WinAC Target for MATLAB/Simulink:

Integrating and calling Simulink models using STEP 7 and WinAC ODK using PID control as an example

WinAC RTX, STEP 7, WinAC ODK

Application • July 2013

Applications & Tools

Answers for industry.



Siemens Industry Online Support

This article is taken from the Siemens Industry Online Support. The following link takes you directly to the download page of this document:

http://support.automation.siemens.com/WW/view/en/56969417

Caution:

The functions and solutions described in this article confine themselves to the realization of the automation task predominantly. Please take into account furthermore that corresponding protective measures have to be taken up in the context of Industrial Security when connecting your equipment to other parts of the plant, the enterprise network or the Internet. Further information can be found under the Entry ID 50203404.

http://support.automation.siemens.com/WW/view/en/50203404

SIEMENS

Automation task

1

Automation solution	2
Basics	3
Function mechanisms of this application	4
Installation	5
MATLAB/Simulink	6
WinAC Target	7
Commissioning with STEP 7	8
External mode	9
Manipulating model parameters using STEP 7	10
Manipulating User- Defined Parameters with STEP 7	11
Error diagnostics	12
References	13
History	14

SIMATIC WinAC Target for MATLAB/Simulink

Integrating and calling Simulink models using STEP 7 and WinAC ODK using PID control as an example

Warranty and liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Preface

Objective of the Application

MATLAB/Simulink from MathWorks is software for modeling and simulating systems. Simulink provides a graphical editor that allows the user to create realistic processes and develop solutions, for example in control engineering. The Embedded Coder add-on is capable of compiling Simulink models directly into C/C++ code. WinAC ODK makes it possible to run C/C++ code in the Windows or real-time environment of WinAC RTX.

WinAC Target supports the user, allowing easy integration of Simulink models in STEP 7.

Main contents of this application note

This application discusses the following main points:

- Creating a simple block (PID controller) using Simulink
- Compiling into a DLL/RTDLL file and SCL source using WinAC Target
- Integrating into a STEP 7 V5.X and V1x project
- Monitoring a model online using Simulink and changing parameters during operation
- Changing model parameters with the STEP 7 program

Validity

This application does not include an in-depth description of:

- Control engineering basics
- MATLAB/Simulink and Embedded Coder
- WinAC RTX and WinAC ODK
- IntervalZero
- C/C++ programming
- STEP 7 programming

Basic knowledge of these topics is required.

Table of Contents

Warra	anty and	liability	4
Prefa	ce		5
Table	e of Conte	ents	6
1	Automat	tion task	8
2	Automat	tion solution	10
	2.1 2.1.1 2.1.2 2.1.3	Overview of the overall solution WinAC Target External mode Manipulating model parameters using STEP 7	10 11 14 15
	2.2	Hardware and software components used	17
3	Basics		19
4	Function	n mechanisms of this application	21
5	Installat	ion	23
	5.1	Hardware installation	23
	5.2	Software installation	24
6	MATLAE	B/Simulink	26
	6.1 6.1.1 6.1.2 6.1.3	Model simulation with MATLAB/Simulink The process The PID controller Simulation of the complete control loop	26 26 28 32
	6.2	Extracting the controller for code generation	35
7	WinAC 1	۲arget	37
	7.1	General	37
	7.2 7.2.1 7.2.2	Activating External mode General Parameterizing External mode	38 38 39
	7.3 7.3.1 7.3.2	Code generation with WinAC Target Settings Generating the code	40 40 41
	7.4	Data type conversion	43
8	Commis	sioning with STEP 7	44
	8.1	General	44
	8.2 8.2.1 8.2.2 8.2.3 8.2.4	STEP 7 V5.5 WinAC ODK Library Contents of the sample program Integrating the Simulink model into STEP 7 V5.5 Commissioning	45 45 45 47 51
q	8.3 8.3.1 8.3.2 8.3.3 8.3.4	STEP 7 V11/V12 (TIA Portal) WinAC ODK Library Contents of the sample program Integrating the Simulink model into STEP 7 V11/V12 Commissioning	52 52 52 53 56
J	LALUIIIdi		

	9.1	Monitoring	. 57
	9.2	Changing internal parameters with Simulink	.60
	9.3	RTDLL Communication Service for External mode	.63
10	Manipula	ating model parameters using STEP 7	.64
	10.1	Enabling parameter access for STEP 7	.64
	10.2	Manipulating internal model parameters using STEP 7 V5.5	.65
	10.3	Manipulating internal model parameters using STEP 7 V11/V12	.67
	10.4	Parameters in Simulink and STEP 7	.69
11	Manipula	ating User-Defined Parameters with STEP 7	.71
	11.1	Creating user-defined parameters	.71
	11.2	Interconnecting user-defined parameters	.74
	11.3	Configuring user-defined parameters	.75
	11.4	Enabling parameter access for STEP 7	.77
	11.5	Manipulating User-Defined Parameters with STEP 7 V5.5	.78
	11.6	Manipulating User-Defined Parameters with STEP 7 V11/V12	. 80
	11.7	User-defined parameters in Simulink and STEP 7	. 82
12	Error dia	ignostics	. 84
	12.1	Encoding with WinAC Target	. 84
	12.2	Connecting to External mode	. 86
	12.3	During operation	. 88
13	Reference	2es	. 89
14	History .		. 90

1 Automation task

Requirement

In automation and control engineering, MATLAB/Simulink is frequently used to simulate processes and create PID controllers or more complex algorithms. The requirement is to run algorithms, functions or systems created in a few steps on a programmable logic controller (PLC).

MathWorks provides the option to compile the Simulink models into C/C++ code using Embedded Coder. This code, in turn, can be integrated into a WinAC ODK project and called and run from the STEP 7 program via the DLL/RTDLL.

General task definition

- 1. Models created in Simulink are to be ported to WinAC RTX and run.
- 2. External mode allows you to monitor the model using Simulink and change parameters during operation.

Figure 1-1: Simulink models on WinAC RTX



Description of the automation task

The sample application shows you how to port models created with Simulink to WinAC RTX. The application is a simple temperature control application with a simulated plant (temperature process).





The PID controller is created using Simulink and ported to WinAC RTX. The temperature process is simulated by a function block.

Simulink allows you to monitor the interfaces of the model and change parameters.

Note The application example is deliberately kept simple to focus on the description of the use of WinAC Target and External mode.

Learning contents of this application

After studying this application, you will have learned the following:

- Examples of simulating processes and creating a controller in Simulink
- Principle of operation of WinAC Target
- Encoding of a Simulink model for porting to WinAC RTX
- Principle of operation of External mode with MATALB/Simulink

2 Automation solution

2.1 Overview of the overall solution

This application serves as an example of temperature control. The process is simulated using the "PROC_C" function block in the user program. The PID controller created in Simulink is called via the ODK interface in the form of a DLL/RTDLL. The process and the PID controller are called in the cyclic OB35. The DLL/RTDLL file is initialized in OB100.

In this application, the runtime system is a SIMATIC IPC427C.



Figure 2-1: Overview of the overall solution

The "Model_Param_DB" data block allows you to optionally access the internal parameters of the Simulink model. This enables you to manipulate these parameters during operation with the STEP 7 user program.

2.1.1 WinAC Target

Porting Simulink models to WinAC RTX

Figure 2-2 and Table 2-1 show the simple process for creating a model in Simulink in order to run it in WinAC RTX using WinAC Target. The gray area shows the engineering components.

Figure 2-2: Porting Simulink models to WinAC RTX



Table 2-1: Instructions

Step	Action	Explanation
1	Create a model in Simulink	In this case: A PID controller
2	Compile the model using WinAC Target	An SCL source and a DLL/RTDLL file will be generated.
3	Integrate the SCL source into the S7 program	The source is manually integrated into STEP 7 V5.5 and STEP 7 V11 using "External source files" in the Project tree.
4	Copy the DLL/RTDLL file to the PC-based controller	The DLL/RTDLL file must be manually copied to the PC system. The path is defined in the SCL source (default path: C:\).
5	Download the S7 program to WinAC RTX	-

When the SCL source is generated, the interfaces of the Simulink model are applied. If a function block is created from the SCL source in the S7 program, it has the same interfaces as the Simulink model. The block is provided with additional parameters (initialize, enable, crea_status, exec_status) for WinAC ODK communication. The "Model_param" and "Write_model_param" inputs will only be additionally created when the "Parameter access with STEP 7" setting is checked in WinAC Target.

Figure 2-3: Simulink model block and STEP 7 function block



WinAC ODK interface with synchronous call

The WinAC ODK interface is used to call the PID controller as a DLL/RTDLL file. Figure 2-4: Schematic diagram of the WinAC ODK functionality



The ODK CCX (Custom Code Extension) interface is used to directly call the DLL/RTDLL from the STEP 7 program. The call is synchronous using SFB65002 (EXEC_COM).





For more information on WinAC ODK, please refer to the following manual. <u>http://support.automation.siemens.com/WW/view/en/19256518</u>

2.1.2 External mode

Simulink External mode provides you with the option to monitor the ported model during operation on the runtime system (here: WinAC RTX on IPC427C) and change block parameters online.

In this case, the External mode interface must be enabled and configured in the code generation settings before porting the model to WinAC RTX.





(1) Monitoring

The **"Scope**" block allows you to graphically display signals in Simulink. In this case, these are the Setpoint, Process_value and PID_out signals.

(2) Changing parameters

Some Simulink blocks (here: "PID Controller") have internal parameters such as P, I, D, etc. You can double-click to open the block screen form and change the parameters during operation.

2.1.3 Manipulating model parameters using STEP 7

In some cases, it is necessary to access the internal parameters of the Simulink models during operation. For example, if parameters have to be adjusted to the running process. Optionally, an additional global data block, "Model_Param_DB", can be created with all parameters of the Simulink model when encoding the SCL source and the DLL/RTDLL file. This data block can be used to manipulate all parameters via the user program in STEP 7.

Figure 2-7: Overview of the overall solution



Note If you want to manipulate the model parameters using STEP 7, you have to enable the "Parameter access with STEP 7" parameter before encoding (see Chapter 10 Manipulating model parameters using STEP 7).

The figure below shows the internal parameters of "PID Controller" in Simulink and the relevant "Model_Param_DB" data block.

