Simulink® PLC Coder™

Getting Started Guide

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.
1 Apple Hill Drive
Natick, MA 01760-2098

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.

Revision History

September 2019	Online only	New for Version 3.1 (Release 2019b)
March 2020	Online only	Revised for Version 3.2 (Release 2020a)
September 2020	Online only	Revised for Version 3.3 (Release 2020b)
March 2021	Online only	Revised for Version 3.4 (Release 2021a)

Contents

Getting Started

Simulink PLC Coder Product Description

Generate IEC 61131-3 Structured Text and Ladder Diagrams for PLCs and PACs

Simulink® PLC Coder™ generates hardware-independent IEC 61131-3 Structured Text and Ladder Diagrams from Simulink models, Stateflow® charts, and MATLAB® functions. The Structured Text and Ladder Diagrams are generated in PLCopen XML and other file formats supported by widely used integrated development environments (IDEs) including 3S-Smart Software Solutions CODESYS, Rockwell Automation® Studio 5000, Siemens® TIA Portal, and OMRON® Sysmac® Studio. 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 and Ladder Diagrams using PLC and PAC IDEs and simulation tools. It also provides code generation reports with static code metrics and bidirectional traceability between model and code. Support for industry standards is available through IEC Certification Kit (for IEC 61508 and IEC 61511).

PLC Code Generation in the Development Process

Simulink PLC Coder software lets you generate IEC 61131-3 compliant Structured Text code from Simulink models. This software brings the Model-Based Design approach into the domain of PLC and PAC development. Using the coder, system architects and designers can spend more time fine-tuning algorithms and models through rapid prototyping and experimentation, and less time on coding PLCs.

Typically, you use a Simulink model to simulate a design for realization in a PLC. Once satisfied that the model meets design requirements, run the Simulink PLC Coder compatibility checker utility. This utility verifies compliance of model semantics and blocks for PLC target IDE code generation compatibility. Next, invoke the Simulink PLC Coder tool, using either the command line or the user interface. The coder generates Structured Text code that implements the design embodied in the model.

Usually, you also generate a corresponding test bench. You can use the test bench with PLC emulator tools to drive the generated Structured Text code and evaluate its behavior.

The test bench feature increases confidence in the generated code and saves time spent on test bench implementation. The design and test process are fully iterative. At any point, you can return to the original model, modify it, and regenerate code.

At completion of the design and test phase of the project, you can easily export the generated Structure Text code to your PLC development environment. You can then deploy the code.

Using Simulink PLC Coder, you can also generate Ladder Diagram code for your applications from a Stateflow chart. The benefits are:

- You can design your application by using states and transitions in a Stateflow chart. Once you complete the design, you can generate Ladder Diagram code in XML or another format. You then import the generated code to an IDE such as CODESYS 3.5 or RSLogix™ AOI 5000 and view the ladder diagram.
- When you test your Stateflow chart by using a set of inputs, you can reuse these inputs to create a test bench for the Ladder Diagram code. You import the test bench to your PLC IDE and compare the results of simulation with the results of running the ladder diagram. If the results match, the original Stateflow chart is equivalent to the generated Ladder Diagram code.

Expected Users

The Simulink PLC Coder product is a tool for control and algorithm design and test engineers in the following applications:

- PLC manufacturing
- Machine manufacturing
- Systems integration

You must be familiar with:

- MATLAB and Simulink software and concepts
- PLCs
- Structured Text language

If you want to download generated code to a PLC IDE, you must also be familiar with your chosen PLC IDE platform. For a list of these platforms, see "Supported IDE Platforms" on page 1-5.

Glossary

Term	Definition
PAC	Programmable automation controller.
PLC	Programmable logic controller.
IEC 61131-3	IEC standard that defines the Structured Text language for which the Simulink PLC Coder software generates code.
PLCopen	Vendor- and product-independent organization that works with the IEC 61131-3 standard. The Simulink PLC Coder product can generate Structured Text using the PLCopen XML standard format. See `https://www.plcopen.org` for details.
Structured Text	High-level textual language defined by IEC 61131-3 standard for the programming of PLCs.
function block	Structured Text language programming concept that allows the encapsulation and reuse of algorithmic functionality.

