SIEMENS

SIMATIC

Windows Automation Center RTX
WinAC RTX (F) 2010

Operating Instructions

07/2010
A5E00486536-04

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

이 기기는 업무용(A급) 전자파 적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며 가정 외의 지역에서 사용하는 것을 목적으로 합니다.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E00486536-04
Ⓟ 07/2010

Table of contents

Introduction 1

Purpose of this document

This documentation provides detailed information about the Windows Automation Center with Real-Time Extensions (WinAC RTX 2010) software package.

Target group

This documentation is intended for engineers, programmers and maintenance personnel with general knowledge regarding automation systems.

Required background

To understand this documentation, you require a general knowledge of automation engineering. Basic knowledge of the following is also necessary:

- Windows operating systems
- Automation systems
- STEP 7 basic software, particularly:
  - Working with SIMATIC Manager
  - Hardware configuration with HW Config

Validity of the documentation

	Version	Order number
WinAC RTX 2010	V4.6	6ES7671-0RC08-0YA0 (standard)
WinAC RTX 2010	V4.6	6ES7671-0RC08-0YE0 (upgrade)
WinAC RTX F 2010	V4.6	6ES7671-1RC08-0YA0 (full version)
WinAC RTX F 2010	V4.6	6ES7671-1RC08-0YE0 (upgrade)

Note

The particular details about WinAC RTX F are available in the section WinAC RTX F (Page 239).

Changes compared to the previous version

Compared to the previous version, WinAC RTX 2010 V4.6 has the following new features:

- Windows 7 support
- Isochronous mode through PROFINET (Page 135): Isochronous reading and outputting of I/O signals. With the help of OB 6x, the user program can be synchronized with the PN I/O processing. Isochronous processing requires IRTtop. Isochronous mode is possible from 250 µs to 4 ms.
- Support of IRT with high performance (Page 134): In addition to reserving of the band width, it is necessary to configure the topology to achieve the highest possible performance.
- Web server support (Page 152)
- Functionality Shared Device (Page 136): To spare devices, 2 controllers can share one device.
- Setting the IP configuration with SFB 104 (Page 113): The "local adjustment" of the IP address parameters and of the "NameOfStation" is possible for I-Device and controller as IP Config. IP address parameters and "NameOfStation" (NoS) can be set via the SIMATIC Manager and via SFB 104.
- Extension of the open communication via PROFINET (ISO on TCP) (Page 147)
- Media redundancy (MRP) (Page 137): Devices that support the media redundancy protocol can be networked to a ring with Ethernet lines. There are two data paths between each two ring nodes.
- Data record routing (Page 122) to PROFIBUS for TCI, PDM and IO-Link
- Support of new communication interfaces (CP 5623, CP 5624) (Page 116)
- Support of new industrial PCs (SIMATIC IPC627C, SIMATIC HMI IPC677C)
- Integration of the system status list SZL0x9C (Page 274): Provides information via PROFINET IO tool changers. Tool changers are IO devices that "manage" the groups of other IO devices (tools).
- More precise input option for the minimum sleep time and minimum cycle time in the tuning panel (Page 79)

Definition

The following terms are used in this documentation:

- **CP 56x1**: This general term encompasses the CP 5611 **and** the CP 5621. Any restrictions are indicated in the individual case.
- **CP 56x3**: This general term encompasses the CP 5603, the CP 5613, **and** the CP 5623. Any restrictions are indicated in the individual case.
- **CP 56x4**: This general term encompasses the CP 5614 **and** the CP 5624. Any restrictions are indicated in the individual case.
- **Distributed I/O**: This term is used for PROFIBUS and PROFINET.

Path information for Windows 7

The operating-system-relevant path information named in the manual relates to Windows XP. This section lists the most important path information also for Windows 7.

- Automation License Manager:

  Start > All Programs > Siemens Automation > Automation License Manager
- PC-based Control:

  Start > All Programs > Siemens Automation > SIMATIC > PC-based Control
- Control Panel: Start > Control Panel
- Creating a rescue medium: Start > All Programs > Maintenance > Backup and Restore
- Documentation:

  Start > All Programs > Siemens Automation > SIMATIC > Documentation

Location of documentation

The WinAC RTX installation includes this documentation as both online help and a PDF online manual and also a PDF online manual for WinAC Time Synchronization. In the Setup program, you can select whether or not you want to install the documentation. If installed, the online help can be opened in the controller panel and all applicable PDF files are accessible with the **Start > SIMATIC > Documentation** menu command.

Position in the information landscape

When working with WinAC RTX and depending on your particular application, you will need to consult the additional documentation listed below.

Documentation for	Brief Description of Relevant Contents
STEP 7 - Programming with *STEP 7*	This manual provides basic information for drafting and programming a STEP 7 user program for WinLC RTX. Available on the Internet (http://support.automation.siemens.com/WW/view/en/18652056).
STEP 7 - System and Standard Functions for *S7-300* and *S7-400*	WinLC RTX includes integrated system functions and organization blocks, which you can use when programming. This manual provides you with descriptions of the system functions, organization blocks and loadable standard functions. Available on the Internet (http://support.automation.siemens.com/WW/view/en/1214574).
STEP 7 - Working with *STEP 7*	This manual explains the usage and the functions of the STEP 7 automation software. The manual provides an overview of the procedure used to configure WinLC RTX and to develop STEP 7 user programs. Available on the Internet (http://support.automation.siemens.com/WW/view/en/18652511).
SIMATIC NET - Commissioning PC stations	This manual supports you when commissioning your SIMATIC NET PC modules in a PC station. All SIMATIC NET software tools and their operation are described (available once SIMATIC NET is installed). Available on the Internet (http://support.automation.siemens.com/WW/view/en/13542666).

Documentation for	Brief Description of Relevant Contents
SIMATIC NET - Industrial Communication with PG/PC, Parts 1 and 2	This manual supports you during setup of industrial communication over PROFIBUS and Industrial Ethernet communication networks (available if you install SIMATIC NET).
WinAC Time Synchronization	This manual describes the configuration and operation of WinAC Time Synchronization. Available on the Internet (http://support.automation.siemens.com/WW/view/en/22205381).
PROFINET System Description	This manual describes the installation, commissioning and operation of a PROFINET system and contains instructions and examples for programming diagnostics for I/O devices. Available on the Internet (http://support.automation.siemens.com/WW/view/en/19292127).
From *PROFIBUS DP* to *PROFINET IO:*	This manual describes differences between PROFIBUS DP and PROFINET IO and supports the migration from PROFIBUS DP to PROFINET IO. Available on the Internet (http://support.automation.siemens.com/WW/view/en/19289930).
PROFINET Getting Started Collection	These manuals contain information for configuring specific PROFINET interfaces and PROFINET IO devices. Available on the Internet (http://support.automation.siemens.com/WW/view/en/19290251).
SIMATIC Communication:	This manual contains a general overview of communication networks and communication technologies used in the automation industry with an emphasis on SIMATIC products. Available on the Internet (http://support.automation.siemens.com/WW/view/en/25074283).
SIMATIC Isochronous Mode	This manual contains a complete overview of isochronous mode: Available on the Internet (http://support.automation.siemens.com/WW/view/en/15218045).
S7-modular Embedded Controller EC31-HMI/RTX	This manual contains information on the operation of the S7-mEC EC31-RTX / EC31-HMI/RTX. Available on the Internet (http://support.automation.siemens.com/WW/view/en/28593853/133300).

Guide

The present document describes how to work with the WinAC RTX software. It consists of instructive sections and a reference section. The documentation includes the following subject areas:

- Product overview
- Installation
- Getting started
- Operating the Controller
- Working in STEP 7
- Communication
- Tuning the controller performance
- Connecting the controller to the SIMATIC NET OPC server
- Operating WinAC RTX with S7-modular Embedded Controller
- WinAC RTX F
- Reference information

Service & Support on the Internet

A guide to the technical documentation offered for the various SIMATIC products and systems is available on the Internet (http://www.siemens.com/simatic-tech-doku-portal).

In addition to our documentation pool, we offer our complete online knowledge base on the Internet (http://www.siemens.com/automation/service&support). There you will find:

- The newsletter, which is constantly updated to provide you with the latest information about your products
- The right documents via our Search function under Service & Support
- The bulletin board, a worldwide knowledge exchange for users and experts.
- Your local representative for Automation & Drives via our representatives database
- Information about on-site services, repairs, spare parts and lots more.

2 Product overview

2.1 Introduction to PC-Based Control

Components

The Windows Automation Center with real-time extensions (WinAC RTX 2010) software package includes the following components:

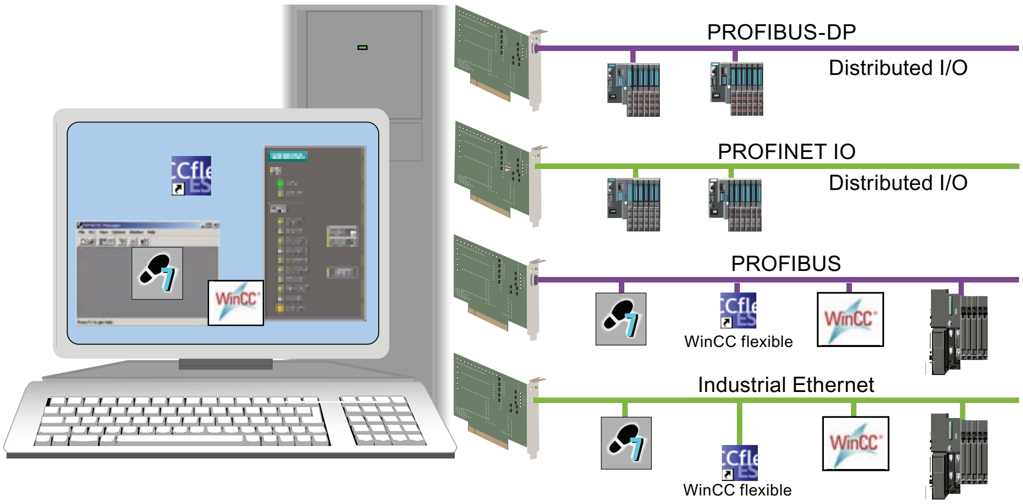
- Windows Logic Controller RTX (WinLC RTX V4.6)
- IntervalZero RTX 2009
- WinAC Time Synchronization V4.2
- Automation License Manager V5.0 SP1
- SIMATIC NET V7.1 SP2 and SIMATIC NET V8.0 including license for Softnet S7 Lean V8.0

You install WinAC RTX and the documentation from the installation DVD included with your release.

Introduction

The PC-based controllers of WinAC (Windows Automation Center) provide the same functionality as SIMATIC S7 CPUs (hardware controllers). The Windows Logic Controller with real-time extensions (WinLC RTX) provides the functionality of a programmable logic controller (PLC) in a real-time, PC-based environment. WinLC RTX uses the IntervalZero RTX (formerly Ardence) real-time extension for Windows and is fully code-compatible with the SIMATIC product family.

As part of the SIMATIC family of automation products, WinLC RTX can also communicate with STEP 7 or other SIMATIC products, such as WinCC flexible or other SIMATIC S7 automation systems, including any of the PC-based controllers, over PROFIBUS or Industrial Ethernet networks. They use the same method to communicate with distributed I/O, for example, with an ET 200S device. They use PG/OP communication (PROFIBUS or Industrial Ethernet) for connecting to STEP 7 or other programming software on another computer.



You can use the same programming languages, program structure, and programming user interface (STEP 7) as for hardware PLCs to develop your process control solution. Programs that have been written for S7 automation systems can be used on PC-based controllers and vice versa. The PC-based controllers contain a controller panel (Page 17) that runs on the PC. With these capabilities, you can use WinLC RTX in a typical factory automation.

2.2 Introduction to the WinAC RTX controller panel

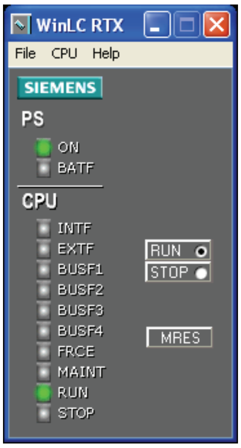
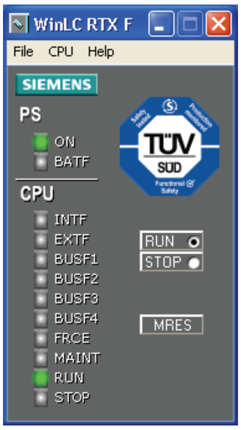
The controller panel corresponds to the front panel of the SIMATIC S7 CPUs.
It is used to start or shut down the controller and to perform other operations.

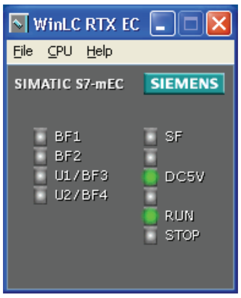
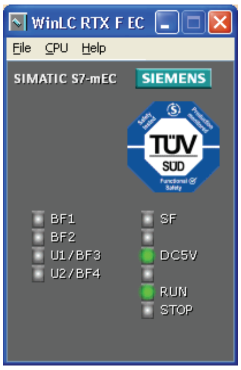
The controller panel is a display window on your PC that contains the following elements for working with the controller:

- Two operating mode selector positions for changing the operating mode of the controller (Page 60) (similar to the mode selector switch on an S7 CPU front panel).
- An MRES switch position for resetting the memory (Page 76)
- Status indicators (Page 77) for the controller
- Menus for controller operation

Different versions of the WinLC RTX controller panel

The user interface of the WinLC RTX controller panel can look different, depending on the platform or software package.

WinAC RTX on a standard PC	WinAC RTX F on a standard PC
	

WinAC RTX on an Embedded Controller	WinAC RTX F on an Embedded Controller
	

2.3 Relationship between the Controller and the Controller Panel

Opening and closing the controller panel

Opening and closing the controller panel does not influence the state of the controller. The status of the switches and the LEDs are stored in the controller.

WinAC RTX icon in the Windows taskbar

The icon is displayed in the Windows taskbar whenever the controller is in operation. You can double-click this icon to open the controller panel when the controller is in operation and the controller panel is closed.

The color of the background frame of this icon provides some additional information about the controller:

- Yellow: The controller is in STOP mode.
- Green: The controller is in RUN mode.
- Red: The controller is in a defective state.

Here, it is the color, rather than the mode selector position, that indicates the actual operating mode. For example, the mode selector can be set to RUN, but the controller can be in STOP mode due to a program execution error or an operating mode change from STEP 7.

2.4 PC-based Control Features

Real-time process control

WinLC RTX is a software version of an S7 controller that adds real-time control provided by a real-time subsystem for the Windows operating system. WinLC RTX executes STEP 7 user programs like other S7 automation systems and allows for easy integration with STEP 7 and standard Windows applications. WinLC RTX runs in two separate environments: processes that run in the real-time system and processes that run in the Windows environment.

- The processes that run in the real-time system execute the STEP 7 user program for WinLC RTX. The process control is assigned the highest priority.
- The processes that run in the Windows environment perform other operations, such as communication and interfaces to Windows systems and applications.

Advantages of real-time extensions (RTX)

WinLC RTX uses real-time extensions (IntervalZero RTX) to provide the following functions:

- Deterministic operation ensures the response is predictable. Execution of the STEP 7 user program occurs entirely in the real-time subsystem, thus reducing "jitter".
- The control process is protected from Windows system errors (Page 97). WinLC RTX is notified of all Windows shutdowns (including those caused by a "blue screen") in order to programmatically shut down in an orderly fashion. You can configure Windows to reboot automatically after a system failure. This option is accessed by the "Startup and Recovery" button under the "Advanced" tab of System Properties in Windows Control Panel.

SIMATIC functionality supported by WinLC RTX

WinLC RTX provides the following features:

- Support for automatic startup of WinAC RTX when the PC boots or during manual startup
- Implementation of a large number of S7 code blocks of SIMATIC controllers: organization blocks (OBs), system function blocks (SFBs) and system functions (SFCs).
- Support for PROFIBUS DP for communication with distributed I/O, including DPV0 and DPV1 (Page 120) slaves (PROFIBUS DPV1 provides enhanced alarm and status reporting, in order to communicate with intelligent slave devices).
- Support for PROFINET (Page 125) communication via Ethernet submodules: PROFINET I/O (Page 128) for communication with distributed I/O and PROFINET CBA (Page 138) (Component Based Automation) for communication with other CBA components.
- Support for a maximum of four individual subnets for connecting the distributed I/O.

- Support for isochronous (Page 123) mode for PROFIBUS DP and PROFINET IO subnets, which allows WinLC RTX to operate with a constant bus cycle.
- Uses S7 communication services and offers compatibility with SIMATIC applications, such as STEP 7, WinCC, and WinCC flexible for tasks such as programming, testing, monitoring, and visualization.
- Enables point-to-point communication (Page 115) between controllers (hardware and software) on the network.
- Supports routing of S7 communication through the submodule CPs of WinLC RTX, which allows STEP 7 on one subnet to connect to an S7 station (such as an S7-400 controller) on a different subnet.
- Supports time synchronization via NTP
- Provides the ability to archive and restore control programs (Page 85).
- Allows you to control the operating mode of the controller and to view status information in the controller panel (Page 17).
- Provides a tuning panel (Page 79) for optimizing system performance.
- Provides time synchronization as either a time master or slave.
- Use of the supplied SIMATIC NET Softnet-S7 Lean license enables the following:
  - Use of the SIMATIC NET OPC server
  - Ability of OPC client applications to access process data
  - SIMATIC communication via the Windows-Ethernet interface of the PC
- The Web server allows you to monitor your PC station via the Internet or via your company's Intranet. This allows evaluation and diagnostics to be carried out remotely.

Use of WinLC RTX with WinAC ODK

WinLC RTX can be used together with WinAC ODK (Open Development Kit) for your total automation solution. WinAC ODK (must be ordered separately) enables you to develop additional software:

- Custom PC applications in a higher-level programming language for the purpose of exchanging data with WinLC RTX via one of the following programming interfaces of the STEP 7 user program: CCX (Custom Code Extension) and SMX (Shared Memory Exchange).
- Software for user interfaces, such as a custom controller panel, to display status information and perform controller operations using the CMI (Controller Management Interface) programming interface.

2.5 System Requirements

To use WinLC RTX, your PC must meet the following requirements:

Category	Requirements
Operating system	Microsoft Windows XP Professional Service Pack 2 and 3 Microsoft Windows 7 Ultimate, Professional and Enterprise **Note:** WinAC RTX (F) only supports the 32-bit variants of the operating systems. IntervalZero (formerly Ardence) RTX 2009 (included in the product package of WinAC RTX) **Note:** Some hardware configurations that are not SIMATIC industrial PCs do not support installation and operation of IntervalZero RTX. The hardware and software requirements for IntervalZero RTX can be found in the RTX Runtime Release Notes on your installation DVD.
Processor and memory	PC system: • 900 MHz or higher • At least 1 Gbyte of RAM • BIOS must support plug-and-play (ACPI, Advanced Configuration and Power Interface) **Note:** Multi-core and hyperthreading systems are also supported.
Supported Windows HALs	HAL (Hardware Abstraction Layer): • MPS multiprocessor PC • ACPI uniprocessor PC • ACPI multiprocessor PC **Note:** WinAC RTX does not support standard HAL, since PIC (Programmable Interrupt Controller) systems are not supported.
Hard disk	Hard disk with 125 Mbytes of free memory space for complete installation. During setup, it can specified that certain components, such as documentation, are not to be installed in order to save memory space. **Note:** We recommend that you use an NTFS type file system. The setup program uses at least 1 Mbyte of available memory on the C:\ drive for the WinLC setup program (setup files are deleted once the installation is complete).
Operator interface	Color monitor, keyboard and mouse or other pointing device (optional) that are supported by Windows
Communication interface	One or more communication interfaces for communication with STEP 7 or other S7 applications or for communication with distributed I/O
Siemens software	Programming and configuration software: STEP 7 V5.5 with installed hardware update for WinLC RTX.
SIMATIC NET (optional)	You only have to install SIMATIC NET from the WinAC RTX installation DVD if you need functionality such as use of the OPC server. The SIMATIC Softnet-S7 Lean license (6GK1704-1LW63-3AA0) is contained in your WinAC RTX installation package. Additional information regarding SIMATIC NET products for PC-based automation can be found in the SIEMENS Mall or the ST PC Catalog.

2.6 Windows User Privileges

You do not need Windows administrator rights (ADMIN) to perform WinAC RTX operations, such as:

- Changing the operating mode (Page 60) of the controller
- Changing the idle time or the minimum cycle time (Page 200) of the controller
- Archiving or restoring (Page 85) control programs
- Setting the security options (Page 70)

As a "power user", "user", or "guest", you can perform any operation from the WinLC RTX controller panel. You can manage the network rights for the PC station within your application and prevent conflicts during installation, commissioning, and operation of a PC-based automation solution.

User change

If you configured WinLC RTX to start at PC boot (controller was started automatically), one user can log off and another user can log on without affecting the controller operation.

If you have **not** configured WinLC RTX to start at PC boot (controller was started manually), the controller will be shut down when the user changes.

Restrictions for Windows XP

The following table shows which functions are restricted for specific Windows user rights.

Operation	Administrator	Power user	User	Guest
Installing WinAC RTX software	Permitted	Not permitted	Not permitted	Not permitted
Configuring or modifying the PC station	Permitted	Permitted	Not permitted	Not permitted
Performing WinAC RTX operations	Permitted	Permitted	Permitted	Permitted

Restrictions for Windows 7

The following table shows which functions are restricted for specific Windows user rights.

Operation	Administrator	TIA Engineer	User
Installing WinAC RTX software	Permitted	Not permitted	Not permitted
Configuring or modifying the PC station	Permitted	Permitted	Not permitted
Performing WinAC RTX operations	Permitted	Permitted	Permitted

Defining access rights under MS Windows 7

When you install WinAC RTX under MS Windows 7, a "Siemens TIA Engineer" user group will be created automatically. This allows the users that are logged in there to configure the PC station and to execute the WinAC RTX operations. These user rights must be assigned by the administrator.

Proceed as follows to include those local users in the "Siemens TIA Engineer" group whose login permits access to WinAC RTX:

1. Open the Control Panel under Windows and select the entry "User accounts".
2. Select the "User accounts" in the navigation window.
3. Select the "Manage user accounts" in the navigation window.
4. In the "Advanced" tab select the Advanced" entry in the "Extended user administration" section.
5. Select the "Local users and groups > Users" in the navigation window. All the users are displayed in the data window.
6. In the "Action" menu select the "New user" command and create a user account with the same login for each user who is to access WinAC RTX.
7. Select the context-specific menu command "Properties" for each created user.
8. In the displayed dialog box select the "Member of" tab and click the "Add" command button.
9. Enter the "Siemens TIA Engineer" user group in the "Enter the object names to be used" field in the "Select groups" dialog box.
10. Confirm your entries with "OK".

Creating the domain-global user group "Siemens TIA Engineer"

In the case of operation in a domain, a domain-global user group can alternatively be set up, which is then mapped to the local user groups "Siemens TIA Engineer".

The following prerequisites must be fulfilled:

- The domain administrator has created a domain-global user group.
- Within the domains the domain administrator has included those users into the domain-global user group under whose login WinAC RTX is accessed.

2.7 Using Help

The online help system provides information about the controller panel and the controller. This topic provides information about using online help:

- Accessing help from the controller panel
- Using the table of contents
- Using the Index
- Using full-text search
- Printing help topics

Accessing help from the controller panel

To access online help from the controller panel, use one of the following methods:

- Click on an entry in the "Help" menu.
- Click the "Help" button in a dialog or message box to view information about that specific dialog or message box.
- Press the F1 key to view context-sensitive help on the currently selected item (for example, a window, dialog, or menu).

The menu commands available in the controller panel Help menu are listed below:

- Help on controller
  The Help > Help on controller menu command displays the first page of the online help for the controller, which is connected to the controller panel. It describes the controller and controller panel operations.
- Introduction
  The Help > Introduction menu command displays a help topic that provides an introduction to PC-based control and the capabilities of the controller.
- Getting Started
  The Help > Getting Started menu command displays a help topic that assists you while working with the controller for the first time.

Using the table of contents

The table of contents is in the left pane of the browser and provides navigation within the online help system:

- Click a book to open it and display the books and topics that it contains.
- Click the book again to close it.
- Click any topic within the table of contents to display that topic.

The topic you are currently viewing is highlighted in the table of contents.

The table of contents can be either hidden or displayed:

- Click the "Hide" button in the browser to close the table of contents.
- Click the "Show" button in the browser and select the "Contents" tab to open the table of contents.

Using the Index

The index provides access to information about a specific subject. Select the "Index" tab to display the index. (If the "Index" tab is not visible, click the "Show" button in the browser.)

Using full-text search

To use the full-text search capabilities of the online help, select the "Search" tab. (If the "Search" tab is not visible, click the "Show" button in the browser.)

The full-text search supports the Boolean operators AND (&), OR (|), and NOT (!), expressions within quotation marks, nesting of expressions with parentheses, and the wildcards * and ? in search expressions.

Printing help topics

To print all or part of the online manual corresponding to the online help, follow these steps:

1. Use the **Start > SIMATIC > Documentation** menu command to open the PDF file.
2. Use the **File > Print** menu command to print all or part of the manual.

Installation 3

3.1 Overview of the Installation Tasks

Requirements

To install WinAC RTX, your computer must meet the following requirements:

- Your PC must meet the system requirements (Page 21).
- You must have Windows administrator (ADMIN) rights.
- All previous versions of WinAC or RTX must be uninstalled on your computer.
- For communication with distributed I/O, your computer must have at least one communication interface (Page 39).

Procedure 1 - A previous version was previously installed on your PC

To install WinAC RTX properly, follow these steps:

1. Make an archive of your existing online STEP 7 user program.
2. Make a note of your custom controller panel settings (optional).
3. Uninstall the following software packages, if present on your computer, in the order indicated:
   - WinLC Basic or WinLC Basic Demo
   - WinLC RTX
   - IntervalZero RTX
4. Restart the computer.
5. Finish the installation by performing steps 1 to 3 of "Procedure 2".

Procedure 2 - A previous version was not previously installed on your PC

To install WinAC RTX properly, follow these steps:

1. Install the WinAC RTX software.
2. To use the OPC server or other SIMATIC NET features, you must install SIMATIC NET from the WinAC RTX installation DVD. Otherwise, you do not need to install SIMATIC NET.
3. License the WinAC RTX installation (Page 32) with the Automation License Manager.

3.2 Setup - Archive data and settings

Because old versions of WinAC RTX must be uninstalled prior to a new installation, programs and settings will be lost.

NOTICE

Data loss

A new installation of WinAC RTX deletes the existing STEP 7 user program on the controller as well as the retentive data and all previous settings of the tuning panel, Station Configuration Editor (specifically for WinAC), WinAC data storage, and other WinAC RTX options.

Procedure

To save your STEP 7 user program and other settings, follow these steps:

1. Archive (Page 85) your STEP 7 user program and the configuration.
2. Make a note of your personal settings in the following areas:
   - Station Configuration Editor
   - WinLC properties
   - Tuning panel
   - Data storage
   - Autostart and password

Result

Following installation, you can restore (Page 85) your STEP 7 user program and the configuration, and reenter the settings you noted.

