

Release Notes for Simulink® PLC Coder™

How to Contact MathWorks



www.mathworks.com Web
comp.soft-sys.matlab Newsgroup
www.mathworks.com/contact_TS.html Technical Support



suggest@mathworks.com Product enhancement suggestions
bugs@mathworks.com Bug reports
doc@mathworks.com Documentation error reports
service@mathworks.com Order status, license renewals, passcodes
info@mathworks.com Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Release Notes for Simulink® PLC Coder™

© COPYRIGHT 2010–2012 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2012b

Workflow for behavioral simulation and code generation of motion instructions for RSLogix 5000 IDE	2
Code generation report with bidirectional traceability between model and code	3
Propagation of block descriptions to generated code comments and RSLogix 5000 AOI/routine descriptions	4
Code generation optimizations for efficient casts and signal reuse	5
Internal signals available as optional AOI outputs for debugging in RSLogix 5000 IDE	6
Rockwell Automation RSLogix 5000 IDE Version 19	7
Absolute Time Temporal Logic	8

R2012a

Code Generation for Rockwell Automation RSLogix 5000 Routines	10
Global Tunable Parameters for Generated Code from Rockwell Automation RSLogix 5000 Add-On Instructions and Routine Formats and Phoenix Contact PC WORX ..	11
Support for Absolute Time Temporal Logic for the Rockwell Automation RSLogix 5000 IDE	12
Integration of Externally Defined Symbols in Generated Code	13
Support for Configuring Tunable Parameters Using Simulink.Parameter Objects	14
Author Creation Data, Descriptions, and Sample Times in Generated Code Header Comments	15
Support for atan2	16
Convenience Dynamic Lookup Table Block	17
New Examples	18

R2011b

Automatic IDE Import of Subsystem Code Without Test	
Bench	20
Subsystem Function Block Code	21
New Demo	23

R2011a

Support for New PLC Target IDEs	26
Generated Code File Name Can Now Be Renamed	27
Generated Code File Header Change	28
Support for Lookup Table Blocks	29
Support for Fixed Point Data Types	30
CORDIC Trigonometric Functions	31
64-Bit Support	32
New Demos	33

R2010b

Support for Triggered Subsystems	36
Support for New PLC Target IDEs	37
Automatic Import of Generated Code	38
New Demo	39

R2010a

New Product	42
-------------------	----

R2012b

Version: 1.4
New Features: Yes
Bug Fixes: Yes

Workflow for behavioral simulation and code generation of motion instructions for RSLogix 5000 IDE

The Simulink® PLC Coder™ software now supports a workflow for the behavioral simulation and code generation of motion instructions for the Rockwell Automation® RSLogix™ 5000 IDE. For more information, see “Simulation and Code Generation of Motion Instructions”.

Code generation report with bidirectional traceability between model and code

Simulink PLC Coder now creates and displays a traceability report file. You can also opt to display the report in a model Web view. See the following Configuration Parameter options.

GUI option	Command-Line Property	Description
Generate traceability report	PLC_GenerateReport	Specify whether to create code generation report.
Generate model Web view	PLC_GenerateWebview	Include the model Web view in the code generation report to navigate between the code and model within the same window. You can share your model and generated code outside of the MATLAB® environment.

For more information, see “Information in Code Generation Reports”.

Propagation of block descriptions to generated code comments and RSLogix 5000 AOI/routine descriptions

The Simulink PLC Coder software now propagates block comments to generated code for all target IDEs. For more information, see “Propagation of Block Descriptions”.

For Rockwell Automation RSLogix 5000 AOI/routine target IDEs, the coder also generates the subsystem block description text as an AOI or routine description L5X XML tag. The IDE can then import the tag as part of AOI and routine definition in the generated code.