System Requirements

For a list of related products, see System Requirements at the MathWorks® website.

Issues with Anti-Virus Software

The Simulink PLC Coder software ships with IDE-specific executables that are used in the "Import Structured Text Code Automatically" workflows. Some anti-virus software identifies these files as malware. However, it has been determined that these cases are false positives and that the files are malware free. You can mark these files as malware free in your antivirus program.

Supported IDE Platforms

IDEs Supported for Structured Text Generation

The Simulink PLC Coder product is tested with these IDE platforms:

- 3S-Smart Software Solutions CODESYS Version 2.3 or 3.3 or 3.5 (SP4 or later)
- B&R Automation Studio 3.0 or 4.0
- Beckhoff TwinCAT 2.11 or 3
- PHOENIX CONTACT Software MULTIPROG® 5.0 or 5.50.

  PHOENIX CONTACT Software GmbH was previously called KW-Software GmbH. The Simulink PLC Coder software supports only the English version of MULTIPROG target IDE.
- OMRON Sysmac Studio Version 1.04, 1.05, 1.09 or 1.12
- Phoenix Contact® PC WORX™ 6.0

  The Simulink PLC Coder software supports only the English version of Phoenix Contact PC WORX target IDE.
- Rexroth IndraWorks version 13V12 IDE
- Rockwell Automation RSLogix 5000 Series Version 17, 18, 19 or 20 and Rockwell Studio 5000 Logix Designer Version 21 or 24

  Simulink PLC Coder can generate code for Add-On instructions (AOIs) and routine code. The software supports automatic import and verification of generated code only for the RSLogix IDEs and not the Studio 5000 IDE.
- Siemens SIMATIC® STEP® 7 Version 5.3, 5.4, or 5.5

  The Simulink PLC Coder software assumes that English systems use English S7 and that German systems use German S7.
- Siemens TIA Portal V14
- Generic
- PLCopen XML
- Selectron CAP1131 IDE

For a list of supported IDEs and platforms, see Supported IDEs at the MathWorks website.

IDEs Supported for Ladder Diagram Code Generation

The Simulink PLC Coder product is tested with these IDE platforms:

- 3S-Smart Software Solutions CODESYS Version 3.5 SP6
- Rockwell Automation RSLogix 5000 Series Version 20 and Rockwell Studio 5000 Logix Designer Version 24
- PLCopen XML

Generate and Examine Structured Text Code

| **In this section...** |
| --- |
| "Generate Structured Text from the Model Window" on page 1-6 |
| "Generate Structured Text with the MATLAB Interface" on page 1-7 |
| "View Generated Code" on page 1-8 |

Generate Structured Text from the Model Window

You must already have set up your environment and Simulink model to use the Simulink PLC Coder software to generate Structured Text code. If you have not yet done so, see "Prepare Model for Structured Text Generation".

1 If you do not have the `plcdemo_simple_subsystem` model open, open it now.

2 In the **Apps** tab, select **PLC Coder**. The **PLC Code** tab appears.

3 In the **PLC Code** tab click **Settings** to open the Configuration Parameters dialog box.

Alternatively, you can right-click the Subsystem block and select **PLC Code > Options**.