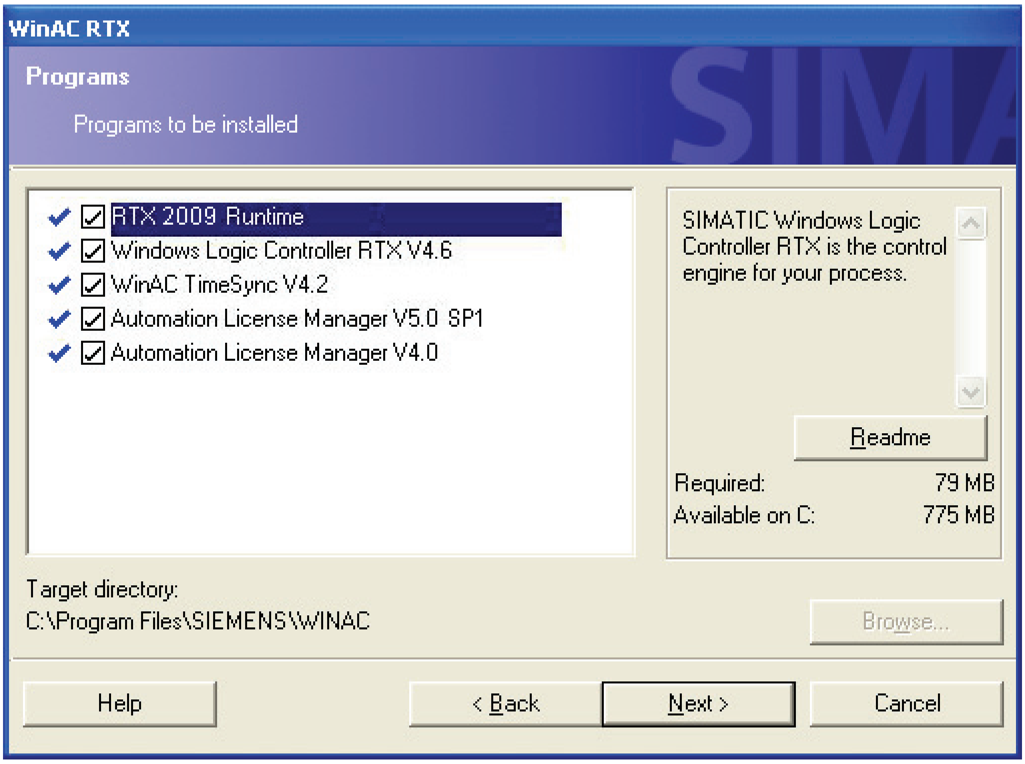
3.3 Installing the WinAC RTX software

To install the WinAC RTX software, including the IntervalZero RTX extensions, uninstall the software listed in Overview of the Installation Tasks (Page 27) in the order indicated.
Insert the WinAC RTX DVD. Follow the instructions of the setup program.
If the setup program does not start automatically, double-click the Setup.exe file on the DVD.

Procedure

To install the WinAC RTX software, follow these steps:

1. Select the language for performing the installation.
2. From the list of WinAC RTX components, select the components to be installed.



WinAC RTX

Programs

Programs to be installed

- ✓ ☑ RTX 2009 Runtime
- ✓ ☑ Windows Logic Controller RTX V4.6
- ✓ ☑ WinAC TimeSync V4.2
- ✓ ☑ Automation License Manager V5.0 SP1
- ✓ ☑ Automation License Manager V4.0

SIMATIC Windows Logic Controller RTX is the control engine for your process.

Required: 79 MB
Available on C: 775 MB

Target directory:
C:\Program Files\SIEMENS\WINAC

3. Confirm with "Next".
   This starts the installation. The progress of the installation is displayed.
4. Follow the instructions in the dialog boxes, which guide you through the installation.

5. Choose which setup type you prefer:
   - Typical: This option installs the complete software and, by default, all documentation in all supported languages.
   - Minimal: This option installs WinLC RTX without documentation. English is preselected as the product language for the installation.
     This option requires the least amount of memory space on the hard disk.
   - Custom: This option installs the languages, online help, and manuals that you select in the subsequent dialog boxes.
6. Choose whether you want to carry out the licensing of WinAC RTX (Page 32) during the installation or at a later time.

Result

The setup program indicates when the installation is finished.

Note

SIMATIC NET message during installation

If SIMATIC NET is installed on your computer and a message is displayed by SIMATIC NET indicating that the CP is configured for use with SIMATIC NET and STEP 7,
confirm with "OK". This is a normal part of the installation process.

3.4 Installing SIMATIC NET

Introduction

SIMATIC NET is available on your WinAC RTX installation DVD. The license key for die Softnet-S7 Lean licenses are located together with the license keys for WinLC RTX on the USB stick that is included in your WinLC RTX installation package.

Install SIMATIC NET and the Softnet-S7 Lean license in accordance with the instruction in the SIMATIC NET readme file.

File location

The path to the file location of SIMATIC NET on the WinAC RTX installation DVD:

- SIMATIC_NET\SIMATIC_NET_CD_2008 (operating system support for Windows XP)
- SIMATIC_NET\SIMATIC_NET_CD_2010 (operating system support for Windows 7)
- SIMATIC_NET\SIMATIC_NET_Manual_CD_08-2010 (SIMATIC NET documentation)

Applications

Using the supplied SIMATIC NET Softnet-S7 Lean license enables the following features:

- Configuring a communication interface in the PC station
- Use of the OPC server
- Use of the Windows Ethernet interface for SIMATIC communication (PG/OP, S7 communication, S7 routing)

You do not need SIMATIC NET to configure communication interfaces as submodules of WinLC RTX.

Compatibility between WinAC RTX and SIMATIC NET

Additional information regarding compatibility between different versions of WinAC RTX and SIMATIC NET can be found on the Internet (http://support.automation.siemens.com/WW/view/en/22304176).

3.5 Licensing WinAC RTX

The WinAC RTX software requires a product-specific license key that you install with the Automation License Manager. Each SIMATIC automation software product (for example, STEP 7) has a separate license key. You must install the license key for each product.

Certificate of license

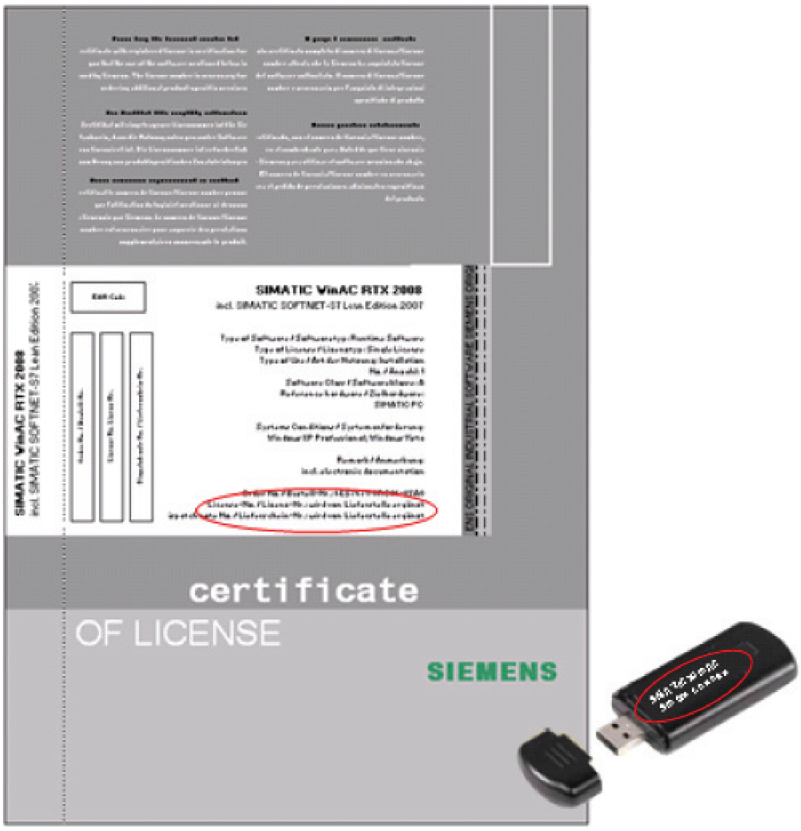
A paper certificate of license is included in the scope of delivery. It contains your unique license number. This serves as proof that you have a valid license key for WinLC RTX. Keep this certificate in a secure place where it can be easily accessed from the location of the computer on which WinLC RTX runs. If you must replace your license key because it was lost or damaged, contact the Siemens hotline (http://www.siemens.com/automation/service&support). Make sure you have your certificate of license available when you contact the hotline. You must have a valid certificate of license to get a replacement license key.

License key

The license key for WinAC RTX is located on a USB stick that is included in the scope of delivery.

If the USB stick containing the license is lost or damaged, you can contact the hotline to obtain a new license key. You need the certificate of license to receive a replacement license key from Siemens.

The following figure shows a certificate of license with a license number for a particular license key. The license key for WinAC RTX is located on a USB stick with an imprinted license number.


SIMATIC WinAC RTX 2008
incl. SIMATIC SOFTNET-S7 Lean Edition 2007
SIMATIC PC
SIMATIC WinAC RTX 2008
incl. SIMATIC SOFTNET-S7 Lean Edition 2007
certificate
OF LICENSE
SIEMENS

Installing the license key during installation

Note

The license key must be installed on a file system of the type NTFS.

When you install WinAC RTX, the Automation License Manager is also installed. If you have not yet installed the Automation License Manager on your computer, select the check box for it to be included as part of the installation. The Automation License Manager will then be installed on your PC during installation of WinAC RTX. A subsequent dialog allows you to choose whether to install the license key during installation. You will find information on installing a license key in the online help of the Automation License Manager. If you do not install the license key, WinLC RTX will display a message at regular intervals indicating that no license key exists.

Installing the license key a later time

If you start the WinAC RTX software and no license key is installed, a prompt is displayed on the screen. If the Automation License Manager is not yet installed on your computer, follow these steps to install the Automation License Manager and the license for WinLC RTX.

1. Insert the WinAC RTX installation DVD and start the setup program.
2. When the "Components" dialog box is displayed, select the "Automation License Manager" check box.
3. After the installation select the menu command **Start > Program Files > Siemens Automation > Automation License Manager**. You can also open the Automation License Manager from the desktop.
4. Continue to install the license key according to the instructions in the online help of the Automation License Manager.

If the Automation License Manager is already installed on your computer, perform the last two steps in the procedure above in order to license WinLC RTX.

Transferring an installed license key

In the Automation License Manager, you can transfer a license key from one computer to another. For more information, refer to the online help in the Automation License Manager.

Working with the WinLC RTX controller without a license key

If a license for WinAC RTX is not available on your computer, the WinLC RTX controller will continue to operate. However, a message will inform you at regular intervals that the license key is missing.

Recovering the License Key if the hard disk is defective

If a fault has occurred on the hard disk or the USB stick containing your license key file, contact your Siemens representative (http://www.siemens.com/automation/service&support). Make sure you have your certificate of license available when you contact the hotline.

3.6 Uninstalling IntervalZero RTX and WinAC RTX

The IntervalZero RTX real-time extension and the WinAC RTX software product can be removed independently of one another.

Note

If you want to remove the real-time extension RTX **and** WinAC RTX,
you must first remove WinAC RTX.

Procedure

To remove the real-time extension IntervalZero RTX or WinAC RTX on your computer, proceed as follows:

1. Double-click the "Add/remove programs" icon in Windows Control Panel.
2. Select the entry of the "WinAC RTX" component in the displayed list of installed software.
3. Click the "Remove" button to remove WinAC RTX.
4. Select the entry of the "IntervalZero RTX" component in the displayed list of installed software.
5. Click the "Remove" button to remove IntervalZero RTX.

If the "Confirm File Deletion" dialog appears, click "No" if you are uncertain.

Note

When you remove WinAC RTX, the controller panel and WinLC RTX are automatically closed, if they were still open. A message is not displayed beforehand.

Removal when using Windows 7

If you are using the Windows 7 operating system, you must restart the PC after removing IntervalZero RTX.

Getting started 4

4.1 Getting Started Overview

The Getting Started section helps you to establish communications between the controller, STEP 7, and I/O devices. You must perform the following tasks:

- Use the Station Configuration Editor to designate a communication interface as a submodule (Page 44) of WinLC RTX.
- Use STEP 7 to configure (Page 52) the hardware and STEP 7 user program and to download the system blocks.

The Getting Started section also helps you understand the basic concepts for setting up a PC-based controller: PC station (Page 35), Communication Interface (Page 39), Index (Page 40), Submodule (Page 41), and Interface (IF) Slot (Page 43).

4.2 Explanation of terms

4.2.1 What is a PC station?

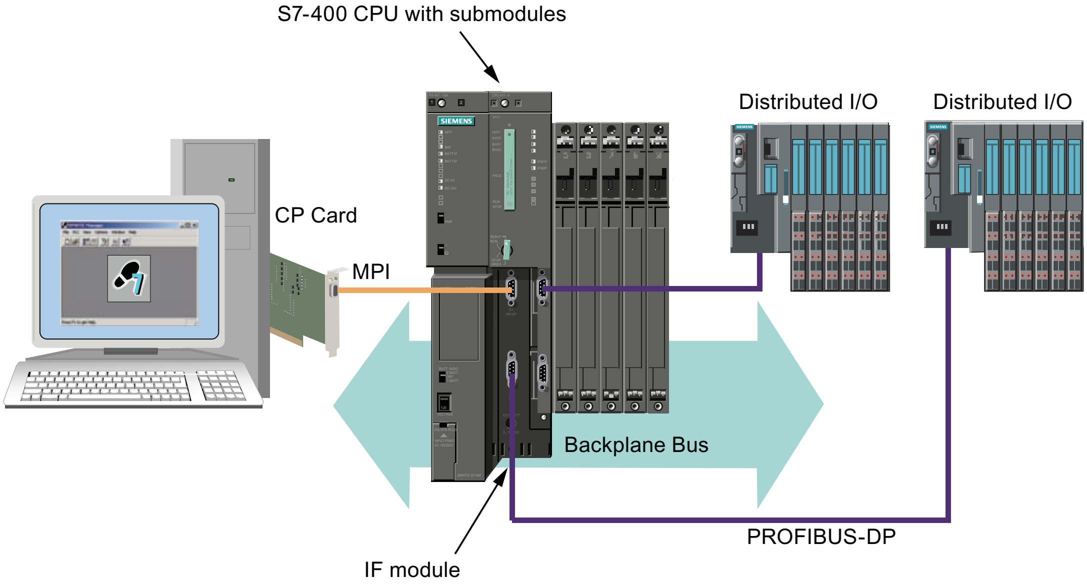
The PC station is a software-based virtual rack that is displayed in the Station Configuration Editor and used for the creation of a PC-based automation system. Like a hardware rack of an S7 CPU-based automation system, the PC station contains space for several modules required for the PC-based automation system.

When you install the WinAC RTX software, the controller appears by default in the second slot (index (Page 40)) of this virtual rack in the Station Configuration Editor. The PC station is also represented in the STEP 7 HW Config (Page 52) editor. The controller in the PC station contains four configurable IF slots for assigning communication interfaces (Page 39) as submodules (Page 41) to be used for communication with distributed I/O, STEP 7, or other S7 applications.

S7-400 communication model

A PC-based controller is similar to an S7-400 hardware controller. The S7-400 controller consists of modules in a rack that communicate over the backplane bus of the rack. Communications for an S7-400 are defined as follows:

- STEP 7 communicates with the controller (in this example, an S7-400 CPU) over an MPI subnet using a CP that is installed in the computer.
- The controller communicates with expansion modules over the backplane bus of the rack.
- The S7-400 CPU uses an integrated submodule interface or an IF module to communicate with distributed I/O (in this example, over a PROFIBUS DP subnet).



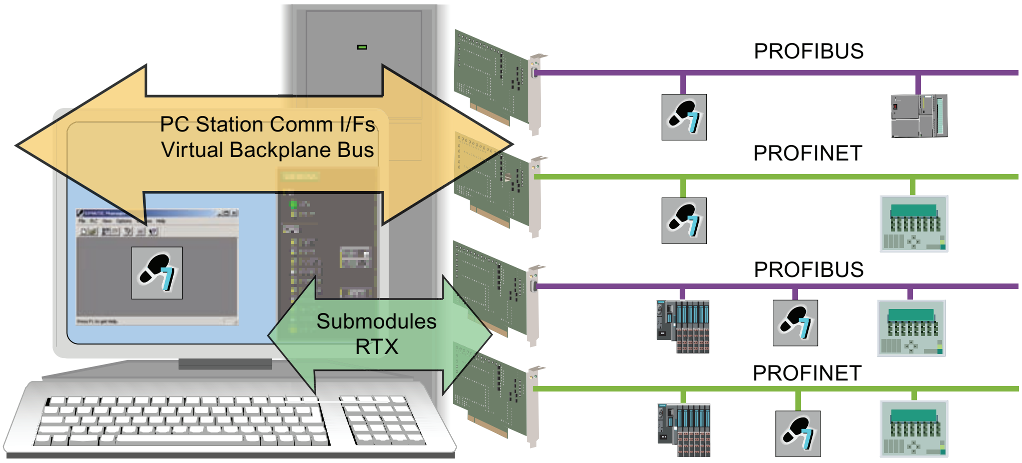
In an S7-400 station, the following types of communication are possible:

Integrated interfaces	CP modules used over backplane bus
Operation of distributed I/O Supported interfaces: • MPI • PROFIBUS • PROFINET	Operation of central I/O Supported communication processors: • PROFIBUS • PROFINET • Industrial Ethernet

Communication model with PC station and PC-based controller

WinLC RTX uses communication interfaces such as the CP 5613 for communication tasks and access to distributed I/O. You can configure and use communication interfaces in WinLC RTX in one of two ways:

- **Configuration as a submodule of WinLC RTX:** A communication interface configured as a submodule is operated in the real-time system and provides optimum performance and stability for communication with distributed I/O. Submodules of WinLC RTX are similar to integrated communication interfaces of an S7-400 controller.
- **Configuration as an interface in the PC station:** A communication interface configured in the PC station is operated in the Windows operating system and is available for many different communication tasks. However, it cannot be used for WinLC RTX communication with distributed I/O. The communication interfaces of a PC station are similar to CP modules that are installed in the rack of an S7-400. WinLC RTX uses a virtual backplane bus that is similar to the S7-CPU backplane bus for communication with components in the PC station and with other PC applications on the computer with WinLC RTX.



Note

No additional software is required to configure communication interfaces as submodules of WinLC RTX. Configuration in the PC station requires the installation of SIMATIC NET, which is a separate software package.

The following table contains the characteristics of the two communication types:

Submodule communication (similar to onboard interface on an S7-CPU)	PC station communication (similar to a CP module that communicates over the backplane bus of an S7-400 station)
Operation exclusively in the real-time system Access to distributed I/O Supported protocols/communication types: **PROFIBUS** • PG/OP communication • S7 communication • S7 routing • PROFIBUS DP **PROFINET** • PG/OP communication • S7 communication • S7 routing • Open user communication (TSEND/TRCV) • PROFINET IO • PROFINET CBA	Operation in the Windows environment No access to the distributed I/O or central I/O Supported protocols/communication types: **PROFIBUS** • PG/OP communication • S7 communication • S7 routing **Industrial Ethernet** • PG/OP communication • S7 communication • S7 routing
Installation of SIMATIC NET is not required	Installation of SIMATIC NET is required
A list of communication interfaces that are supported by WinLC RTX can be found in "What is a communication interface?".	A list of supported communication interfaces can be found in the SIMATIC NET documentation.

Configuration for a PC-based controller

You use the Station Configuration Editor to configure components of the PC station.
To configure submodules (Page 44), you edit the properties of WinLC RTX in the Station Configuration Editor.

In the same way that you use STEP 7 to create the system and program blocks for an S7-400, you use the STEP 7 HW Config (Page 52) tool to configure the components that you installed in the PC station.

Once you have completed the hardware configuration in STEP 7 and the submodule configuration (Page 44) in the Station Configuration Editor, you can download your STEP 7 user program to the controller.

Note

To use the CP to communicate both with STEP 7 as well as with distributed I/O, you may need an additional software license. You can obtain information about this from your Siemens representative.

4.2.2 What Is a Communication Interface?

A communication interface is:

- A communication interface module, such as a CP 5611 or CP 5613 for PROFIBUS
- An Industrial Ethernet card
- An integrated PROFIBUS or PROFINET interface on a Siemens Box PC, Rack PC, or Panel PC
- Any card or service that is supported by SIMATIC NET for communication purposes

Applications for a communication interface

You can use SIMATIC NET to configure Industrial Ethernet or other communication interfaces in the PC station. You can use these communication interfaces for S7 communication, but not for communication with distributed I/O.

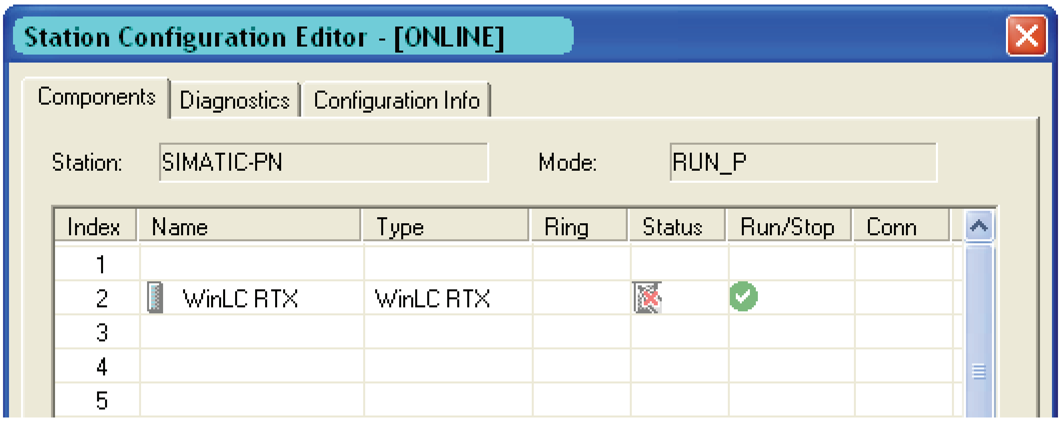
Communication interfaces enable communication between WinLC RTX and STEP 7 or other S7 applications.

4.2.3 What Is an Index?

An index is a numbered slot on the virtual rack of the PC station. The PC station (Page 35) provides slots for WinLC RTX and the SIMATIC components of a PC-based automation solution. The following list shows some (but not all) of the typical SIMATIC components that can be assigned an index:

- Communication interface modules such as CP 5611 or CP 5613 for PROFIBUS (requires installation of SIMATIC NET)
- SIMATIC HMI (requires installation of SIMATIC HMI)
- SIMATIC NET OPC server (requires installation of SIMATIC NET)

Each slot in the PC station corresponds to a number or index. When you install WinLC RTX, the setup program configures the controller in the second index slot by default.
The Station Configuration Editor shows the configuration of your PC station.



Station: SIMATIC-PN

Mode: RUN_P

Index	Name	Type	Ring	Status	Run/Stop	Conn
1						
2	WinLC RTX	WinLC RTX		☒	✓	
3						
4						
5						

You can choose any index number for a component. However, the index number in the Station Configuration Editor must be the same as the slot number in the STEP 7 HW Config tool for the same component.

Note

If you have deleted WinLC RTX in the Station Configuration Editor, there is no entry for WinLC RTX in the **Start > SIMATIC > PC Based Control** menu. To restore this option in the menu, you must configure WinLC RTX in an index in the Station Configuration Editor.

4.2.4 What Is a Submodule?

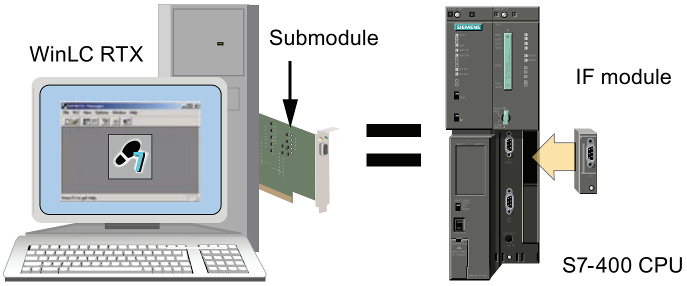
Definition

A submodule is a configured communication interface (Page 39) that enables communication between WinLC RTX and distributed I/O or between WinLC RTX and STEP 7 or other S7 applications.

Introduction

In order for WinLC RTX to communicate with distributed I/O devices in a PROFIBUS DP or PROFINET IO network, you must specify one communication interface as a submodule for the controller (Page 44). With this submodule approach, WinLC RTX has full control over the distributed I/O communications, providing optimum performance and determinism for operating the I/O. WinLC RTX supports up to four submodules configured in any of the four IF slots (Page 43).

Configuring a communication interface (Page 39) as a submodule of WinLC RTX is like installing an IF module into a slot of an S7-400 CPU.



Total number of supported submodules

If you use a CP 56x1 or an integrated CP 56x1 communication interface as a submodule of WinLC RTX, take into account that you can only plug one CP as a submodule. Likewise, you can only insert one Ethernet submodule (CP 1604, CP 1616 or IE General). However, you can configure up to four CP 56x3 communication interfaces, that is, the total number of submodules supported by WinLC RTX.

Note

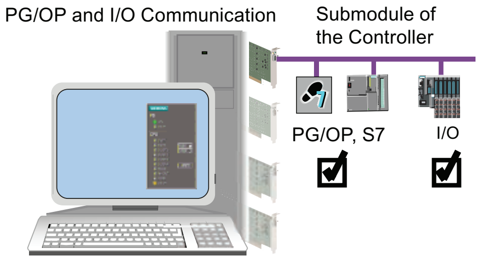
In order for a submodule to be used for SIMATIC communication with an application other than WinLC RTX (on the PC station), the second application must be a configured component of the PC station.

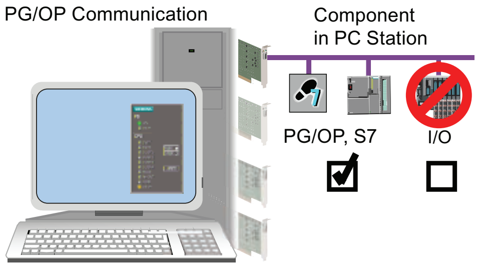
Communication interface as a component of the PC station

Configuring a communication interface as a component of the PC station requires the installation of SIMATIC NET. As a component of the PC station, you can use the communication interface only for SIMATIC communications with STEP 7, SIMATIC HMI or other SIMATIC controllers. For example, you can download a program from STEP 7 to WinLC RTX. A communication interface configured as a component of the PC station cannot be used for WinLC RTX communications with distributed I/O.

The following comparison shows the difference between a communication interface (in this case, a CP) as a submodule of WinLC RTX and as a component of the PC station:

WinLC RTX submodule communication	PC station communication
If a communication interface is configured as a submodule, WinLC RTX can communicate both with STEP 7 on a remote computer (via PG/OP communication) as well as with distributed I/O.	If a communication interface is configured as a component of the PC station, WinLC RTX can communicate with STEP 7 on a remote computer but cannot communicate with distributed I/O.