Figure 2-8: Model	parameters in	Simulink and	in the data	block in S	STEP 7

Function Block Parameters: PID Controller			×	
PID Controller			*	
This block implements continuous- and discrete-time PID anti-windup, external reset, and signal tracking. You can (requires Simulink Control Design).	control algorithms and includes advanced une the PID gains automatically using the	features such 'Tune' butto	as n	
Controller: PID	Form: Parallel			
Time domain:	Discrete-time settings			
	Integrator method:	ward Euler	-	
Continuous-time	Filter method: For	ward Euler		
Discrete-time	Sample time (-1 for inherited): 0.1			
Main PID Advanced Data Types State Attribute	5			
Controller parameters			E .	
Proportional (P): 1.9	<u>Compensator fo</u>	rmula		
Integral (I): 0.065				
Derivative (D): 8.5	$P+I.T$ _1 +	nN		
Filter coefficient (N): 5	z-1	$1 + N \cdot T_s -$	1	
		*		
	Sune			
C3-0899		Tarnet DI	D Project PID simple	e dii fash 10809
Initial conditions	"Model_Parameter_DB" WinAG	C_Target_PI	D_Project\PID_simple	e_dll_finah_\D899
Initial conditions CP DB99 Source: internal Address	"Model_Parameter_DB" WinAd	C_Target_PI	D_Project\PID_simpl	c_dll_finaf\D899
Initial conditions CD D899 Source: internal Addree	"Model Panmeter_DB" WinA4	Target_PI	D_Project\PID_simple Initial value	e_dll_finah_\D899
Initial conditions Source: internal Integrator: 0 Filter: 0	Model Parmeter DB" WinAG	Target_PI	D_Project\PID_simple Initial value 0.000000e+000	c_dl[finaf\D899
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: none	Model Parmeter_D8" WinAf	Type STRUCT REAL PEAL	D_Project/PID_simple Initial value 0.000000e+000 0.00000e+000	cdW_finaf\D899
Initial conditions Source: internal Integrator: 0 Filter: 0 External rest: none Integrators: 0 Filter: 0 F	* Model Parameter_DB* WinAd * Hane 0 ProportionalGain_Gain Integrator_gainval Integrator_com	Type STRUCT 22AL 22AL 22AL 22AL	D_Project/PID_simple Initial value 0.000000+000 0.000000+000 0.000000+000	cdlfmaf_\D899
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: none Type reset when linearizing Construction	Model Pattereb" WinAd Name ProportionalGain_Gain Integrator_IC DerivativeGain_Gain Filger anismol	Type STRUCT REAL REAL REAL REAL	D_Project\PID_simple Initial value 0.000000e+000 0.00000e+000 0.00000e+000	c_dllfinah_\D899
Initial conditions Source: internal Integrator: 0 Filter: 0 Filter: 0 Gamma Source Integrator: 0 Filter: 0	Model Parmeter_D8" WinAd Name ProportionalGain_Gain Integrator_IC DerivativeGain_Gain Filter_gainval Distry TC	Type SIRUCT REAL REAL REAL REAL REAL REAL	D_Project/PID_simple Initial value 0.000000+000 0.000000+000 0.000000+000 0.000000+000	c_dl[finalDB99 Comment Simulink Model Parameters: ProportionalGein_Gain Simulink Model Parameters: Integrator_gainval Simulink Model Parameters: Integrator_IC Simulink Model Parameters: DerivativeGein_Gain Simulink Model Parameters: Filter_gainval Simulink Model Parameters: Filter_gainval
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: none Ignore reset when linearizing Ignore reset when linearizing Ignobe zero-crossing detection	Model Parmeter_D8" WinAG Hame ProportionalGain_Gain ProportionalGain_Gain Integrator_gainval Filter_IG Filter_Gainval Filter Filter_Gainval Filter	Type STRUCT REAL REAL REAL REAL REAL REAL REAL REAL	D_Project/PID_simple Initial value 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000	cdlfinal_\D899
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: inone ignore reset when linearizing C Enable zero-crossing detection * * * * * * * * * * * * * * * * * * *	* Model Pahaneter_DB* WinAd * Hame 0 ProportionalGain_Gain Integrator_gainval 10 Integrator_IC 0 Filter_gainval 7 Filter_IC FilterCoefficient_Gain Saturation UpperSat	Target_PI Type STRUCT REAL REAL REAL REAL REAL REAL REAL REAL	D_Project/PID_simpl Initial value 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000	cdl[mafD899
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: none 1 Janore reset when linearizing C Enable zero-crossing detection 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Model Pahyneter_DB" WinAd Hame I Hame I TroportionalGain_Gain Integrator_IC DerivativeGain_Gain Filter_gainval Filter_JIC Filter_Gainval Filter_IC Saturation_UpperSat Saturation_LopperSat	Target PI Type STRUCT REAL REAL REAL REAL REAL REAL REAL REAL	D_Project/PID_simple Initial_value 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000	cdlfinaf\D899
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: none grant reset: none grant reset: none tenable zero-crossing detection tenable zero-crossing detection tenable zero-crossing detection tenable zero-trossing detection tenable z	 Model Palmeter_DB" WinAG Name ProportionalGain_Gain Integrator_IC DerivativeGain_Gain Filter_gainval Filter_TIC Filter_Coefficient_Gain Saturation_UpperSat Saturation_LowerSat Integratorian_Gain 	Target PI Type STRUCT REAL REAL REAL REAL REAL REAL REAL REAL	D_Project/PID_simple Initial value 0.000000+000 0.00000+000 0.00000+000 0.00000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000	cdlfinat_D0899 Comment Simulink Model Parameters: ProportionalGein_Gain Simulink Model Parameters: Integrator_gainval Simulink Model Parameters: DerivativeGein_Gein Simulink Model Parameters: Filter_geinval Simulink Model Parameters: Filter_IC Simulink Model Parameters: FilterCoefficient_Gein Simulink Model Parameters: Saturation_UpperSat Simulink Model Parameters: Saturation_LowerSat Simulink Model Parameters: Saturation_LowerSat Simulink Model Parameters: Saturation_LowerSat Simulink Model Parameters: Saturation_LowerSat Simulink Model Parameters: Naturation_LowerSat Simulink Mo
Initial conditions Source: internal Integrator: 0 Filter: 0 Filter: 0 Integrator: 0 Filter: 0 Fi	Model Parmeter_D8" WinAd Name Vane	Type STRUCT 22AL 22AL 22AL 22AL 22AL 22AL 22AL 22A	D_Project/PID_simpl Initial value 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000	cdlfinat_D0899 Comment Simulink Model Parameters: ProportionalGein_Gein Simulink Model Parameters: Integrator_geinval Simulink Model Parameters: DerivativeGein_Gein Simulink Model Parameters: Filter_geinval Simulink Model Parameters: Filter_IC Simulink Model Parameters: FilterCoefficient_Gein Simulink Model Parameters: Saturation_UpperSat Simulink Model Parameters: Saturation_LowerSat Simulink Model Parameters: IntegralGein_Gein
Initial conditions Source: internal Integrator: 0 Filter: 0 External reset: none ignore reset when linearizing C Enable zero-crossing detection +2 +2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	* Model Pahyneter_DB* WinAG * Hame 0 ProportionalGain_Gain 10 Integrator_gainval 10 Filter_Gainval 10 Filter_IC Filter_IC FilterCoefficient_Gain Saturation_LowerSat IntegralGain_Gain 10 10 10 10 10 10 10 10 10 10	Type STRUCT REAL REAL REAL REAL REAL REAL REAL REAL	D_Project/PID_simple Initial value 0.000000+000 0.00000+000 0.00000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000 0.000000+000	cdlfinal_\D899 Comment Simulink Model Parameters: ProportionalGain_Gain Simulink Model Parameters: Integrator_gainval Simulink Model Parameters: Integrator_IC Simulink Model Parameters: Fitter_IC Simulink Model Parameters: Fitter_Gainval Simulink Model Parameters: Fitter_Gainval Simulink Model Parameters: Staturation_DoperSat Simulink Model Parameters: Staturation_DoperSat Simulink Model Parameters: IntegralGain_Gain Simulink Model Parameters: IntegralGain_Gain

2.2 Hardware and software components used

2.2 Hardware and software components used

The application was created with the following components:

Hardware components

Table 2-2: SIMATIC components

Component	Qty.	Order no.	Note
SIMATIC IPC427C	1	6ES7675-1D1	Different bundles are available for the IPC427C, see <u>http://support.automatio</u> <u>n.siemens.com/WW/vie</u> <u>w/en/37954208</u> . The bundle must include WinAC RTX 2010 UPD 3.

Software components

Table 2-3: SIMATIC software components

Component	Qty.	Order no.	Note
STEP 7 V5.5	1	6ES7810-4C.10	-
S7-SCL V5.3	1	6ES7811-1CC05	-
STEP 7 Prof. V11 SP2 Update 4	1	6ES7822-1A.01	V11 projects can also be processed with STEP 7 V12
WinAC ODK V4.2 SP1	1	6ES7806-1CC03-0BA0	-

Table 2-4: Third-party software

Component	Qty.	Order no.	Note
MATLAB V8.0 (R2012b)	1	-	www.mathworks.com
Simulink V8.0	1	-	
Embedded Coder V6.3	1	-	Embedded Coder requires
MATLAB Coder V2.3	1	-	that MATLAB Coder and
Simulink Coder V8.2	1	-	installed.
Microsoft Visual Studio 2008/2010 Professional	1	-	www.microsoft.com
IntervalZero SDK V9.1.2	1	-	www.intervalzero.com (optional, for generating RTDLL files)

2.2 Hardware and software components used

Sample files and projects

The following list contains all files and projects that are used in this example.

Tal	ble	2-5
1 G		20

Component	Note	
Setup: WinAC Target	Setup for installing WinAC Target for Simulink	
Compressed (zipped) folder: WinAC_Target_Examples_v11	 This zip archive contains the following files: 01_Simulink_Model "PID_Process_Model.mdl" Simulink model "PID_Model.mdl" Simulink model 02_WinAC_Target_Example_Code Files generated with WinAC Target, including External mode for STEP 7 V5.x with DLL STEP 7 V5.x with RTDLL STEP 7 V1x with DLL STEP 7 V1x with DLL 03_STEP7_V5x_Project Sample project for STEP 7 V5.5 04_STEP7_V11_Project Sample project for STEP 7 V11 	

3 Basics

For all basic topics on the software components used in this application, please use the following links.

MathWorks

MATLAB

Program to solve mathematical problems and graphically display the results. In MATLAB, programming is done in a proprietary programming language. It offers a great scope of mathematical functions.

Simulink

Add-on for MATLAB for graphical programming of models and simulations. Simulink offers a comprehensive library with functions to create, for example, control engineering applications.

• Embedded Coder

Add-on for Simulink for compiling models into C/C++ code. Embedded Coder requires that MATLAB Coder and Simulink Coder be installed.

www.mathworks.com

Microsoft Visual Studio 2008/2010

Visual Studio is a development environment with integrated high-level languages such as C, C++, C#, Basic.

www.microsoft.com

IntervalZero

IntervalZero offers a real-time environment for PC systems: IntervalZero RTX. To create RTDLL files using Microsoft Visual Studio, you need the appropriate IntervalZero RTX SDK (software development kit).

www.intervalzero.com

STEP 7

STEP 7 is the development environment for programming and commissioning SIMATIC programmable logic controllers.

http://www.automation.siemens.com/mcms/simatic-controllersoftware/en/Pages/Default.aspx

Copyright © Siemens AG 2013 All rights reserved

SIMATIC WinAC RTX and WinAC ODK

• WinAC RTX

SIMATIC WinAC RTX is the SIMATIC software controller for PC-based automation solutions and allows real-time capable, deterministic control on the PC.

WinAC ODK

Via three different interfaces, the WinAC option Open Development Kit (ODK) allows flexible use of all PC resources from the control program, providing a high-performance extension of the PLC functionality. All Windows operating system functions and system resources are available to the programmer and therefore also access to external hardware and software components.

http://www.automation.siemens.com/mcms/programmable-logiccontroller/en/software-plc/Pages/Default.aspx

4 Function mechanisms of this application

The figure below shows the complete function mechanism of the application: From the creation of the PID controller with Simulink to the execution on the target system (here: WinAC RTX on IPC427C).

Figure 4-1: Complete overview of the application



- 1. An IPC427C where WinAC RTX is installed is used as the runtime system.
- 2. The STEP 7 program:
 - **OB100** (startup OB for restart (warm restart)): Calls the ported Simulink model once to initialize the DLL/RTDLL file.
 - **OB35** (time interrupt OB with 100ms cycle)): Calls the simulated process (PROC_C) and the ported Simulink model (PID controller).
 - Optional **Model_Param_DB** (global data block to manipulate internal model parameters)
- 3. **FB PID_Model** (ported Simulink model) reads inputs, communicates with the DLL/RTDLL file and writes outputs.
- 4. WinAC Target encodes the Simulink model and generates the **SCL source** and the **DLL/RTDLL file**.

5 Installation

5.1 Hardware installation

In general, all PC systems where WinAC RTX is installed can be used for this application. A SIMATIC IPC427C was used in this example.

Figure 5-1: Hardware configuration



The programming device (PG/PC) must be connected to the runtime system (IPC427C) using an Ethernet cable.

The following Ethernet addresses were used for this application:

- PG/PC interface: Ethernet (192.168.0.100)
- IPC427C interface: PROFINET CP1616 (192.168.0.1)
 Interface to WinAC RTX
- IPC427C interface: Industrial Ethernet (192.168.0.2)
 Interface to Windows

Subnet mask 255.255.255.0 was parameterized in both devices.

For more information on the IPC427C PC system, please refer to this manual: http://support.automation.siemens.com/WW/view/en/37028954

Note In order to use External mode with Simulink, TCP/IP communication must be established between the PG/PC and the runtime system. For this purpose, the Windows interface must be parameterized in WinAC Target (here: 192.168.0.2).

5.2 Software installation

5.2 Software installation

Installing the basic software

Install the following software on the programming device. Table 5-1: Basic software installation instructions

No.	Action	Remark
1	Install MATLAB V8.0 (R2012b) Simulink V8.0 MATLAB Coder V2.3 Simulink Coder V8.3 Embedded Coder V6.3	-
2	Install Microsoft Visual Studio 2008 or 2010 Professional	-
3	Optionally, install IntervalZero SDK V9.1.2	Only required for RTDLL generation
4	Install STEP V5.5 and S7-SCL V5.3 Alternatively, install STEP 7 V11 Professional with Service Pack 2 Update 4 or	Manual: http://support.automation.s iemens.com/WW/view/en/ 10805384/133300
	STEP 7 V12 Professional	
5	Install WinAC ODK V4.2 SP1	Manual: http://support.automation.s iemens.com/WW/view/en/ 12840073/133300

Install the following software on the runtime system (here: IPC427C).

No.	Action	Remark
1	Install WinAC RTX 2010 Update 3 on a PC system	Manual: http://support.automation.s iemens.com/WW/view/en/ 10805641/133300
		Update: http://support.automation.sie mens.com/WW/view/en/1522 7402

Installing WinAC Target

Install the following software on the programming device.

