integrity. This course describes techniques for creating a verification project, reviewing and understanding verification results, emulating target execution environments, handling missing functions and data, managing unproven code, applying MISRA-C rules and reporting analysis results.

Prerequisites

Strong knowledge of C or C++

Detailed course outline »

TOPICS

Day 1

- Polyspace Workflow Overview
- Polyspace Bug Finder Analysis
- Analyzing Polyspace Code Prover Results
- Code Verification Checks

Day 2

- Managing Polyspace Code Prover Verifications and Results
- Adding Precision to Polyspace Code Prover Verifications
- Integration Analysis
- Application Analysis

Day 3

- (optional, available with private training only)
- Hands-On Instruction (Optional)

The Value of an Experienced Training Expert

Our training courses are developed by MathWorks' team of training engineers with exclusive product knowledge gained from working closely with product developers. They acquire significant hands-on experience by using new products months before they are released and are always current on new capabilities.

Learn Relevant Skills

Each course contains a set of learning objectives designed to help participants quickly master necessary skills. Our hands-on approach allows participants to practice, apply, and evaluate their knowledge in the classroom.

Receive Expert Instruction

Our training employs industry-accepted best practices for adult learning and technical instruction, and has developed course content that facilitates a “Presentation, Practice, Test” approach to learning. All training engineers have been selected based on their theoretical knowledge, technical education, experience, and teaching ability.

Increase Team Success Rates

According to post-training surveys, teams who receive 40 hours of training meet project objectives three times as often as those who receive 30 hours or less. This increase in training time raises the likelihood of meeting objectives by 90%.

sciengineer

Expand your knowledge