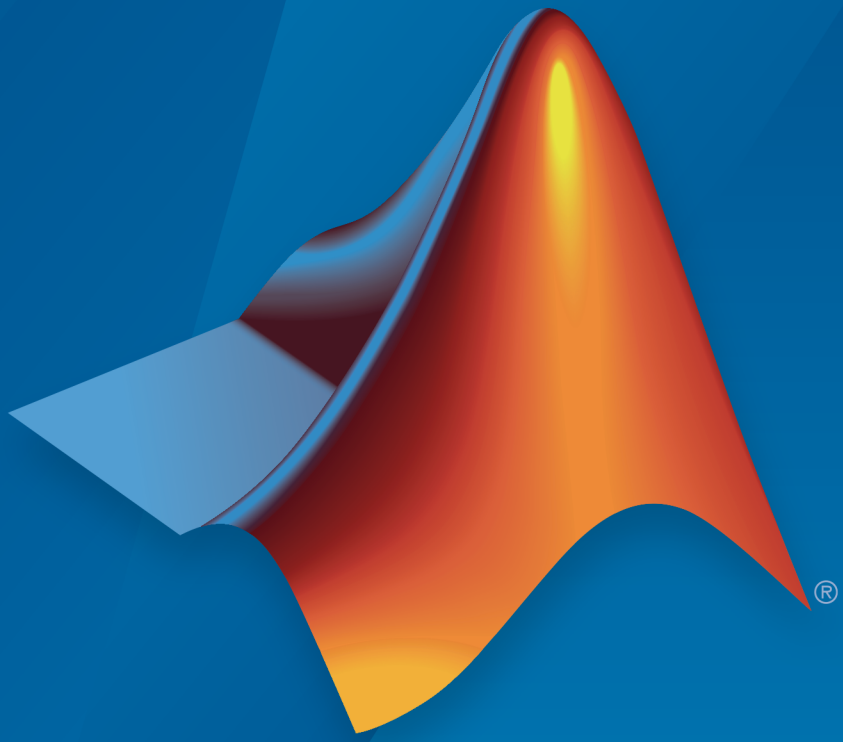
Simulink® PLC Coder™ Release Notes



MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

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

Simulink® PLC Coder™ Release Notes

Contents

R2017a

R2016b

R2016a

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2017a

Version: 2.3

New Features

Bug Fixes

Code Optimization for Initialization Code: Optimize generated code by removing redundant timer initialization calls

In R2017a, Simulink® PLC Coder™ generates optimized initialization code when Stateflow® charts have absolute time temporal logic.

Previously, if the Stateflow chart had absolute time temporal logic, the generated code had multiple instances of the timer initialization call. With this change, the software scans for and finds consecutive redundant timer initialization calls and removes these statements, resulting in efficient code.

`rand` Function Support: Generate code for `rand` functions on PLC IDEs that support `uint32`

Previously, Simulink PLC Coder did not support random number functions.

In R2017a, Simulink PLC Coder generates code for MATLAB Function blocks that use `rand` functions from the library. PLC IDEs must support the `uint32` data type. The software has conformance checks to report diagnostics for incompatible targets.

Syntax Highlighting in Code Generation Report: Read generated code more easily with syntax highlighting

In 2017a, enhancements in syntax highlighting greatly improve the readability of the code generated with Simulink PLC Coder.

Previously, the HTML code generation report used only two highlighting colors: blue for code and green for comments. With this change, PLC-specific keywords are highlighted in blue and the rest of the code is in black. Comments are still highlighted in green.

Code Optimization for Unused Stateflow Events: Generate more efficient code for unused events

In R2017a,when you have unused events in Stateflow charts, Simulink PLC Coder optimizes the generated code to remove these unused events.

FB Call ssmethod output assignment optimization

In R2017a, Simulink PLC Coder optimizes the `ssmethod` call output value assignments by generating code for step and output methods only when the output values are used.

STEP7 and TIA Portal INOUT var check

STEP® 7 and TIA Portal IDE do not support `INOUT` variables for single precision targets.

In R2017a, if the Simulink model has MATLAB Function blocks that use inplace variables, Simulink PLC Coder generates code by converting `INOUT` variables to standard input and output variables. However, if the MATLAB Function block is marked as reusable, conversion is not possible and a conformance check error is issued.

B&R Automation Studio 4 and Beckhoff TwinCAT 3: Generate code for these IDEs

In R2017a, you can generate structured text code dedicated to these IDEs:

- B&R Automation Studio™ 4
- Beckhoff® TwinCAT® 3

The Target IDE drop down list in Simulink Configuration Parameters shows these IDEs as options. You can also use the command line API to select these two targets and generate code.