Communication interfaces that can be configured as WinLC RTX submodules

Information regarding the respective communication interfaces can be found in Chapter Communication (Page 115):

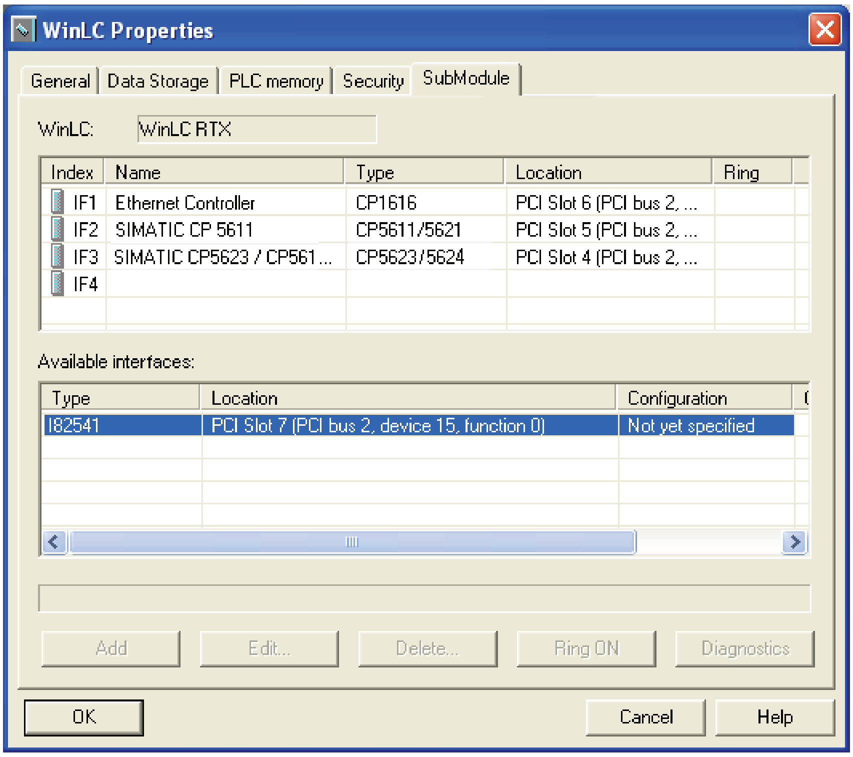
- PROFIBUS interfaces (Page 116)
- PROFINET interfaces (Page 125)

4.2.5 What is an IF slot?

Definition

WinLC RTX provides four interface slots (IF slots) for declaring communication interfaces (Page 39) as submodules (Page 41). WinLC RTX has exclusive control over any card configured in an IF slot. The submodules enable the controller to communicate with distributed I/O, or with STEP 7 or other S7 applications.

In order for WinLC RTX to communicate with I/O, you must configure at least one communication interface (Page 39) as a submodule of WinLC RTX. In the "WinLC Properties (Page 73)" dialog box, you assign a communication interface to one of the four interface slots, IF1 to IF4:



WinLC: WinLC RTX

Index	Name	Type	Location	Ring
IF1	Ethernet Controller	CP1616	PCI Slot 6 (PCI bus 2, ...	
IF2	SIMATIC CP 5611	CP5611/5621	PCI Slot 5 (PCI bus 2, ...	
IF3	SIMATIC CP5623 / CP561...	CP5623/5624	PCI Slot 4 (PCI bus 2, ...	
IF4				

Available interfaces:

Type	Location	Configuration
I82541	PCI Slot 7 (PCI bus 2, device 15, function 0)	Not yet specified

The IF slot number is independent of the PCI hardware slot. However, the IF slot number for the submodule in the WinLC properties must match the IF slot number in HW Config of STEP 7.

Information on submodule configuration can be found in: Designating a communication interface as a submodule (Page 44).

4.3 Configuring communication interfaces

4.3.1 Designating a communication interface as a submodule

Overview

Submodule communication interfaces enable WinLC RTX to communicate with distributed I/O as well as with STEP 7 or other S7 applications.

To configure a communication interface as a submodule of WinLC RTX, follow these steps:

- In the Station Configuration Editor, insert a communication interface in an IF slot of WinLC RTX.
- Configure WinLC RTX, the submodules, and all other components of the PC station in STEP 7.

Note

WinLC RTX supports a maximum of one CP 56x1 card or one integrated CP 56x1 interface as a submodule, a maximum of one PROFINET interface (CP 1616, CP 1604, or the like) as a submodule, and a maximum of four submodules in total. Of the four submodules, any number can be CP 56x3 communication interfaces. You can configure any combination of communication interfaces in the four slots as long as you observe these restrictions.

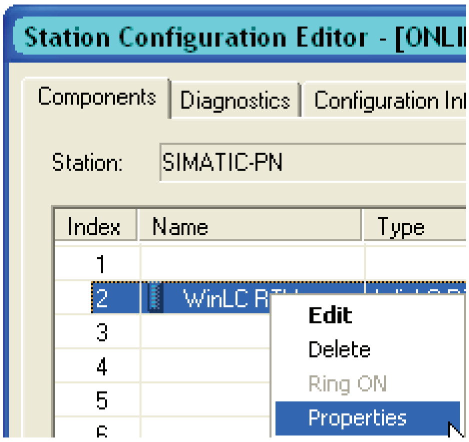
Requirements

Shut down the controller (with "Shut Down Controller").

Procedure

To configure a communication interface as a submodule (Page 41), follow these steps:

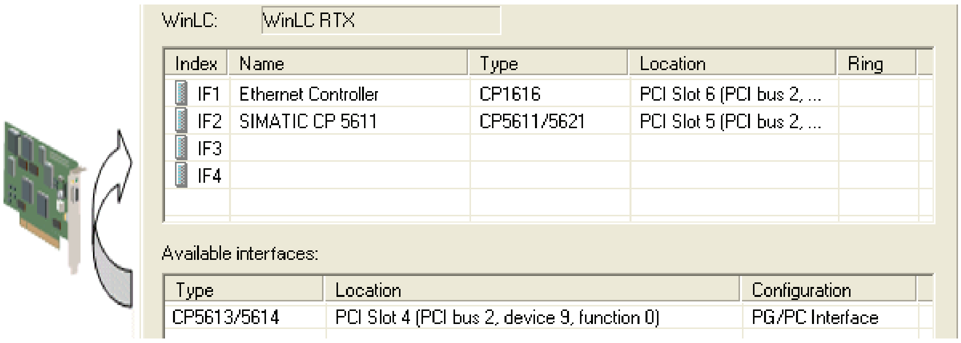
1. In the Windows taskbar, double-click the 🖥 icon to open the Station Configuration Editor.
2. Right-click WinLC RTX and in the context menu select the "Properties" command to open the "WinLC Properties" dialog box.



Station: SIMATIC-PN

Index	Name	Type
1		
2	WinLC R	
3		
4		
5		
6		

In the WinLC Properties (Page 73) dialog box, the four submodule interfaces (IF1 to IF4) are displayed in the top pane and a list of available communication interfaces is displayed in the bottom pane. The following example shows a CP 1616 and a CP 56x1 that are already configured in IF slots 1 and 2, and a CP 56x3 that is added as a submodule in IF slot 3.



WinLC: WinLC RTX

Index	Name	Type	Location	Ring
IF1	Ethernet Controller	CP1616	PCI Slot 6 (PCI bus 2, ...	
IF2	SIMATIC CP 5611	CP5611/5621	PCI Slot 5 (PCI bus 2, ...	
IF3				
IF4				

Available interfaces:

Type	Location	Configuration
CP5613/5614	PCI Slot 4 (PCI bus 2, device 9, function 0)	PG/PC Interface

3. In the bottom pane, select the communication interface that you want to configure as a submodule. (An integrated PROFIBUS or PROFINET interface is represented as a CP with the "System module" position.)
4. Drag the selected device to an empty interface slot (IF slot) in the upper pane or click the "Add" button in order to add the card to the first available interface slot. If there is more than one card, repeat the steps listed above as required.
5. Select an occupied IF slot and click the "Edit" button to change the assignment of the IF slot for a configured communication interface.
6. Use the up/down arrow keys on the keyboard to move a submodule to a different IF slot.
7. In the "WinLC Properties" dialog box, select "OK" to accept your changes and to configure the submodules. This configuration can take a few seconds.

Result

You have configured a submodule as a communication interface.

WinLC RTX response to submodule changes

WinLC RTX can detect if a configured submodule is no longer accessible, for example, if it has been physically removed from the PC or has failed.

In the predecessor versions of WinAC RTX 2005 SP2, WinLC RTX deleted the STEP 7 user program and the configuration in this case. Starting with WinAC RTX 2005 SP2, WinLC RTX signals the detected change. The diagnostic buffer contains an error "STOP due to I/O error", which indicates the deleted or failed submodule.

If you then want to access the WinLC properties, you will be informed by WinLC RTX that a submodule is no longer accessible. You will be prompted to confirm removal of the submodule. If you choose "OK", WinLC RTX deletes the submodule in the WinLC properties. If you select "Cancel", WinLC RTX keeps the submodule configuration and retains the current STEP 7 user program as well as the configuration.

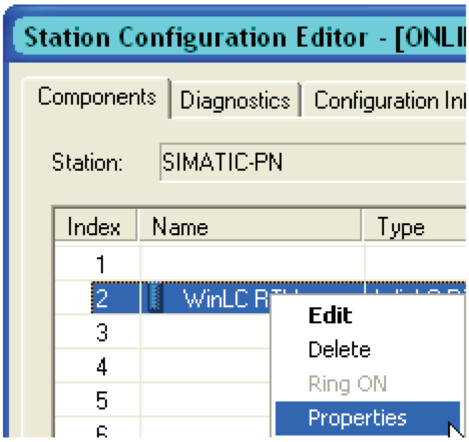
4.3.2 Removing a communication interface as a submodule

In the WinLC Properties (Page 73) dialog box, you can move a communication interface configured as a submodule of WinLC RTX to the list of available cards on your computer.

Procedure

To remove a communication interface from the WinLC RTX submodule configuration, shut down the controller and follow these steps:

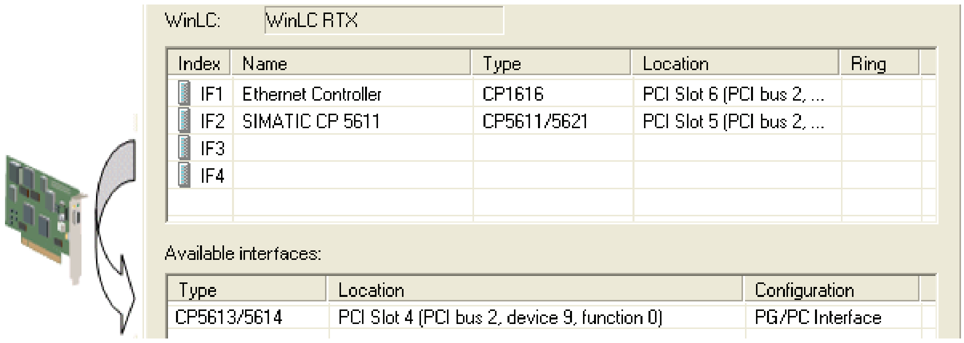
1. In the Windows taskbar, double-click the icon to open the Station Configuration Editor.
2. Right-click WinLC RTX and in the context menu select the "Properties" command to open the WinLC Properties (Page 73) dialog box.



Station: SIMATIC-PN

Index	Name	Type
1		
2	WinLC R…	
3		
4		
5		
6		

In the "WinLC Properties" dialog box, the four submodule interfaces (IF1 to IF4) are displayed in the top pane and a list of available communications cards is displayed in the bottom pane. The following example shows how to remove a CP 5613 or CP 5614 as a submodule. One CP 1616 and one CP 56x1 remain in the IF slots 1 and 2.



WinLC: WinLC RTX

Index	Name	Type	Location	Ring
IF1	Ethernet Controller	CP1616	PCI Slot 6 (PCI bus 2, ...	
IF2	SIMATIC CP 5611	CP5611/5621	PCI Slot 5 (PCI bus 2, ...	
IF3				
IF4				

Available interfaces:

Type	Location	Configuration
CP5613/5614	PCI Slot 4 (PCI bus 2, device 9, function 0)	PG/PC Interface

3. In the top pane, select the communication interface that you want to remove as a submodule.
4. Drag the selected card to an available position in the bottom pane, or right-click the selected card and select "Delete".
   The card is removed as a submodule and displayed again in the list of available cards.
5. In the "WinLC Properties" dialog box, click "OK".
6. Restart the computer.

Result

WinLC RTX removes the communication interface from the configured submodules and displays it again in the list of available cards.

The new configuration is active after the next restart of the computer.

Note

Changing the configuration of a communication interface

Once you have removed a communication interface from an IF slot of WinLC RTX or from a slot of the PC station, you must restart the computer. The communication interface is then available for a new configuration.

4.4 Configuring the controller in STEP 7

4.4.1 Connecting STEP 7 to the controller

You must establish a connection between STEP 7 and the controller in order to load the configuration and blocks of the STEP 7 user program. This type of communication is called PG/OP communication. The controller can be connected with STEP 7 via one of the following interfaces:

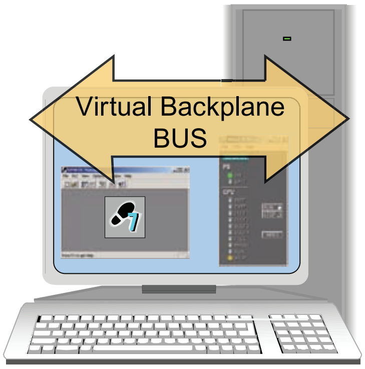
- Virtual backplane bus with STEP 7 on the same computer as the controller
- Submodule (Page 41)communication interface (Page 39) with STEP 7 on a different computer
- Communication interface of a PC station (Page 35) with STEP 7 on a different computer

Requirements for both options

Configuring a communication interface as a PC station and not as a submodule requires the installation of SIMATIC NET, an additional software package.

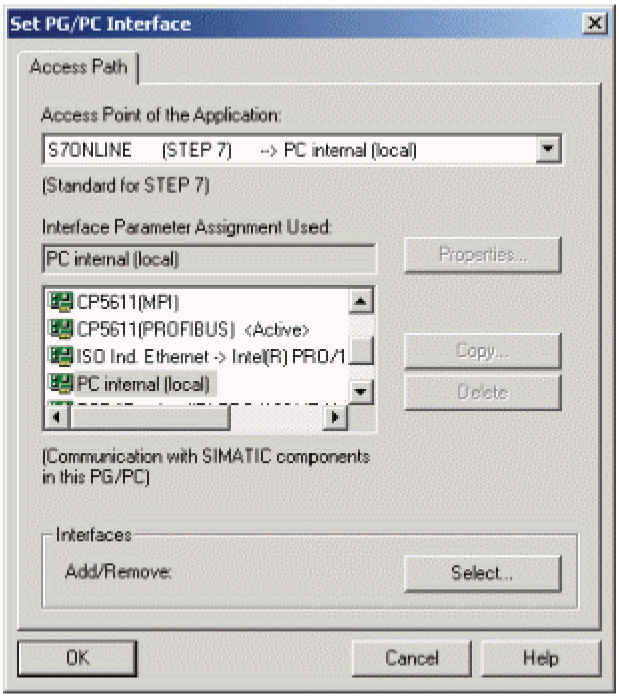
Option 1: Connecting STEP 7 to the controller on the same computer

On the same computer, STEP 7 and the controller communicate across the virtual backplane bus:



To configure the communication between the controller and STEP 7 on the same computer, follow these steps:

1. Open the "Set PG/PC interface" dialog box.
2. Select the access point "PC internal":



Set PG/PC Interface

Access Path

Access Point of the Application:

S7ONLINE (STEP 7) --> PC internal (local)

(Standard for STEP 7)

Interface Parameter Assignment Used:

PC internal (local)

- CP5611(MPI)
- CP5611(PROFIBUS) <Active>
- ISO Ind. Ethernet -> Intel(R) PRO/1
- PC internal (local)

(Communication with SIMATIC components in this PG/PC)

Interfaces

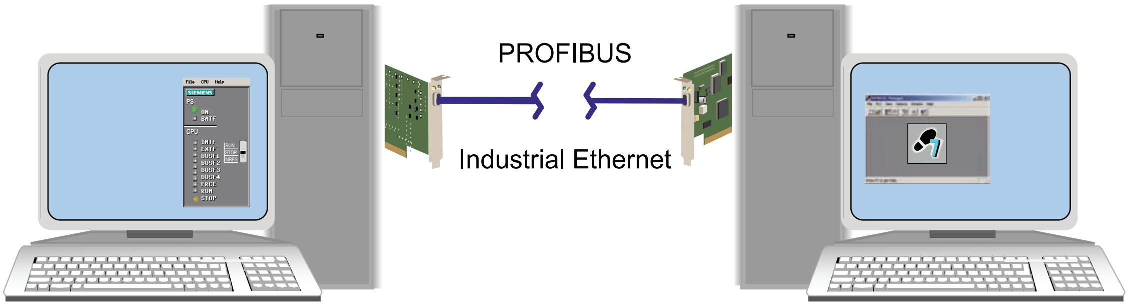
Add/Remove:

Option 2: Connecting STEP 7 to the controller on a different computer

STEP 7 can communicate with WinAC RTX on a different computer or programming device via a communication interface that is configured as a submodule of the controller or via a communication interface that is configured in the PC station.

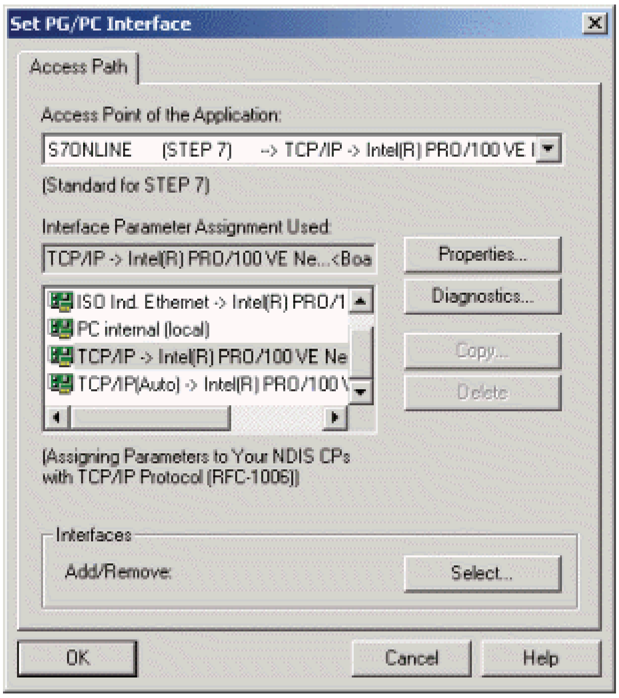
The following communication types are supported:

- PROFIBUS: for CP 56x3, CP 56x1, or integrated PROFIBUS interface, configured as a submodule
- PROFINET: for CP 1616, CP 1604, IE General or integrated PROFINET interface, configured as a submodule
- Industrial Ethernet: for an IE card configured in the PC station



To configure the communication between the controller and STEP 7 on a different computer or programming device, follow these steps:

1. Open the "Set PG/PC interface" dialog box.
2. Set the access point for the specific communication interface and the type of communication for the programming device / PC interface, for example an Industrial Ethernet card with TCP/IP protocol:



Set PG/PC Interface

Access Path

Access Point of the Application:
S7ONLINE (STEP 7) --> TCP/IP -> Intel(R) PRO/100 VE I
(Standard for STEP 7)

Interface Parameter Assignment Used:
TCP/IP -> Intel(R) PRO/100 VE Ne...<Boa

- ISO Ind. Ethernet -> Intel(R) PRO/1
- PC internal (local)
- TCP/IP -> Intel(R) PRO/100 VE Ne
- TCP/IP(Auto) -> Intel(R) PRO/100 V

(Assigning Parameters to Your NDIS CPs with TCP/IP Protocol (RFC-1006))

Interfaces
Add/Remove:

Alternative connection method (Option 2)

This connection method is an alternative for connecting STEP 7 to a decentralized WinAC RTX via a PROFIBUS, PROFINET, or Industrial Ethernet interface as described previously. You can connect STEP 7 to WinAC RTX on a different computer via the "PC Internal" interface if the following conditions are met:

- A local installation of WinAC RTX is installed on the computer on which STEP 7 is installed.
- The computer with STEP 7 and a local installation of WinAC RTX are connected to a network to which WinAC RTX is connected on the other computer.

If these conditions are met, you can change the S7ONLINE access point to "PC Internal" on the computer with STEP 7 and establish a connection to the decentralized WinAC RTX via the "PC Internal" interface.

4.4.2 Configuring the Hardware in STEP 7

You configure the STEP 7 project for a PC station (Page 35) with a PC-based controller in STEP 7 in exactly the same way as for an S7 hardware controller. Detailed information on this subject can be found in the help and in the STEP 7 documentation.

Creating a project and PC station with the SIMATIC Manager

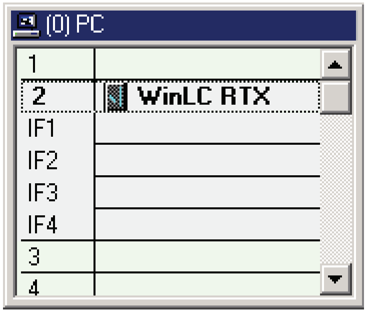
To create a project and PC station, follow these steps:

1. Select the **File > New** menu command in SIMATIC Manager to create a new project.
2. To insert a PC station in the project, select the **Insert > Station > SIMATIC PC Station** menu command.
3. Change the name (Page 265) of the PC station such that it corresponds to the name of the PC station that was configured in the Station Configuration Editor on the computer on which WinLC RTX is located. If you want to search for the station name, open the Station Configuration Editor and click the "Station Name" button.

Working with the STEP 7 hardware configuration

To configure the PC-based controller for the PC station, follow these steps:

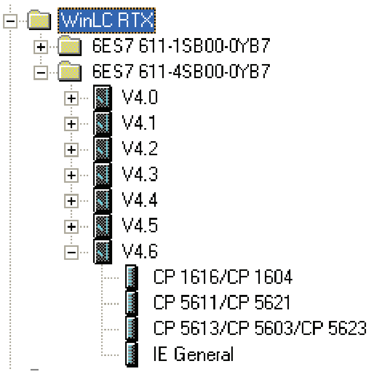
1. Open the PC station folder in the project and double-click the symbol for the configuration to open STEP 7 HW Config.
2. Navigate to your specific controller under the SIMATIC PC Station.
3. Drag the controller to the same index it occupies in the Station Configuration Editor on the target computer.



(0) PC

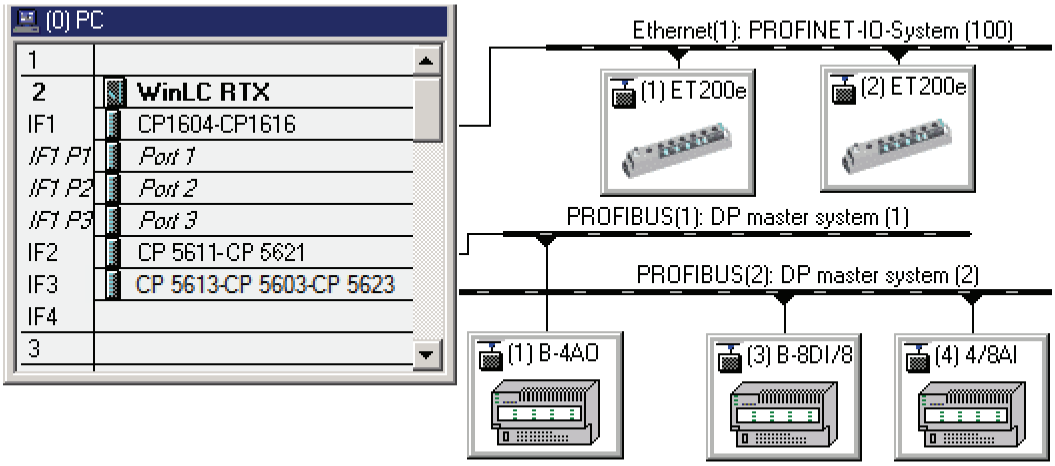
1	
2	WinLC RTX
IF1	
IF2	
IF3	
IF4	
3	
4	

4. Verify that the name of the controller matches the name of the controller configured in the Station Configuration Editor.
5. Drag the communication interfaces of the submodule from the hardware catalog to the interface slots (IF slots (Page 43)) of the WinLC RTX controller. The WinLC RTX folder in the hardware catalog lists the available selection. (Select a CP 56x1 for integrated PROFIBUS interfaces on SIEMENS PCs.)
   The submodule cards do not have to have the same name as in the PC station configuration, but it is recommended to assign the same name. They must have the same type and the same interface number (IF) as in the Station Configuration Editor.
   If you use a CP 56x1 or the integrated CP 56x1 PROFIBUS interface as a submodule of WinLC, take into account that you can only plug one CP as a submodule. Likewise, you can only insert one CP 1604, CP 1616, or Industrial Ethernet interface as a submodule.



```
WinLC RTX
  6ES7 611-1SB00-0YB7
  6ES7 611-4SB00-0YB7
    V4.0
    V4.1
    V4.2
    V4.3
    V4.4
    V4.5
    V4.6
      CP 1616/CP 1604
      CP 5611/CP 5621
      CP 5613/CP 5603/CP 5623
      IE General
```

6. Configure the distributed I/O for each submodule network:



(0) PC

1	
2	WinLC RTX
IF1	CP1604-CP1616
IF1 P1	*Port 1*
IF1 P2	*Port 2*
IF1 P3	*Port 3*
IF2	CP 5611-CP 5621
IF3	CP 5613-CP 5603-CP 5623
IF4	
3	

Ethernet(1): PROFINET-IO-System (100)
(1) ET200e
(2) ET200e
PROFIBUS(1): DP master system (1)
PROFIBUS(2): DP master system (2)
(1) B-4AO
(3) B-8DI/8
(4) 4/8AI

Note

Standard Ethernet network cards must be configured in the STEP 7 hardware configuration with IE General. In the Station Configuration Editor, they are displayed with their standard name, such as "Intel PRO/1000 GT (PCI)".

Further options in the hardware configuration

The following actions are optional and depend on the specific application:

1. Insert the CPs that are required for your application into the PC station.
2. Insert any HMI devices, for example, text displays or operator panels.
3. Configure WinLC RTX for point-to-point communication:
   - In SIMATIC Manager, select the name of the controller.
   - In the right pane, double-click the symbol for connections.
   - Configure the network with NetPro.

Note

SIMATIC NET must be installed in order to configure communication interfaces in the PC station with the Station Configuration Editor. You do not need SIMATIC NET to configure the CPs as submodules of WinLC RTX.

Result

After you have configured WinLC RTX in STEP 7, you can develop and load your STEP 7 user program in SIMATIC Manager.

⚠ **CAUTION**

If you load a STEP 7 user program that is too large for the PC memory, the PC may crash or WinLC operation may be unstable such that devices may be damaged and/or persons injured.

For STEP 7 and WinLC, there is no limit regarding the number of blocks or the size of the STEP 7 user program, but your computer has a limit that is based on the available memory space on the drive and the amount of available RAM. The maximum size of the STEP 7 user program and the maximum number of blocks for your PC can only be determined by testing a configured system to the requirements of your controller application.

After you have loaded your program to the controller, you can start the controller and monitor and change the process variables with STEP 7.

4.5 Verifying the Configuration

A complete configuration of a PC-based automation project consists of a configuration in the Station Configuration Editor and WinLC properties (Page 44) conforming to the configuration in STEP 7 (Page 52).