The Configuration Parameters dialog box is displayed.
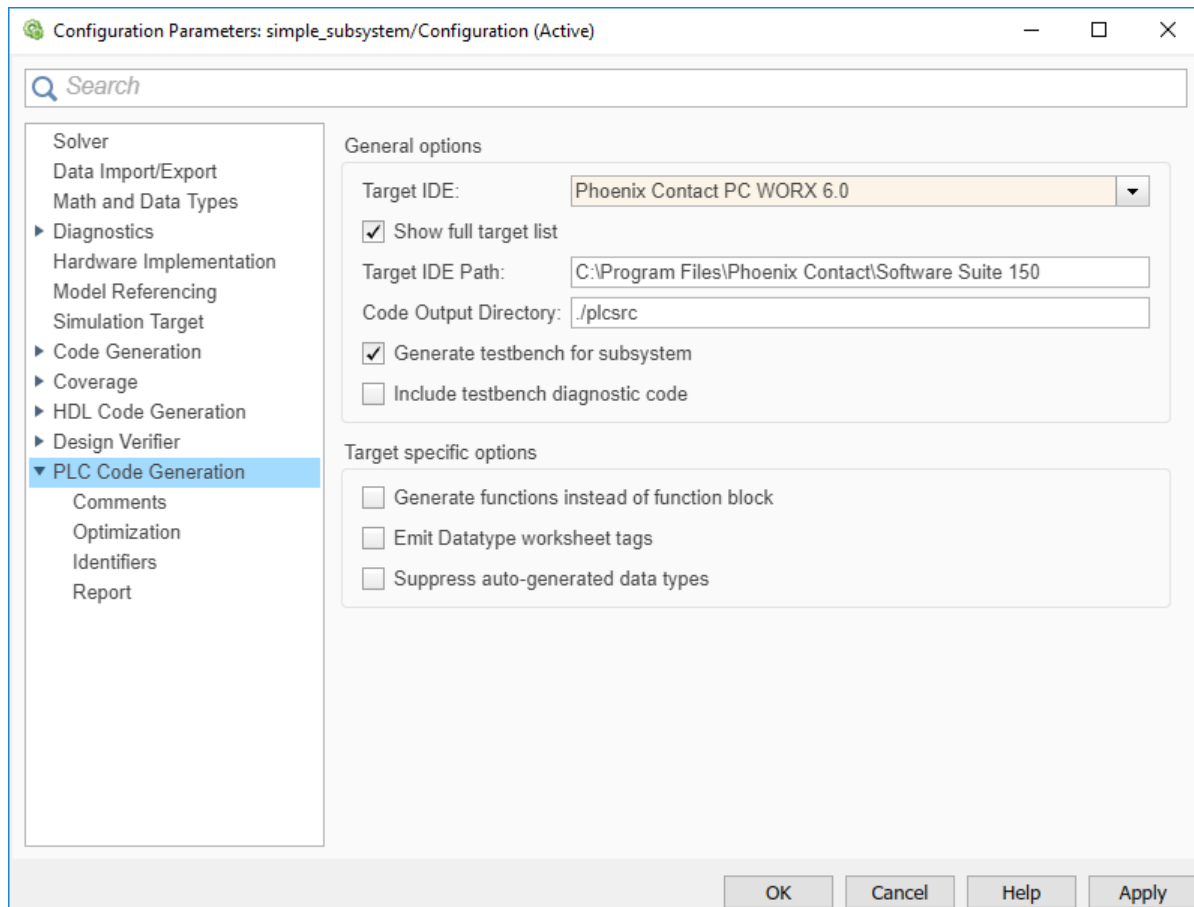
4   On the **PLC Code Generation** pane, select an option from the **Target IDE** list, for example, 3S CoDeSys 2.3.

The default **Target IDE** list displays the full set of supported IDEs. To see a reduced subset of the target IDEs supported by Simulink PLC Coder, disable the option **Show full target list**. To customize this list, use the plccoderpref function.

5   Click **Apply**, then **OK**.

6   To generate code for the Subsystem, select the Subsystem block. In the **PLC Code** tab, verify that the Subsystem name appears in the **Code for** option. To remember the selection, you can pin this option. Click **Generate PLC Code** [icon].

This button:

- Generates Structured Text code (same as the **PLC Code > Generate Code for Subsystem** option)
- Stores generated code in *model_name*.exp (for example, plcdemo_simple_subsystem.exp)