Table 5-3: WinAC Target installation instruction

No.	Action	Remark
1	Install	Prerequisite:
	winAC larget (run setup.exe)	Windows XD Deefeesien al OD2
		or
		Windows 7 (32-bit)
		STEP 7 V5.5 and S7-SCL V5.3 or
		STEP 7 V11 Professional with Service Pack 2
		or STEP 7 V12 Professional
		WinAC ODK V4.2 SP1

Integrating WinAC Target into Simulink

In order to use WinAC Target in Simulink, the software has to be integrated first. Perform the following steps once.

Table 5-4: WinAC Target into Simulink – installation instructions

No.	Action	Remark
1	Open " MATLAB ".	-
2	In the "Current Folder", navigate to the WinAC Target installation folder . It contains the " startup.m " file. Right-click and select Run to run this file. WinAC Target will be integrated into Simulink.	-

Note

This has to be done only once when using WinAC Target for the first time.

Installing the sample projects

Table 5-5: Sample projects installation instruction

No.	Action	Remark
1	Unzip the "56969417_WinAC_Target_Examples_v11.zip" file	-

6 MATLAB/Simulink

6.1 Model simulation with MATLAB/Simulink

6.1.1 The process

This application simulates a process that behaves similarly to a temperature control application. The response of the process to a step from 0°C to 50°C is very slow and aperiodic approaches the value 50 (Figure 6-1). The Y-axis represents the temperature in degrees Celsius [°C] and the X-axis represents the time in seconds [s].

Figure 6-1: Step response of the process



As a mathematical model, there is a continuous PT3 plant with the following formula:

$$G(s) = \frac{1}{10s+1} \cdot \frac{1}{10s+1} \cdot \frac{1}{5s+1}$$

or

Table 6-1: Plant parameters of the continuous process

Plant parameters	Remark
GAIN = 1	Gain
TM_LAG_1 = 10 seconds	Time lags
TM_LAG_2 = 10 seconds	
TM_LAG_3 = 5 seconds	

In Simulink, the process is simulated in the form of three PT1 functions connected in series.

Figure 6-2: Simulink model: Step response of the continuous process



This model is a continuous process. Later the process and the PID controller will be called in the PLC (here: WinAC RTX) in a cyclic organization block with the T = 100ms cycle. This means it is a discrete process simulation and a discrete PID controller. For this reason, the continuous plant must be converted to a discrete plant.

The TUSTIN transformation is used in this case:

$$s = \frac{1}{T} \cdot \ln(z) = \frac{1}{T} \left[(z-1) - \frac{(z-1)^2}{2} + \dots \right] \Rightarrow s \approx \frac{1}{T} \cdot (z-1)$$

T = 100ms = 0.1

The continuous process ...

$$G(s) = \frac{1}{10s+1} \cdot \frac{1}{10s+1} \cdot \frac{1}{5s+1}$$

after the TUSTIN transformation ...

$$G(z) = \frac{1}{10 \cdot \frac{1}{0.1} \cdot (z-1) + 1} \cdot \frac{1}{10 \cdot \frac{1}{0.1} \cdot (z-1) + 1} \cdot \frac{1}{5 \cdot \frac{1}{0.1} \cdot (z-1) + 1}$$

becomes the discrete process:

$$G(z) = \frac{1}{100z - 99} \cdot \frac{1}{100z - 99} \cdot \frac{1}{50z - 49}$$

The model in Simulink then looks as follows: Figure 6-3: Simulink model: Step response of the discrete process



6.1.2 The PID controller

The following section shows you how to create a model in Simulink. In this case, this is a simple PID controller with the process described in Chapter 6.1 Model simulation with MATLAB/Simulink.

Note Alternatively, you can also open the "PID_Process_Model.mdl" Simulink model.

Creating a model with PID controller in Simulink

Table 6-2: Creating a Simulink model

Step	Instruction
1.	Open " MATLAB ".
2.	Create a new model using Simulink and save it under the name "PID_Process_Model". MATLAB R2012b FROT Flas Find Flas Script Strg+N SIMATIC > Dot Function Function Function Figure Graphical User Interface Command Shortcut Simulink Model Simulink Model Simulink Project



6 MATLAB/Simulink

Step	Instruction		
6.	Use drag and drop to insert the following blocks from the Simulink library into the subsystem:		
	More input: "Simulink – Sources – In1"		
	 Adding block: "Simulink – Math Operations – Sum" 		
	 PID controller: "Simulink – Discrete – Discrete PID Controller" 		
7.	Double-click to open the block parameters of "Sum" and change the "List of signs" parameter to " +- ".		
8.	Then close the parameterization screen form.		
9.	Double-click to open the block parameters of " Discrete PID controller " and apply the following values.		
	Function Block Parameters: Discrete PID Controller		
	This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune' button (requires Simulink Control Design).		
	Controller: PID Form: Parallel		
	Time domain: Discrete-time settings		
	Continuous-time		
	Discreta-time Sample time (-1 for inheritarf): 0.1		
	Main PID Advanced Data Types State Attributes E		
	Proportional (P): 1.9		
	Integral (I): 0.065		
	Derivative (D): 8.5 $P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{z-1}$		
	Filter coefficient (N): 5		
	Initial conditions		
	Source: internal		
	Integrator: 0		
	Filter: 0		
	External reset: Inone		
	Ignore reset when linearizing		
	OK Cancel Help Apply		
	Note:		
	• Sample time is set to 0.1 for 100ms. The controller and the process will later be called in WinAC RTX in OB35 (cycle: 100ms).		
	• The PID values were determined empirically for this application. A good		
	response to setpoint changes with little overshoot is achieved with this		
	parameterization.		
	I he sample time in the Simulink model must match the sample time later desired in the PLC application		

Step Instruction					
10.	In the "PID Advanced" tab, set a limit	t from 0 to 100 for t	he PID output.		
	Function Block Parameters: Discrete PID Controller				
	This block implements continuous- and discrete-time PID co anti-windup, external reset, and signal tracking. You can tur (requires Simulink Control Design).	introl algorithms and includes a ne the PID gains automatically u	dvanced features such as sing the 'Tune' button		
	Controller: PID	Form: Parallel	•]		
	Time domain:	Discrete-time settings			
		Integrator method:	Forward Euler 🔹		
	Continuous-time	Filter method:	Forward Euler •		
	Discrete-time	Sample time (-1 for inherite	ed): 0.1		
	Main PID Advanced Data Types State Attributes	1			
	Output saturation				
	Z Limit output				
	Upper saturation limit:	Anti-windup	o method:		
	100.0	none			
	Lower saturation limit:				
	Tonore saturation when linearizing				
	Tedies and				
	Final Franking mode				
	Tracking coefficient (kt):				
	1				
11.	Close the parameterization screen for	rm.			
12	Wire the blocks and assign the name	s as shown below			
12.					
	PID_Process_Model/PID_Controller_Subsystem				
			• » 🖉 • 🗰 •		
	PID_Controller_Subsystem				
	PID_Process_Model Pa PID_Controller_Subsystem	m	•		
	Q				
	Setpoint A		PID_out		
		TID CONTOICI			
	Process value				
	× × × × × × × × × × × × × × × × × × ×				
	Ready	50%	ode45		
			000043		
3.	Close the subsystem.				
3.	Close the subsystem.				

6.1.3 Simulation of the complete control loop

The simulation of the process and the created PID controller allows you to check the response of the complete control loop directly in Simulink. To do so, proceed as follows.

Table 6-3: Simulation of the complete control loop

Step	Instruction	
	Use drag and drop to insert three blocks of the "Discrete Transfer Fcn" type from the Simulink library into the Simulink model: • "Simulink – Discrete" – "Discrete Transfer Fcn"	
	Successively open the " Discrete Transfer Fcn " blocks and parameterize them as follows.	
	Parameterization for blocks France 2. Function Block Parameters: Discrete Transfer Fcn Discrete Transfer Fcn Implement a z-transform transfer function. Specify the numerator and denominator coefficients in descending powers of z. The order of the denominator must be greater than or equal to the order of the numerator. Main Data Types State Attributes Data Source Value Numerator: Dialog [1] Denominator: Dialog [1]	

Step	Instruction
4.	Parameterization for block 3:
	Function Block Parameters: Discrete Transfer Fon
	powers of z. The order of the denominator must be greater than or equal to the order of the numerator. Main Data Types State Attributes Data Source Value Numerator: Dialog [1] Denominator: Dialog [50 - 49]
	Initial states: Dialog
	Optimize by skipping divide by leading denominator coefficient (a0)
	QK Qancel Help Apply Close the parameterization screen forms.
5.	 In addition, insert the following blocks: "Simulink – Sources – Constant" for the setpoint "Simulink – Sinks – Scope" to visualize the control loop "Simulink – Signal Routing – Mux" to merge the signals
6.	Double-click to open the " Mux " block and change the value of the " number of inputs " parameter to " 3 ". Close the parameterization screen form.
7.	Double-click to open the " Constant " block and change " Constant Value " to 50 and " Sample time " to 0.1 for 100ms. Close the parameterization screen form.

6 MATLAB/Simulink

6.1 Model simulation with MATLAB/Simulink

Step	Instruction
8.	Wire the complete control loop and assign the names as shown below.
	PID_Process_Model File Edit View Display Diagram Simulation Analysis Code Tools Help PID_Process_Model PID_Process_Model PID_Process_Model Setpoint PID_out 1 1002:09 PID_Controller_Subsystem PIT+Element2 PT+Element3 Scope Ready 100% ode45
9.	 Double-click to open the "Scope" block. Select 150 for "Simulation stop time" and use the "Start simulation" icon in the Simulink model menu bar to start the simulation. Click the "Autoscale" icon in the menu bar of the "Scope" block. As a result, the simulation of the complete control loop is displayed.

Note Before porting the PID controller to the runtime system, you have to extract the model. This means that the plant simulation is removed from the model as the complete model is compiled when encoding. The next chapter describes the extraction of the controller.

6.2 Extracting the controller for code generation

6.2 Extracting the controller for code generation

To port the PID controller to WinAC RTX, the model has to be adjusted. This is necessary as only the controller is to run on the runtime system and not the complete control loop with controller and process.

Note Conversion with WinAC Target always compiles the complete Simulink model.

Step	Instruction
1.	Open the " PID_Process_Model " model.
2.	Create a new model and save it under the name "PID_Model".
3.	Double-click the " PID_Controller_Subsystem " block to open the subsystem from the " PID_Process_Model ".
4.	Select the entire content of the subsystem and use drag and drop to move it to the newly created " PID_Model ".

To do so, follow the steps in the table below: Table 6-4: Extracting the controller

Note Alternatively, you can also open the "PID_Model.mdl" Simulink model.

6 MATLAB/Simulink

6.2 Extracting the controller for code generation

Step	Instruction
5.	Open the "Simulink Library Browser".
	PID_Model File Edit View Display Diagram Simulation Analysis Code Tools Help PID_Model PID_Model PID_Model PID_Model PID_Model PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller PID_controller
	Ready 100% ode45
6.	Use drag and drop to insert the following blocks from the Simulink library into the new "PID_Model": "Simulink – Signal Attributes – Signal Conversion" "Simulink – Sinks – Scope"
	• "Simulink – Signal Routing – Mux"
7.	Double-click to open the " Mux " block and change the value of the " number of inputs " parameter to " 3 ". Close the parameterization screen form.
8.	Wire the model and assign the names as shown below.
	PID_Model PID_Model PID_Model PID_form PID form PID form <

The model now only includes the PID controller with the links to the variables: "Setpoint", "Process_value" and "PID_out". To graphically display the signals, the model also includes the "Signal Conversion" and "Scope" blocks.

Now the model can be encoded for WinAC RTX and run (see 7.3 Code generation with WinAC Target).
7 WinAC Target

7.1 General

WinAC Target is a Siemens add-on for Simulink that generates the DLL/RTDLL file and the SCL source from Simulink models.

To integrate the Simulink model into a STEP 7 project, WinAC Target automatically generates all necessary blocks and files. An SCL source and a DLL/RTDLL file are created from the generated C/C++ code.

The SCL source must be manually integrated into the STEP 7 project. This is done using the "External Sources" function in the Project tree of the STEP 7 project. Finally, you have to download the STEP 7 program and the DLL or RTDLL to a PC system where WinAC RTX is installed.

- **Note** If you want to use External mode, you have to activate it before encoding (see Chapter 7.2 Activating External mode).
- **Note** If you want to manipulate the model parameters using STEP 7, you have to enable the "Parameter access with STEP 7" parameter before encoding (see Chapter 10 Manipulating model parameters using STEP 7).