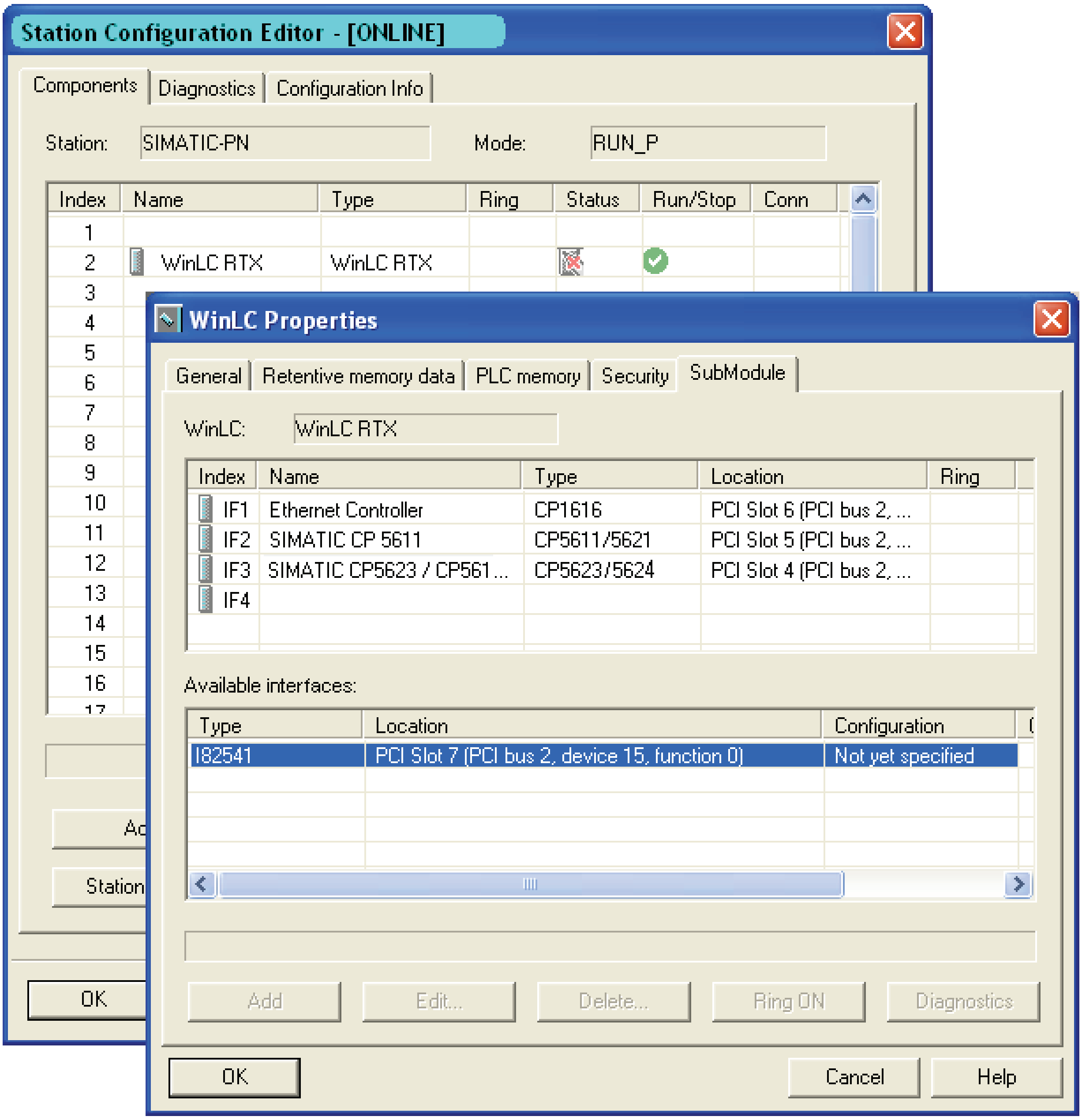
Example of a PC station configuration

The following configuration is an example of a PC-based automation project:

- WinLC RTX Controller in Index 2 of the PC station
- CP 1616 configured as a WinLC RTX submodule in IF slot 1, connected to PROFINET IO
- CP 561x1 configured as a WinLC RTX submodule in IF slot 2, connected to PROFIBUS DP I/O
- CP 56x3 or CP 56x4 configured as a WinLC RTX submodule in IF slot 3, connected to PROFIBUS DP
- STEP 7 on the same computer as WinLC RTX

Station Configuration Editor and WinLC properties

The project configuration is displayed in the Station Configuration Editor and in the "WinLC Properties" dialog box:



Station Configuration Editor - [ONLINE]

Station: SIMATIC-PN

Mode: RUN_P

Index	Name	Type	Ring	Status	Run/Stop	Conn
1						
2	WinLC RTX	WinLC RTX		✗	✓	
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

WinLC Properties

WinLC: WinLC RTX

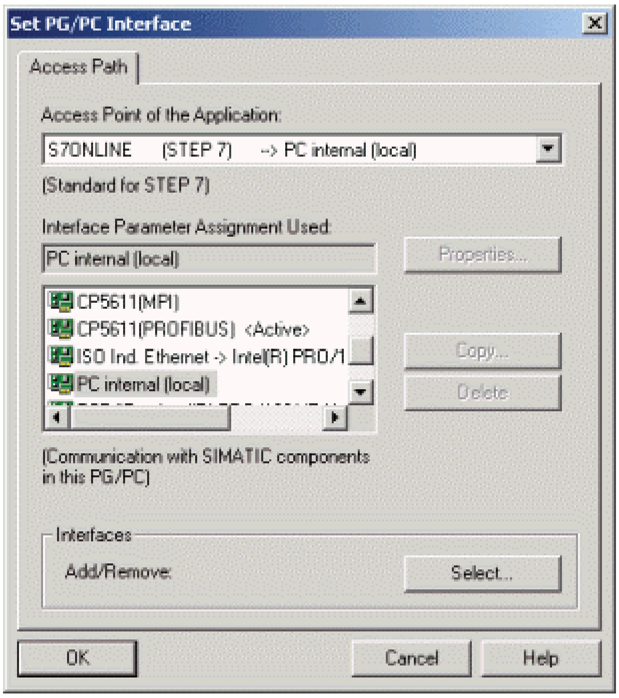
Index	Name	Type	Location	Ring
IF1	Ethernet Controller	CP1616	PCI Slot 6 (PCI bus 2, ...	
IF2	SIMATIC CP 5611	CP5611/5621	PCI Slot 5 (PCI bus 2, ...	
IF3	SIMATIC CP5623 / CP561...	CP5623/5624	PCI Slot 4 (PCI bus 2, ...	
IF4				

Available interfaces:

Type	Location	Configuration
I82541	PCI Slot 7 (PCI bus 2, device 15, function 0)	Not yet specified

STEP 7 PG/PC interface

The STEP 7 PG/PC interface shows the access point "PC internal":



Set PG/PC Interface

Access Point of the Application:

S7ONLINE (STEP 7) --> PC internal (local)

(Standard for STEP 7)

Interface Parameter Assignment Used:

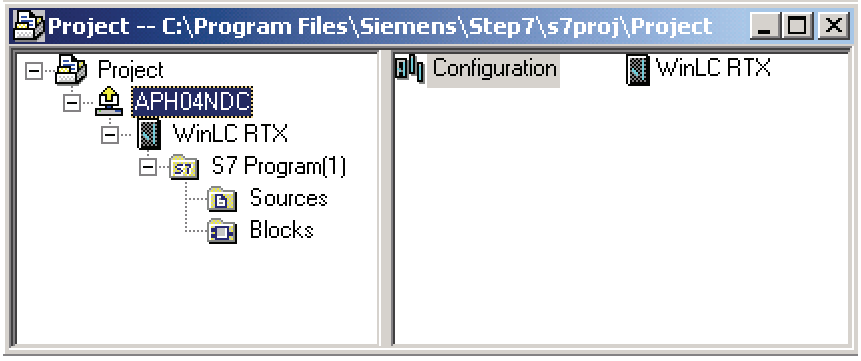
PC internal (local)

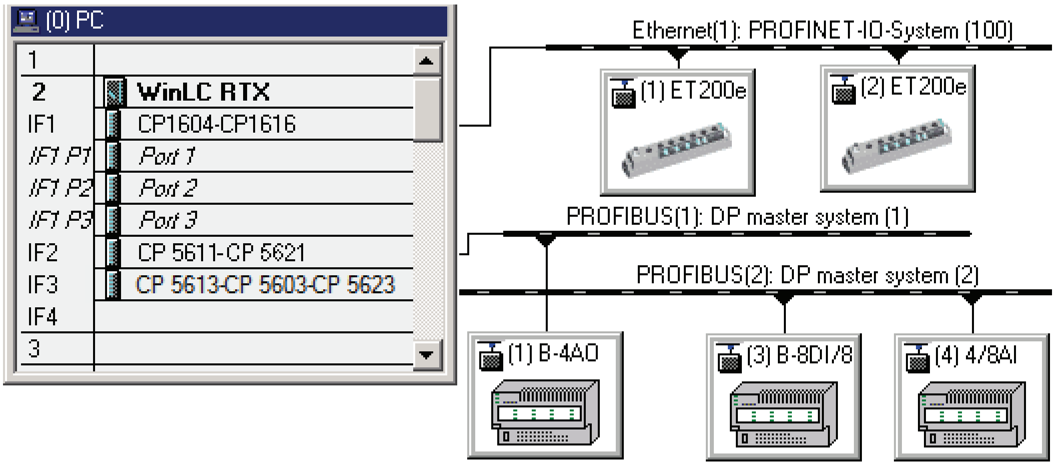
- CP5611(MPI)
- CP5611(PROFIBUS) <Active>
- ISO Ind. Ethernet -> Intel(R) PRO/1
- PC internal (local)

(Communication with SIMATIC components in this PG/PC)

STEP 7 HW Config

The STEP 7 hardware configuration shows WinLC RTX in slot 2 of the hardware configuration and the three submodules of WinLC RTX with I/O:




(0) PC
Ethernet(1): PROFINET-IO-System (100)
(1) ET200e
(2) ET200e
PROFIBUS(1): DP master system (1)
PROFIBUS(2): DP master system (2)
(1) B-4AO
(3) B-8DI/8
(4) 4/8AI

1	
2	**WinLC RTX**
IF1	CP1604-CP1616
IF1 P1	*Port 1*
IF1 P2	*Port 2*
IF1 P3	*Port 3*
IF2	CP 5611-CP 5621
IF3	CP 5613-CP 5603-CP 5623
IF4	
3	

Operating the Controller 5

5.1 Starting and closing the controller

The controller operates independently from the controller panel. The controller panel is the visible interface to the controller. The controller can be operating or shut down.
This is independent of whether or not the panel is displayed.

- Opening the panel starts the controller if it is not already in operation.
- Closing the panel (menu command **File > Exit**) does **not** shut down the controller.
- Shutting down the controller does **not** close the panel.

When the controller is in operation, your Windows taskbar displays a WinLC RTX icon regardless of whether the controller panel is open. The icon will be surrounded by a yellow border when the controller is in STOP mode, and by a green border when the controller is in RUN mode.

The following settings affect the starting or shutting down of the controller:

- Selecting the Autostart function (Page 63)
- Configuring the controller for start at PC boot (Page 63)

Starting WinLC RTX.

If the controller panel is not open, use one of the following methods to start WinLC RTX:

- Select the **Start > SIMATIC > PC Based Control** menu command. Then select the name of your WinLC controller. (After you have downloaded the STEP 7 user program to WinLC, the name in the menu matches the name in STEP 7.)
- Double-click the desktop icon for WinLC RTX:

Note

If the WinLC RTX menu command or icon is missing, then WinLC RTX has been deleted from the Station Configuration Editor. If this is the case, insert WinLC RTX into Slot 2 of the Station Configuration Editor. The menu command and desktop icon will then be available.

If the controller panel is open, but the controller is shut down, select the **CPU > Start Controller** menu command.

Result: The WinLC RTX controller starts.

Note

When the controller is in operation and the controller panel is closed, you can double-click the icon to open the controller panel.

Shutting down WinLC RTX

To shut down the WinLC controller, select the **CPU > Shut Down Controller** menu command. This action does not close the controller panel. This command is only available in the controller panel when the controller is in operation. After you shut down the controller, you can still change the WinLC properties (Page 62).

5.2 Changing the Operating Mode of the Controller

The controller panel (Page 17) provides a mode selector that allows you to change the operating mode of the controller. The mode selector positions on the controller panel correspond to the mode selector positions of an S7 hardware controller:

- RUN: The controller executes the STEP 7 user program.
- STOP: The controller does not execute the STEP 7 user program. Outputs are set to a "safe" state.

Specific controller actions are allowed or prohibited based on the operating mode.

Procedure

In order to change the operating mode of the controller, use one of the following methods:

- Switch the mode selector to the RUN or STOP position.
- Select the **CPU > RUN** or **CPU > STOP** menu command.

Operating mode (RUN/STOP) and status indicators

The mode selector on the controller panel functions like the manual mode selector on an S7 hardware controller. For both hardware controllers and PC-based controllers, the RUN and STOP status indicators (Page 77) show the current operating mode of the controller.

If the status indicator shows a different operating mode than the mode selector position, this can be caused by one of the following:

- The controller may have changed operating mode due to an error in the program
- You have changed the operating mode in STEP 7

Note

If you used the STOP switch on the controller panel to switch the controller to STOP mode, you cannot change WinLC RTX to RUN mode in STEP 7.

Memory reset in RUN mode

The controller automatically goes to STOP mode when you reset the memory from the controller panel. To perform a memory reset from STEP 7, you must first switch the controller to STOP mode.

Allowed and prohibited actions

The operating mode allows or prohibits access to the controller for some types of operations as shown in the following table:

Operating mode	Description
RUN	Allowed: • Uploading a program from the controller to your computer • Downloading a program to the controller • Downloading individual blocks to the controller • Using STEP 7 to modify program variables and changing the operating mode of the controller • Performing a memory reset from either the controller panel or STEP 7
	Not allowed: • Archiving and restoring a STEP 7 user program
STOP	Allowed: • Uploading a program from the controller to your computer or programming device • Downloading a program or individual blocks to the controller • Using STEP 7 to modify program variables • Performing a memory reset from either the controller panel or STEP 7 • Archiving and restoring a STEP 7 user program
	Not allowed: • Using STEP 7 to change the operating mode to RUN, if you previously used the STOP selector on the controller panel to switch WinLC RTX to STOP mode

5.3 Setting the properties

5.3.1 Setting properties

"WinLC Properties" dialog

You enter settings in the "WinLC Properties" dialog that affect the controller panel or that directly affect the controller.

Basic procedure

The following methods are available to open the "WinLC Properties" dialog:

Option 1

1. Open the Station Configuration Editor.
2. Click the controller in index 2.
3. Select the **Properties** command in the context menu.

   The dialog opens. The "Submodule" tab is active.

Option 2

1. Open the controller panel.
2. Select the **CPU > Properties** menu command.

   The dialog opens. The "General" tab is active.

Possible settings

The "WinLC Properties" dialog provides the following tabs containing the following possible settings:

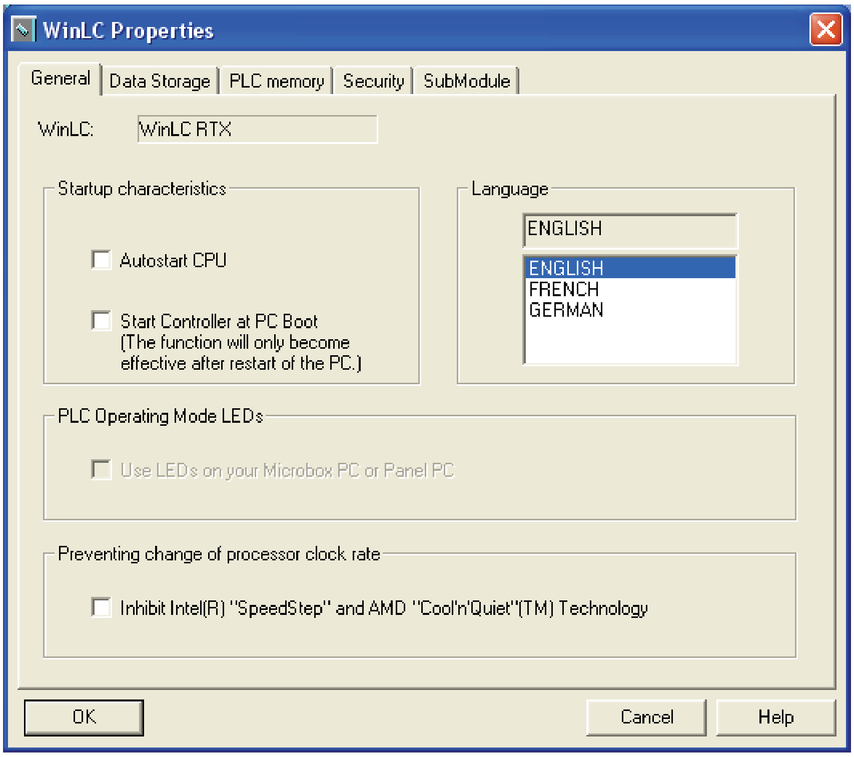
Tab	Setting
"General" Tab (Page 63)	Startup characteristics Language Use LEDs on your PCs Preventing change of processor clock rate
"Data Storage" tab (Page 66)	Storage location and behavior
"PLC Memory" tab (Page 69)	Management of STEP 7 project size
"Security" tab (Page 70)	Access protection options
"SubModule" tab (Page 73)	Configuring submodules

5.3.2 "General" Tab

Introduction

You can make the following settings in the "General" tab:

- Startup characteristics
- Language
- PLC Operating Mode LEDs
- Preventing change of processor clock rate



WinLC Properties

General | Data Storage | PLC memory | Security | SubModule

WinLC: WinLC RTX

Startup characteristics

□ Autostart CPU

□ Start Controller at PC Boot (The function will only become effective after restart of the PC.)

Language

ENGLISH

ENGLISH
FRENCH
GERMAN

PLC Operating Mode LEDs

□ Use LEDs on your Microbox PC or Panel PC

Preventing change of processor clock rate

□ Inhibit Intel(R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology

Startup characteristics

By default, you must start the controller manually after the computer boots. However, you can register the controller to start automatically during the Windows boot sequence prior to user login.

Autostart CPU

The "Autostart CPU" option starts the controller in the same operating mode as before the shutdown. Following a blue screen, the controller starts in the operating mode indicated by the mode selector before the shutdown.

Start controller at PC boot

If only the "Start controller at PC boot" option is selected, the controller starts up in STOP mode when the PC boots. The activation or deactivation of this option comes into effect only after a restart of the PC.

Note

Both options can be selected at the same time. This means the controller will start up in the RUN operating state, if the "Start controller at PC start" option is activated.

Language

The "Language" field displays the current language setting for the controller panel display.

The selection list displays all the languages that have been installed for the controller panel. Select the language in which the controller panel is to be displayed.

Note

In order to install the languages for the Controller Panel run the Setup program and select the languages.

PLC Operating Mode LEDs

Through the selection of the operating state LEDs of the PLC you can use the LEDs at a Microbox PC 427B/SIMATIC IPC427C or a Panel PC 477B to display the operating state. This option is only available for these PCs. WinLC RTX will light the LEDs to display RUN/STOP mode or a potential fault condition if you select this option and restart your computer. Status indicators (Page 77) describes the LED status displays.

Preventing change of processor clock rate (power saving function)

⚠ CAUTION

A difference in the power saving option can result in faulty operation of the controlled application. This can lead to physical injuries and/or property damage.

Set the power saving function as described above.

Power saving functions such as "SpeedStep" (computers with Intel processors) and "Cool'n'Quiet" (computers with AMD processors) save power by dynamically changing the clock rate of the processor. The clock rate is reduced at lower processor utilization ("throttling") and increased again as the utilization increases. However, the dynamic change in the clock rate can cause long latencies that adversely affect the performance of the RTX real-time extension.

The "General" tab of the "WinAC Properties" dialog contains an option (**Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology**) that is used to disable changing of the processor clock rate.

On SIMATIC PCs the power saving function of the processor is deactivated by default in the BIOS. The option **Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology** can therefore remain deactivated.

The power saving function may be activated in the BIOS on other computers. In this case the option **Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology** should be activated manually.

The option **Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology** is not activated by default. The changing of the processor cycle clock is therefore not disabled by default. If you do not use a SIMATIC PC, check whether the power saving function is activated in the BIOS. If this is the case, the option **Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology** has to be activated.

WinAC RTX is configured so that a constantly high processor load of 100% is displayed in the Tuning Panel and Task Manager by manually disabling the power saving function. The processor utilization is only simulated artificially. It does not affect the performance of other applications on the PC.

Changes to WinLC RTX options take effect only after a restart of WinLC RTX.

5.3.3 "Data Storage" tab

You can configure the following in the "Data Storage" tab:

- A path for storing the STEP 7 user program and the configuration data
- A path for storing retentive data (Page 89)

If NVRAM (Page 90) is available, you can specify whether the retentive data are to be stored on the hard disk or in NVRAM.

Note

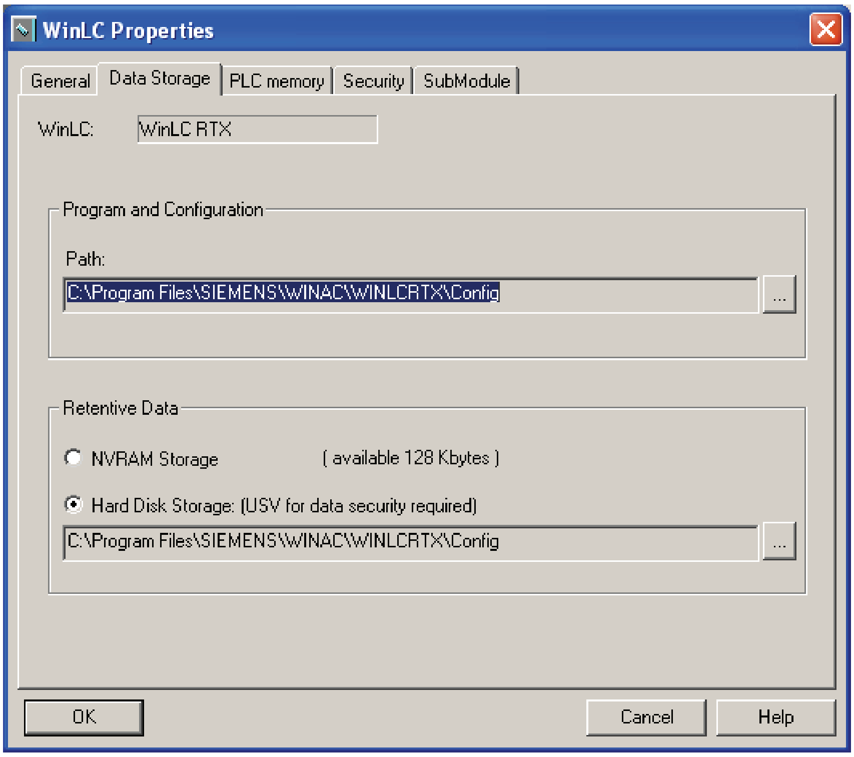
Do not use a swappable memory medium

For storing data on the hard disk, do not use a memory medium that can be swapped during operation, such as a USB stick, MMC, or external hard disk.

Configuring memory options

To configure the options for data storage, follow these steps:

1. Archive (Page 85) your STEP 7 user program and the configuration.
2. Select the **CPU > Properties** menu command.
3. Open the "Data Storage" tab:



WinLC: WinLC RTX

Program and Configuration

Path: C:\Program Files\SIEMENS\WINAC\WINLCRTX\Config

Retentive Data

○ NVRAM Storage (available 128 Kbytes)

● Hard Disk Storage: (USV for data security required)

C:\Program Files\SIEMENS\WINAC\WINLCRTX\Config

4. Accept the default path in the "Program and Configuration" field or use the ... button to find the folder where you want to store the user program and the STEP 7 configuration.
5. In the "Retentive Data" field, accept the default path or use the ... button to find the folder where you want to store the retentive data.
6. Choose between "NVRAM Storage" or "Hard Disk Storage" for the retentive data, provided your computer has NVRAM memory. The dialog displays the amount of available NVRAM memory. Otherwise, only "Hard Disk Storage" can be selected.
7. Confirm with "OK".

   After a change, the button changes to "Confirm and restart controller".
8. Restore the STEP 7 user program and the configuration you archived in step 1.

Refer to "Available options for WinLC data storage (Page 88)" for an overview of the memory options for the supported PC systems.

In NVRAM storage (Page 90), you will find the PCs that have an NVRAM. In rare cases, the "Data Storage" tab will display NVRAM as available even though it is not.
Verify the availability of NVRAM on your particular computer before you select NVRAM for the data storage.

NOTICE

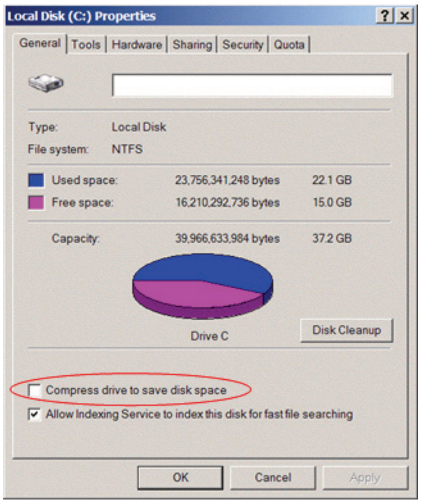
"Hard Disk Storage" option only with use of a UPS

If you choose "Hard Disk Storage" for retentive data, you must use an uninterruptible power supply (UPS) to save data following a power failure. Without a UPS, retentive data in "Hard Disk Storage" are lost after a power failure. If you choose "NVRAM Storage", the use of a UPS is strongly recommended for the following situations:

- Other applications such as databases and archives must also save data in the event of a power failure.
- Operation of Windows XP or Windows XP Embedded Standard with Enhanced Write Filter disabled. There is a risk that the operating system will be destroyed.

No compressed file system for file storage

Do not specify a compressed file system for the "Program and Configuration" path or the "Retentive Data" path. To determine whether or not a file system is compressed, open the system properties of the drive that you have specified. Make sure that the "Compress drive to save disk space" check box is not selected.



Type: Local Disk
File system: NTFS

🟦	Used space:	23,756,341,248 bytes	22.1 GB
🟪	Free space:	16,210,292,736 bytes	15.0 GB
	Capacity:	39,966,633,984 bytes	37.2 GB

☐ Compress drive to save disk space
☑ Allow Indexing Service to index this disk for fast file searching

Use of Enhanced Write Filters (EWF, FBWF)

For SIMATIC PCs with Windows XP Embedded Standard on CompactFlash card and for EC31-RTX, you can use an enhanced write filter, for example, Enhanced Write Filter (EWF)) or File-Based Write Filter (FBWF) in order to prevent the CompactFlash card from being used up prematurely due to frequent write accesses.