Code generation optimizations for efficient casts and signal reuse

An Optimization pane has been added to the Configuration Parameters dialog box PLC Coder node. This pane contains the following parameters:

GUI option	Command-Line Property	Description
Signal storage reuse	PLC_PLCEnableVarReuse	Reuse signal memory.
Remove code from floating-point to integer conversions that wraps out-of-range values	PLC_PLCEnableEfficientCast	Enable code removal for efficient casts.
Loop unrolling threshold	PLC_RollThreshold	Specify the minimum signal or parameter width for which a for loop is generated.

For more information, see “Model Architecture and Design”.

Internal signals available as optional AOI outputs for debugging in RSLogix 5000 IDE

The Simulink PLC Coder software now generates code where test point outputs to the top level subsystem are mapped to optional AOI outputs for RSLogix 5000 IDE. In the generated code, the variable tags that correspond to the test points have the property `Required=false`.

For more information on signals with test points, see “What Is a Test Point?”.

Rockwell Automation RSLogix 5000 IDE Version 19

The Simulink PLC Coder software now supports Rockwell Automation RSLogix 5000 IDE Version 19.

Absolute Time Temporal Logic

The Simulink PLC Coder product now enables absolute time temporal logic for all supported IDEs. In previous releases, this capability was supported only for the Rockwell Automation RSLogix IDE. For more information, see “Integrate Absolute Time Temporal Logic Code”.

R2012a

Version: 1.3
New Features: Yes
Bug Fixes: Yes

Code Generation for Rockwell Automation RSLogix 5000 Routines

The Simulink PLC Coder software now generates code for routines from the Rockwell Automation RSLogix 5000 IDE.

- Load the code generated from routines without first restarting the Rockwell Automation RSLogix 5000 PLC. You can now:
- Take advantage of RSLogix user defined types (UDTs) to preserve model hierarchy in routine code and represent model.
- Observe that reusable subsystems become separate routine instances and access instance data in program UDTs.

To accommodate this capability:

- In the Configuration Parameters dialog box **PLC Code Generation > Target IDE** parameter, the Rockwell RSLogix 5000 17, 18: Routine option was added.
- In the Configuration Parameters dialog box **PLC Code Generation > Target IDE** parameter, the Rockwell RSLogix 5000 17, 18 option was changed to Rockwell RSLogix 5000 17, 18: AOI. This renamed option continues to generate code for Add-On instruction constructs, as in previous releases.
- In the command-line PLC_TargetIDE parameter, the rslogix5000_routine option was added.

For more information, see Target IDE.

Global Tunable Parameters for Generated Code from Rockwell Automation RSLogix 5000 Add-On Instructions and Routine Formats and Phoenix Contact PC WORX

The Simulink PLC Coder software supports global tunable parameters for generated code from Rockwell Automation RSLogix 5000 Add-On instructions (AOIs) and routine formats and Phoenix Contact® PC WORX™. For more information on how tunable parameters are mapped, see About Tunable Parameters in the Simulink PLC Coder Environment in the Simulink PLC Coder User's Guide.

Support for Absolute Time Temporal Logic for the Rockwell Automation RSLogix 5000 IDE

The Simulink PLC Coder software now supports absolute time temporal logic in Stateflow® charts for the Rockwell Automation RSLogix 5000 IDE. The coder does not support absolute time temporal logic for all other target IDEs.

Note If your model uses absolute time temporal logic, you cannot create test bench code for that model.

Integration of Externally Defined Symbols in Generated Code

You can now suppress symbol definitions in the generated code. This suppression allows the generated code to refer to these symbols. You must then provide these definitions when importing the code into the PLC IDE. For more information, see [Integrating Externally Defined Symbols](#).

Support for Configuring Tunable Parameters Using Simulink.Parameter Objects

You can now configure tunable parameters using `Simulink.Parameter` objects. In previous releases, you could only configure tunable parameters using the Configuration Parameters dialog box. For more information, see [Working with Tunable Parameters in the Simulink PLC Coder Environment](#).

Author Creation Data, Descriptions, and Sample Times in Generated Code Header Comments

The Simulink PLC Coder generated code header now includes:

- Author names from model properties
- Creation dates from model properties
- Model descriptions from model properties
- Fundamental sample times for the model and the subsystem block for which you generate code

Support for atan2

The Simulink PLC Coder software now supports the math function atan2.