When code generation is complete, a **View diagnostics** hyperlink appears at the bottom of the model window. Click this hyperlink to open the Diagnostic Viewer window.
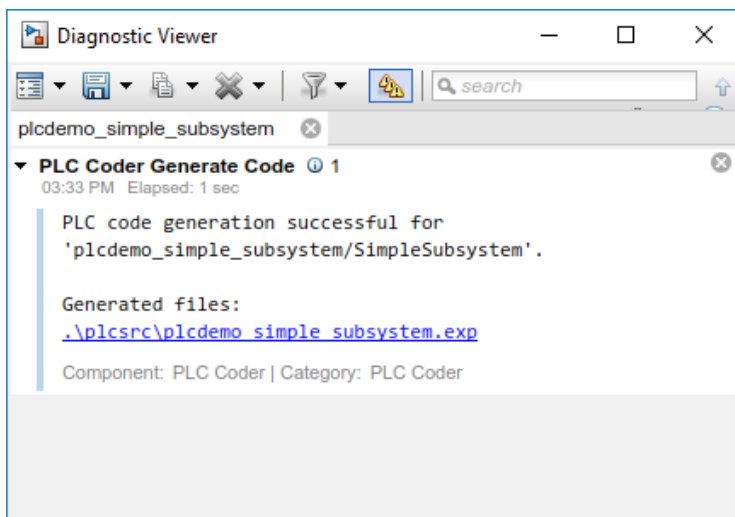
This window has links that you can click to open the associated files. For more information, see "Files Generated by Simulink PLC Coder".

Generate Structured Text with the MATLAB Interface

You can generate Structured Text code for a subsystem in the Command Window with the plcgeneratecode function. You must have already configured the parameters for the model or, alternatively, you can use the default settings.

For example, to generate code from the SimpleSubsystem subsystem in the plcdemo_simple_subsystem model:

1   Open the plcdemo_simple_subsystem model:

```
plcdemo_simple_subsystem
```

2   Open the Configuration Parameters dialog box using the `plcopenconfigset` function:

```
plcopenconfigset('plcdemo_simple_subsystem/SimpleSubsystem')
```

3   Select a target IDE.

4   Configure the subsystem as described in "Prepare Model for Structured Text Generation".

5   Generate code for the subsystem:

```
generatedfiles = plcgeneratecode('plcdemo_simple_subsystem/SimpleSubsystem')
```

When using `plcgeneratecode` for code generation, diagnostic messages are printed to the MATLAB command window.

View Generated Code

After generating the code, you can view it in the MATLAB Editor. For a description of how the generated code for the Simulink components map to Structured Text components, see "PLC Code Generation Basics". In addition, note the following:

- Matrix data types: The coder converts matrix data types to single-dimensional vectors (column-major) in the generated Structured Text.

- Generated code header: If your model has author names, creation dates, and model descriptions, the generated code contains these items in the header comments. The header also lists fundamental sample times for the model and the subsystem block for which you generate code.

- Code comments: You can choose to propagate block descriptions to comments in generated code. See "Propagate Block Descriptions to Code Comments".

The figure illustrates generated code for the CoDeSys Version 2.3 PLC IDE. Generated code for other platforms, such as Rockwell Automation RSLogix 5000, is in XML or other format and looks different.

```
15  FUNCTION_BLOCK SimpleSubsystem
16  VAR_INPUT
17      ssMethodType: SINT;
18      U: LREAL;
19  END_VAR
20  VAR_OUTPUT
21      Y: LREAL;
22  END_VAR
23  VAR
24      UnitDelay_DSTATE: LREAL;
25  END_VAR
26  VAR_TEMP
27      rtb_Gain: LREAL;
28  END_VAR
29  CASE ssMethodType OF
30      SS_INITIALIZE:
31          (* InitializeConditions for UnitDelay: '<S1>/Unit Delay'  *)
32          UnitDelay_DSTATE := 0;
33
34      SS_OUTPUT:
35          (* Gain: '<S1>/Gain' incorporates:
36           *  Inport: '<Root>/U'
37           *   Sum: '<S1>/Sum'
38           *   UnitDelay: '<S1>/Unit Delay'
39           *)
40          rtb_Gain := (U - UnitDelay_DSTATE) * 0.5;
41
42          (* Outport: '<Root>/Y'  *)
43          Y := rtb_Gain;
44
45          (* Update for UnitDelay: '<S1>/Unit Delay'  *)
46          UnitDelay_DSTATE := rtb_Gain;
47
48  END_CASE;
49  END_FUNCTION_BLOCK
```

If you are confident that the generated Structured Text is good, optionally change your workflow to automatically generate and import code to the target IDE. For more information, see "Import Structured Text Code Automatically".