



## Course Title:

### **Integrating Code with Simulink**

## Course Purpose

This one-day course presents multiple methods for integrating code into Simulink models. The focus is on integrating C and MATLAB code. Topics discussed include C MEX S-functions, MATLAB code, and the Legacy Code Tool for wrapping external C functions into Simulink. This course is intended for intermediate to advanced Simulink users.

## Pre- requisites

Simulink for System and Algorithm Modeling course, MATLAB Fundamentals course, and knowledge of C programming.



- ✓ 1 training day
- ✓ Hours: 09:00-17:00
- ✓ Total training hours: 8

## Teaching method

The course combines lectures, demonstrations and practical exercises in MATLAB, using original training books from MathWorks. The course is in Hebrew but the training materials are in English.

עמוד מס' 1

### **Training Center Systematics - Contact information:**

**Phone number:** 03-7660111 Ext: 5 **Email:** training@systematics.co.il

**Website:** <http://www.systematics.co.il/mathworks>



## Course Objective:

### Code Integration Methods

**Objective:** Become familiar with the various code integration methods and discuss how a Simulink model interacts with user-defined blocks.

- Overview of all methods of code integration
- Introduction to S-functions

### Transitioning from MATLAB to Simulink

**Objective:** Integrate MATLAB code into Simulink models.

- Writing a MATLAB function in a MATLAB Function block
- Converting a MATLAB function to a MATLAB Function block
- MATLAB Function block coding standards

### Calling External Routines

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

- Calling an external C routine with the Legacy Code Tool
- Calling an external C routine in a MATLAB Function block

### Writing Wrapper S-Functions

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

- Writing a C MEX S-function
- Calling external code from a C MEX S-function
- Work vectors
- Inheriting input and output port dimensions
- Additional macros

### Code Generation Considerations

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

- Generating code from a MATLAB Function block
- Generating code from C MEX S-functions (Legacy Code Tool)

עמוד מס' 2

### **Training Center Systematics - Contact information:**

**Phone number:** 03-7660111 Ext: 5 **Email:** training@systematics.co.il

**Website:** <http://www.systematics.co.il/mathworks>



### Code Integration Methods Review

**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

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

- Review of work vectors
- Unit delay object
- C++ S-functions

### Appendix B: MATLAB S-Function

- Writing a Level 2 MATLAB S-function
- Converting a MATLAB function to a Level 2 MATLAB S-function

### Appendix C: S-Function Builder

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

### Appendix D: C Code Integration with Stateflow

- Incorporating external C code with Stateflow

### Appendix E: Fortran S-Functions

- Calling Fortran code from the wrapper S-function
- Compiling the Fortran object code
- Compiling and linking the C and Fortran code together

עמוד מס' 3

#### **Training Center Systematics - Contact information:**

**Phone number:** 03-7660111 Ext: 5 **Email:** training@systematics.co.il

**Website:** <http://www.systematics.co.il/mathworks>