Figure 7-1: Code generation – schematic diagram

7.2 Activating External mode

Note IntervalZero SDK (software development kit) is required for generating an RTDLL.

http://www.intervalzero.com/

7.2 Activating External mode

7.2.1 General

Simulink External mode provides you with the option to monitor the ported model during operation on the runtime system (here: WinAC RTX on IPC427C) and change internal model parameters online.

To enable Simulink to communicate with the ported model on the runtime system via External mode, External mode must be activated before encoding. In this case, the code is extended by the communication code to allow data exchange between Simulink and the model in the runtime system. Communication is implemented based on TCP/IP.





- **Note** When External mode is activated for encoding, the size of the generated DLL/RTDLL file increases and it needs more time to run. The reason for this is the additional code for communication.
- **Note** If communication with an RTDLL file is to take place, an additional service must be started on the runtime system (see Chapter 9.3 RTDLL Communication Service for External mode).

7.2.2 Parameterizing External mode

If you want to use External mode, you have to make the following settings. Table 7-1: Parameterizing External mode

Step	Instruction		
1.	Open the following Simulink model: "PID_Model".		
2.	Set "Simulation stop time" to "inf" and "Simulation Mode" to "External".		
	PID_Model File Edit View Display Diagram Simulation Analysis Code Tools Help PID_Model PID_MODEL PID_NODEL		
	Ready 100% Fixe		
3.	In the menu, click "Code – C/C++ Code – Code Generation Options".		
4.	A window with a navigation bar appears. "Code Generation" is checked by default.		
5.	Set the following parameters:		
	 "Code Generation" navigation item System target file: WinAC_Target.tlc "Interface" navigation item Data exchange Interface: External Mode Transport layer: tcpip MEX-file arguments: '192.168.0.2' (IP address of the runtime system, Windows) 		

7.3 Code generation with WinAC Target

7.3 Code generation with WinAC Target

7.3.1 Settings

In this application, the following settings are made for WinAC Target. Table 7-2: Settings for code generation

Step	Instruction	
1.	Open the following Simulink model: " PID_Model ".	
2.	In the menu, click "Code – C/C++ Code – Code Generation Options".	
3.	A window with a navigation bar appears. The "Code Generation" item is checked by default. Set the following parameters: "Code Generation" navigation item - System target file: WinAC_Target.tlc	
4.	 "Solver" navigation item Type: Fixed-step Solver: discrete (no continuous states) Fixed-step size: auto "Interface" navigation item Data exchange Interface: External Mode Transport layer: tcpip MEX-file arguments: '192.168.0.2' 	
	 "Data Import/Export" navigation item Save to workspace: Uncheck the Time, States, Output, Final states variables "WinAC Options" navigation item In "WinAC ODK project type", you can select whether you want to generate a Windows DLL (DLL) or a real-time DLL (RTDLL). A Windows DLL is selected by default. In "STEP 7 Version", you can select the automation software you want to use later. The STEP 7 V5.5 setting is selected by default. In "Data cycle time (Extended mode) [ms]", you can customize the cycle time for External mode. The default setting is "50". In "STEP 7 Function Block name", you can select the name of the function block. The default setting is the name of the Simulink model. In "Parameter access with STEP 7", you can enable parameter access to the model parameters with the STEP 7 user program. 	

7.3 Code generation with WinAC Target

 Note
 If the development environment and WinAC RTX are installed on one computer, MEX-file arguments must be modified. Change the IP address to 'localhost'.

 If the IP address of your target system differs from the one in this application example, change MEX-file arguments.

 Note
 When selecting WinAC Target, External mode will be activated by default. If you do not want to use External mode, proceed as follows:

 In the menu, click "Code – C/C++ Code – Code Generation Options..."

 "Interface" navigation item

 In the "Data exchange" menu option, set "Interface" to "None"

7.3.2 Generating the code

Note Alternatively, you can also use the code from the sample files:

02_WinAC_Target_Example_Code

Files generated using WinAC Target with External mode for

- STEP 7 V5.x with DLL
- STEP 7 V5.x with RTDLL
- STEP 7 V1x with DLL
- STEP 7 V1x with RTDLL

WinAC Target Output

To compile a Simulink model into C++ code and generate a DLL/RTDLL and an SCL file, proceed as follows:

Table 7-3: Settings for code generation

Step		Instruction
1.	In the Simulink model menu, c	lick "Code – C/C++ Code – Build Model".
2.	In this case, the "PID Controlle (PID_Model.mdl) is compiled in DLL/RTDLL file and the SCL s	er" block from the Simulink model nto C/C++. Then WinAC Target generates the source from this source code.
3.	The "\PID_Model_WinAC_T contains the generated code. • "Binary" folder: • "SCL_Source" folder:	Farget_Output" Simulink model directory DLL/RTDLL file SCL source

7.3 Code generation with WinAC Target

Figure 7-3: Files generated with WinAC Target

MATLAB >	PID_Model_WinAC_Target_Ou	tput 🕨	▼ +j
Organisieren 🔻 — 🎇 Öffner	n Freigeben für 💌 »	⊯ • ⊡	0
Name	Тур	Größe	-
🍌 Binary	Dateiordner		
📕 SCL_Source	Dateiordner		
魪 buildInfo.mat	MATLAB Data	16 KB	
😰 C_ODK.obj	Object File	1 KB	
😢 C_ODKfunc.obj	Object File	3 KB	
1 codeInfo.mat	MATLAB Data	5 KB	
📄 defines.txt	Textdokument	1 KB	
🕶 ert main.cpp	C++ Source	2 KB	

Addition to the SCL source

The interfaces were applied when porting the Simulink code. In STEP 7, the block has the same interfaces as the Simulink model.

The additional interfaces (initialize, enable, crea_status, exec_status) for WinAC ODK communication will be described in the next chapter.

The "Model_param" and "Write_model_param" inputs will only be additionally created when the "Parameter access with STEP 7" setting is checked in WinAC Target.

Figure 7-4: Simulink model block and STEP 7 function block



7.4 Data type conversion

As STEP 7, WinAC ODK and Embedded Coder use different data types, type conversion must take place on the interfaces of the programs. WinAC Target recognizes the data types of the Embedded Coder C/C++ program and adjusts the data in the WinAC ODK program and the STEP 7 program (SCL source). Type conversion takes place when running WinAC Target. The following table shows the relevant data types.

C/C++ (Embedded Coder)	C/C++ (WinAC ODK)	Bytes	STEP 7
boolean_T	bool	1 bit	BOOL
int8_T, unint8_T, char_T, uchar_T, byte_T	char	1	CHAR
int16_T, uint16_T	short	2	INT
int32_T, uint32_T, int_T, uint_T	long	4	DINT
ulong_T	long	4	DINT
real_T	float	4	→ REAL (4 Byte) Notice: Loss of data
real32_T	float	4	REAL
"type"_T ["length"]	"type" ["length"]	Depending on array size and data type	Array [lo hi] of type

Table 7-4: Data types overview

If the data types used in the WinAC ODK program are larger than the ones used in STEP 7, loss of data may occur. The figure below shows the interfaces and data types of this application. In this case, data conversion with loss of data takes place from the [double] to the [float] type in the WinAC ODK program. However, this does not affect the application example as the accuracy is sufficiently high despite the loss of data.

Figure 7-5: Example of data type conversion



8.1 General

8 Commissioning with STEP 7

This chapter describes the programming and commissioning with STEP 7 V5.5 and STEP 7 V11 (TIA Portal).

8.1 General

Figure 8-1: Simulink model in the STEP 7 program

MATLAB		STEP 7 program
		OB100 startup
Simulink		FB (SCL source) Simulink model initialization Instance DB OB35 cyclic
System target file: WinAC_Target.tlc	source	FB (SCL source)
Language: C++ Description: WinAC Target		execution
		Nonteen recenteer

When porting Simulink models in STEP 7, the following steps are recommended: Table 8-1: Tips for porting in STEP 7

Instruction	Note
Initialization The ported Simulink model (FB from SCL source) is to be initialized in OB100 (startup).	The initialization operation of the DLL/RTDLL file may take several seconds (depending on the operating system and the complexity of the DLL/RTDLL). If initialization in the time interrupt OB is performed, for example, with the 100 ms cycle, the runtime system will go to STOP due to a timeout.
During operation During operation, it is standard practice to call the Simulink model in a cyclic time interrupt OB. This ensures an equidistant call.	When creating the Simulink model, you should know from the beginning in which cycle the model is called on the runtime system. Accordingly, the model in Simulink and the time interrupt OB are parameterized with the same cycle time.

8.2.1 WinAC ODK Library

Note To integrate models, the WinAC ODK software has to be installed. In addition, the **WinAC ODK Library** must be integrated in the SIMATIC Manager as the SFB65001 (CREA_COM) and SFB65002 (EXEC_COM) blocks are required to run the DLL/RTDLL.

For more information on WinAC ODK, please refer to <u>http://www.automation.siemens.com/mcms/programmable-logic-</u>controller/en/software-plc/simatic-winac-odk/Pages/Default.aspx

WinAC ODK Library	Objektname	Symbolischer Name
🗇 🛐 Organization Blocks	SFB65001	CREA_COM
Blocks	SFB65002	EXEC_COM
E-st System Function Blocks	🕞 SF865003	ASYN_COM

8.2.2 Contents of the sample program

The "...\03_STEP7_V5x_Project" folder contains the sample project for STEP7 V5.5 with the following contents:

• WinAC RTX

This is only the hardware configuration of a PC station with WinAC RTX. The device is empty. This configuration serves as a template.

• PID Simple

This example provides a prepared program for integration by WinAC Target. The following blocks have already been created:

- **OB100 (Complete Restart)** is called when starting the PLC. The PID controller is initialized in this OB.
- **OB35 (CYC_INT5)** as a cyclic OB with a 100 ms cycle. The simulated plant and the PID controller (PID_Controller) are called in this block.
- **DB35 (DATA)** as a global data block. It includes all necessary variables:
 - Setpoint [Real],
 - PID_output [Real],
 - Process_value [Real],
 - crea_status [Word] (provides the status via CREA_COM),
 - exec_status[Word] (provides the status via EXEC_COM),
 - initialize [Bool]
 - enable [Bool]
 - Model_param [Any]
 - Write_model_param [Bool]

- FB100 (PROC_C) with instance DB100 Simulated PT3 process (parameterized like the process in Simulink)
- SFB65001 (CREA_COM) to initialize the DLL/RTDLL file
- SFB65002 (EXEXC_COM) to run the DLL/RTDLL file
- VAT_1 Variable table with the variables of DB35 (DATA) and DB1 (Model_Param_DB)
- PID_Simple_dll_final

•

This program includes ready-to-use programming with integration of the PID controller from Simulink by WinAC Target. This program uses a DLL call.

• PID_Simple_rtdll_final

This program includes ready-to-use programming with integration of the PID controller from Simulink by WinAC Target. This program uses an RTDLL call.

Note It is recommended to initialize the ported Simulink model in an OB100 as the time required for this operation can be significantly longer than the normal call.

8.2.3 Integrating the Simulink model into STEP 7 V5.5

The following steps complete the "**PID_simple**" program with the "**PID_Model**" block from the Simulink model:

Step	Instruction
1.	Open the SIMATIC Manager.
2.	Open the "WinAC_Target_PID_Project" sample project in the "\03_STEP7_V5x_Project" directory.
3.	Expand the tree under "WinAC_Target_PID_Project".
4.	Expand the tree under "PID_simple".
5.	Click the Source folder in the " PID_simple " S7 program.
6.	Then use " Insert – External Source " to add the SCL source generated by WinAC Target.
7.	Open the SCL source and click at the menu "Options – Customize"
	"OK"
	OK Cancel Help

Step	Instruction
9.	Generate the function block with "File – Compile".
10.	Open OB 100 and insert FB " PID_Model " into " Network 1: Initializing the DLL ". Interconnect the inputs and outputs as shown in the figure. Network 1: Initializing the DLL Comment:
	DB2 "PID_ Nodel_DB" FB1 "PID_Nodel"
	EN ENO
	DB35 DEX16 exec
	.0 status "DATA". initializ INIT - e
	enable
	param
	Write_ model_ param



Note If you want to use a different path to store the DLL/RTDLL file, the path also has to be modified in the SCL source:

Variable: "DLL_name"

Alternatively, you can also change the variable directly in the generated function block

"Interface - Static - DLL_name"

Note If an RTDLL is to be called in WinAC RTX, it must be registered in IntervalZero Runtime after copying to the runtime system.

In the PC system, select "**Start - Programs - IntervalZero - RTX 2009 - Tools – RtssRun**". Register the RTDLL file as shown in the figure.

