Integrating C Code with Simulink

Training Objectives

This one-day course presents multiple methods for integrating C code into Simulink® models. Topics discussed include the C Caller and C Function blocks, Legacy Code Tool for wrapping external C functions into Simulink, and manually written C MEX S-functions. This course is intended for intermediate to advanced Simulink users.

Prerequisites

Simulink Fundamentals, *MATLAB Fundamentals*, and knowledge of C programming

Products

- Simulink

Course Outline

Day 1 of 1

Integrating External C Code Using Simulink Blocks (2.0 hrs)

Objective: Integrate C code into Simulink models using the C Caller and C Function blocks.

- Integrating algorithmic C code
- Integrating C code with custom data types
- Integrating code with states
- Sharing custom C code blocks

Creating S-Functions from Legacy Code (2.0 hrs)

Objective: Integrate C code into a Simulink model using automated tools.

- Calling external C functions with Legacy Code Tool
- Handling states
- Sharing compiled S-functions

Writing Wrapper S-Functions (2.0 hrs)

Objective: Integrate C code into a Simulink model by manually writing C MEX S-functions.

- Writing C MEX S-functions
- Calling external code from C MEX S-functions
- Work vectors
- Debugging C MEX S-functions
- Multirate C MEX S-functions

Deploying Integrated C Code (0.75 hrs)

Objective: Explore the procedures and limitations for automatically generating code with Simulink Coder™.

- Generating code from C Caller and C Function blocks
- Generating code from C MEX S-functions
- Function inlining
- Integrating target-specific code

C Code Integration Methods Review (0.25 hrs)

Objective: Review code integration methods and discuss the pros and cons of each.

- Review of all methods of code integration
- How to choose a code integration method

Appendix A: Integrating C++ Code (0.5 hrs)

Objective: Create S-functions that are defined using the C++ language.

- Review of work vectors
- Unit delay object
- Creating a C++ S-function

Appendix B: C Code Integration with the MATLAB® Function Block (1.0 hrs)

Objective: Call external C functions from inside the MATLAB Function block.

- Calling external C routines from a MATLAB Function block

Appendix C: Integrating C Code with S-Function Builder (1.0 hrs)

Objective: Create S-functions that call external C code using S-Function Builder.

- Writing a C MEX S-function with S-Function Builder
- Calling an external C routine with S-Function Builder

Appendix D: Integrating C Code with Stateflow® (0.5 hrs)

Objective: Call external C functions within action and conditions statements in Stateflow charts.

- Importing external C code to Stateflow®
- Calling C code from C action language charts
- Calling C code from MATLAB® action language charts