Convenience Dynamic Lookup Table Block

As a convenience, the DynamicLookup block has been added to the plclib/Simulink/Lookup Tables sublibrary. In previous releases, you could achieve the dynamic lookup behavior using the Prelookup block with the Interpolation Using Prelookup block. Going forward, use the plclib/Simulink/Lookup Tables/DynamicLookup block.

New Examples

The following examples are new:

- **Speed Cruise Control System Using Variable-Step Continuous Solver** — Illustrates code generation for the variable-step continuous solver. In this example, the controller subsystem has a fixed sample time, while the model has a variable-step continuous solver.
- **Mapping Tunable Parameters Defined Using Simulink.Parameter Objects to Structured Text** — Illustrates the specification of tunable parameters using Simulink.Parameter objects in the MATLAB base workspace.
- **Generating Structured Text for Stateflow Chart with Absolute Time Temporal Logic** — Illustrates code generation for Stateflow Chart blocks with absolute time temporal logic. This example requires the Rockwell Automation RSLogix AOI or routine format.
- **Integrating User Defined Function Blocks, Data Types, and Global Variables into Generated Structured Text** — Illustrates how to integrate user defined function blocks, data types, and global variables and constants into generated structured text.

R2011b

Version: 1.2.1
New Features: Yes
Bug Fixes: Yes

Automatic IDE Import of Subsystem Code Without Test Bench

The Simulink PLC Coder software now generates and imports subsystem code into target IDEs without the test bench. To use this feature:

- 1** In the Configuration Parameters dialog box, clear the **Generate testbench for subsystem** check box.
- 2** In the Simulink editor, right-click the subsystem and select **PLC Code Generation > Generate and Import Code for Subsystem**.

In previous releases, the coder generated and imported test bench code into the target IDE regardless of the setting of the **Generate testbench for subsystem** check box.

Subsystem Function Block Code

Compatibility Considerations: Yes

In generated code, the function block code of the top-level subsystem has been simplified. The coder now generates the function block code depending on whether or not the top-level subsystem has internal state. In previous releases, the coder always generated the function block code with the `ssMethodType` parameter for top-level subsystems.

Compatibility Considerations

This release simplifies the function block code of the top-level subsystem for generated code.

- If the top-level subsystem in the Simulink model has internal state, the generated function block for the block will have an extra first parameter `ssMethodType` of integer type. This extra parameter is in addition to the function block I/O parameters mapped from Simulink block I/O ports.

To use the function block:

- 1** Initialize the block by calling the function block with `ssMethodType` set to integer constant `SS_INITIALIZE`.
- 2** If the IDE does not support symbolic constants, set `ssMethodType` to integer value 0.
- 3** For each follow-up invocation, call the function block with `ssMethodType` set to constant `SS_STEP`.
- 4** If the IDE does not support symbolic constants, set `ssMethodType` to integer value 1.

These settings cause the function block to initialize or compute and return output for each time step.

- If the top-level subsystem does not have internal state, the function block code has only parameters mapped from Simulink block I/O ports. There is no `ssMethodType` parameter. To use the function block in this case, call the function block with I/O arguments.

For non-top-level subsystems, either with or without internal state, the function block code has the `ssMethodType` parameter. The generated code might have other `ssMethodType` constants to implement Simulink semantics.

New Demo

The following demo is new:

- **Generating Structured Text for a Simple Simulink Subsystem without Internal State** — Illustrates changes for function block prototypes in generated code.

R2011a

Version: 1.2
New Features: Yes
Bug Fixes: Yes

Support for New PLC Target IDEs

The Simulink PLC Coder software now supports code generation and automatic import of code for the Phoenix Contact PC WORX IDE.

See Supported IDE Platforms in the Simulink PLC Coder User's Guide for more information.

Generated Code File Name Can Now Be Renamed

You can now specify a custom name for the code file that you generate with Simulink PLC Coder. Use the **Function name options** parameter in the Subsystem block.