RTSSrun: ["C:\PID_Model.rtdll"	
Suppress warnings	
lodes	
Run RTSS process	Register RTDLL
Use nonpaged pool	Share between processes
F Register to run at boot time	Perform load\unload test
Ideal processor 7	Force Registration
Affinity mask	

8.2.4 Commissioning

Once you have downloaded the program to WinAC RTX, perform the following steps for commissioning.

Table 8-3: Commissioning with STEP 7 V5.5

Step	Instruction
1.	Set WinLC to RUN mode.
2.	Open the " VAT_1 " variable table (see Figure 8-2).
3.	Click the "Monitor Variable" icon.
4.	Change the value of the "ENABLE" variable to "TRUE" and click the "Modify Variable" icon.
5.	Change the value of the " Setpoint " variable to the desired value, e.g. 50, and click the " Modify Variable " icon.
	Due to the setpoint change, the PID controller will output an output value. The process will change accordingly until it matches the " Setpoint ".

Figure 8-2: Variable table

-	Tabelle	Bear	beite	en <u>E</u> infügen <u>Z</u> ielsystem <u>V</u> aria	ble <u>Ansicht</u>	Extras F	3 er <u>H</u> ilfe	- 8
▰▯▰◼◓▯▯▫▫▾◥◾ਃਲ਼: ◷~~~~~~~~								
Τ	Copera	nd		Symbol	Anzeigeform	Statuswert	Steuerwert	
1	DB35.0	BD	0	"DATA".Setpoint 5	GLEITPUNKT		50.0	
2	DB35.0	BD	4	"DATA".PID_output	GLEITPUNKT			
3	DB35.0	BD	8	"DATA".Process_value	GLEITPUNKT			
4	DB35.0	BW	12	"DATA".crea_status	HEX			
5	DB35.0	BW	14	"DATA".exec_status	HEX			
6	DB35.0	BX	16.0	"DATA".INIT	BOOL			
7	DB35.0	BX 1	16.1	"DATA".ENABLE	BOOL		true	
8	DB35.0	BX	16.2	"DATA".WRITE_Parameters	BOOL			
9	DB99.0	BD	0	"Model_Parameter_DB".Proportional	GLEITPUNKT			
10	DB99.0	BD	4	"Model_Parameter_DB".Integrator_g	GLEITPUNKT			
11	DB99.0	BD	8	"Model_Parameter_DB".Integrator_I	GLEITPUNKT			

Note

The following variables allow you to diagnose errors.

The "crea_status" variable corresponds to SFB65001.Status The "exec_status" variable corresponds to SFB65002.Status

For the error codes, please refer to the WinAC ODK manual. http://support.automation.siemens.com/WW/view/en/35948966

8.3 STEP 7 V11/V12 (TIA Portal)

8.3.1 WinAC ODK Library

For STEP 7 V11 or higher, all necessary ODK blocks have already been integrated in the engineering.

8.3.2 Contents of the sample program

The ""...**\04_STEP7_V11_Project**" folder contains the sample project for STEP7 V11 with the following contents:

• PID_Simple

This example provides a prepared program for integration by WinAC Target. The following blocks have already been created:

- **OB100 (Complete Restart)** is called when starting the PLC. The PID controller is initialized in this OB.
- **OB35 (CYC_INT5)** as a cyclic OB with a 100 ms cycle. After integration, the simulated plant and the PID controller (PID_Controller) are called in this block.
- **DB35 (Data)** as a global data block. It includes all necessary variables:
 - Setpoint [Real],
 - PID_output [Real],
 - Process_value [Real],
 - crea_status [Word] (provides the status via CREA_COM),
 - exec_status[Word] (provides the status via EXEC_COM),
 - initialize [Bool]
 - enable [Bool]
 - Model_param [Any]
 - Write_model_param [Bool]
- FB100 (PROC_C) with instance DB100 Simulated PT3 process (parameterized like the process in Simulink)
- SFB65001 (CREA_COM) to initialize the DLL/RTDLL file
- SFB65002 (EXEXC_COM) to run the DLL/RTDLL file
- Watch table_1
 Variable table with the variables of DB35 (DATA) and DB1 (Model_Param_DB)

• PID_Simple_dll_final

This program includes ready-to-use programming with integration of the PID controller from Simulink by WinAC Target. This program uses a DLL call.

• PID_Simple_rtdll_final

This program includes ready-to-use programming with integration of the PID controller from Simulink by WinAC Target. This program uses an RTDLL call.

8.3.3 Integrating the Simulink model into STEP 7 V11/V12

The following steps complete the "**PID_simple**" program with the "**PID_Model**" block from the Simulink model:

Step	Instruction
1.	Open TIA Portal V11.
2.	Open the "WinAC_Target_PID_Project" sample project in the "\04_STEP7_V11_Project" directory.
3.	Go to the " Project View " (bottom left).
4.	Expand the tree under "PID_Simple [IPC427C PN]".
5.	Expand the tree under "WinAC RTX [CPU]".
6.	Double-click to select "Add new external file" under the "External source files" folder.
7.	Select "SCL Sources (*.scl)".
8.	If you have already generated the SCL source and the DLL/RTDLL for STEP 7 V11 using WinAC Target, navigate to the directory where the SCL source is located.
	Alternatively, open the SCL source under
	"\02_ WinAC_Target_Code\PID_Model_V11_dII" for the DLL call or
	"\02_WinAC_Target_Code\PID_Model_V11_rtdII" for the RTDLL call
9.	The SCL source is now located in the STEP 7 V11 project. Right-click the " PID_Model.scl " SCL source.
10.	In the context menu, select "Generate Blocks". A function block is generated from the SCL source and stored in the "Program Blocks" folder.
11.	Open " COMPLETE RESTART [OB100] " and use drag and drop to insert the generated FB from the Project tree into network 1.
12.	Select " OK " to confirm the data block generation.

Table 8-4: Integrating the Simulink model into STEP 7 V11 / V12



Step	Instruction
18.	Now copy the DLL (PID_Model.dll) or RTDLL (PID_Model.rtdll) to the C: \ directory of the PC system with WinAC RTX. RTDLL files have to be registered (see notes below).
19.	Set WinLC RTX to RUN mode.

Note If you want to use a different path to store the DLL/RTDLL file, you have to modify the path also in the SCL source and regenerate the function block.

Variable: "DLL_name"

Alternatively, you can also change the variable directly in the generated function block

"Interface - Static - DLL_name"

Note If an RTDLL is to be called in WinAC RTX, it must be registered in IntervalZero Runtime after copying to the runtime system.

In the PC system, select "**Start - Programs - IntervalZero - RTX 2009 - Tools – RtssRun**". Register the RTDLL file as shown in the figure.

TSSrun: "C:\PID_Model.rtdll"	
Suppress warnings	
odes	
C Run RTSS process	Register RTDLL
Use nonpaged pool	Share between processes
☐ Register to run at boot time	Perform load/unload test
Ideal processor 7	Force Registration
Affinity mask	

8.3.4 Commissioning

Once you have downloaded the program to WinAC RTX and copied the DLL/RTDLL file, perform the following steps for commissioning.

Table 8-5: Commissioning with	STEP	7 V	11/V12
-------------------------------	------	-----	--------

Step	Instruction
1.	Set WinLC to RUN mode.
2.	In "Watch and force table", open the "Watch table_1" variable table.
3.	Click the "Monitor all" icon
4.	Change the value of the "enable" variable to "True" and click the "Modify all selected values once and now" icon.
5.	Change the value of the " Setpoint " variable to the desired value, e.g. 50.0, and click the " Modify all selected values once and now " icon. Due to the setpoint change, the PID controller will output an output value. The
	process will change accordingly until it matches the set " Setpoint ".

Figure 8-3: Variable table

dll_fin	al [IPC427C PN] > WinAC RTX_dll [CPU] >	Beobachtung	s- und Forceta	abellen 🕨 Wa	tch table_1	_ # #×
10	4 3 9, %, % ⁹ , ⁹ , ⁹ , ⁹					
i	Name	Adresse	Anzeigeforma	Beobachtung	Steuerwer 🔗	Kom
1	*Data*.Setpoint	%DB35.DBD0	Gleitpunktzahl		50.0 5	
2	"Data".PID_output	%D835.D8D4	Gleitpunktzahl			
3	"Data".Process_value	%DB35.DBD8	Gleitpunktzahl			
4	"Data".crea_status	%DB35.DBW12	Hex			
5	"Data".exec_status	%DB35.DBW14	Hex			
6	*Data*.INIT	%DB35.DBX16.0	Bool			
7	*Data*.ENABLE	%DB35.DBX16.1	Bool 💌		TRUE 4	
8	"Data".WRITE_Parameters	%DB35.DBX16.2	Bool			
9	"Model_Parameter_DB".ProportionalGain_Gain	%DB1.DBD0	Gleitpunktzahl			
10	*Model_Parameter_DB*.Integrator_gainval	%DB1.DBD4	Gleitpunktzahl			
11	"Model_Parameter_DB".Integrator_IC	%DB1.DBD8	Gleitpunktzahl			
12	"Model_Parameter_DB".DerivativeGain_Gain	%DB1.DBD12	Gleitpunktzahl			
13	Model Parameter DB Filter gainval	%DB1.DBD16	- Cleitpunktzahl	ALL ALLAND	A	

Note

The following variables allow you to diagnose errors.

The "crea_status" variable corresponds to SFB65001.Status The "exec_status" variable corresponds to SFB65002.Status

For the error codes, please refer to the WinAC ODK manual. http://support.automation.siemens.com/WW/view/en/35948966

9 External mode

- **Note** If you want to monitor and change parameters with External mode, you must have parameterized WinAC Target accordingly before encoding (see 7.2 Activating External mode).
- **Note** If you want to use External mode in conjunction with an RTDL file, additionally start the RTDLL Communication Service (see 9.3 RTDLL Communication Service for External mode).
- **Note** To use External mode, port 17725 must be enabled on the runtime system (here: IPC427C) when a firewall is enabled.

9.1 Monitoring

Table 9-1: Monitoring with External mode



9 External mode

9.1 Monitoring



9.1 Monitoring



9.2 Changing internal parameters with Simulink

9.2 Changing internal parameters with Simulink

Step Instruction 1. Open the "PID_Model.mdl" model created with WinAC Target and transferred to WinAC RTX Set "Simulation stop time" to "inf" and "Simulation Mode" to "External". 2. PID_Model File Edit View Analysis Code Display Diagram Simulation Tools Help **_** 0 External • inf PID Model ۲ Ð PID(z) 1 $\overline{1}$ Setpoint PID_out K 7 PID Controlle Signal Conversio ⇒ (2 Process_value Ai ≫ Ready 100% Fixed 3. Use the "Connect To Target" button to connect Simulink to the controller model on WinAC RTX. PID_Model File Edit View Display Diagram Simulation Analysis Code Help Tools 00 0 External 0 ü inf PID_Model ۲ Q PID(z) •1 1 Setpoint PID_out K X PID Controller Signal C onversio = (2 Process value A >> Ready 100% Fixed

Table 9-2: Changing parameters with External mode

9.2 Changing internal parameters with Simulink

Step	Instruction
4.	To change controller block parameters, double-click to open "PID Controller".
	Function Block Parameters: PID Controller
	PID Controller
	This block implements continuous- and discrete-time PID control algorithms and includes advap ³ anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using (requires Simulink Control Design).
	Controller: PID - Form: Parallel
	Time domain:
	Continuous-time Integrator method:
	Filter method:
	Discrete-time Sample time (-1 for inherited):
	Main PID Advanced Data Types State Attributes
	Controller parameters
	Proportional (P): 1.9
	Integral (I): 0.065
	Derivative (D): 8.5 $P+I.T$
	Filter coefficient (N): 5
	In this dialog box, you can adjust the controllor parameters
	"Disconnect From Target" button.
	Setpoint PID Controller
	⇒ 2 Signal Process_value Conversion
	Image: Strain and Strain an

9 External mode

9.2 Changing internal parameters with Simulink

Note Parameter changes with External mode are valid only until the next restart of WinAC RTX.

If you want to permanently save parameter changes, re-encode the DLL/RTDLL and save it on the runtime system. In this case, it is not necessary to reintegrate the SCL source into the S7 program as the model interfaces have not changed.

Note A general WinLC RTX reset closes the port for communication with the Simulink model. This operation may take longer. As soon as the general reset has been completed and the port has been closed, the STOP LED lights up.

9.3 RTDLL Communication Service for External mode

If you are using External mode in conjunction with an RTDLL file, you have to start the RTDLL Communication Service on the runtime system before going online with External mode. This program enables a service that implements the communication between the IntervalZero real-time environment and Windows. This allows data exchange between Simulink and the RTDLL file. To enable the RTDLL Communication Service, perform the following steps.

