

Embedded Coder for Production Code Generation

OEM: MathWorks • Duration: 3 Days (24 hrs) • Code: MW-EMBEDDED-CODER-FOR-PRODUCTION-CODE-GE

COURSE MODULES & TOPICS

Day 1 of 3

Generating Embedded Code

- Architecture of an embedded application
- System specification
- Generating code
- Code modules
- Logging intermediate signals
- Data structures in generated code
- Verifying generated code
- Embedded Coder® build process

Optimizing Generated Code

- Optimization considerations
- Removing unnecessary code
- Removing unnecessary data support
- Optimizing data storage
- Profiling generated code
- Code generation objectives

Integrating Generated Code with External Code

- External code integration overview
- Model entry points
- Creating an execution harness
- Integrating generated code into an external project
- Controlling code destination
- Packaging generated code

Controlling Function Prototypes

- Default model function prototype
- Modifying function prototypes
- Generated code with modified function prototypes
- Model function prototype considerations

- Reusable function interface
- Function defaults

Day 2 of 3

Customizing Data Characteristics Using Data Objects

- Simulink® data objects overview
- Controlling data types with data objects
- Creating reconfigurable data types
- Controlling storage classes with data objects
- Controlling data type and variable names
- Data dictionaries

Creating Storage Classes

- User-defined storage classes
- Creating storage classes
- Using user-defined storage classes
- Sharing user code definitions

Customizing Generated Code Architecture

- Simulink model architecture
- Controlling code partitioning
- Generating reusable subsystem code
- Generating variant components
- Code placement options

Model Referencing and Bus Objects

- Creating reusable model references
- Controlling data type of bus signals
- Controlling storage class of bus signals
- Model Reference software testing

Day 3 of 3

Scheduling Generated Code Execution

- Execution schemes for single-rate and multirate systems
- Generated code for single-rate models
- Multirate single-tasking code
- Multirate multitasking code
- Generating exported functions

Testing Generated Code on Target Hardware

- Hardware support overview

- Arduino setup
- Validating generated code on target
- Target optimization overview
- Profiling generated code on target
- Using code replacement libraries
- Creating code replacement tables

Deploying Generated Code

- Embedded application architecture
- Creating a deployment harness
- Using device driver blocks
- Running a real-time application
- External mode

Integrating Device Drivers

- Device drivers overview
- Using Legacy Code Tool
- Customizing device driver components
- Developing a device driver block for Arduino

Improving Code Efficiency and Compliance

- Model Advisor
- Hardware implementation parameters
- Compliance with standards and guidelines