Generated Code File Header Change

The comment header in the code file that you generate with Simulink PLC Coder now includes a sample time field for the model.

Support for Lookup Table Blocks

Simulink PLC Coder models can now generate code for lookup table blocks.

Support for Fixed Point Data Types

Simulink PLC Coder models can now generate code for fixed point data types. For more information, see Fixed-Point Data Type Limitations in the Simulink PLC Coder User's Guide.

CORDIC Trigonometric Functions

The Simulink PLC Coder product now supports code generation for CORDIC trigonometric functions. This support enables you to use trigonometric functions for PLCs that do not support these functions in built-in libraries.

To generate code for CORDIC trigonometric functions:

- 1** Add the Simulink Trigonometric Function block to the coder subsystem.
- 2** Configure the block to the desired trigonometric function.
- 3** From the **Approximation method** parameter, select CORDIC.
- 4** Generate code for the atomic subsystem.

64-Bit Support

The Simulink PLC Coder product supports 64-bit systems. You can still use the Simulink PLC Coder product with 32-bit IDEs.

See the MathWorks Web site at [Supported IDEs](#) for a list of supported IDEs and platforms.

New Demos

The following demos are new:

- Airport Conveyer Belt Control System — Illustrates code generated for an airport conveyer belt.
- Generating Structured Text for Simulink Model with Fixed-Point Data Types — Illustrates generating fixed-point code in the Simulink PLC Coder environment.

R2010b

Version: 1.1
New Features: Yes
Bug Fixes: No

Support for Triggered Subsystems

You can now use the Simulink PLC Coder software to generate code from Simulink triggered subsystems. Use the Triggered Subsystem block. See How Triggered Subsystem Code Maps to Function Blocks in the Simulink PLC Coder User's Guide.

Support for New PLC Target IDEs

The Simulink PLC Coder software now supports:

- Siemens® SIMATIC® STEP® 7 IDE
- KW-Software MULTIPROG® 5.0 IDE

See Supported IDE Platforms in the Simulink PLC Coder User's Guide for more information.

Automatic Import of Generated Code

You can now automatically import structured text code, generated by the Simulink PLC Coder software, to your PLC IDE. In previous releases, you imported the generated code manually according to the instructions provided by the PLC IDE manufacturer.

You can take advantage of this capability for the following PLC IDEs:

- CoDeSys IDE V2.3
- Rockwell Automation RSLogix 5000 IDE
- Siemens SIMATIC STEP 7 IDE
- KW-Software MULTIPROG 5.0 IDE

See *Automatically Importing Structured Text Code in the Simulink PLC Coder User's Guide* for more information.

New Demo

A new Simulink PLC Coder demo, Speed Cruise Control System Using Simulink and Stateflow, illustrates code generated for a cruise control controller subsystem using a triggered subsystem.

R2010a

Version: 1.0
New Features: Yes
Bug Fixes: No

New Product

Simulink PLC Coder generates hardware-independent IEC 61131-3 structured text from Simulink models, Stateflow charts, and Embedded MATLAB® functions. The structured text is generated in PLCopen and other file formats supported by widely used integrated development environments (IDEs). As a result, you can compile and deploy your application to numerous programmable logic controller (PLC) and programmable automation controller (PAC) devices.

Simulink PLC Coder generates test benches that help you verify the structured text using PLC and PAC IDEs and simulation tools.

Key features:

- Automatic generation of IEC 61131-3 structured text
- Simulink support, including reusable subsystems, PID controller blocks, and lookup tables
- Stateflow support, including graphical functions, truth tables, and state machines
- Embedded MATLAB support, including if-else statements, loop constructs, and math operations
- Support for multiple data types, including Boolean, integer, enumerated, and floating-point, as well as vectors, matrices, buses, and tunable parameters
- IDE support, including B&R Automation Studio®, PLCopen, Rockwell Automation RSLogix 5000, and Smart Software Solutions CoDeSys
- Test-bench creation