Step	Instruction
1.	Click the engineering PC where WinAC Target is installed:
	Windows 7 "Start->All Programs -> SIMATIC -> WinAC Target -> RTDLL Communication Service"
	Windows XP: "Start -> SIMATIC -> WinAC Target -> RTDLL Communication Service"
2.	Copy the entire contents of the open window to the runtime system.
3.	Run the " RTDLL_Com_Service.exe " file.
4.	Select the "Start service" button to start the service.

Table 9-3: External mode	with an	RTDLL	file
--------------------------	---------	-------	------

Figure 9-1: WinLC RTX panel / RTDLL Communication Service for External mode



10.1 Enabling parameter access for STEP 7

10 Manipulating model parameters using STEP 7

10.1 Enabling parameter access for STEP 7

To allow access to the internal parameters of the Simulink models in the STEP 7 user program, parameter access for STEP 7 must be enabled in WinAC Target before generating the DLL/RTDLL file.

Figure 10-1: WinAC Options

😵 Contiguistion Parameters: PII	ID_Model/Configuration (Active)	
Soloct: Sniver Data Import/Export Diagnostics Hardwara Implementat Model Referencing Sninulation Targa.	WINAC ODK project type: Windows DLI -i1 -P / Verano: V5.X Data cycle time (External mode) [ms]: 50 STEP 7 Function Block name: FID_Model V Parameter access with SL-P /	
-Report -Curriments -Symhols -Curriments -Curriments -Code Code Debug -Thierfane -Verification -Code Style -Templates -Code Placement -Code Placement	Jiep	
•'	π.:	-
0	OK Carce Help	Apply

Note The maximum length for a data block comment in STEP 7 V5.x is 79 characters. When composing the comment from the Simulink model information, this limit may be exceeded. Compiling the SCL source generates all blocks and reduces the relevant location in the comment to the maximum number. The complete comment is retained in the SCL source. A respective warning is output when compiling the SCL source.

Note In data blocks for STEP 7 V5.x, the maximum length of variable names is limited to 24 characters. The original parameter names from the Simulink model are used for the variable names in STEP 7. If these names are too long for STEP 7 V5.x, they will be automatically truncated. For security, the original parameter name is in the respective comment.

10.2 Manipulating internal model parameters using STEP 7 V5.5

10.2 Manipulating internal model parameters using STEP 7 V5.5

Once you have downloaded the program to WinAC RTX, perform the following steps for manipulation.

Table 10-1: Commissioning with STEP 7 V5.5

Step	Instruction
1.	Set WinLC to RUN mode.
2.	Open the " VAT_1 " variable table (see Figure 8-2).
3.	Click the "Monitor Variable" icon.
4.	Change the value of the "ENABLE" variable to "TRUE" and click the "Modify Variable" icon. The Simulink model parameters are read and saved in the "Model_Param_DB" data block.
5.	Change the value of the "WRITE_Parameters" variable to "TRUE" and click the "Modify Variable" icon.
6.	Change the values of " Model_Param_DB " to the desired values and click the " Modify Variable " icon. The PID controller is run with these changed parameter values.
7.	Change the value of the " Setpoint " variable to the desired value, e.g. 50 , and click the " Modify Variable " icon. Due to the setpoint change, the PID controller will output an output value. The process will change accordingly until it matches the " Setpoint ".

10.2 Manipulating internal model parameters using STEP 7 V5.5

Fiaure	10-2:	Variable	table
90.0		v anabio	(abio

				9 6 w	60° ×7 //	
	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert	
1	DB35.DBD 0	"DATA".Setpoint	GLEITPUNKT		50.0	
2	DB35.DBD 4	"DATA".PID_output	GLEITPUNKT			
3	DB35.DBD 8	"DATA".Process_value	GLEITPUNKT			
	DB35.DBW 12	"DATA".crea_status	HEX			
5	DB35.DBW 14	"DATA".exec_status	HEX			
3	DB35.DBX 16.0	"DATA".INIT	BOOL			
1	DB35.DBX 16.1	"DATA".ENABLE	BOOL		true	
3	DB35.DBX 16.2	"DATA".WRITE_Parameters	BOOL		true	
	DB99.DBD 0	"Model_Param_DB".ProportionalGain_Gain	GLEITPUNKT			
0	DB99.DBD 4	"Model_Param_DB".Integrator_gainval	GLEITPUNKT			
11	DB99.DBD 8	"Model_Param_DB".Integrator_IC	GLEITPUNKT			
12	DB99.DBD 12	"Model_Param_DB".DerivativeGain_Gain	GLEITPUNKT	\frown		
13	DB99.DBD 16	"Model_Param_DB".Filter_gainval	GLEITPUNKT	(6)		
4	DB99.DBD 20	"Model_Param_DB".Filter_IC	GLEITPUNKT			
15	DB99.DBD 24	"Model_Param_DB".FilterCoefficient_Gain	GLEITPUNKT			
6	DB99.DBD 28	"Model_Param_DB".Saturation_UpperSat	GLEITPUNKT			
7	DB99.DBD 32	"Model_Param_DB".Saturation_LowerSat	GLEITPUNKT			
8	DB99.DBD 36	"Model Param DB".IntegralGain Gain	GLEITPUNKT			

Note Parameter changes with STEP 7 are valid only until the next restart of WinAC RTX.

If you want to permanently save parameter changes, re-encode the DLL/RTDLL and save it on the runtime system. In this case, it is not necessary to reintegrate the SCL source into the S7 program as the model interfaces have not changed.

NOTICE When manipulating Simulink model parameters using STEP 7, these changes will not be updated when using External mode in Simulink. Furthermore, changes to parameters cannot be made via External mode as long as the "WRITE_Parameters" variable is set to "TRUE".

10.3 Manipulating internal model parameters using STEP 7 V11/V12

10.3 Manipulating internal model parameters using STEP 7 V11/V12

Once you have downloaded the program to WinAC RTX and copied the DLL/RTDLL file, perform the following steps for commissioning.

Step	Instruction
1.	Set WinLC to RUN mode.
2.	In "Watch and force table", open the "Watch table_1" variable table.
3.	Click the "Monitor all" icon.
4.	Change the value of the "ENABLE" variable to "TRUE" and click the "Modify all selected values once and now" icon. The Simulink model parameters are read and saved in the "Model_Param_DB" data block.
5.	Change the value of the "WRITE_Parameters" variable to "TRUE" and click the "Modify all selected values once and now" icon.
6.	Change the values of " Model_Param_DB " to the desired values and click the " Modify all selected values once and now " icon. The PID controller is run with these changed parameter values.
7.	Change the value of the " Setpoint " variable to the desired value, e.g. 50.0, and click the " Modify all selected values once and now " icon. Due to the setpoint change, the PID controller will output an output value. The process will change accordingly until it matches the set " Setpoint ".

Table 10-2: Commissioning with STEP 7 V11/V12

Figure 10-3: Variable table

1 1 2 2 m m								
i	Name	Adresse	Anzeigeformat	Beobachtun	Steuerwert	9		
1	*Data*.Setpoint	%D835.D8D0	Gleitpunktzahl		50.0			
2	"Data".PID_output	%DB35.DBD4	Gleitpunktzahl		-			
3	"Data".Process_value	%DB35.DBD8	Gleitpunktzahl					
4	*Data*.crea_status	%DB35.DBW12	Hex					
5	"Data".exec_status	%D835.D8W14	Hex					
6	"Data".INIT	%DB35.DBX16.0	Bool					
Ż	*Data*.ENABLE	%DB35.DBX16.1	Bool		TRUE			
в	*Data*.WRITE_Parameters	%DB35.DBX16.2	Bool	14	TRUE			
9	"Model_Param_DB".ProportionalGain_Gain	%DB1.DBD0	Gleitpunktzahl					
10	"Model_Param_DB".Integrator_gainval	%DB1.DBD4	Gleitpunktzahl					
11	"Model_Param_DB".integrator_IC	%DB1.DBD8	Gleitpunktzahl					
12	"Model_Param_DB".DerivativeGain_Gain	%D81.D8D12	Gleitpunktzahl			-		
13	*Model_Param_DB*.Filter_gainval	%DB1.DBD16	Gleitpunktzahl	6				
14	"Model_Param_DB".Filter_IC	%DB1.DBD20	Gleitpunktzahl					
15	*Model_Param_DB*.FilterCoefficient_Gain	%DB1.DBD24	Gleitpunktzahl					
6	*Model_Param_DB*.Saturation_UpperSat	%D81.D8D28	Gleitpunktzahl					
7	*Model_Param_DB*.Saturation_LowerSat	%DB1.DBD32	Gleitpunktzahl					
18	*Model_Param_DB*.IntegralGain_Gain	%D81.D8D36	Gleitpunktzahl					
19		<hinzufügen></hinzufügen>						

10.3 Manipulating internal model parameters using STEP 7 V11/V12

Note Parameter changes with STEP 7 are valid only until the next restart of WinAC RTX.

If you want to permanently save parameter changes, re-encode the DLL/RTDLL and save it on the runtime system. In this case, it is not necessary to reintegrate the SCL source into the S7 program as the model interfaces have not changed.

NOTICE When manipulating Simulink model parameters using STEP 7, these changes will not be updated when using External mode in Simulink. Furthermore, changes to parameters cannot be made via External mode as long as the "WRITE_Parameters" variable is set to "TRUE".

10.4 Parameters in Simulink and STEP 7

10.4 Parameters in Simulink and STEP 7

To allow exact assignment between the parameters in the Simulink model und the parameters in STEP 7, the "**Model_Param_DB**" data block was provided with comments.

The comments are structured as follows:

Table 10-3: Structure of "Model_Param_DB" comments

Step	Instruction
1.	Name of the subsystem where the parameter is located
2.	Name of the block where the parameter is located
3.	Complete internal name of the Simulink model parameter

Figure 10-4: "Model_Param_DB" data block in STEP 7

Name	Тур	Anfangs	Kommenta
	STRUCT		
ProportionalGain_Gain	REAL	0.00000	PID Controller Proportional Gain]: ProportionalGain Gain
Integrator_gainval	REAL	0.00000	[PID Controller/Integrator]: Integrator_gainval
Integrator_IC	REAL	0.00000	[PID Controller/Integrator]: Integrator_IC
DerivativeGain_Gain	REAL	0.00000	[PID Controller/Derivative Gain]: DerivativeGain_Gain
Filter_gainval	REAL	0.00000	[PID Controller/Filter]: Filter_gainval
Filter_IC	REAL	0.00000	[PID Controller/Filter]: Filter_IC
FilterCoefficient_Gain	REAL	0.00000	[PID Controller/Filter Coefficient]: FilterCoefficient_Gain
Saturation_UpperSat	REAL	0.00000	[PID Controller/Saturation]: Saturation_UpperSat
Saturation_LowerSat	REAL	0.00000	[PID Controller/Saturation]: Saturation_LowerSat
IntegralGain_Gain	REAL	0.00000	[PID Controller/Integral Gain]: IntegralGain_Gain
	END STR		

10.4 Parameters in Simulink and STEP 7

Figure 10-5: Simulink model parameters



Note The maximum length for a data block comment in STEP 7 V5.x is 79 characters. When composing the comment from the Simulink model information, this limit may be exceeded. Compiling the SCL source generates all blocks and reduces the relevant location

in the comment to the maximum number. The complete comment is retained in the SCL source.

A respective warning is output when compiling the SCL source.

Note In data blocks for STEP 7 V5.x, the maximum length of variable names is limited to 24 characters. The original parameter names from the Simulink model are used for the variable names in STEP 7. If these names are too long for STEP 7 V5.x, they will be automatically truncated. For security, the original parameter name is in the respective comment.

11.1 Creating user-defined parameters

11 Manipulating User-Defined Parameters with STEP 7

In MATLAB, you can create variables and interconnect them as parameters in the Simulink model. WinAC Target and STEP 7 then enable you to manipulate these user-defined parameters.