R2016b

Version: 2.2

New Features

Bug Fixes

Ladder Logic Support: Generate ladder diagrams from Stateflow charts for CODESYS 3.5 IDE and Rockwell Automation AOIs

In R2016b, you can generate Ladder Diagram code in PLC Open XML format from Stateflow charts. You can import the generated code to IDEs such as CODESYS 3.5 and Rockwell Automation® AOIs, and view them as ladder diagrams. You can also validate the generated code using one of the following:

- Create a validation model containing the Ladder Diagram code. You can compare the output of the model with the original Stateflow chart.
- Create testbench code. You can import the generated code and the testbench code to the CODESYS 3.5 IDE and verify your generated code against the testbench.

For more information, see Ladder Diagram Generation for PLC Controllers.

Rockwell Automation IDE Support: Generate code for RSLogix 5000 V20 and Studio 5000 Logix Designer V24 IDEs

In R2016b, you can generate structured text for newer versions of the following Rockwell Automation IDEs:

- RSLogix 5000 (version 20)
- Studio 5000 Logix Designer (version 24)

For more information, see Target IDE.

Multirate Support: Generate code from multirate models for Siemens IDEs and Rockwell Automation AOIs

In R2016b, you can generate code from multirate models for the following Siemens® IDEs:

- Siemens SIMATIC® STEP 7
- Siemens TIA Portal
- Rockwell Automation Studio 5000 Logix Designer for AOI format
- Rockwell Automation RSLogix™ 5000 for AOI format

For information on support for multirate models in Simulink PLC Coder, see Generated Code Structure for Multirate Models and Multirate Model Limitations.

Global Variables for Rockwell Automation IDEs: Generate code for global variables by using INOUT variables for Rockwell Automation AOIs

In R2016b, if you generate code for Rockwell Automation IDEs using Add-On Instruction (AOI) constructs, your models can use global data, for instance, in Data Store Memory blocks. The global data appear as `INOUT` variables in generated code. The target IDEs supported for this feature are:

- Rockwell Automation Studio 5000 Logix Designer for AOI format
- Rockwell Automation RSLogix 5000 for AOI format

Previously, you could not use models with global data to generate code for Rockwell Automation IDEs using Add-On Instruction (AOI) constructs, because global variables are not supported in AOIs.

For more information on the workflow, see Generate Global Variables from Signals in Model.

Improved Code for Reusable Subsystems: Generate better reusable code for reusable subsystems

In R2016b, you can generate better reusable code for reusable subsystems.

Previously, if the same subsystem had multiple instances and some instances had constant inputs, the software generated separate function blocks for each instance. With this change, the software does not consider whether the inputs to the subsystem are constant and generates one function block for the multiple instances.

For more information, see Generate reusable code.

R2016a

Version: 2.1

New Features

Bug Fixes

`INOUT` Variable Support: Generate `INOUT` variables for MATLAB Function and Truth Table blocks that use the same name for input and output data

In R2016a, you can generate code for models that contain a MATLAB® Function block or a Truth Table block when the block has input and output variables with the same name. The generated code uses `InOut` parameters for those variables.

This capability is not supported if you select:

- **Rockwell RSLogix 5000: Routine** as your target IDE, because the Rockwell Automation RSLogix 5000 routines do not support input, output or `InOut` parameters. For this target IDE, you cannot generate code if a MATLAB Function block or a Truth Table block has input and output variables with the same name.
- **KW-Software MULTIPROG 5.0** or **Phoenix Contact PC WORKX 6.0** as your target IDE, and the variables have array or structure data types.

Alias Data Type Support: Optionally preserve alias names for data types in generated code to help integration with target-specific data types

In R2016a, you can preserve alias data types from your model in generated code.

You can create an alias for a built-in Simulink data type by using the Simulink.AliasType class. If you assign an alias data type to signals and parameters in your model, you can preserve the alias data type in your generated code.

For instance, you can use alias names that denote safe data types in your generated code. Using these safe data types, you can conform to PLCopen safety specifications that require clear differentiation between safety-relevant signals and standard signals.

For more information, see Preserve Alias Type Names for Data Types.

Simulink Requirements Links: Embed requirements links as comments in generated code

In R2016a, if you create links to requirements documents from your model with the Simulink Verification and Validation™ software, the links appear in generated code

comments. When you view the code in the Code Generation Report, you can open the links from the comments.

See View Requirements Links from Generated Code.

Simulink Design Verifier Integration: Generate code with multiple test benches from test harness models created with Simulink Design Verifier

In R2016a, if the input to your subsystem consists of multiple signal groups, you can generate code with multiple test benches.