Recommendation: Specify a drive without an activated write filter for the data storage, since WinLC RTX creates the following components every time after a restart:

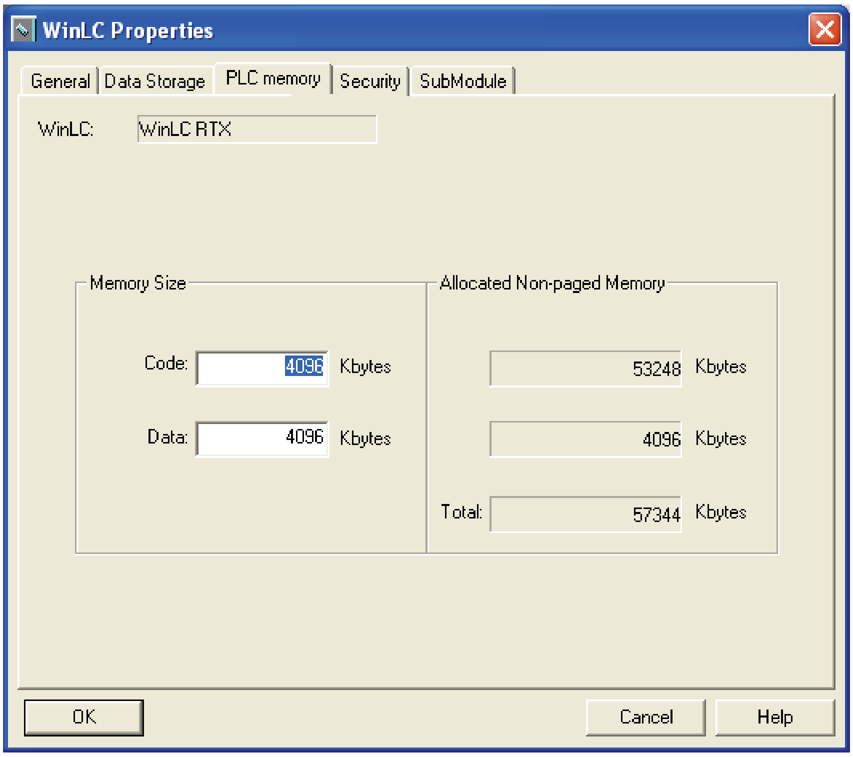
- STEP 7 user program
- STEP 7 configuration
- Retentive data

If you specify a drive for data storage that is to be protected by an enhanced write filter, you must first disable the write filter to allow the data to be stored on the CompactFlash card. This ensures that the stored data are available for WinLC RTX when it restarts.

5.3.4 "PLC Memory" tab

STEP 7 project size

In the "PLC Memory" tab, the preset size of a STEP 7 project is displayed in the "Memory Size" field. This is the recommended maximum size of a STEP 7 project that can be downloaded to WinLC.



WinLC: WinLC RTX

Memory Size

Code: 4096 Kbytes

Data: 4096 Kbytes

Allocated Non-paged Memory

53248 Kbytes

4096 Kbytes

Total: 57344 Kbytes

WinLC subdivides the overall size of a STEP 7 project into the following components:

- Size of the code (for example, OBs)
- Size of the data (for example, DBs)

Downloading a larger STEP 7 project

If you want to download a STEP 7 project in which one of the components exceeds the set size, the download will be canceled.

The recommended maximum size can be manually increased.

CAUTION

If the size of a STEP 7 project differs from the recommended default size, this can result in faulty operation of the controlled application. This can lead to physical injuries and/or property damage.

Observe the recommended default size of a STEP 7 project.

5.3.5 "Security" tab

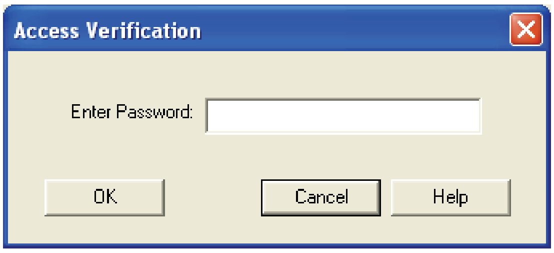
You can use the **CPU > Properties** menu command in the "Security" tab to change the options for access protection.

Basic procedure

To change the access protection options, follow these steps:

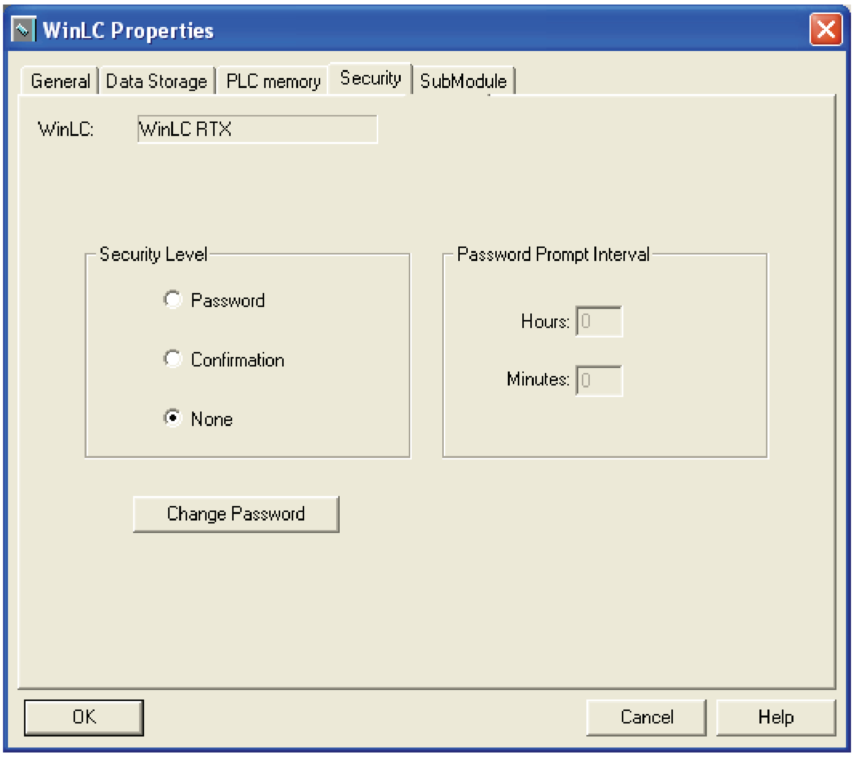
1. Click the "Security" tab.

   The "Access Verification" dialog opens.



2. Enter your password. The default password is an empty field.
3. Confirm with "OK".

   The "Security" tab is active.


WinLC: WinLC RTX

Security Level

○ Password

○ Confirmation

● None

Password Prompt Interval

Hours: 0

Minutes: 0

4. Enter the settings for the access protection of the controller.
5. Confirm with "OK".

Security level

In the "Security" tab, you can set up levels of password security that limit access to the controller. The following security access options are provided:

- Password: When you select "Password", certain controller panel operations, such as changing the operating mode and archiving and restoring a STEP 7 user program, require that the user enter a password.
- Confirmation: When you select "Confirmation", operating mode changes require that the user acknowledge a prompt for confirmation.
- None: When you select "None", no confirmation or password is required.

Password prompt interval

You can set the password prompt interval to a time interval of your choice, from 0 to a maximum of 23 hours, 59 minutes. After you have entered your password, you are not prompted for it again until this time interval has expired. The default setting of 0 means that you must enter the password for each protected operation.

Shutting down and starting the controller does not affect the expiration of the password prompt interval; however, it is reset whenever you shut down the controller panel. The next time you start the controller panel and access a password-protected operation, you will be prompted for password entry.

Change password

Click the "Change Password" button to display the "Change Password" dialog.
The "Change password" dialog allows you to change the current password.

Note

The default password is an empty field without characters. To enter the default password, press the Enter key.

Procedure

Use the following procedure to change the password:

1. In the "Old Password" field, enter the old password.
2. In the "New Password" field, enter the new password (maximum length of 12 characters).
3. In the "Confirm New Password" field, enter the new password again.
4. Confirm with "OK".

Result

You have configured the password for accessing controller panel operations such as changing the operating mode and archiving and restoring STEP 7 user programs. To subsequently access the Security options, you must enter the password in the "Access Verification" dialog box.

Operating the controller without a password

If you create a password, but set the security level to "None" (disabling the password), you still need to enter the configured password before you can access the "Security" dialog box again.

⚠ WARNING

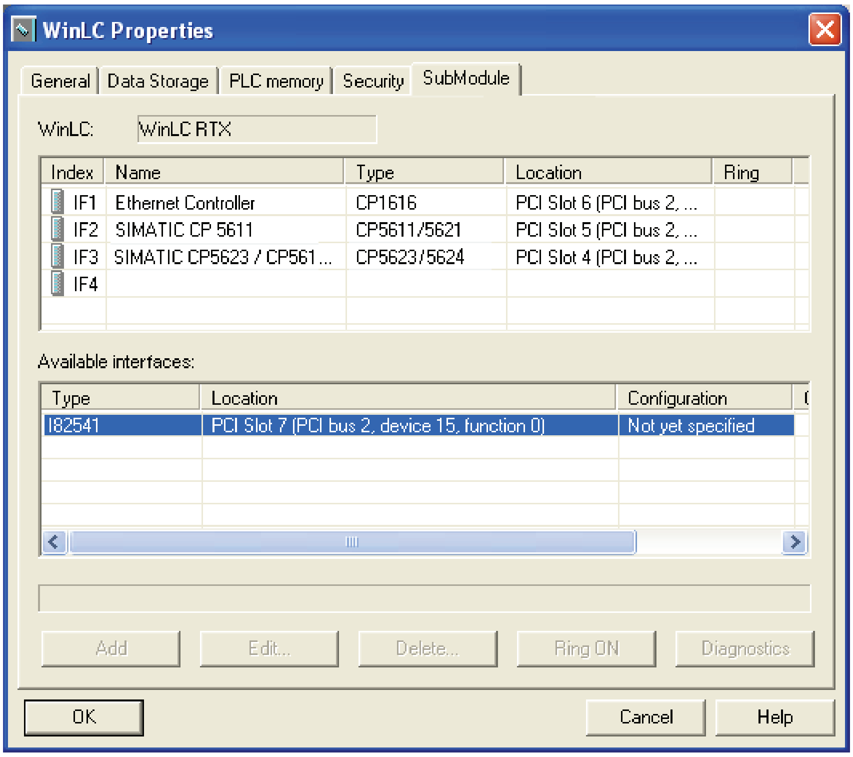
Running the controller without a confirmation or password protection increases the risk that an operator may change the controller mode inadvertently, which could cause the process or equipment to operate unpredictably, resulting in potential damage to equipment and/or death or serious injury to personnel.

Exercise caution to ensure that you do not inadvertently change the operating mode. Permit only authorized persons to access machines and processes. Install a physical emergency stop circuit for your machine or process.

5.3.6 "SubModule" tab

Introduction

The "SubModule" tab displays the current configuration of the four interface slots (Page 43) (IF slots) for WinLC RTX submodules and a list of available communication cards that can be configured as submodules. In this tab, you can add, edit, or delete communication cards as submodules of WinLC RTX. You can test the submodule assignment of a CP 56x3 and display the diagnostic data for a CP 56x1 submodule or a CP 56x3 card.



WinLC: WinLC RTX

Index	Name	Type	Location	Ring
IF1	Ethernet Controller	CP1616	PCI Slot 6 (PCI bus 2, ...	
IF2	SIMATIC CP 5611	CP5611/5621	PCI Slot 5 (PCI bus 2, ...	
IF3	SIMATIC CP5623 / CP561...	CP5623/5624	PCI Slot 4 (PCI bus 2, ...	
IF4				

Available interfaces:

Type	Location	Configuration
I82541	PCI Slot 7 (PCI bus 2, device 15, function 0)	Not yet specified

Note

WinLC RTX supports a maximum of one CP 56x1 card or one integrated CP 56x1 PROFIBUS interface as a submodule, a maximum of one Industrial Ethernet interface (CP 1616, CP 1604, or IE General) as a submodule, and a maximum of four submodules in total. Of the four submodules, any number can be CP 56x3 communication interfaces. You can configure any combination of communication interfaces in the four slots as long as you observe these restrictions.

Available functions if WinLC RTX is shut down

The following functions are available in the "Submodule" tab when WinLC RTX is closed:

- Add: You use the "Add" button to choose a communication card from the list of available cards and insert it into an empty IF slot. The communication card is configured as a submodule of WinLC RTX in the first free interface slot.
- Edit: You can select an occupied IF slot and click the "Edit" button to change the assignment of the IF slot for a configured DP interface or the name.
- Delete: You can choose any IF slot and click the "Delete" button in order to delete the configuration of a communication card as a submodule of WinLC RTX. The card is deleted from the IF slot and is then displayed as an available card in the bottom pane.
  It is now no longer a submodule of WinLC RTX. The "Del" key on the keyboard has the same functionality as the "Delete" button in this dialog.

Available functions if WinLC RTX is running

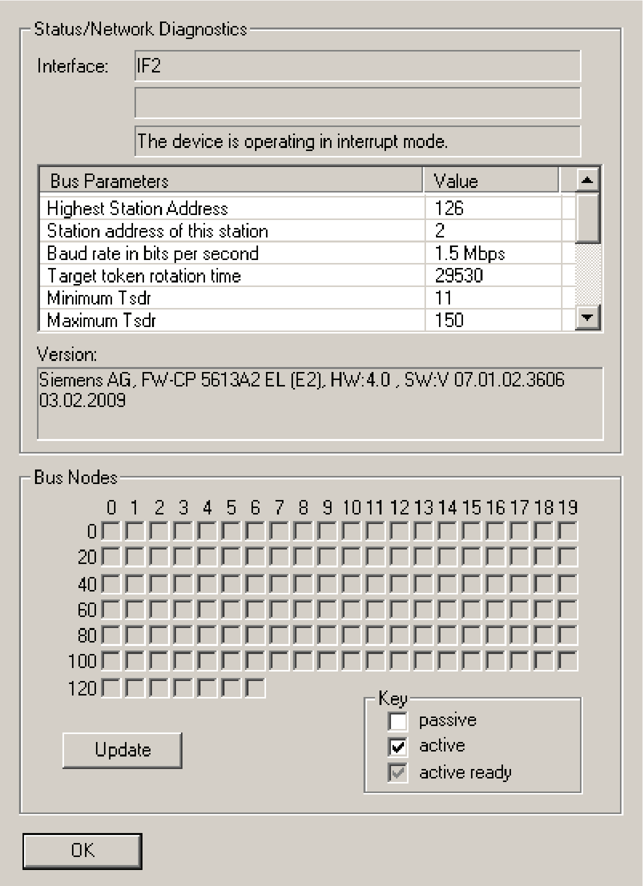
The following functions are available in the "WinLC Properties" for CP 56x3 if WinLC RTX is running: The following buttons are not available for CP 56x1 and for
Industrial Ethernet cards:

- Ring ON: You can choose any IF slot and click the "Ring ON" button in order to test the configuration of a CP 56x3 for a particular IF slot. WinLC RTX does not require that the physical PCI slot in the computer and the configured IF slot match. The Ring ON function allows you to test whether you have configured the CP 56x3 for an IF slot that corresponds to the actual PC slot. During this test with the Ring ON function, the LEDs on the configured CP 56x3 flash in an alternating pattern. These LEDs are visible on the back of the computer so you can verify that you have configured the correct CP.
  The computer also emits an audible beep if the CP is functioning. Click the "Ring OFF" button in the "WinLC Properties" dialog box to end the test.

  "Ring ON" is not available for a CP 56x1.

- **Diagnostics:** You can choose an IF slot occupied with a CP and click the "Diagnostics" button to display communication information for the CP. The "Submodule Network Diagnostics" dialog box displays the current version of the selected CP and the bus parameters.

  You can also display all of the nodes in the communication network and the status of each node. Click the "Update" button to open this display. This display is not available all of the time, because querying each node places an additional load on the network.



Status/Network Diagnostics

Interface: IF2

The device is operating in interrupt mode.

Bus Parameters	Value
Highest Station Address	126
Station address of this station	2
Baud rate in bits per second	1.5 Mbps
Target token rotation time	29530
Minimum Tsdr	11
Maximum Tsdr	150

Version:
Siemens AG, FW-CP 5613A2 EL (E2), HW:4.0 , SW:V 07.01.02.3606
03.02.2009

Bus Nodes

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
20	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
40	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
60	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
80	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
100	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
120	□	□	□	□	□	□	□													

Key
□ passive
☑ active
☑ active ready

5.4 Resetting the memory

Introduction

The memory reset functions like a master reset of the controller by resetting the controller to its initial (default) state. A memory reset deletes the STEP 7 user program and the system data (configuration), and also disconnects any online communications, for example, STEP 7, WinCC, WinCC flexible, PROFIBUS, or S7 communications.

MRES (CPU menu)

You typically execute the MRES command before downloading a new program to the controller. You **must** perform a memory reset when the STOP indicator on the controller panel is flashing slowly to alert you to one of the following conditions:

- Errors were detected in the work memory, for example, the size of the user program exceeds the work memory.
- A power cycle followed a defective state of the controller.

Procedure

Use one of the following methods to reset the memory:

- Click the MRES button on the controller panel (Page 17).
- Select the **CPU > MRES** menu command.
- Press the key combination ALT+C+M.
- Perform the memory reset in STEP 7.

The STOP indicator flashes while the memory reset is in progress.

Result

The MRES command switches the controller to STOP mode, if necessary, and then performs the following tasks:

- Deletes the entire STEP 7 user program (OBs, DBs, FCs, FBs, and the system data) from both the work memory and the load memory
- Resets the memory areas (I, O, M, T, and C) to 0
- Reloads the default system configuration (for example, the size of the process image areas and the size of the diagnostic buffer)
- Deletes all active communication jobs (for example, TIS) and all open communications

The MRES command does not affect the submodule network addresses or the content of the diagnostic buffer.

After a memory reset

After the memory has been reset, the diagnostic buffer is resized to its default size.
The memory areas of inputs (I) and outputs (O) are also resized to their default sizes.
After a memory reset, you must then reconfigure these values to your own specifications.

5.5 Status indicators

The status indicators on the controller panel display the current operating mode and are helpful in troubleshooting an error condition. These indicators correspond to the LED displays on an S7 hardware PLC.

You cannot change the status of the controller by clicking the status indicators.

Description of the status indicators

The table below describes the different status indicators on the controller panel:

Indicator	Description
ON	Power supply. Lights up (steady) when you start the controller. Turns off when you shut down the controller.
BATF	Battery fault. Always off.
INTF	This indicator lights up to show error conditions within the controller, such as programming errors, arithmetic errors, timer errors, and counter errors. If the STEP 7 user program handles the error by executing OB 80 or OB 121, the INTF indicator goes off after 3 seconds if there is no subsequent error condition.
EXTF	This indicator lights up to show error conditions that exist outside the controller, such as hardware faults, parameter assignment errors, loss of communication or other communication errors, download of a STEP 7 program with configured CP cards to a WinLC RTX with no CP cards or faulty CP cards, and I/O faults. This indicator lights up together with a BUSF indicator to show that a CP is defective. If the STEP 7 user program handles the error by executing OB 122, the EXTF indicator goes off after 3 seconds if there is no subsequent error condition.
BUSF1 BUSF2 BUSF3 BUSF4	These indicators light up (flashing) to identify an error during communication with the distributed I/O. The number of the BUSF LED corresponds to the IF number of the submodule in which the error occurred.
FRCE	This indicator is never lit. WinLC RTX does not support force commands.
MAINT	This indicator lights up if a PROFINET IO Controller or I/O device requires maintenance. For additional information, refer to the STEP 7 online help.
RUN STOP	Lights up to show the operating mode (RUN or STOP). The RUN indicator flashes and the STOP indicator is lit during a transition from STOP to RUN mode. When the STOP indicator turns off, the outputs are activated. **Note:** The RUN and STOP indicators show the actual operating mode of the controller. The RUN and STOP mode selector positions show the selected mode (similar to the mode selector switch on an S7 CPU front panel), which can differ from the current operating mode. Example: Changing the operating mode with STEP 7 causes the status indicators to change, but the mode selector does not change.

For detailed information about the behavior of the LEDs, refer to the chapter Troubleshooting Network Problems (Page 261).

Flashing indicators

Flashing patterns of the RUN and STOP indicators provide additional information about the controller or the STEP 7 user program:

RUN indicator	STOP indicator	Description
flashing 2 Hz	flashing 2 Hz	The controller is defective. All status indicators flash.
flashing 0.5 Hz	on	The STEP 7 user program has stopped at a breakpoint.
flashing 2 Hz	on	A cold or warm restart is in progress. The RUN indicator continues to flash until the restart is complete. The time required for the restart depends on the time required to execute the startup OB.
off	flashing 0.5 Hz	The controller requires a memory reset (MRES).
off	flashing 2 Hz	An memory reset (MRES) is in progress.

Corrective action if all status indicators are flashing

If all of the status indicators are flashing at the same time, the controller is in a defective state and has encountered an error condition that cannot be corrected by resetting the memory with the MRES menu command. To recover from this condition, you must perform the following steps:

1. Select the **CPU > Shut Down Controller** menu command to shut down the controller.
2. Restart the controller.

   The STOP indicator flashes and the RUN indicator is switched off.
3. Use the MRES command to reset the memory.
4. Use STEP 7 to download the STEP 7 user program and the system configuration, or to restore an archived STEP 7 user program.

Note

If either shutting down or restarting the controller does not resolve the problem, you may need to reboot your computer.

5.6 Tuning panel

Tuning panel (CPU menu)

The tuning panel is designed for adjusting the parameters and verifying the performance of WinLC. The tuning panel displays information about the scan cycle (Page 183), such as the execution time and the sleep time. By adjusting these values, you can tune the performance of the controller.

Note

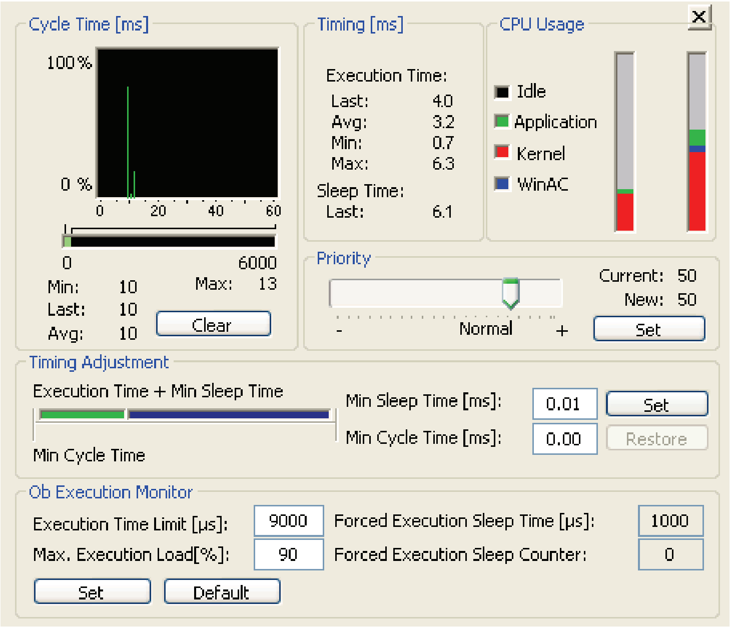
The tuning panel causes an additional load on the computer resources. Therefore, do not leave the tuning panel open during normal operation of WinLC.

Procedure

To open the tuning panel, follow these steps:

1. Select the **CPU > Tuning Panel** menu command.

WinLC RTX opens the tuning panel, as shown below.



Min: 10 Max: 13
Last: 10
Avg: 10

Execution Time:
Last: 4.0
Avg: 3.2
Min: 0.7
Max: 6.3
Sleep Time:
Last: 6.1

■ Idle
🟩 Application
🟥 Kernel
🟦 WinAC

Current: 50
New: 50

Min Sleep Time [ms]: 0.01
Min Cycle Time [ms]: 0.00

Execution Time Limit [µs]: 9000
Max. Execution Load[%]: 90
Forced Execution Sleep Time [µs]: 1000
Forced Execution Sleep Counter: 0

Functional areas in the tuning panel

The tuning panel contains the following functional areas:

Area	Description
Cycle time (Page 183)	• This area displays a histogram of execution times of the scan cycle over a 60 ms range. • This histogram tracks the minimum (shortest) and maximum (longest) scan cycle time as well as the percentage of the scan cycles that fall into various ranges of the cycle times. • "Clear" button: Clears the historical data and starts a new histogram. The cycle time display is reset by: • Operating mode transition from STOP to RUN • Closing and reopening the tuning panel
CPU Usage (Page 204)	Shows the following values: • Percentage that is used by applications, the kernel, or WinLC RTX.
Priority (Page 193)	• Sets the priority level for the execution of WinLC RTX relative to other RTX applications running on your computer Setting a higher priority means: • The operating system responds to WinLC RTX before executing other, lower-priority tasks. • Less jitter in the start times and execution times of the OBs in your program **Note:** WinLC RTX has a higher priority than all other Windows applications. The change in priority affects WinLC RTX only if other RTX applications are running.
Timing Adjustment (Page 200)	The first write-protected field shows the following information about the cycle: • **Execution Time** displays the execution time for the last (most current) scan cycle, the average cycle time, the minimum (shortest) cycle time, and the maximum (longest) cycle time. • **Sleep Time** displays the amount of sleep time for the last (most current) scan cycle. The cycle can be set in the editable fields by entering one of the following values: • Values for the minimum sleep time • Values for the minimum cycle time These parameters determine the amount of sleep time that is added at the end of the free cycle. **Note:** These values can be specified as floating point numbers with two post-decimal points. After you have confirmed your entry by clicking the "Set" button, only one post-decimal point is still displayed. **"Set" button:** applies the entered values **"Restore" button:** resets the entered values to the values currently used by the controller • Tuning panel stores newly entered values for the controller • Effects of these values on the execution of your control program can be monitored **Note:** To ensure that the minimum cycle time controls the sleep time for the controller, you must configure the cycle-time monitoring and minimum cycle time parameters in the "Cycle/Clock Memory" tab of the "Properties" dialog box in STEP 7. Set the minimum cycle time to a value less than the value for the scan cycle monitoring time. (The default cycle time is 6 seconds.)
OB Execution Monitor (Page 204)	Ensures that the controller does not exceed a configurable maximum execution load for the CPU usage within a monitor interval.

Working with the tuning panel

Values other than the minimum cycle time are unique to WinLC RTX and are not stored in the system configuration. Using the tuning panel to enter a value for the minimum cycle time does not change the configuration of the controller.

Switching the controller from STOP to RUN mode resets the minimum cycle time parameter to the value that you configured in STEP 7. To make any changes in the tuning panel permanent, you must adapt your STEP 7 configuration accordingly.

Execution time and sleep time

If the controller does not provide sufficient sleep time for the other applications to run, the computer can become unresponsive to operator input, or the controller and other applications can operate incorrectly. In addition, the execution of the STEP 7 user program can experience non-deterministic behavior (jitter) such that execution times can vary and start events can be delayed.

Always provide an EMERGENCY STOP circuit. In addition, always tune the sleep time and manage the performance of the controller so that your STEP 7 user program executes consistently.

⚠ CAUTION

Variation in the execution time or response time of the STEP 7 user program could potentially create a situation where the application being controlled can operate erratically and possibly cause damage to equipment or injury to personnel.

Ensure sufficient sleep time in your STEP 7 user program.

Reference

Additional information regarding the scan cycle, jitter, controller priority, and sleep time can be found in the Chapter Tuning the controller performance (Page 183).

5.7 Using the Diagnostic Buffer

Diagnostic buffer (CPU menu)

The diagnostic buffer allows you to view system diagnostic information without using the SIMATIC STEP 7 programming software.