11.1 Creating user-defined parameters

To allow access to the user-defined parameters of the Simulink models in the STEP 7 user program, the MATLAB variables must be created in the MATLAB workspace before generating the DLL/RTDLL file. Table 11-1: Creating user-defined parameters

Step			Instructi	on		
1.	Right-click of create a new	on a free space in the w MATLAB variable.	MATLAB	workspa	ace and clic	< on " New" to
	Workspace				¢	0
	Name 🔺	Value	Min	Max	Class	
		New	Strg+N			
		Save Clear Workspace	Strg + S			
		Refresh	F5			
		Choose Columns	-			
		Sort By	•			
		Paste	Strg +V			
		Select All	Strg +A			
		Print	Strg + P			
	Right-click d	Page Setup on the MATLAB varia	ble you hav	ve creat	ed and sele	ct " Rename '
	Right-click o Change the Workspace	Page Setup on the MATLAB varia name to " Proportio	ble you hav nal_Value'	ve creat	ed and sele	ct "Rename'
	Right-click of Change the Workspace Name	Page Setup on the MATLAB varia name to " Proportio Value	ble you hav nal_Value' Min	ve creat	ed and sele	ct "Rename'
	Right-click of Change the Workspace Name	Page Setup on the MATLAB varia name to " Proportio Value 0 Open Selection Strg+D Save As	ble you hav nal_Value' Min 0	ve creat '. Max 0	ed and sele Class double	ct "Rename'
	Right-click o Change the Workspace Name	Page Setup on the MATLAB varia name to " Proportio Value 0 Open Selection Strg+D Save As Copy Strg+C Duplicate	ble you hav nal_Value' Min 0	ve creat Max 0	ced and sele Class double	ct "Rename'
	Right-click of Change the Workspace Name	Page Setup on the MATLAB varia name to " Proportio Value 0 Open Selection Strg+D Save As Copy Strg+C Duplicate Delete Entf	ble you hav nal_Value' Min 0	Ve creat	ced and sele Class double	ct "Rename'
	Right-click o Change the Workspace Name	Page Setup on the MATLAB varia name to " Proportio Value 0 Open Selection Strg+D Save As Copy Strg+C Duplicate Delete Entf Rename	ble you hav nal_Value' Min 0	ve creat 7. Max 0	ced and sele	ct "Rename'
	Right-click of Change the Workspace Name	Page Setup on the MATLAB varia name to " Proportio Value 0 Open Selection Strg+D Save As Copy Strg+C Duplicate Delete Entf Rename Edit Value	ble you hav nal_Value' Min 0	Max 0	ced and sele Class double	ct "Rename'
<u>.</u>	Right-click o Change the Workspace Name	Page Setup on the MATLAB varia name to "Proportio Value 0 Open Selection Strg+D Save As Copy Strg+C Duplicate Delete Entf Rename Edit Value Plot Catalog	ble you hav nal_Value' Min 0	ve creat 2. Max 0	ced and sele	ct "Rename'

11 Manipulating User-Defined Parameters with STEP 7

11.1 Creating user-defined parameters

Instruction								
Right-click on the M Change the value to	IATLAB variable you h o " 1.9" .	ave creat	ed and s	elect "Edit	: Value".			
Workspace								
Name 🔺	Value	Min	Max	Class				
Proportions ¹⁰⁴⁶¹	o Open Selection Strg+D Save As	0	0	double				
	Copy Strg+C Duplicate Delete Entf							
	Rename							
	Edit Value	8						
	Plot Catalog							
Perform these step variables and assig	s for the " Integral_Va n the values as follows	l ue" and ' S.	"Derivat	ive_Value [:]				
Name 🔺	Value	Min	Max	Class				
Derivative_Value Integral_Value Proportional_Value	8.5000 0.0650 ie 1.9000	8.5000 0.0650 1.9000	8.5000 0.0650 1.9000	double double				
	Right-click on the M Change the value to Workspace Proportion Perform these step variables and assig Workspace Name Derivative_Value Integral_Value Proportional_Value	Right-click on the MATLAB variable you h Change the value to "1.9". Workspace Name A Value Proportion Open Selection Strg+D Save As Copy Strg+C Duplicate Delete Entf Rename Edit Value Plot Catalog Perform these steps for the "Integral_Val variables and assign the values as follows Workspace Name A Value Derivative_Value 8,5000 Integral_Value 0.0650 Proportional_Value 1.9000	Instruction Right-click on the MATLAB variable you have creat Change the value to "1.9". Workspace Name • Value Min Proportion Open Selection Strg+D Save As Copy Strg+C Duplicate Delete Entf Rename Edit Value Plot Catalog Workspace Workspace Value Min Operivative_Value 0.0650 0.0650 Proportional_Value 1.9000 1.9000	Instruction Right-click on the MATLAB variable you have created and s Change the value to "1.9". Workspace Name A Value Min Max Open Selection Strg+D 0 0 Save As Copy Strg+C Duplicate Delete Entf Rename Edit Value Plot Catalog 0 0 Perform these steps for the "Integral_Value" and "Derivativativativative_Value 8.5000 8.5000 8.5000 Name A Value Min Max 9 Derivative_Value 8.5000 8.5000 9.5000 Integral_Value 0.0650 0.0650 0.0650 0.0650 1.9000 <	Instruction Right-click on the MATLAB variable you have created and select "Edit Change the value to "1.9". Workspace Name Walue Min Max Class 0 0 double Open Selection Strg+D 0 0 double Save As Copy Strg+C Duplicate Duplicate Delete Entf Rename Edit Value Plot Catalog Perform these steps for the "Integral_Value" and "Derivative_Value" variables and assign the values as follows. Workspace Workspace Value Min Max Class Derivative_Value 8,5000 8,5000 double Proportional_Value 1,9000 1,9000 double			

Note The created MATLAB variables are not saved in the Simulink model. After restarting MATLAB, they are no longer available. The MATLAB variables need to be created again.

You can export MATLAB variables. For more information, please refer to the MATLAB Help.
11.1 Creating user-defined parameters

NOTE External mode is not available with structures as user-defined parameters in a Simulink model.

Simulink resp.External Mode does not support uploading of signal with structure type.

11.2 Interconnecting user-defined parameters

11.2 Interconnecting user-defined parameters

The figure below shows the interconnection of the created MATLAB variables with the block parameters of "PID Controller".

Figure 11-1: Interconnecting us	ser-defined parameters
---------------------------------	------------------------

Function Block Parame	eters: PID Controller					
-PID Controller)
This block implements anti-windup, external r (requires Simulink Con	continuous- and discret eset, and signal tracking trol Design).	e-time PID cont g. You can tune	rol algo the PID	rithms and gains aut	l includes advan omatically using	ced feature the 'Tune
Controller: PID		•	Form:	Parallel		2
-Time domain:			Discre	ete-time se	ettings	
🗇 Continuous-time			Integrator method: Forward E			Forward E
			Filter	Filter method:		Forward E
Oiscrete-time	Oiscrete-time			Sample time (-1 for inherited): 0.1		
Main PID Advanced	d Data Types Stat	te Attributes				Ź
-Controller parameters						
Proportional (P):	Proportional_Value				⊡ <u>Compensat</u>	<u>or formula</u>
Integral (I):	Integral_Value					Ę
Derivative (D):	Derivative_Value				$P + I \cdot T_s =$	$\frac{1}{1} + D \longrightarrow$
Filter coefficient (N):	5				2 ·	$^{-1}$ 1+N
			T	une		
M-Mar Marine	\sim	~~~ <u>~</u> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	*	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	$\sim\sim\sim\sim\sim\sim\sim\sim$	$\sim \sim \sim$

11.3 Configuring user-defined parameters

11.3 Configuring user-defined parameters

To allow access to the user-defined parameters of the Simulink models in the STEP 7 user program, the MATLAB variables must be configured as model parameters before generating the DLL/RTDLL file.

Table 11-2: Configuring	user-defined	parameters
-------------------------	--------------	------------

Step	Instruction
5.	In the Simulink model, open "Code Generation Options"
6.	Open the "Optimization " navigation item and select the "Signals and Parameters " item
7.	Check "Inline Parameters" and click on "Configure"

11 Manipulating User-Defined Parameters with STEP 7

11.3 Configuring user-defined parameters

Step	Instruction
8.	Select the listed variables from the "MATLAB workspace" and click on "Add to table >>"
	Model Parameter Configuration: PID_Model Description Define the global (limable) parameters for your model. These parameters affect. 1. the simulation by providing the ability to tune parameters during execution, and 2. the generated code by enabling access to parameters by other models. Source lis: Global (lumable) parameters MATLAB workspace Name Storage class Storage type qualifier Name 1 Dervative_Value 2 ntegral_Value
	3 Proport onal_Value
	Reacy OK Cancel Help Apply
9.	Close the dialog with "OK".

11.4 Enabling parameter access for STEP 7

11.4 Enabling parameter access for STEP 7

To allow access to the user-defined parameters of the Simulink models in the STEP 7 user program, parameter access for STEP 7 must be enabled with WinAC Target before generating the DLL/RTDLL file.

Figure 11-2: Enabling parameter access for STEP 7

Helenn:		_ ^
Co ver	WinAC ODK project type: Windows DLL	•
Data Import/Export	STEP 7 varsion: V5.x	-
∃- Optimization		
H- Jiaghostics	Data cycle time (External moce) [ms]: 50	
	STEP 7 Function Block name: PIC_Model	
Dimulation Target	Devenueter server with CTED 7	
∃-Code Generation		
-Report	Неф	
		-U
Debug		
Interface		
Verification		
Code Style		
- Terrpiates Coda Place root		
Data Type Replace		
Memory Sections		
^t WINAC Options		
	II	
9		\$
la data blacka	of CTED 7 VE w the merimum length of veriable nemes is limite	1
In data blocks	of STEP 7 V5.x, the maximum length of variable names is limite	b
In data blocks to 24 character	of STEP 7 V5.x, the maximum length of variable names is limite ars. The original parameter names from the Simulink model are	b
In data blocks to 24 character used for the va	of STEP 7 V5.x, the maximum length of variable names is limiteers. The original parameter names from the Simulink model are ariable names in STEP 7. If these names are too long for STEP 7	d ,
In data blocks to 24 character used for the va	of STEP 7 V5.x, the maximum length of variable names is limite ers. The original parameter names from the Simulink model are ariable names in STEP 7. If these names are too long for STEP 7 be automatically truncated. For security, the original parameter	d
In data blocks to 24 character used for the va V5.x, they will	of STEP 7 V5.x, the maximum length of variable names is limite ers. The original parameter names from the Simulink model are ariable names in STEP 7. If these names are too long for STEP 7 be automatically truncated. For security, the original parameter	d ,
In data blocks to 24 character used for the va V5.x, they will name is in the	of STEP 7 V5.x, the maximum length of variable names is limite ers. The original parameter names from the Simulink model are ariable names in STEP 7. If these names are too long for STEP 7 be automatically truncated. For security, the original parameter respective comment.	d ,
In data blocks to 24 character used for the va V5.x, they will name is in the	of STEP 7 V5.x, the maximum length of variable names is limite ers. The original parameter names from the Simulink model are ariable names in STEP 7. If these names are too long for STEP 7 be automatically truncated. For security, the original parameter respective comment.	d

Note The maximum length for a data block comment in STEP 7 V5.x is 79 characters. When composing the comment from the Simulink model information, this limit may be exceeded. Compiling the SCL source generates all blocks and reduces the relevant location in the comment to the maximum number. The complete comment is retained in the SCL source. A respective warning is output when compiling the SCL source.

Note

11.5 Manipulating User-Defined Parameters with STEP 7 V5.5

11.5 Manipulating User-Defined Parameters with STEP 7 V5.5

Once you have downloaded the program to WinAC RTX, perform the following steps for manipulation.

Step	Instruction
1.	Set WinLC to RUN mode.
2.	Open the " VAT_1 " variable table.
3.	Click on the " Monitor Variable " icon.
4.	Change the value of the "ENABLE" variable to "TRUE" and click on the "Modify Variable" icon. The Simulink model parameters are read and saved in the "Model_Param_DB" data block.
5.	Change the value of the " WRITE_Parameters " variable to " TRUE " and click on the " Modify Variable " icon.
6.	Change the values of " Model_Param_DB " to the desired values and click on the " Modify Variable " icon. The PID controller is run with these changed parameter values.
7.	Change the value of the " Setpoint " variable to the desired value, e.g. 50 , and click on the " Modify Variable " icon.
	process will change accordingly until it matches the "Setpoint".

Table 11-3: Commissioning with STEP 7 V5.5

Figure 11-3: Variable table

sele	Ţ	abelle Bearbeit	en Einfügen Zielsystem Variable	Ansicht Ext	ra 3 enster	Hilfe _	5
-0-	1			8 N? 0	9 60° 403	60°1 10°1 /400	
	1	Operand	Symbol 7	Anzeigeformat	Statuswert	Steuerwert	
1		DB35.DBD 0	"DATA".Setpoint	GLEITPUNKT		50.0	
2		DB35.DBD 4	"DATA".PID_output	GLEITPUNKT			
3		DB35.DBD 8	"DATA".Process_value	GLEITPUNKT			
4		DB35.DBW 12	"DATA".crea_status	HEX			
5		DB35.DBW 14	"DATA".exec_status	HEX			
6		DB35.DBX 16.0		BOOL			
7		DB35.DBX 16.1	"DATA".ENABLE	BOOL		true	
8		DB35.DBX 16.2	"DATA".WRITE_Parameters	BOOL		true	
9		DB99.DBD 0	"Model_Param_DB".Derivative_Value	GLEITPUNKT			
10		DB99.DBD 4	"Model_Param_DB".integral_Value	GLEITPUNKT	6		
11		D899.080 8	"Model_Param_DB".Proportional_Value	GLEITPUNKT			
12							

11.5 Manipulating User-Defined Parameters with STEP 7 V5.5

Note Parameter changes with STEP 7 are valid only until the next restart of WinAC RTX.