To provide multiple signal groups as inputs, do one of the following:

- Use a Signal Builder block with multiple signal groups to provide inputs to the subsystem.
- Create a test harness model from the subsystem with Simulink Design Verifier™. In the test harness model, a Signal Builder block with one or more signal groups is created to test the model. Copy this block from the test harness model and use it to provide inputs to your subsystem.

For more information on generating multiple test benches, see Verify Generated Code with Multiple Test Benches.

64-bit Windows 7 Support for Siemens STEP 7 and RSLogix 5000 IDEs: Generate, import, and verify code for these IDEs

In R2016a, you can generate, import, and verify code in Siemens STEP 7 and RSLogix 5000 IDEs for 64-bit Windows® 7.

CODESYS 3.5 POU Block Description Support: Generate block descriptions as POU descriptions in code generated for CODESYS 3.5

In R2016a, when you generate code for the CoDeSys 3.5 IDE, Simulink PLC Coder can propagate block descriptions from the model into the `documentation` XML tag. When you import the generated code into the CoDeSys 3.5 IDE, the IDE parses the content of this tag and provides readable descriptions of the function blocks in your code.

For more information, see Propagate Block Descriptions to Code Comments.

Code generation for models containing `enum` to Integer Conversion

In R2016a, you can generate code from models that convert constant values from `enum` to integer data types using Data Type Conversion blocks.

Code Generation for subsystems with no input or output

In R2016a, you can generate code for non-empty subsystems that do not have inputs or outputs. Previously, you encountered an error when generating code from such subsystems. For code generation from a subsystem with no inputs or outputs, you must set the **Function packaging** parameter of the block to `Reusable function`.

However, you cannot generate testbench code for such subsystems because testbench code requires output from subsystems.

Support for KW-Software MULTIPROG 5.5 IDE

Simulink PLC Coder now supports version 5.5 of the KW-Software MULTIPROG® IDE.

R2015b

Version: 2.0

New Features

Bug Fixes

SIEMENS TIA Portal V12 and V13 IDE Support: Generate code for these IDEs

Simulink PLC Coder supports the Siemens TIA Portal target IDE. Versions V12 and V13 are supported.

For more information on:

- How to select the IDE, see Target IDE.
- How to integrate generated code with an existing Siemens TIA Portal project, see Integrate Generated Code with Siemens TIA Portal Projects.

Streamlined Target IDE Selection: Choose target IDE more quickly

In Simulink Configuration Parameters, on the **PLC Code Generation** pane, the default **Target IDE** list shows a reduced subset of target IDEs for easier navigation. You can customize this reduced **Target IDE** list and specify more frequently used IDEs using the `plccoderpref` function. For more information, see plccoderpref.

To see all target IDEs supported by Simulink PLC Coder, select **Show full target list**. For more information, see Show full target list.

Absolute Time Temporal Logic by Using IEC 61131 Timer: Generate code for this Stateflow construct

Simulink PLC Coder supports code generation from models with Stateflow charts that use Absolute Time Temporal Logic.

For more information, see Stateflow Chart with Absolute Time Temporal Logic.

Global Variables for Siemens IDEs: Generate code for global data store memory using `Simulink.Signal` objects for Siemens STEP 7 and TIA Portal IDEs

Simulink PLC Coder supports code generation for global data store memory using Simulink.Signal objects for Siemens STEP 7 and TIA Portal IDEs.

Additional Math Function Support: Generate code for hyperbolic functions

Simulink PLC Coder supports code generation from models that contain hyperbolic functions such as sinh, tanh, etc.

Code Optimizations: Generate more efficient code for type casts

Simulink PLC Coder generates more efficient code by removing unnecessary type casts.

Linked Subsystems Code Verification: Verify that generated code results match simulation results

Simulink PLC Coder supports test bench code generation from linked subsystems.

For more information on test bench code, see Verification.

Improved code for global data store memory using `Simulink.Signal` objects

Simulink PLC Coder generates more efficient code for global data store memory using Simulink.Signal objects.

Code generation from models with atomic subcharts

Simulink PLC Coder allows code generation from models that contain Stateflow charts with atomic subcharts in them. However, before code generation, the software converts the atomic subcharts to regular subcharts.

Unnecessary variables removed from generated code for Data Store Memory blocks

If your model contains Data Store Memory blocks, the code generated by Simulink PLC Coder no longer contains unnecessary variables related to code coverage.

R2015a

Version: 1.9

New Features

Bug Fixes

Code generation for 3S-Smart Software Solutions CoDeSys V3.5 IDE

Simulink PLC Coder supports CoDeSys IDE V3.5.

Generation of code that preserves variable names in MATLAB Function blocks

The generated structured text from Simulink PLC Coder retains the names of variables defined in MATLAB functions.