Procedure

To display the SIMATIC diagnostic buffer, follow these steps:

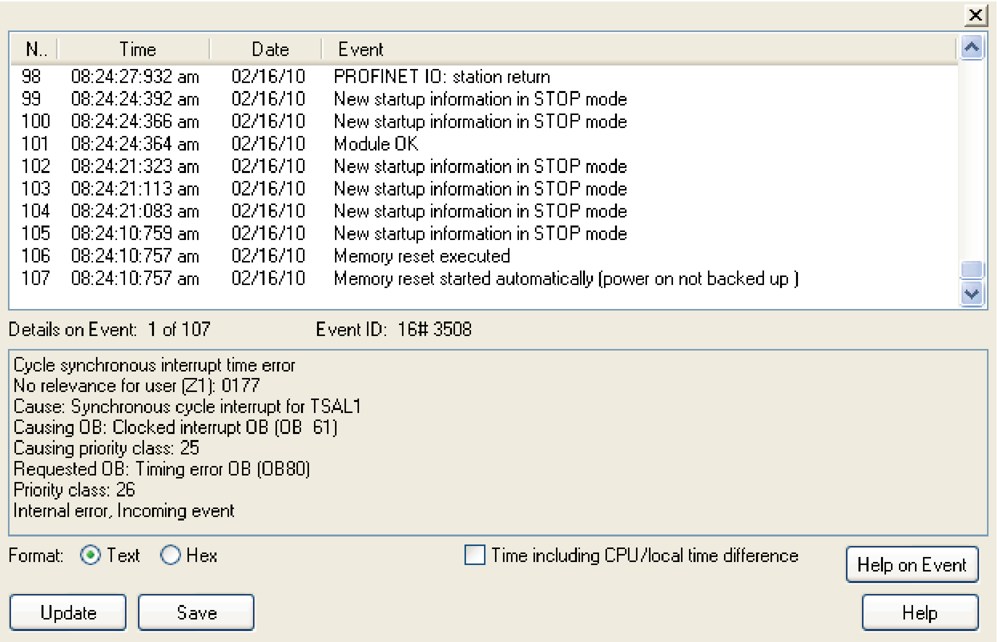
1. Select the **CPU > Diagnostic Buffer** menu command.

   WinLC RTX opens the diagnostic buffer.

Structure of the diagnostic buffer

The diagnostic buffer displays various diagnostic information.

- An event list is displayed in the top pane of the diagnostic buffer.
- Specific details regarding the events are displayed in the bottom pane.



N..	Time	Date	Event
98	08:24:27:932 am	02/16/10	PROFINET IO: station return
99	08:24:24:392 am	02/16/10	New startup information in STOP mode
100	08:24:24:366 am	02/16/10	New startup information in STOP mode
101	08:24:24:364 am	02/16/10	Module OK
102	08:24:21:323 am	02/16/10	New startup information in STOP mode
103	08:24:21:113 am	02/16/10	New startup information in STOP mode
104	08:24:21:083 am	02/16/10	New startup information in STOP mode
105	08:24:10:759 am	02/16/10	New startup information in STOP mode
106	08:24:10:757 am	02/16/10	Memory reset executed
107	08:24:10:757 am	02/16/10	Memory reset started automatically (power on not backed up)

Details on Event: 1 of 107   Event ID: 16# 3508

Cycle synchronous interrupt time error
No relevance for user (Z1): 0177
Cause: Synchronous cycle interrupt for TSAL1
Causing OB: Clocked interrupt OB (OB 61)
Causing priority class: 25
Requested OB: Timing error OB (OB80)
Priority class: 26
Internal error, Incoming event

Format: ● Text ○ Hex   ☐ Time including CPU/local time difference

Task of the diagnostic buffer

The diagnostic buffer is implemented as a ring buffer that contains single event entries. The events are displayed in descending order by time. If the ring buffer is full, a new event overwrites the oldest entry in the buffer.

The diagnostic buffer displays the following information:

- Event list (upper pane): This list displays all the events in the diagnostic buffer. The following information is shown for each event:
  - Number of the entry (column can be sorted)
  - Data and time of the event
  - Brief description of the event (column can be sorted)
- Event ID (between the upper and lower panes): Displays the ID number of a selected event.
- Event Details (lower pane): Displays the event details in either text or hexadecimal format.

Selecting the format

You can display the diagnostic information in the lower pane in text or hexadecimal format. In hexadecimal format, the hexadecimal values of the 20 bytes of the selected event are displayed.

Text format: the following details about the selected event are displayed:

- A brief description
- Additional information, depending on the event, such as the address of the instruction that caused the diagnostic event and the mode transition that was caused by the event.
- The event state (incoming or outgoing)

If a single parameter of text cannot be identified, the diagnostic buffer displays the string "###". If no text exists for new modules or new events, the event numbers and the single parameters are displayed as hexadecimal values.

Hexadecimal format: the hexadecimal values for the selected event are displayed.

Selecting the time type

If you select the "Time including CPU/local time difference" check box, the diagnostic buffer applies a correction value to the time of day.

If you do not select the "Time including CPU/local time difference" check box, the diagnostic buffer displays the entries with the time of day of the module. Use this setting if the time of day of the module is the same as the time at the location of the user reading the diagnostic buffer (same time zone).

If you change the settings, the diagnostic buffer immediately updates the time stamps of the entries.

Updating the diagnostic buffer

To display the most up-to-date information in the window, select the "Update" button.

Saving the diagnostic buffer

To save a text file containing the event list and the detailed information for every event, click the "Save" button. The text file contains the information either in text or in hexadecimal format.

Displaying help

To display help on the diagnostic buffer, click the "Help" button. To display help on a specific event:

1. Select the event in the upper pane.
2. Click the "Help on Event" button.

5.8 Archiving and restoring STEP 7 user programs

Introduction

The archive command enables you to save the configuration and STEP 7 user program to an archive file (*.wld). The archive file allows you to easily restore the configuration and STEP 7 user program for the controller.

The archive file functions like the removable memory module (EEPROM card) of an S7-CPU; however, it differs in that the controller does not automatically restore the archive file after a memory reset (MRES). You must manually restore the archive file.

Requirements

- The controller is in STOP mode.

Creating an archive file

To create an archive file, follow these steps:

1. Select the **File > Archive...** menu command.

   The Save As... dialog opens.
2. Navigate to the storage location of the archive file.
3. Name the archive file.

Result

The controller then creates the archive file with the extension *.wld.

The archive file stores the current STEP 7 user program, the current system configuration, and the current values of the DBs. The archive file does **not** store the configuration of the PC station in the component configurer.

Note

You can also use SIMATIC Manager of STEP 7 to create an archive file by selecting the **File> Memory Card File > New** menu command.

Restoring an archive file

NOTICE
To load a wld file, the stored configuration in the wld file must agree with the configuration of the submodules in the component configurer. The configuration of the component configurer has to be created manually. If the configurations do not agree, the wld file is rejected.

To restore an archive file, follow these steps:

1. Select the **File > Restore** menu command.
2. Select the archive file with the *.wld extension you want to restore.
3. Confirm with "OK".

Result

The STEP 7 user program and the configuration for the controller are downloaded again.

5.9 Selecting the restart method

The restart method determines which startup OB (Page 105) the controller executes whenever a change from STOP mode to RUN mode occurs. The startup OB allows you to initialize your STEP 7 user program and variables. WinLC RTX supports two restart methods:

- Warm restart: The controller executes OB 100 before starting the free cycle (OB 1).
  A warm restart resets the peripheral inputs (PI) and changes the peripheral outputs (PQ) to a pre-defined safe state (default is 0). The warm restart also saves the current value for the retentive memory areas for the memory bits (M), timers (T), counters (C), and data blocks (DBs).
- Cold restart: The controller executes OB 102 before starting the free cycle (OB 1). Like a warm restart, a cold restart resets the peripheral inputs (PI) and changes the peripheral outputs (PQ) to a pre-defined safe state (default is 0). However, a cold restart does **not** save the retentive memory (M, T, C, and DB), but sets these areas to their default settings.

You use STEP 7 to configure the default restart method for the controller. The default restart method is stored in the configuration (system data) for the controller that you download with your STEP 7 user program. WinLC RTX uses this restart method when WinLC RTX is configured for Autostart (Page 63) and returns to RUN mode following a power cycle.

Whenever you click (using the left mouse button) the RUN mode selector on the panel to change from STOP mode to RUN mode, WinLC RTX performs a warm restart, executing OB 100.

To select a specific restart method, choose one of the following options to change the controller from STOP mode to RUN mode:

Option 1

1. Select the **CPU > RUN** menu command to change the controller from STOP to RUN mode.
2. If required, confirm your selection or enter the password.
3. Select either warm or cold restart in the "Restart Method" dialog box.

Option 2

1. Right-click the "RUN" mode selector position.
2. If required, confirm your selection or enter the password.
3. Select either warm or cold restart in the "Restart Method" dialog box.

> **Note**
>
> If you have configured the confirmation security option, you must acknowledge a confirmation dialog before the controller panel displays the "Restart Method" dialog.
> If you have configured the password security option (Page 70) and the password prompt interval is either 0 or has expired, the controller panel displays the "Access Verification" dialog for you to enter the password. After verifying successful password entry, the controller panel displays the "Restart Method" dialog.

Result

After executing OB 100 (warm restart) or OB 102 (cold restart), the controller executes the free cycle (OB 1).

5.10 Saving retentive data

5.10.1 Available options for WinLC data storage

Introduction

The controller can store retentive data in the following ways:

- In a file system on your hard disk
- In NVRAM (Page 90)

To configure the retentive data storage, use the options in "Data Storage" tab (Page 66).

Options for storing retentive data

The following table provides an overview of the options for storing retentive data:

PC system	Storage	Size	UPC necessary	Note
Any PC	On hard disk	Limited only by the size of the hard disk on the computer	Yes	Retentive data cannot be saved to the hard disk in the event of a blue screen.
SIMATIC Box PC: Microbox PC 427B Box PC 627B *) Box PC 827B *) IPC427C IPC627C **) IPC827C **)	Integrated NVRAM	128 Kbytes	No	*) DC Power supply and hardware variant PROFIBUS and PROFINET IRT required **) Hardware variant PROFIBUS and PROFINET IRT required
SIMATIC Panel PC: Panel PC 477B Panel PC 677B *) HMI IPC477C (Pro) HMI IPC577C *) HMI IPC677C **)	Integrated NVRAM	128 Kbytes	No	*) DC Power supply required **) Hardware variant PROFIBUS and PROFINET IRT required
SIMATIC Modular Embedded Controller: S7-mEC, EC31	Integrated NVRAM	512 Kbytes	No	

CAUTION

Recommended use of a UPS

Power loss without a shutdown of the operating system can cause file systems of Windows XP Professional to become corrupted. For this reason, use a UPS (Page 95) to protect the file systems in these operating systems.

5.10.2 What Information about the Controller Does WinLC RTX Store?

Retentive data

WinLC RTX stores the following information as retentive data:

- Operating mode of controller

  WinLC RTX stores both the current operating mode of the controller and the mode selector setting on the controller panel.

- Retentive memory areas and diagnostic buffer

  When you configure WinLC RTX in STEP 7, you can specify the areas of retentive data for the timers (T), counters (C), bit memory (M), and retentive data blocks (DBs). When you perform a normal shutdown of WinLC RTX, the controller saves this retentive data and the diagnostic buffer. A normal shutdown of the Windows operating system, whether initiated by user action or a UPS signal, also causes WinLC RTX to save these data.

The retentive data are saved during operation. You specify the options (Page 88) for saving data in the "Data Storage (Page 66)" tab of the controller properties. The saved retentive data are loaded the next time WinLC RTX starts up. For a description of how WinLC RTX loads the saved data at startup, refer to "How WinLC RTX loads memory areas during startup (Page 93)".

When does WinLC RTX save retentive data?

The following table shows the actions that cause WinLC RTX to save retentive data:

Retentive data	Action that causes WinLC RTX to save this data
Operating mode of controller	Operating mode changes made using STEP 7 or the mode selector on the controller panel Normal shutdown of WinLC RTX (shut down controller) Termination, for example, due to power failure or blue screen, if the WinLC data are stored in NVRAM
Retentive memory areas (T, C, M, and DBs), diagnostic buffer	Normal shutdown of WinLC RTX or termination, for example, due to power failure or blue screen, if the WinLC data are stored in NVRAM

Note

You must select the "NVRAM (Page 90) Storage" option on the "Data Storage" tab of the controller properties in order to ensure that retentive data will be stored in the event that WinLC terminates, for example, due to power-down of PC, power failure, or Windows Stop Error ("Blue Screen (Page 98)").

5.10.3 NVRAM storage

Storage of retentive data in NVRAM protects against loss of important program data after a power failure.

Possible NVRAM sizes

The following types of NVRAM are available:

- 128 Kbytes of NVRAM on an optional, plug-in SIMATIC WinAC NV128 card
- 128 Kbytes of integrated NVRAM
- 512 Kbytes of S7-mEC, EC31-RTX NVRAM, EC31-HMI/RTX NVRAM

NVRAM memory allocation

The following types of information share the available NVRAM. The various areas of the NVRAM memory can be configured according to the available size:

Item	Memory consumption	Default	
System startup information	1 Kbyte	1 Kbyte	
Diagnostic buffer	Number of entries * 20 bytes	2400 bytes	120 entries
Flag memory (M)	Number of flag bytes	16 bytes	MB0 – MB15
S7 timers	Number of timers * 2 bytes	0 bytes	No time is retentive by default
S7 counters	Number of counters * 2 bytes	16 bytes	Z0 - Z7
Retentive DBs configured with STEP 7 or created by means of SFC 85 with ATTRIB = 0x00	Number of Kbytes in retentive DBs	Configuration of the user program	
Overhead for DBs created by means of SFC 85	Number of DBs * 45 bytes	0 bytes	

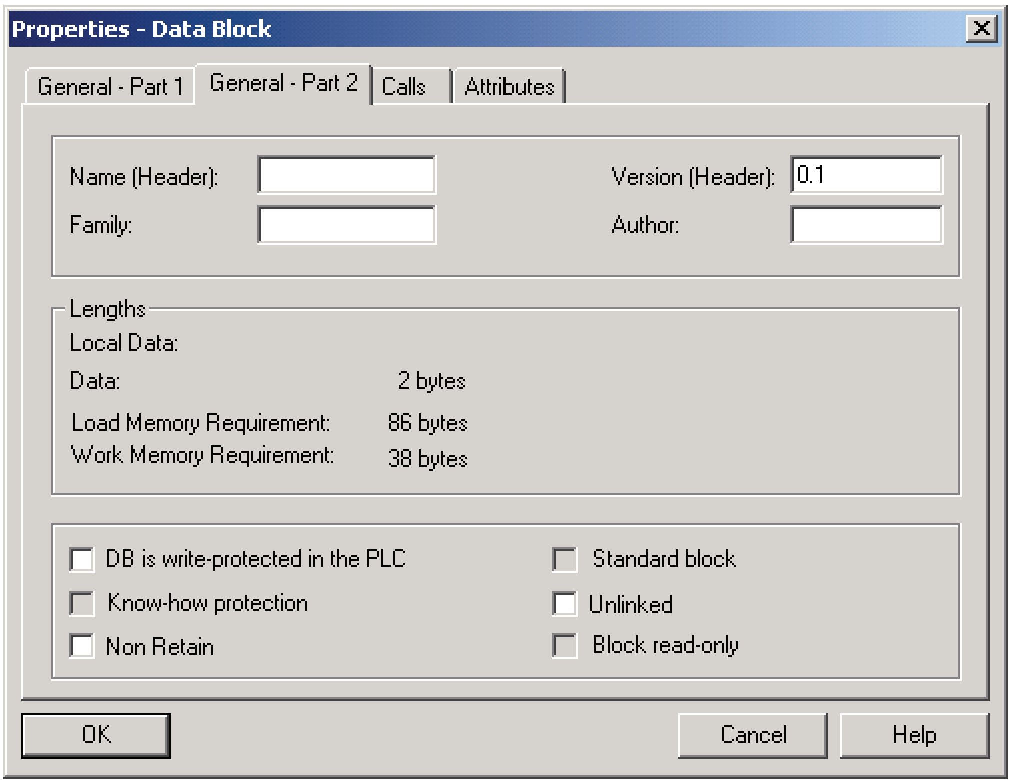
Adapting the configuration of retentive areas for diagnostic buffer, bit memory, timers, and counters

The size of the diagnostic buffer and the sizes of the retentive areas for bit memory, timers, and counters can be set in the WinLC RTX properties dialog in STEP 7 HW Config.
For details on these settings, refer to the STEP 7 documentation.

Display and configuration of retentive data for data blocks

By default, STEP 7 configures all data blocks as retentive. In the "Properties - Data Block" dialog, the following three check boxes are **not** selected for a retentive data block:

- DB is write-protected in the PLC
- Non Retain
- Unlinked



Name (Header):
Version (Header): 0.1
Family:
Author:

Lengths
Local Data:
Data: 2 bytes
Load Memory Requirement: 86 bytes
Work Memory Requirement: 38 bytes

□ DB is write-protected in the PLC
□ Standard block
□ Know-how protection
□ Unlinked
□ Non Retain
□ Block read-only

If you select any of these three check boxes, the data block is not retentive. The NVRAM restrictions do not affect the DB.

Data blocks created with SFC 85

A data block created with SFC 85 is retentive when parameter ATTRIB = 0x00 is set. If you allow for use of the NVRAM, these data blocks require memory for the overhead and for the retentive data. For data blocks created with SFC 85 with parameter ATTRIB not equal to 0x00, the data block only requires memory for the overhead.

Particularities for use of a SIMATIC WinAC NV128 card

To use a SIMATIC WinAC NV128 card, plug the card into any available PCI slot when the computer is shut down. When you power on the computer, the Windows Plug-and-Play Manager detects the card and allocates memory for it. Whether you install WinAC RTX before or after you install the SIMATIC WinAC NV128 card, WinAC RTX will automatically detect the card and make it available for WinLC data storage.

NOTICE

You cannot use multiple SIMATIC WinAC NV128 cards in WinLC RTX. Likewise, you cannot use a SIMATIC WinAC NV128 card in conjunction with integrated NVRAM.

If you have saved retentive data on a SIMATIC WinAC NV128 card and you remove the card when your computer is off, the next time you start WinLC RTX, it will start without retentive data. The controller panel lights up the INTF status indicator, and the diagnostic buffer contains the "Unbuffered startup" error.

To recover, you must either shut down WinLC RTX and your computer and insert and install a new SIMATIC WinAC NV128 card or you must switch to Hard Disk Storage for your retentive data.

Exceeding the NVRAM storage

If you switch from "Hard Disk Storage" to "NVRAM Storage" for your retentive data, and the retentive data in your STEP 7 user program requires more memory than the NVRAM supports, no retentive data can be reloaded after a startup. A message in the diagnostic buffer indicates that an unbuffered startup occurred.

You must either reduce the size of the retentive data in your STEP 7 user program or select the "Hard Disk Storage" option instead of "NVRAM Storage" for your retentive data.
In the online view of STEP 7, the current memory utilization is displayed on the "Memory" tab of the "Module Information" dialog.

5.10.4 How WinLC RTX loads memory areas during startup

Introduction

Upon startup, WinLC RTX determines whether the controller was shut down correctly.
It performs the following tasks:

- Loads the blocks of the STEP 7 user program
- Restores the state of the controller, based on the saved operating mode and Autostart configuration and sets the mode selector on the controller panel to the last position saved.
- Restores the work memory, depending on whether or not a valid retentive data area is available.

Note

Accessing data from previous releases

WinLC RTX cannot read the saved retentive data of a previous release of WinLC RTX or WinAC Basis. However, you can restore a STEP 7 user program and a configuration that were archived in a previous release.

Initializing the memory with valid retentive data

The controller was shut down normally and the data were saved on the hard disk or in NVRAM. WinLC RTX loads the operating data of the controller as follows:

- During startup, WinLC RTX loads the retentive S7 memory areas, the current values of the data blocks (work memory), and the contents of the diagnostic buffer. If you have configured the controller for a cold restart (OB 102), WinLC RTX resets the process variables and the S7 memory areas to the initial values from the load memory.
- Based on the autostart settings, WinLC RTX sets the state of the controller to either STOP or RUN mode.
- WinLC RTX sets the mode selector to the setting when WinLC RTX last saved the operating mode of the controller.

Note

Once these values are read in and startup is complete, WinLC RTX deletes the old retentive data.

Initializing the memory without valid retentive data

The controller was not shut down normally and the data were not saved in NVRAM. In this case, the work memory is restored with the initial state from the load memory. The diagnostic buffer is empty after the controller is restarted.

WinLC RTX performs the following tasks when the controller is restarted:

- The load memory is read and the system configuration, the process variables, and the S7 memory areas are reloaded with the initial values configured in STEP 7.
- An unbuffered startup is triggered. WinLC RTX generates a startup event, which you can read out in OB 100. Based on the Autostart settings, WinLC RTX sets the controller to either STOP or RUN mode.
- WinLC RTX sets the mode selector to the setting when WinLC RTX last saved the operating mode of the controller.

Problems when starting the controller

If WinLC RTX cannot read an element in the retentive memory (for example, state of the controller), the controller goes to STOP and the mode selector is set to STOP. WinLC RTX still contains the STEP 7 user program and the configuration, but no retentive data. This problem can be caused by hardware faults in your computer, among other things.
To recover from this error condition, you must reload your control program and system data from STEP 7.

Note

The mode selector of the controller is set to STOP. You can download the control program and system data from a remote computer, but you cannot use the remote computer to set the controller to RUN mode. You must set the mode selector (Page 60) to RUN on the local computer for WinLC RTX in order to place the controller in RUN mode.

5.10.5 Uninterruptible power supply (UPS)

You can use a UPS to supply emergency power to your computer. A UPS system helps ensure that WinLC RTX shuts down correctly and saves the powered-down state (Page 89) in case of a power failure. Siemens strongly recommends the use of a UPS for operation with the Windows XP Professional operating system.

The WinAC RTX installation DVD contains an executable file ("WinLC_Shutdown.exe"), which is installed on your PC. To ensure reliable data retention, WinAC RTX must be shut down by the installed UPS system with the aid of this shutdown file.

Note

For information regarding suitable UPS systems and help on creating an executable batch file, refer to the FAQs (http://support.automation.siemens.com/WW/view/en/31410255) on the Internet.

Setting up the UPS for your computer

Microsoft Windows provides a dialog box for configuring the UPS for your computer:

1. Select the **Start > Settings > Control Panel** menu command to display Control Panel.
2. Double-click the "Power Options" icon to open the "Power Options Properties" dialog box.
3. Open the "UPS" tab and enter the parameters for your UPS system.
4. Click "Apply" or "OK" to set the UPS properties.

Consequences of a power loss without operating system shutdown

Power loss without a shutdown of the operating system can cause file systems of Windows XP Professional to become corrupted. For this reason, use a UPS system to protect the file systems in these operating systems.

In addition, some SIMATIC PCs detect a power failure and send a power failure signal to WinLC RTX. WinLC RTX can then initiate a fast shutdown and save retentive data to NVRAM if so configured. See "Retentive data storage in the event of power failure or blue screen (Page 98)" for a list of the SIMATIC PCs that support the power failure signal, and a description of how WinLC RTX responds.

Systems with Windows XP Embedded that use a compact flash file system that is protected with the Enhanced Write Filter are stable against unexpected loss of power.

5.10.6 Buffering of Data with SFCs

You can use SFC 82 (CREA_DBL), SFC 83 (READ_DBL), and SFC 84 (WRIT_DBL) to save data at significant events in your process.

Note

You must consider the possibility of a Windows Stop Error when using SFC 22, SFC 23, SFC 82, SFC 83, SFC 84, or SFC 85.

Example

When changing a recipe, you store the recipe values in load memory without downloading new blocks for the STEP 7 user program to the CPU.

Principle

SFC 82 and SFC 84 modify the data for the STEP 7 user program that is stored in the load memory. Saving the blocks in load memory instead of keeping the values in work memory ensures that these blocks are available even if WinLC RTX cannot save the powered-down state when shutting down the controller.

When executed in the STEP 7 user program, SFC 82 (CREA_DBL), SFC 83 (READ_DBL), and SFC 84 (WRIT_DBL) create and update blocks that are stored as part of your STEP 7 user program in load memory.

Application

SFC 82, SFC 83, and SFC 84 are asynchronous SFCs that run in the background. In order to use asynchronous SFCs, you must allow sufficient sleep time to allow WinLC RTX to process the SFCs without encountering jitter (Page 187).

If you call SFC 82, SFC 83, or SFC 84 from the startup OB (OB 100 or OB 102), WinLC RTX executes these SFCs synchronously. This differs from the operation of a hardware PLC.

Avoiding jitter

Do not use a polling loop that waits for completion of an asynchronous SFC, especially for SFC 82, SFC 83, and SFC 84. Because the asynchronous SFCs are being executed in the background, you do not have to wait until they are finished. You can continue with further program processing,

Whenever the STEP 7 user program calls SFC 82, SFC 83, or SFC 84, the SFC reads or writes the data on the hard disk. If you call these SFCs in each scan cycle (fro example, from OB 1) or from a cyclical OB that is executing rapidly, the constant reading and writing on the disk can cause the disk to fail or can add jitter.

Recommendation

You should only call SFC 82, SFC 83, or SFC 84 to record an important process event, e.g., a changed recipe.

See also

What Information about the Controller Does WinLC RTX Store? (Page 89)

5.11 Operation of WinLC RTX after a Windows Stop Error (blue screen)

5.11.1 WinLC RTX Response to a Blue Screen

Introduction

WinLC RTX supports OB 84 (CPU hardware fault), which allows you to initiate orderly shut down of your process in case a Windows Stop Error (blue screen) occurs while WinLC RTX is in operation. During a blue screen, communication interfaces (Page 39) configured as submodules (Page 41) continue to function.

Response of operating mode and OB 84 to a Windows Stop Error

In the following cases, WinLC RTX can still operate after Windows has initiated the system shutdown procedure, provided the memory used by the real-time system has not been corrupted:

- WinLC RTX is in RUN mode and OB 84 is loaded

  WinLC RTX remains in RUN mode until:

  – The STEP 7 user program calls SFC 46 (STP) to place the controller in STOP mode.

  – A programming device or communication partner that accesses WinLC RTX by means of a submodule communication interface initiates a change to STOP mode

- WinLC RTX is in RUN mode and OB 84 is not loaded

  WinLC RTX changes the operating mode to STOP mode. Windows then completes the shutdown of the system.

- WinLC RTX is not in RUN mode

  Windows is completely shut down.

Note

You can configure Windows and WinLC RTX to automatically restart following a blue screen.

Restrictions

The following restrictions apply when Windows is shutting down:

- The WinLC RTX controller panel is not available.
- Some system functions are disabled, including SFC 22, SFC 23, SFC 82, SFC 83, SFC 84, and SFC 85. (Page 99)
- Block operations fail, returning an error code.
- Communication with Windows applications is unavailable; however, communication with submodules of WinLC RTX is not affected.
- Communication with external systems (such as HMI devices or programming devices) is only available if the network is connected to a configured submodule of WinLC RTX.
- If the retentive data are not stored in NVRAM, a restart of the computer followed by a restart of WinLC RTX initializes all of the program variables to their default values and empties the diagnostic buffer. If the retentive data are stored in NVRAM, WinLC RTX can restore the data when it restarts. Refer to "NVRAM storage (Page 90)" for an overview of which SIMATIC PCs have NVRAM for storing retentive data.