If you want to permanently save parameter changes, re-encode the DLL/RTDLL file and save it on the runtime system. In this case, it is not necessary to reintegrate the SCL source into the S7 program as the model interfaces have not changed.

NOTICE	When manipulating Simulink model parameters using STEP 7, these
	changes will not be updated when using External mode in Simulink.
	Furthermore, changes to parameters cannot be made via External mode as
	long as the "WRITE_Parameters" variable is set to "TRUE".

11.6 Manipulating User-Defined Parameters with STEP 7 V11/V12

11.6 Manipulating User-Defined Parameters with STEP 7 V11/V12

Once you have downloaded the program to WinAC RTX and copied the DLL/RTDLL file, perform the following steps for commissioning.

Step	Instruction
1.	Set WinLC to RUN mode.
2.	In "Watch and force table", open the "Watch table_1" variable table.
3.	Click on the "Monitor all" icon.
4.	Change the value of the "ENABLE " variable to "TRUE " and click on the "Modify all selected values once and now " icon. The Simulink model parameters are read and saved in the "Model_Param_DB " data block.
5.	Change the value of the "WRITE_Parameters" variable to "TRUE" and click on the "Modify all selected values once and now" icon.
6.	Change the values of " Model_Param_DB " to the desired values and click on the " Modify all selected values once and now " icon. The PID controller is run with these changed parameter values.
7.	Change the value of the " Setpoint " variable to the desired value, e.g. 50.0, and click on the " Modify all selected values once and now " icon. Due to the setpoint change, the PID controller will output an output value. The process will change accordingly until it matches the set " Setpoint ".

Table 11-4: Commissioning with STEP 7 V11/V12

Figure 1111-4: Variable table

	5					
🦻 lo	9, 9, 2 " "					
i	Name	Adresse	Anzeigeformat	Beobachtun	Steuerwert	9
	"Data".Setpoint	%DB35.DBD0	Gleitpunktzahl		50.0 7	
	"Data".PID_output	%DB35.DBD4	Gleitpunktzahl		_	E
2	"Data".Process_value	%DB35.DBD8	Gleitpunktzahl			
0	"Data".crea_status	%DB35.DBW12	Hex			
	"Data".exec_status	%DB35.DBW14	Hex			E
	"Data".INIT	%DB35.DBX16.0	Bool			
	"Data".ENABLE	%DB35.DBX16.1	Bool		TRUE	E
	"Data".WRITE_Parameters	%DB35.DBX16.2	Bool		TRUE	E
	"Model_Param_DB".Derivative_Value	%DB1.DBD0	Gleitpunktzahl			E
0	"Model_Param_DB".Integral_Value	%DB1.DBD4	Gleitpunktzahl	6		IC
1	"Model_Param_DB".Proportional_Value	%DB1.DBD8	Gleitpunktzahl			
2		<hinzufügen></hinzufügen>				E

11.6 Manipulating User-Defined Parameters with STEP 7 V11/V12

Note Parameter changes with STEP 7 are valid only until the next restart of WinAC RTX.

If you want to permanently save parameter changes, re-encode the DLL/RTDLL file and save it on the runtime system. In this case, it is not necessary to reintegrate the SCL source into the S7 program as the model interfaces have not changed.

NOTICE When manipulating Simulink model parameters using STEP 7, these changes will not be updated when using External mode in Simulink. Furthermore, changes to parameters cannot be made via External mode as long as the "WRITE_Parameters" variable is set to "TRUE".

11.7 User-defined parameters in Simulink and STEP 7

11.7 User-defined parameters in Simulink and STEP 7

To allow exact assignment between the parameters in the Simulink model and the parameters in STEP 7, the "**Model_Param_DB**" data block was provided with comments.

The comments are structured as follows:

Table 11-5: Structure of "Model_Param_DB" comments

Step	Instruction
1.	Name of the subsystem where the parameter is located
2.	Name of the block where the parameter is located
3.	Complete internal name of the user-defined Simulink model parameter

Figure 11-5: "Model_Param_DB" data block in STEP 7

Datei	Bearbeiten Einfügen	Zielsystem	Test Ar	nsicht Extras Fenster Hilfe 🔤 🖻
0 🖨 🖁	~ 🖬 🥌 X 🖻 🖻	10 01	011 🏜 🕞	⊐ ≗_ &' !< ≫! □ □ \ ₽
ldresse]	Name	7 _{YP}	Infango	Kommentar
0.0		STRUCT		
+0.0	Derivative_Value	REAL	0.00000	PID Controller Derivative Gain : PID_Mcdel_P.Derivative_Value
14.0	Integral_Value	REAL	0.00000	[PIF Controller/Integral Gain]: PID_Model_P.Integral_Value
+8.0	Proportional_Value	REAL	3.00000	[PIB Controller/Proportional Gain]: PID_Model_P.Proportional_Valu
=12.0		END STRU		

11.7 User-defined parameters in Simulink and STEP 7



Figure 11-6: Simulink model parameter

Note In data blocks in STEP 7 V5.x, the maximum length of variable names is limited to 24 characters. The original parameter names from the Simulink model are used for the variable names in STEP 7. If these names are too long for STEP 7 V5.x, they will be automatically truncated. For security, the original parameter name is in the respective comment.

Note The maximum length for a data block comment in STEP 7 V5.x is 79 characters. When composing the comment from the Simulink model information, this limit may be exceeded. Compiling the SCL source generates all blocks and reduces the relevant location in the comment to the maximum number. The complete comment is retained in the SCL source. A respective warning is output when compiling the SCL source. 12.1 Encoding with WinAC Target

12 Error diagnostics

Different problems may occur when using WinAC Target. The aim of this chapter is to help you with diagnostics.

12.1 Encoding with WinAC Target

When generating codes with WinAC Target, the following error reports may occur.

Error screen

Figure 12-1: Error screen

Message Model error	Source	Reported By	Summary
Model error	the end of the second rest rest of		
	PID_Model	Simulink	The call to WinAC_Target_make_rtw_hook, during th.
A			*****
The call to WinAC WinAC ODK V4.2 The build process Caused by: WinAC ODK V4.2	C_Target_make 2.1 or higher is 5 will terminate 2.1 or higher is	e_rtw_hook, dur required. as a result. required.	ing the before_make hook generated the following error:

Error diagnostics

This error message appears during code generation when WinAC ODK V4.2.1 is not installed.

Install the missing software.

12.1 Encoding with WinAC Target

Error screen

Figure 12-2: Error screen

TID_INIOUCI			
iew <u>F</u> ont Size	9		
Message	Source	Reported By	Summary
Model error	PID_Model	Simulink	The call to WinAC_Target_make_rtw_hook, during the befor
			(10000)
The call to WinAG IntervalZero RTX The build process	C_Target_make (SDK 9.1.2 or H s will terminate	_rtw_hook, durin higher is required as a result.	g the before_make hook generated the following error:
Caused by: IntervalZero RTX	(SDK 9.1.2 or)	igher is required	

Error diagnostics

This error message appears during code generation when IntervalZero RTX SDK 9.1.2 is not installed and the "Real-Time DLL (RTDLL)" option was selected in WinAC Target.

Install the missing software or select "DLL" in the WinAC Target options.

12.2 Connecting to External mode

12.2 Connecting to External mode

When connecting to External mode, the following error messages may occur.

Error screen

Figure 12-3: Error screen

	~		
Message	Source	Reported By	Summary
Model error	PID_Model	Simulink	Error occurred while executing External Mode MEX-file 'ext_com
PID_Model			
PID_Model	while executin	g External Mod	e MEX-file 'ext_comm': OStream. More detail may be reported in the MATLAB command
PID_Model Error occurred v An error occurre window	while executin ed attempting	g External Mod to open an rtI(e MEX-file 'ext_comm': OStream. More detail may be reported in the MATLAB command
PID_Model Error occurred v An error occurre window	while executin ed attempting	g External Mod to open an rtI	e MEX-file 'ext_comm': OStream. More detail may be reported in the MATLAB command
PID_Model Error occurred v An error occurre window	while executin ed attempting	g External Mod to open an rtI(e MEX-file 'ext_comm': DStream. More detail may be reported in the MATLAB command
PID_Model Error occurred v An error occurre window	while executin ed attempting	g External Mod to open an rtI(e MEX-file 'ext_comm': OStream. More detail may be reported in the MATLAB command

Error diagnostics

This error message indicates that Simulink could not connect to the set communication partner.

Table 12-1: Error diagnostics

No.	Instruction
1.	Make sure that you have set the correct IP address in the code generation options. (See Chapter 7.3.1)
2.	Make sure that the DLL runs with compiled External mode in WinAC RTX.
3.	When using a real-time DLL (RTDLL), make sure that the Communication Service is started.
4.	Make sure that port 17725 of your firewall is enabled on the runtime system.

12.2 Connecting to External mode

Error screen

Figure 12-4: Error screen

Tiem Fout 212	e		
Message	Source	Reported By	Summary
Model error	PID_Model	Simulink	Error occurred while executing Exte
PID_Model			
Error occurred v Timed-out waith	while execution ng for first cor	g External Mod nnect response	e MEX-file 'ext_comm': packet.

Error diagnostics

Table 12-2: Error diagnostics

No.	Instruction
1.	Make sure that the S7 program has been downloaded to the PLC.
2.	Make sure that the DLL/RTDLL has been initialized.
3.	Make sure that WinLC is in RUN mode and that the controller is running.

12.3 During operation

12.3 During operation

During operation, the following error messages may occur.

Error screen

Figure 12-5: Error screen

	* :			
Message	Source	Reported By	Summary	
Model error	Unknown	Simulink	Error occurred while executing Exte	
Model error	PID_Model	Simulink	Error occurred while executing Exte	
			4444# %	
Q				
Error occurred v ExtTargetPktPe	while executin nding() call fa	g External Mod lied while check	e MEX-file 'ext_comm': ing for target pkt	

Error diagnostics

If this error message is displayed when running External mode, make sure that the development computer and the target system are connected to one another.

13 References

This list is not complete and only presents a selection of relevant information. Table 13-1: Internet links

	Торіс	Title
1	Reference to the entry	http://support.automation.siemens.com/WW/view/en/ 56969417
2	Windows Automation Center Open Development Kit (WinAC ODK) V4.1	http://support.automation.siemens.com/WW/view/en/ 19256518
3	SIMATIC Embedded Automation SIMATIC IPC427C Bundles, SIMATIC HMI IPC477C Bundles	http://support.automation.siemens.com/WW/view/en/ 37954208
4	MathWorks home page	http://www.mathworks.com/
5	Microsoft home page	http://www.microsoft.com/
6	IntervalZero home page	http://www.mathworks.com/
7	PLC programming software for the entire SIMATIC controller range	http://www.automation.siemens.com/mcms/simatic- controller-software/en/Pages/Default.aspx
8	SIMATIC PC-based Controller	http://www.automation.siemens.com/mcms/programm able-logic-controller/en/software- plc/Pages/Default.aspx
9	SIMATIC Industry PC SIMATIC IPC427C	http://support.automation.siemens.com/WW/view/en/ 37028954
10	STEP 7 – manuals	http://support.automation.siemens.com/WW/view/en/ 10805384/133300
11	SIMATIC WinAC ODK – manuals	http://support.automation.siemens.com/WW/view/en/ 12840073/133300
12	SIMATIC WinAC RTX – manuals	http://support.automation.siemens.com/WW/view/en/ 10805641/133300
13	Windows Automation Center RTX Open Development Kit (WinAC ODK) – manual	http://support.automation.siemens.com/WW/view/en/ 35948966
14	PC-based Automation: Basics for the Solution of Automation Tasks Based on WinAC RTX	http://support.automation.siemens.com/WW/view/en/ 21004765
15	PC-Based Automation: Linking Windows Applications to WinAC RTX with WinAC ODK Using the Example of SIMATIC Vision Sensors	http://support.automation.siemens.com/WW/view/en/ 21572937
16	WinAC ODK Tips and Tricks	http://support.automation.siemens.com/WW/view/en/ 67217690

14 History

Table 14-1: Document versions

Version	Date	Modification
V1.0	04/2013	First version
V1.1	06/2013	Update: Chapter 11 Manipulating User-Defined Parameters with STEP 7