This behavior allows you to easily map variables defined in your MATLAB code to the ones in the generated structured text. Earlier, certain optimizations during code generation caused reuse of variable names.

R2014b

Code generation for Rexroth IndraWorks version 13V12 IDE

Simulink PLC Coder supports Rexroth IndraWorks version 13V12 IDE.

Code generation for OMRON Sysmac Studio v1.09 IDE

Simulink PLC Coder supports OMRON® Sysmac® Studio v1.09 IDE.

Code generation support for exported functions in Stateflow

Simulink PLC Coder supports code generation for Stateflow exported functions. Functions can be defined in one Stateflow chart, and then exported and called by other charts.

Code generation support for global data store memory using Simulink.Signal object

Simulink PLC Coder supports code generation for global data store memory using `Simulink.Signal` objects. This support applies to IDEs which support global variables.

Variable names preserved for function block inputs and outputs

For PLC code generation, Simulink PLC Coder preserves function block input and output variable names in generated code. If variable names conflict with reserved names or keywords in the target, they are changed.

R2014a

Version: 1.7

New Features

Bug Fixes

Static code metrics report

Simulink PLC Coder reports key characteristics of the generated code, such as number of variables and lines of code in each function block. For more information, see Generate a Static Code Metrics Report.

Code generation for Siemens STEP 7 V5.5 IDE, B&R Automation Studio 4 IDE, and Beckhoff TwinCAT 3 IDE

Simulink PLC Coder supports Siemens STEP 7 version 5.5, B&R Automation Studio® 4, and Beckhoff TwinCAT 3.

Model block description in generated code for CoDeSys 2.3 IDE

Simulink PLC Coder generates a model block description in the code for the CoDeSys 2.3 IDE.

Simulink.Parameter description in generated code for Codesys 2.3 IDE

Simulink PLC Coder generates a `Simulink.Parameter` description in the code for the CoDeSys 2.3 IDE.

Change in Diagnostic Viewer launch behavior after code generation

After PLC code generation, the Diagnostic Viewer window showing the PLC Code Generation log no longer launches automatically. Instead, a "View diagnostics" hyperlink appears at the bottom of the Simulink model window. This link opens the Diagnostic Viewer and the PLC Code Generation log with links to the generated code.

R2013b

Version: 1.6

New Features

Bug Fixes

Masked parameters for atomic subsystems

Mask parameters for subsystems now map to function block inputs in the generated code.

To see how mask parameters map to generated code, see Generated Code Structure for Subsystem Mask Parameters.

Reusable code for intrinsic functions

For internal MATLAB functions, such as `atan2`, Simulink PLC Coder generates a single copy of the function in the generated code.

Millisecond and microsecond units with absolute-time temporal logic

You can generate code for Stateflow absolute-time temporal logic that uses the millisecond (`msec`) and microsecond (`usec`) time units.

PC WORX IDE support improvements including enhanced support for global tunable parameters

Simulink PLC Coder generates code for the PC WORX™ IDE with the following improvements:

- Global tunable parameters that are structures and array data types are initialized in a `PLC_INIT_PARAMETERS` function block. For more information, see Global Tunable Parameter Initialization for PC WORX.
- `floor` and `ceil` rounding is supported.
- Array type names incorporate data type description and size.

  For example, if you have a 51-element array of real data, the generated code for the array data type is:

  ```
  PLC_ARRAY_0_50_LREAL: ARRAY [0..50] OF LREAL;
  ```
- Code generation header comments and block description comments are in the body of the generated code, and are therefore visible when you import the code into the IDE.

Temporary variable minimization

The coder minimizes the number of temporary variables. This optimization improves code quality and memory usage.

Relative tolerance for test bench data comparison

When checking single and double data type values, the test bench now uses a relative error tolerance. Integer data type comparisons in the test bench still use an absolute tolerance.

To learn more about test bench data comparison, see How Test Bench Verification Works.

R2013a

Version: 1.5

New Features

Bug Fixes

Code generation for OMRON Sysmac Studio IDE

The Simulink PLC Coder software now supports OMRON Sysmac Studio Version 1.04 or later.

Code generation for multirate models in single-tasking mode

The Simulink PLC Coder software can now generate code for multirate models in single-tasking mode. For more information, see Generated Code Structure for Multirate Models and Multirate Model Restrictions.

To open an example that shows how to generate code from a multirate model, at the command line, enter:

```
plcdemo_multirate
```

R2012b

Version: 1.4

New Features

Bug Fixes

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, see Internal Signals for Debugging in RSLogix 5000 IDE.

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

Bug Fixes

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 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

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

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

Bug Fixes

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® website 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

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

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