Undetected Windows blue screen

WinLC RTX cannot guarantee in all cases that it can detect a Windows blue screen and continue operation. Operation is only possible if the cause of the blue screen does not corrupt memory that WinLC RTX or the real-time operating system uses.

If WinLC RTX does not detect the blue screen, it cannot call OB 84 or continue running. You must reboot your computer to continue operating.

If you specified that retentive data are to be stored in NVRAM (SIMATIC WinAC NV128 card or integrated PC NVRAM) and an undetected Windows blue screen occurs, WinAC RTX will start with an unbuffered startup after the reboot. The controller panel lights up the INTF status indicator, and the diagnostic buffer contains the "Unbuffered startup" error.

5.11.2 Retentive data storage in the event of power failure or blue screen

Previous topics discussed the following:

- WinLC RTX response to a blue screen (Page 97)
- What controller information does WinLC RTX store? (Page 89)
- Available options for WinLC data storage (Page 88)
- Configuring the data storage (Page 66)
- Uninterruptible power supply (UPS) (Page 95)

All of these variants in features, configuration choices, and types of Windows or power failures will interact together. The behavior of your particular system depends on these variations in combination.

Conclusion

Retentive data can be saved in the event of a power failure or a blue screen for SIMATIC PCs with integrated NVRAM. A list of the corresponding PCs can be found in Chapter Available options for WinLC data storage (Page 88)

Retentive data can generally **not** be saved in the event of a power failure or a blue screen for SIMATIC PCs without integrated NVRAM.

5.11.3 Considerations for SFC 22, SFC 23 and SFC 82 to 85

If a Windows blue screen (Page 97) occurs when WinLC RTX is in RUN mode, it attempts to stay in RUN mode and initiates OB 84; however, the operation of WinLC RTX during a blue screen can be adversely affected by SFC 22, SFC 23, SFC 82, SFC 83, SFC 84, or SFC 85.

Under most circumstances, SFC 22, SFC 23, SFC 82, SFC 83, SFC 84, and SFC 85 return error code 8092 in the event of a Windows blue screen. Applications that need to continue operating after a Windows Stop Error can check for this error code. If, however, one of these SFCs is in a Windows call at the time of the Stop Error, the SFC is not able to return the 8092 error code and WinLC RTX cannot initiate OB 84.

Effects of active SFCs

Certain SFCs, if active at the time of a Windows Stop Error, can cause either WinLC RTX or other functions to become unresponsive and lock up.

- If SFC 22, SFC 23, or SFC 85 is in a call of a Windows function at the time of the Stop Error, the SFC cannot return from the SFC call and WinLC RTX fails to maintain control of the process. If this occurs, the I/O time monitoring disables the inputs and outputs.
- If SFC 82, SFC 83, or SFC 84 is in a call of a Windows function at the time of the Stop Error, WinLC RTX attempts to remain in RUN mode (continuing to control the process), but background operations including some communication functions can lock up. Setting WinLC RTX to STOP mode, whether by program action or by user intervention from a remote system, can affect the shutdown sequence of the computer.

⚠ WARNING

A blue screen that results in locking up of either the controller or background functions can cause damage to process equipment or injury to personnel. You can prevent this by taking proper precautions when designing your STEP 7 user program.

Possible remedy

If your process application needs to survive a Windows Stop Error, call these SFCs (SFC 22, SFC 23, SFC 82, SFC 83, SFC 84, or SFC 85) only during initialization (during the execution of OB 100 or OB 102) or during non-critical parts of the control process.

5.11.4 Configuring automatic Windows restart after a blue screen

Procedure

To configure automatic Windows restart, follow these steps:

1. Open the Windows Control Panel and double-click "System".
2. On the "Advanced" tab in the "System Properties" dialog, click "Startup and Recovery".
3. Select the "Automatic restart" check box.
4. Click "OK" to confirm the "Startup and Recovery" dialog box and the "System Properties" dialog box.

Result

The Windows operating system will restart automatically the next time a blue screen occurs.

5.11.5 Restart behavior of WinLC RTX after a blue screen

If Windows is configured to restart automatically after a Windows blue screen (Stop Error), WinLC RTX starts if it is configured to start at PC boot.

Retentive data, "NVRAM storage" setting

WinLC RTX restarts with the current STEP 7 user program and uses the retentive data stored in NVRAM, provided the following conditions are met:

- Your computer supports NVRAM configuration for WinLC data storage
- Data storage in NVRAM was configured before the blue screen occurred
- WinLC RTX was able to save the retentive data when the blue screen occurred

In addition, the actions described in "Initializing the memory with valid retentive data" in the chapter How WinLC RTX loads memory areas during startup (Page 93) are performed.

Retentive data, "Hard Disk Storage" setting

WinLC RTX restarts with the STEP 7 user program as it was last downloaded and executes OB 100 if it is present. WinLC RTX executes OB 100 with event 1382 (hex) after a blue screen, even if OB 102 "Cold Start" is configured in STEP 7 HW Config. The diagnostic buffer shows the current/last startup type as "automatic warm reboot after non-backup power-on with system memory reset".

Reference: Detailed information about the use of OB 100 can be found in the STEP 7 online help or in the *System Software for S7-300/400 System and Standard Functions* reference manual. If you want to open this manual on a PC on which STEP 7 is installed, select **Start > SIMATIC > Documentation > English** and double-click "STEP 7 - System and Standard Functions for S7-300 and S7-400".

No retentive data - data not stored in NVRAM

If in the event of a blue screen, the WinLC data cannot be stored in NVRAM, WinLC RTX restores the state of the controller prior to the Windows Stop Error.

Note

WinLC RTX generates a startup event that identifies the type of startup: buffered or unbuffered. You can program OB 100 to read this start event. For an unbuffered startup, the variable OB100_STOP at address LW6 is set to W#16#4309.

Start behavior: The controller was configured for Autostart. If the controller mode selector was in the RUN position at the time of the shutdown, WinLC RTX restarts in RUN mode. If OB 84 has responded to a Windows blue screen and has placed the controller in STOP mode before the shutdown, WinLC RTX still starts up in RUN mode because without NVRAM data storage, WinLC RTX could not store the state of the controller.

If you do not have NVRAM data storage and you do not want the controller to start in RUN mode after a Windows Stop Error, you must include code in the startup OB (OB 100 or OB 102) to detect that a WinLC RTX shutdown occurred without saving the retentive data and to set the controller to STOP mode when restarted. A value of 0010 xxxx in bits 7 - 0 of the OB_STR_INFO variable in the startup OB indicates that retentive memory areas are not available on the hard disk.

Working in STEP 7 6

6.1 Using STEP 7 with the Controller

STEP 7 provides programming and configuration tools for working with WinLC RTX.
You perform the following tasks with STEP 7:

- Define the controller and distributed I/O configuration through the STEP 7 Hardware Configuration (Page 52) tool
- Develop a STEP 7 user program using any of the STEP 7 control programming languages
- Configure operational parameters (Page 103) and I/O addresses for the controller
- Download your configuration and STEP 7 user program to the controller

Refer to your STEP 7 documentation for additional information.

6.2 Configuring the Operational Parameters for the Controller

STEP 7 provides a Hardware Configuration application for configuring the operational parameters for the controller. This configuration is then stored in various SDBs in the System Data container.

After you download the System Data, the controller uses the configured parameters for the following events:

- Whenever you start the controller
- On the transition to RUN mode (if you modified the hardware configuration online while the controller was in STOP mode)

To configure the operational parameters from the STEP 7 Hardware Configuration application, right-click the controller entry in the station window and select Object Properties. From the Properties dialog, you configure the operational parameters.

Accessing Operational Parameters

To configure any of these operational parameters in STEP 7, open the SIMATIC Manager and follow these steps:

1. In the SIMATIC Manager, select the PC station.
2. Click the Configuration icon.
3. Right-click the controller in the station window and select Object Properties.
4. Click the tab with the name of the parameter that you want to configure (such as Cyclic Interrupt) and enter the appropriate values in the dialog.
5. Click OK to confirm your configuration.

Refer to your STEP 7 documentation for specific information about configuring the controller properties and the operational parameters.

6.3 Logic Blocks Supported by WinLC RTX

Introduction

Like other S7 controllers, WinLC RTX provides several types of logic blocks for processing the user program:

- Organization blocks (OBs) (Page 105)
- System functions (SFCs) (Page 109)
- System function blocks (SFBs) (Page 113)

These blocks are an integral part of WinLC RTX.

Additional S7 blocks

In addition to these system blocks, you can also use the following S7 blocks to create your STEP 7 user program:

- Functions (FCs): WinLC RTX supports up to 65536 FCs (FC 0 to FC 65535).
  Each function can contain up to 65570 bytes.
- Function blocks (FBs): WinLC RTX supports up to 65536 FBs (FB 0 to FB 65535).
  Each function block can contain up to 65570 bytes.
- Data blocks (DBs): WinLC RTX supports up to 65535 DBs (DB 1 to DB 65535).
  (DB 0 is reserved.) Each data block can contain up to 65534 bytes.

The number and size of the FCs, FBs, and DBs are also limited by the amount of available system memory.

6.4 Organization Blocks (OBs)

Definition

Organization blocks (OBs) are the interface between the controller operating system and the STEP 7 user program. You use OBs to execute specific components of your STEP 7 user program for the following events:

- When the controller starts and restarts
- Cyclically or at a specific time interval
- At certain times or on certain days.
- After running for a specified period of time
- When errors occur
- When a process interrupt occurs

The program logic in an OB can contain up to 65570 bytes.

Organization blocks are processed according to the priority assigned to them.

List of OBs

The following table lists the OBs that WinLC RTX supports:

OB	Description	Priority class
OB 1	Free scan cycle	1 (lowest)
OB 10	Time-of-day interrupt	0, 2 to 24
OB 20	Time-delay interrupt	0, 2 to 24
OB 30 to OB 38	Cyclic interrupt	0, 2 to 24
OB 40	Process interrupt	0, 2 to 24
OB 52 to OB 54	ODK interrupt	15
OB 55	Status interrupt	0, 2 to 24
OB 56	Update interrupt	0, 2 to 24
OB 57	Manufacturer-specific interrupt	0, 2 to 24
OB 61 and OB 62	Synchronous cycle interrupt	0, 2 to 26 Default: 25
OB 80	Time error	26
OB 82	Diagnostic interrupt	24 to 26 (or 28)**
OB 83	Insert/remove-module interrupt	24 to 26 (or 28)**
OB 84	CPU hardware fault	24 to 26 (or 28)**
OB 85	Priority class error	24 to 26 (or 28)**
OB 86	Rack (DP slave) failure	24 to 26 (or 28)**
OB 88	Processing interrupt	28
OB 100	Warm restart	27
OB 102	Cold restart	27
OB 121	Programming error	Priority class of the OB causing the error
OB 122	I/O access error	
** Priority class 28 during STARTUP, user-configurable priority class (from 24 to 26) in RUN mode		

OBs for the free scan cycle of the program and for cold and warm restarts

The following table shows the OBs for the free scan cycle of the program and for cold and warm restarts. WinLC RTX provides OB 1 (free scan cycle) for continuously executing the STEP 7 user program. On the transition from STOP mode to RUN mode, WinLC RTX executes OB 100 (warm restart) or OB 102 (cold restart), based either on the hardware configuration for WinLC RTX or on the restart type selected in the WinLC RTX controller panel. After OB 100 (or OB 102) has been successfully executed, WinLC RTX executes OB 1.

Organization block (OB)		Start event (in hex)	Priority class
Main program cycle	OB 1	1101, 1103, 1104	1
Warm restart	OB 100	1381, 1382	27
Cold restart	OB 102	1385, 1386	27

Interrupt OBs

WinLC RTX provides a variety of OBs that interrupt the execution of OB 1. The following table lists the different interrupt OBs that are supported by WinLC RTX. These interrupts occur according to the type and configuration of the OB.

The priority class determines whether the controller suspends the execution of the STEP 7 user program (or other OBs) and executes the interrupt OB. You can change the priority class of the interrupt OBs.

Interrupts		Start event (in hex)	Default priority class
Time-of-day interrupt	OB 10	1111	2
Time-delay interrupt Range: 1 ms to 60000 ms	OB 20	1121	3
Cyclic interrupt Range: 1 ms to 60000 ms Recommended: > 10 ms	OB 30 OB 31 OB 32 OB 33 OB 34 OB 35 OB 36 OB 37 OB 38	1131 1132 1133 1134 1135 1136 1137 1138 1139	7 8 9 10 11 12 13 14 15
Process interrupt	OB 40	1141	16
Status interrupt	OB 55	1155	2
Update interrupt	OB 56	1156	2
Manufacturer-specific interrupt	OB 57	1157	2

If WinLC RTX has been configured to execute a particular interrupt OB, but that OB has not been downloaded, WinLC RTX reacts in the following manner:

- If OB 10, OB 20, OB 40, OB 55, OB 56, or OB 57 is missing and OB 85 has not been downloaded, WinLC RTX changes operating mode (from RUN to STOP).
- WinLC RTX remains in RUN mode if a cyclic interrupt OB (OB 32 to OB 36) is missing. If these OBs cannot be executed at the specified time and OB 80 has not been downloaded, WinLC RTX changes from RUN mode to STOP mode.

Notes on cyclic interrupt OBs

Based on the time interval that you configure in the operating parameters (Page 103) for the cyclic interrupt, WinLC RTX starts the execution of the cyclic interrupt OB at the appropriate time. The optimum time interval for your application depends on the processing speed of your computer and the execution time of the cyclic OB. Jitter (Page 187) can cause an occasional overrun in the start event for a cyclic OB, which might cause WinLC RTX to go to STOP mode. Other factors that affect the execution of the OB include the following situations:

- The program in the OB takes longer to execute than the interval allows. If the execution of the program consistently overruns the start event of the cyclic OB, WinLC RTX can go to STOP mode (unless OB 80 is loaded).
- Programs in other priority classes frequently interrupt or take longer to execute, which prevents the controller from executing the cyclic OB at the scheduled time. If this occasionally causes an overrun, WinLC RTX starts the cyclic OB as soon as the first OB finishes.
- STEP 7 performs some task or function that causes the controller not to execute the cyclic OB at the scheduled time.

The sleep time of the WinLC RTX scan cycle does not affect the execution of a cyclic interrupt OB: WinLC RTX attempts to execute the OB at the appropriate interval regardless of the amount of sleep time that you configure for the scan cycle. WinLC RTX provides several types of sleep time management (Page 195) for the free scan cycle. If a cyclic interrupt OB runs too frequently or requires too much of the time allotted for the total scan cycle, it could cause the watchdog to signal an error (calling OB 80 or placing the controller in STOP mode).

If you schedule a cyclic interrupt OB (OB 30 to OB 38) to be executed at a specific interval, make certain that the program can be executed within the time frame and also that your STEP 7 user program can process the OB within the allotted time.

Error OBs

WinLC RTX provides a variety of error OBs. Some of these error OBs have the configured (user-assigned) priority class, while others (OB 121 and OB 122) inherit the priority class of the block where the error occurred.

The local variables for OB 121 and OB 122 contain the following information that can be used by the STEP 7 user program to respond to the error:

- The block type (byte 4) and the block number (bytes 8 and 9) where the error occurred
- The address within the block (bytes 10 and 11) where the error occurred

If the start event occurs for a particular error OB that has not been downloaded, WinLC RTX changes the operating mode (from RUN to STOP).

Error		Start event (in hex)	Default priority class
Time error	OB 80	3501, 3502, 3505, 3507	26
Diagnostic interrupt	OB 82	3842, 3942	26
Insert/remove-module interrupt	OB 83	3861, 3863, 3864, 3865, 3961	26
CPU hardware fault (Windows blue screen (Page 97))	OB 84	3585	26 (or 28)
Priority class error: Start event occurs for an OB that has not been downloaded. During the I/O cycle, WinLC attempts to access a module or slave that is defective or not connected.	OB 85	35A1, 35A3, 39B1, 39B2	26
Rack failure (distributed I/O): a node in the subnet has failed or has been restored.	OB 86	38C4, 38C5, 38C7, 38C8, 39C4, 39C5	26 (or 28)
execution of a program block has been aborted.	OB 88	3571, 3572, 3573, 3575, 3576, 3578, 357A	28
Programming error (For example: the user program attempts to address a timer that does not exist.)	OB 121	2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 253A; 253C, 253E	Same priority class as the OB in which the error occurred.
I/O access error (For example: the user program attempts to access a module that is defective or not connected.)	OB 122	2942, 2943	

Reference

Detailed information about the OBs can be found in the STEP 7 online help or in the *S ystem Software for S7-300/400 System and Standard Functions* reference manual. If you want to open this manual on a PC on which STEP 7 is installed, select **Start > SIMATIC > Documentation > English** and double-click "STEP 7 - System and Standard Functions for S7-300 and S7-400".

6.5 System Functions (SFCs)

Introduction

WinLC RTX provides SFCs, which are system functions that perform various tasks.
The STEP 7 user program calls the SFC and passes the required parameters. The SFC then performs its task and returns the result.

Listing of SFCs

The following table lists the SFCs that WinLC RTX supports:

SFC	Name	Description
SFC 0	SET_CLK	Sets the system clock
SFC 1	READ_CLK	Reads the system clock
SFC 2	SET_RTM	Sets the runtime meter
SFC 3	CTRL_RTM	Starts or stops the runtime meter
SFC 4	READ_RTM	Reads the runtime meter
SFC 5	GADR_LGC	Queries the logical address of a channel (PROFIBUS DP only; see also SFC 70)
SFC 6	RD_SINFO	Reads the start information of an OB
SFC 9	EN_MSG	Enables block-related and symbol-related messages as well as group status messages
SFC 10	DIS_MSG	Disables block-related and symbol-related messages as well as group status messages
SFC 11	DPSYNC_FR	Synchronizes groups of DP slaves (not available for PROFINET IO)
SFC 12	D_ACT_DP	Deactivates and activates slaves (PROFIBUS DP or PROFINET IO)
SFC 13	DPNRM_DG	Reads the diagnostic data of a DP slave (PROFIBUS DP only; for similar functionality for PROFINET IO, see SFB 54 and SFB 52) DP configuration tested: one ET 200M slave with one 8-input/8-output module and one 16-output module
SFC 14	DPRD_DAT	Reads the consistent data from a DP slave
SFC 15	DPWR_DAT	Writes the consistent data to a DP slave
SFC 17	ALARM_SQ	Generates an acknowledgeable block-related message
SFC 18	ALARM_S	Generates a permanently acknowledgeable block-related message
SFC 19	ALARM_SC	Queries the acknowledgement status for the last message (SFC 17 or SFC 18)
SFC 20	BLKMOV	Copies variables
SFC 21	FILL	Initializes a memory area 1 word 50 words 100 words

SFC	Name	Description
SFC 22 (Page 99)	CREAT_DB	Creates a retentive data block in work memory The current values of the DB are retained after a warm restart.
SFC 23 (Page 99)	DEL_DB	Deletes a data block WinLC RTX allows an application to delete a non-sequence-relevant data block.
SFC 24	TEST_DB	Provides information about a data block For WinLC RTX, SFC 24 can return the DB length and write protection flags for non-sequence-relevant data blocks, although it returns error code 80B2 for non-sequence-relevant data blocks.
SFC 26	UPDAT_PI	Updates the process-image input table
SFC 27	UPDAT_PO	Updates the process-image output table
SFC 28	SET_TINT	Sets the time-of-day interrupt (OB 10)
SFC 29	CAN_TINT	Cancels the time-of-day interrupt (OB 10)
SFC 30	ACT_TINT	Activates the time-of-day interrupt (OB 10)
SFC 31	QRY_TINT	Queries the time-of-day interrupt (OB 10)
SFC 32	SRT_DINT	Starts the time-delay interrupt (OB 20)
SFC 33	CAN_DINT	Cancels the time-delay interrupt (OB 20)
SFC 34	QRY_DINT	Queries the time-delay interrupt (OB 20)
SFC 36	MSK_FLT	Masks synchronous errors
SFC 37	DMSK_FLT	Unmasks synchronous errors
SFC 38	READ_ERR	Reads the error register
SFC 39	DIS_IRT	Disables the processing of new interrupt events
SFC 40	EN_IRT	Enables the processing of new interrupt events
SFC 41	DIS_AIRT	Delays higher priority interrupts and asynchronous errors
SFC 42	EN_AIRT	Enables the processing of new interrupt events with higher priority than the current OB
SFC 43	RE_TRIGR	Retriggers cycle-time monitoring
SFC 44	REPL_VAL	Transfers a substitute value to ACCU1 (Accumulator 1)
SFC 46	STP	Changes the operating mode to STOP mode
SFC 47 (Page 203)	WAIT	Delays the execution of the STEP 7 user program by the specified number of microseconds, rounded up to the nearest millisecond
SFC 49	LGC_GADR	Queries the module slot belonging to a logical address (PROFIBUS DP only; see also SFC 71)
SFC 50	RD_LGADR	Queries all of the logical addresses of a module
SFC 51	RDSYSST	Reads all or part of a system status list
SFC 52	WR_USMSG	Writes a user-defined diagnostic event to the diagnostics buffer
SFC 54	RD_DPARM	Reads the defined parameters (PROFIBUS DP only, see also SFB 81)
SFC 55	WR_PARM	Writes the dynamic parameters (PROFIBUS DP only, see also SFB 53)
SFC 56	WR_DPARM	Writes the default parameters (PROFIBUS DP only, see also SFB 53 and SFB 81)
SFC 57	PARM_MOD	Assigns the parameters to a module (PROFIBUS DP only, see also SFB 53 and SFB 81)
SFC 58	WR_REC	Writes a data record

SFC	Name	Description
SFC 59	RD_REC	Reads a data record
SFC 62	CONTROL	Checks the status of a connection belonging to an SFB instance
SFC 64	TIME_TCK	Reads the system time
SFC 70	GEO_LOG	Determines the start address of a module
SFC 71	LOG_GEO	Determines the slot belonging to a logical address
SFC 78	OB_RT	Reports OB runtime information with a resolution to the nearest microsecond
SFC 79	SET	Sets a range of outputs
SFC 80	RESET	Resets a range of outputs
SFC 82 (Page 99)	CREA_DBL	Creates a data block in load memory
SFC 83 (Page 99)	READ_DBL	Copies data from a block in load memory
SFC 84 (Page 99)	WRIT_DBL	Writes to a load memory block so that the data are stored immediately Load memory blocks that are used to recover from an abnormal termination can be updated while the program is running. Use SFC 84 only for larger segments of a database, not for frequent variable processing.
SFC 85 (Page 99)	CREA_DB	Creates a DB that can be either retentive or non-retentive, depending on the input parameter: If retentive, the current values of the DB are retained after a warm restart (OB 100 (Page 105)). If non-retentive, the current values of the DB are not retained after a warm restart (OB 100).
SFC 87	C_DIAG	Determines the current status of all S7 connections
SFC 100	SET_CLKS	Sets the time of day of your CPU and sets the time-of-day status
SFC 105	READ_SI	Reads out the amount of system resources currently used for message generation with SFC 107 and SFC 108
SFC 106	DEL_SI	Deletes the system resources currently used
SFC 107 SFC 108	ALARM_DQ ALARM_D	Generates a message at each call, to which an accompanying value can be appended
SFC 112	PN_IN	Copies input data from shadow memory of PROFINET CBA components to the associated interface DB
SFC 113	PN_OUT	Copies output data to shadow memory of PROFINET CBA components from the associated interface DB
SFC 114	PN_DP	Updates connections between PROFINET CBA components on the local PROFIBUS and updates connections between PROFINET CBA components on the local PROFIBUS and external PROFINET CBA components
SFC 126	SYNC_PI	Updates the process-image input table in the synchronous cycle
SFC 127	SYNC_PO	Updates the process-image output table in the synchronous cycle

Note

Some SFCs require special consideration regarding the possibility of a Windows blue screen. For more information, refer to "Considerations for SFC 22, SFC 23, and SFCs 82 to 85 (Page 99)".

Reference

Detailed information about the SFCs can be found in the STEP 7 online help or in the *System Software for S7-300/400 System and Standard Functions* reference manual.
If you want to open this manual on a PC on which STEP 7 is installed, select **Start > SIMATIC > Documentation > English** and double-click "STEP 7 - System and Standard Functions for S7-300 and S7-400".

Running asynchronous SFCs concurrently

WinLC RTX restricts the number of asynchronous SFCs that can be running concurrently according to the following rules:

- A maximum of 5 instances of the asynchronous system function SFC 51 (Index B1, B3) may run in WinLC RTX.
- A maximum of 20 asynchronous SFCs from the following SFCs may run in WinLC RTX: SFC 11, SFC 13, SFC 55, SFC 56, SFC 57, SFC 58, and SFC 59.
- A maximum of 32 asynchronous SFCs in any combination from the following SFCs may run in WinLC RTX: SFC 82, SFC 83, and SFC 84.

SFCs that can cause the scan cycle to vary

The following SFCs can cause the scan cycle to vary ("jitter (Page 187)"):

- SFC 22 (CREAT_DB)
- SFC 23 (DEL_DB)
- SFC 52 (WR_USMG)
- SFC 85 (CREA_DB)

Notes for SFC 82, SFC 83, and SFC 84

In contrast to the S7-300, WinLC RTX supports a synchronous interface for SFC 82, SFC 83, and SFC 84 in STARTUP. WinLC allows both the first call (with REQ = 1) and the second call (with REQ = 0) in STARTUP mode so the action can be completed in STARTUP.

The normal STEP 7 error codes apply to SFC 82, SFC 83, and SFC 84. The error code 80C3 is also returned. These SFCs return the error code 80C3, if WinLC RTX exceeds the limit of 32 pending SFC 82, SFC 83, and SFC 84 jobs.

6.6 System Function Blocks (SFBs)

Definition

System function blocks are logic blocks (similar to SFCs) that perform basic tasks when called by the STEP 7 user program. You require a data block (DB) to call an SFB.

List of SFBs

The following table lists the SFBs that WinLC RTX supports:

SFB	Name	Description
SFB 0	CTU	Counts up
SFB 1	CTD	Counts down
SFB 2	CTUD	Counts up/down
SFB 3	TP	Generates a pulse
SFB 4	TON	Generates an ON delay
SFB 5	TOF	Generates an OFF delay
SFB 8	USEND	Sends a data packet of CPU-specific length (two-way), uncoordinated with receiving partner
SFB 9	URCV	Asynchronously receives a data packet of CPU-specific length (two-way)
SFB 12	BSEND	Sends a segmented data block of up to 64 Kbytes (two-way)
SFB 13	BRCV	Receives a segmented data block of up to 64 Kbytes (two-way)
SFB 14	GET	Reads data up to a CPU-specific maximum length (one-way) from a remote CPU
SFB 15	PUT	Writes data up to a CPU-specific maximum length (one-way) to a remote CPU
SFB 19	START	Initiates a warm or cold restart on a remote device
SFB 20	STOP	Changes a remote device to STOP mode
SFB 22	STATUS	Queries the status of a remote device
SFB 23	USTATUS	Receives the status of a remote device
SFB 31	NOTIFY8P	Generates block-related messages without acknowledgement display for eight signals
SFB 32	DRUM	Implements a sequencer
SFB 33	ALARM	Generates block-related messages with acknowledgement display
SFB 34	ALARM_8	Generates block-related messages without values for eight signals
SFB 35	ALARM_8P	Generates block-related messages with values for eight signals
SFB 36	NOTIFY	Generates block-related messages without acknowledgement display
SFB 37	AR_SEND	Sends archive data to operator control and monitoring systems that have logged on for this purpose
SFB 52	RDREC	Reads a data record
SFB 53	WRREC	Writes a data record

SFB	Name	Description
SFB 54	RALRM	Receives alarm data for a PROFIBUS DP slave or PROFINET IO device
SFB 81	RD_DPAR	Reads predefined parameters
SFB 104	IP_CONF	Is used for program-controlled configuration of the integrated PROFINET interface.
SFB 65001	CREA_COM	(WinAC ODK CCX)
SFB 65002	EXEC_COM	(WinAC ODK CCX)
SFB 65003	ASYNC	(WinAC ODK CCX)

Reference

Detailed information about the SFBs can be found in the STEP 7 online help or in the *System Software for S7-300/400 System and Standard Functions* reference manual. If you want to open this manual on a PC on which STEP 7 is installed, select **Start > SIMATIC > Documentation > English** and double-click "STEP 7 - System and Standard Functions for S7-300 and S7-400".

Communication 7

7.1 Overview

Overview

Like other S7 PLCs, WinLC RTX provides S7 communication between controllers on the network. WinAC RTX supports the following communication models:

- PROFIBUS
- PROFINET
- Open user communication (OUC)

Communication blocks

A few blocks specifically support communication. Like other S7 PLCs, WinLC RTX provides open user communication blocks for exchanging data with other TCP/IP communication partners on a PROFINET submodule network. The following blocks are important for communication:

SFB (Page 113)	SFC (Page 109)	FB (Page 147) (open user communication)
SFB 8	SFC 12	FB 63
SFB 9	SFC 14	FB 64
SFB 12	SFC 15	FB 65
SFB 13	SFC 58	FB 66
SFB 14	SFC 59	FB 67
SFB 15	SFC 62	FB 68
SFB 19	SFC 87	
SFB 20		
SFB 22		
SFB 23		
SFB 37		
SFB 81		

Further information about the S7 communication can be found in the STEP 7 documentation.

7.2 Using PROFIBUS

7.2.1 Supported communication interfaces

Definition

A submodule is a configured communication interface that enables communication between WinLC RTX and distributed I/O or between WinLC RTX and STEP 7 or other S7 applications.

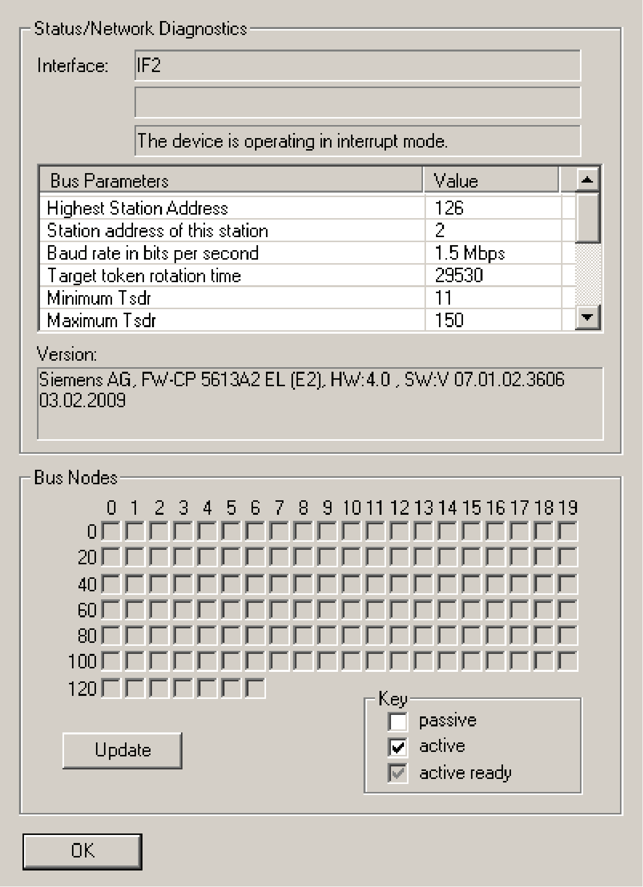
Communication interfaces that can be configured as WinLC RTX submodules

Network adapter	Designation in STEP 7 HW Config
CP 5603	CP 5613/CP 5603/CP 5623
CP 5613 V3 or CP 5613 V6 and higher	CP 5613/CP 5603/CP 5623
CP 5613 A2	CP 5613/CP 5603/CP 5623
CP 5611 A2	CP 5611/CP 5621
CP 5614 A2 (master only)	CP 5613/CP 5603/CP 5623
CP 5614 FO	CP 5613/CP 5603/CP 5623
CP 5621 (PCI Express)	CP 5611/CP 5621
CP 5623 (PCI Express)	CP 5613/CP 5603/CP 5623
CP 5624 (PCI Express, master only)	CP 5613/CP 5603/CP 5623
SIEMENS PC with integrated CP 5611 PROFIBUS interface: ASPC2 STEP E2 or ASPC2 STEP R ASIC	CP 5611/CP 5621

7.2.2 Displaying PROFIBUS submodule diagnostics

You can view communication information for a PROFIBUS submodule.
The "Submodule Network Diagnostics" dialog box displays the following information:

- The current version of the selected CP
- The bus parameters
- All nodes in the communication network
- The status of individual nodes



Status/Network Diagnostics

Interface: IF2

The device is operating in interrupt mode.

Bus Parameters	Value
Highest Station Address	126
Station address of this station	2
Baud rate in bits per second	1.5 Mbps
Target token rotation time	29530
Minimum Tsdr	11
Maximum Tsdr	150

Version:
Siemens AG, FW-CP 5613A2 EL (E2), HW:4.0 , SW:V 07.01.02.3606
03.02.2009

Bus Nodes

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
20	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
40	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
60	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
80	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
100	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
120	□	□	□	□	□	□	□													

Key
□ passive
☑ active
☑ active ready

Procedure

To display the submodule diagnostics, follow these steps:

1. From the Station Configuration Editor, double-click WinLC RTX to open the "WinLC Properties" dialog box.
2. Select an interface slot (IF slot) that a communication interface occupies.
3. Click the "Diagnostics" button.
4. In the "Submodule Network Diagnostics" dialog box, click the "Update" button.
   This action builds the node status display. Querying each node puts an additional load on the communication network.

Result

You are able to monitor diagnostics for a PROFIBUS submodule and view the status of each node in the subnet.

Interrupt mode for a communication interface

For some operations, such as isochronous mode, the communication interface (Page 39) must operate in interrupt mode. You can improve the performance of the communication interface by changing the IRQ settings. Refer to the information about this in Improving the Performance of a Communication Interface (Page 262), Section "Troubleshooting", in the Reference Information.

Note

The submodule diagnostics is available for the CP 56x3 and the CP 56x1, including integrated PROFIBUS interfaces on Siemens PCs. Submodule diagnostics are not available for CP 1604 and CP 1616, including integrated PROFINET interfaces on Siemens PCs.

7.2.3 Testing the configuration of a CP 5613

You can check whether a submodule CP 56x3 is properly configured with the ring test in the WinLC properties. This test is especially important if you have several CPs 56x3 installed in your computer.

Note

This test is not available for CP 56x1.

Procedure

To check the operation of the submodule CP card, follow these steps:

1. Start WinLC RTX if it is not already started. (The ring test is only available when WinLC RTX is operating.)
2. In the Windows taskbar, double-click the 🖥 icon to open the Station Configuration Editor.
3. Double-click the WinLC RTX index entry to display the "WinLC Properties" dialog box.
4. Select the interface slot (IF slot) containing the CP card to be tested.
5. Click the "Ring ON" button.

Result

The LEDs on the CP at the back of your computer flash in an alternating pattern so you can verify that you have configured the correct CP. The computer also emits an audible beep if the CP is functioning.

Ending the test

Select the "Ring OFF" button to end the test of the CP.

7.2.4 Using PROFIBUS DPV1

7.2.4.1 PROFIBUS DPV1

DPV1 extensions to PROFIBUS-DP allow the enhanced communication required by complex slave devices. This enhanced communication includes acyclic data exchange, alarm and status messaging, and the transmission of complex data types. WinLC RTX provides support for the following DPV1 functionality:

- DP-Norm, DP-S7, DPV1, and DPV1 S7-compliant slaves
- Alarm and status OBs for processing DPV1-defined events, including:
  - OB 40 (process alarm)
  - OB 55 (status alarm)
  - OB 56 (update alarm)
  - OB 57 (manufacturer-specified alarm)
  - OB 82 (diagnostic alarm)
  - OB 83 (module pull/plug alarm)
- Data set read and write function blocks:
  - SFB 52 (RDREC), Read Data Set
  - SFB 53 (WRREC), Write Data Set
  - Execution of SFB 54 (RALRM), read alarm data, in the context of the triggering alarm
- Station and interface address
- Buffering of alarms received in DP mode CLEAR

For WinLC RTX to support DPV1, configure the submodule communication interface (Page 39) to be a DP Master (Page 121) in STEP 7.

7.2.4.2 Selecting a DP Master

Procedure

To select a DP Master, follow these steps from the SIMATIC Manager:

1. Open the Hardware Configuration for your PC Station.
2. Double-click your submodule DP interface in the corresponding submodule slot of WinLC RTX.
3. Select the Operating Mode tab of the CP card Properties dialog.
4. Select DP Master and set the DP mode to DPV1.

Result

You have configured a submodule to be a DP Master.

Refer to your STEP 7 documentation for specific information about DPV1 functionality.

Refer also to the following topics in this documentation:

- What Is a Communication Interface? (Page 39)
- Designating a communication interface as a submodule (Page 44)
- Configuring the Hardware in STEP 7 (Page 52)

7.2.5 Data set routing

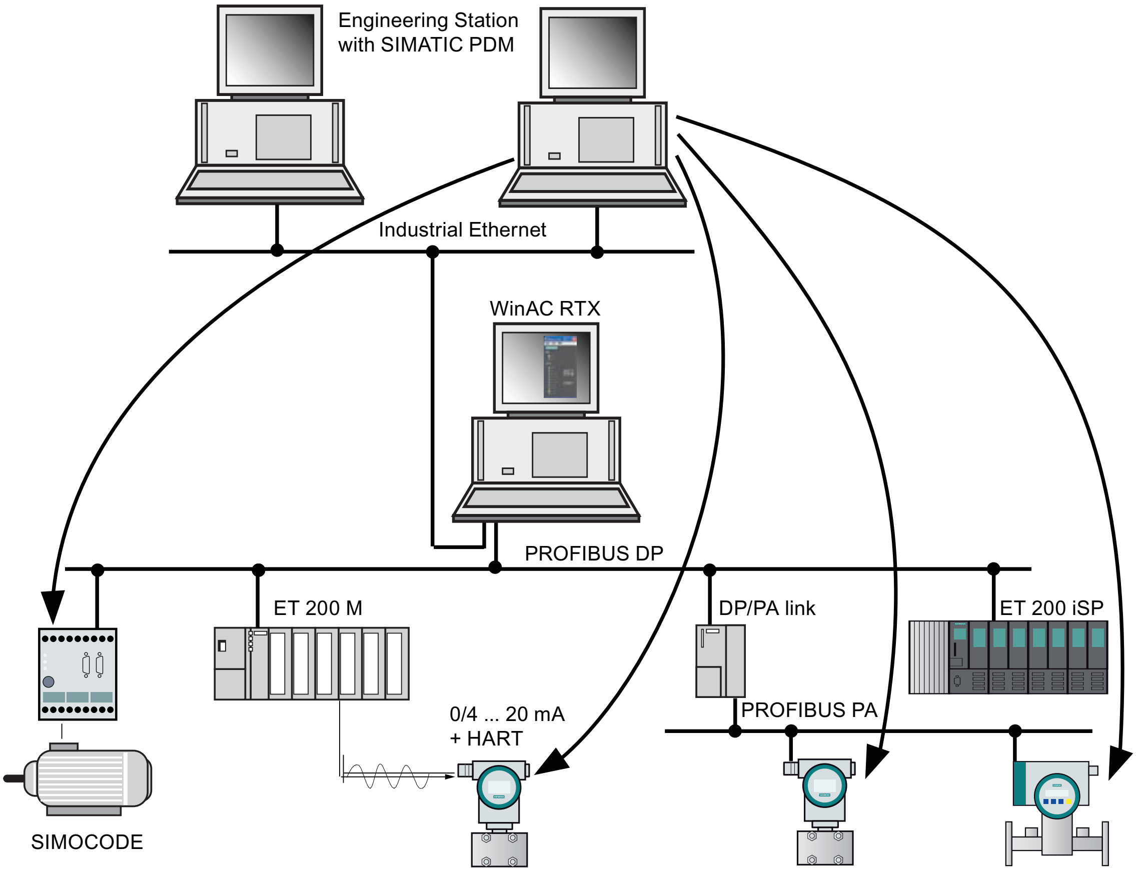
Routing

Routing is the transfer of data beyond network boundaries. You can send information from a transmitter to a receiver across several networks.

Data set routing

Data record routing is a special version of routing. Data record routing supports online functions such as configuration, parameter assignment and diagnostics of PROFIBUS standard devices (also 3rd party).

The following figure shows the access of the engineering station (with associated configuration and diagnostics tools) to various PROFIBUS standard devices. The engineering station is connected to the WinAC RTX via Industrial Ethernet in this scenario. The engineering tools can be operated on the same PC as WinAC RTX. WinAC RTX communicates with the PROFIBUS standard devices using the PROFIBUS.



Note

The structure of the target address for data record routing depends on the data contents, i.e. the device to which the data is sent. The PROFIBUS standard devices themselves do not have to support data record routing, because they do not forward the information received.

Use of PDM, TCI and IO-Link with WinAC RTX

The data record routing functionality enables communication with PROFIBUS for TCI, PDM and IO-Link.

- **TCI** (Tool Calling Interface) for integration of manufacturer-specific engineering tools in STEP 7 to use standard devices (3rd party). Use TCI for configuration, parameter assignment and diagnostics of standard devices.
- A **PDM** (Process Device Manager) embedded in STEP 7 can communicate with corresponding PROFIBUS standard devices on the PROFIBUS.
- An S7-PCT (Port Configuration Tool) embedded in STEP 7 can assign parameters to an **IO-Link** master and the IO-Link Devices in the PROFIBUS network of the WinAC RTX.

Additional information

For additional information about TCI refer to the STEP 7 online help.

You can find additional information on SIMATIC PDM in the *Process Device Manager* manual.

7.2.6 Isochronous Mode for a Constant Bus Cycle

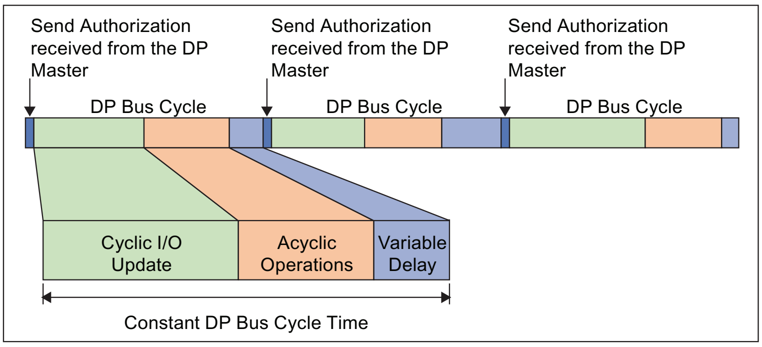
With WinLC RTX, you can operate the DP Master in an isochronous mode to maintain a constant bus cycle time.

Note

WinLC RTX allows you to use isochronous mode on more than one PROFIBUS DP subnet. Assign an exclusive interrupt to the communication interface CP 56x1 and the PCI cards of the communication interface CP 56x3. This assigned interrupt may not be used together with other devices of the Windows operating system (for example with a video card). Check the technical documentation for your specific PC regarding interrupts and PCI slots.

To implement an isochronous DP cycle, you assign a synchronous interrupt OB (OB 61 or OB 62) with an associated process image to the DP master for synchronous update. Each isochronous DP cycle contains the following elements:

- A global control command (Send Authorization) notifies the slave devices of the start of the bus cycle.
- The cyclic inputs and outputs are updated.
- Any acyclic operations are performed.
- A variable delay allows the next DP cycle to start on the next multiple of the configured cycle time.


Send Authorization received from the DP Master
Send Authorization received from the DP Master
Send Authorization received from the DP Master
DP Bus Cycle
DP Bus Cycle
DP Bus Cycle
Cyclic I/O Update
Acyclic Operations
Variable Delay
Constant DP Bus Cycle Time

During the bus cycle, two events signal the STEP 7 user program:

- At the end of the I/O update, an interrupt schedules the synchronous OB for execution.
- At the start of the succeeding cycle (when the Send Authorization command is being transmitted to the slave devices), an event signals WinLC RTX that further execution of SFC 126 and SFC 127 will return an error.

Between the two events (between the interrupt and the transmission of the global control command), the synchronous OB can call SFC 126 and SFC 127 to execute the synchronous updating of the process image partition that was assigned to the synchronous OB.
If these SFC calls execute without error, the I/O update is synchronized to the process image partition update and occurs at a constant interval between updates.

You can configure the DP bus cycle when you configure network properties for the DP master.

System requirements for an isochronous DP cycle

For a isochronous DP cycle, your system requires a CP 56x1 or a CP 56x3 card release version 6 or higher, which is operated in interrupt mode (Page 262). If you use a CP 56x1 communication interface, then the used interrupt may not be shared with other devices.

To determine the revision level, you can use either the "Set PG/PC Interface" utility of STEP 7 or you can view the submodule diagnostics (Page 117).

7.3 Using PROFINET

7.3.1 Supported communication interfaces

CPs for Industrial Ethernet

WinAC RTX 2010 supports the following Ethernet cards:

Network adapter	Chipset	Designation in STEP 7 HW Config
CP 1616 hardware revision 8 and higher (*)	ERTEC 400-1	CP 1616/CP 1604
CP 1604 hardware revision 7 and higher (*)	ERTEC 400-1	CP 1616/CP 1604
S7-mEC CP1616/ERTEC400_EC (integrated)	ERTEC 400-1	PN-IO
SIMATIC PC 427B/477B (integrated) CP 1616	ERTEC 400-1	CP 1616/CP 1604
SIMATIC PC 427C/477C (integrated) CP 1616	ERTEC 400-1	CP 1616/CP 1604
SIMATIC PC 627B/677B (integrated) CP 1616	ERTEC 400-1	CP 1616/CP 1604
SIMATIC Rack PC 847B integrated CP 1616	ERTEC 400-1	CP 1616/CP 1604
SIMATIC PC 627C/677C (integrated) CP 1616	ERTEC 400-1	CP 1616/CP 1604
SIMATIC PC 827C/877C (integrated) CP 1616	ERTEC 400-1	CP 1616/CP 1604
SIMATIC Microbox PC 427B/Panel PC 477B integrated Intel PRO/1000 PL	Intel 82573L	IE General
SIMATIC Box PC 627B/Panel PC 677B integrated Intel PRO/1000 PL	Intel 82573L	IE General
SIMATIC Rack PC 847B integrated Intel PRO/1000 PL	Intel 82573L	IE General
Intel PRO/1000 GT (PCI)	Intel 82541PI	IE General
Intel PRO/1000 PL (integrated)	Intel 82573L	IE General
Intel PRO/1000 PT Dual Port Server Adapter (PCI-Express)	Intel 82571EB	IE General
Intel 9301CT Gigabit CT Desktop Adapter (PCI-Express)	Intel 82574L	IE General
SIMATIC IPC427C/SIMATIC HMI IPC477C integrated Intel 9301CT Gigabit CT Desktop Adapter	Intel 82574L	IE General
SIMATIC IPC627C/SIMATIC HMI IPC677C	Intel 82574L	IE General

Use depends on the required quantity structure and possible fields of application.
Only one of the network adapters can be configured as a submodule of the WinLC.

(*) Port 4 on the hardware is inoperative with WinAC RTX. Only 3 ports are available for use in the hardware configuration of STEP 7.

Note

Operation of the CP 1604, CP 1616, and Intel PRO/1000 GT (PCI) interface modules as a PROFINET interface with WinAC RTX requires an exclusive PCI interrupt (non-shared IRQ). For more information, refer to Chapter "Improving the Performance of a Communication Interface (Page 262)".

7.3.2 Introduction to PROFINET

What is PROFINET?

Within the context of Totally Integrated Automation (TIA), PROFINET is the systematic development of the following systems:

- PROFIBUS DP, the well-established field bus
- Industrial Ethernet, the communication bus for the cell level.

Experiences from both systems have been and are being integrated in PROFINET.

PROFINET as an Ethernet-based automation standard from PROFIBUS International (PROFIBUS Nutzerorganisation e.V.) thereby defines a cross-vendor communication and engineering model.

What is PROFINET IO?

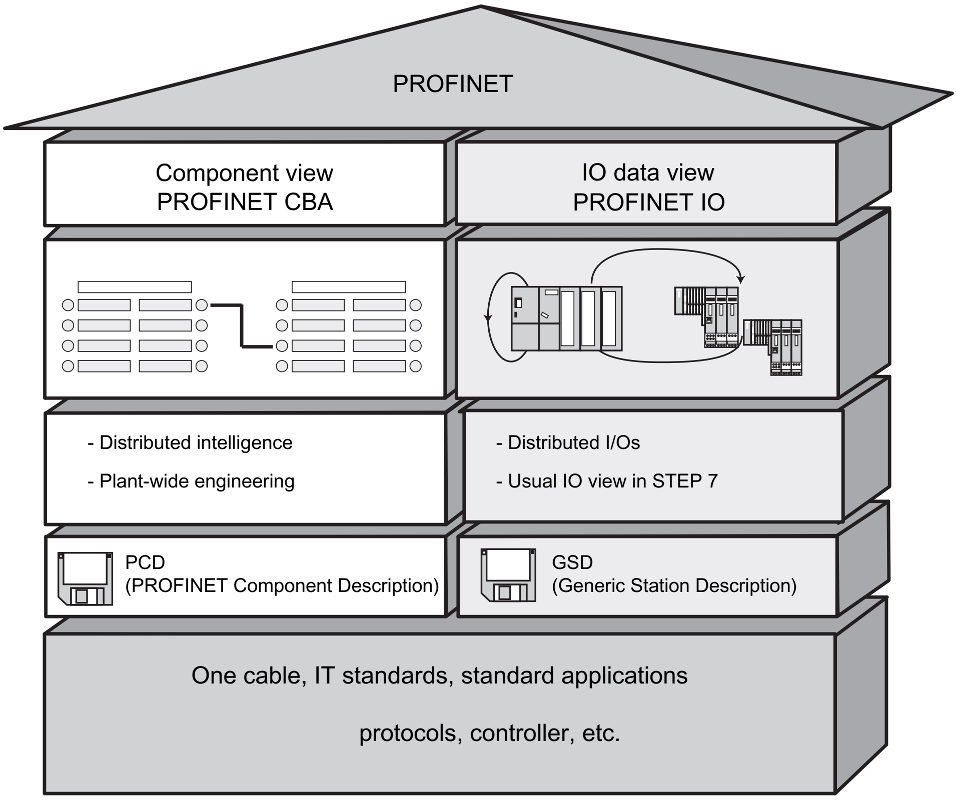
As part of PROFINET, PROFINET IO is a communication concept that is used to implement modular, distributed applications. PROFINET IO allows you to create automation solutions which are familiar to you from PROFIBUS.

WinLC is used as an IO controller for communication via PROFINET IO.

What is PROFINET CBA?

Within the framework of PROFINET, PROFINET CBA is an automation concept for the implementation of applications with distributed intelligence. Component Based Automation allows you to use complete technological modules as standardized components in large systems. PROFINET CBA lets you create distributed automation solutions, based on default components and partial solutions.

The WinLC controller acts like a DP master with proxy functionality during communication via PROFINET CBA.


PROFINET
Component view
PROFINET CBA
IO data view
PROFINET IO
- Distributed intelligence
- Plant-wide engineering
- Distributed I/Os
- Usual IO view in STEP 7
PCD
(PROFINET Component Description)
GSD
(Generic Station Description)
One cable, IT standards, standard applications
protocols, controller, etc.

Note

Only special features of PROFINET if used with WinAC RTX are described in the next chapters.

Reference

Further information on the topic of PROFINET can be found:

- in the "PROFINET System Description" system manual.
- in the "PROFINET IO, Getting Started: Collection" manual.

Additional information on system migration from PROFIBUS DP to PROFINET IO can be found:

- in the "PROFINET IO, From PROFIBUS DP to PROFINET IO" programming manual.

This manual also provides a clear overview of the new PROFINET blocks and system status lists.

7.3.3 PROFINET IO

7.3.3.1 Using PROFINET IO

As part of PROFINET, PROFINET IO is a communication concept that is used to implement modular, distributed applications.

PROFINET IO allows you to create automation solutions which are familiar to you from PROFIBUS.

The STEP 7 engineering tool helps you to structure and configure an automation solution. In STEP 7 you have the same application view, regardless of whether you are configuring PROFINET devices or PROFIBUS devices. You will program your STEP 7 user program in the same way for both PROFINET IO and PROFIBUS DP because you will use the extended blocks and system status lists for PROFINET IO.

PROFINET uses TCP/IP and IT standards as the Ethernet-based automation standard. PROFINET IO ensures communication between IO controllers and IO devices.

Devices capable of PROFINET (PN) communication

- PROFINET IO devices (for example, interface module IM 151-3 PN in an ET 200S)
- PROFINET CBA components
- I-Devices can be assigned to PROFINET IO systems (for example, SIMATIC ET 200S, IM151-8 PN/DP CPU firmware V3.2 or higher, CPU S7-31x (F))
- S7-300 (F)/S7-400 (F) with PROFINET interface (for example, CPU 317-2 PN/DP or CP 343-1)
- Active network components (a switch, for example)
- PG/PC with Ethernet card
- IE/PB-Link

Properties of the PROFINET interface

IEEE standard	802.3
Connector design	RJ45
Transmission rate	Up to 100 Mbps
Protocols and communication functions	PROFINET IO PROFINET CBA PROFINET-Standard according to IEC 61784-2, Conformance Class A and B open block communication via TCP, ISO on TCP and UDP S7 Communication PG Functions SNMP LLDP time synchronization using the NTP method as a client

Converting a STEP 7 user program to PROFINET IO

You can convert PROFIBUS DP STEP 7 user programs to PROFINET IO. This requires only a few changes to be made. The program does not have to be created from scratch.

If the OPC interface is used, the dynamics of the PROFINET IO OPC server and handling of items for services remains the same.

Only conversion tasks typical for SIMATIC CPUs such as the S7-300 and the S7-400 need to be performed.

7.3.3.2 Overview - PROFINET IO and WinAC RTX

The communications model of PROFINET IO was developed to benefit from the communication system Industrial Ethernet without redesigning the STEP 7 user programs and without losses in performance and deterministics.

With this cyclic data exchange, the WinLC controller is the master (IO controller) and the distributed I/Os take on the role of the slave (IO device).

PROFINET devices can be integrated in a PC-based PLC if the following requirements are met:

- WinAC RTX software package with a WinLC RTX V4.4 (or higher) controller
- STEP 7 V5.4 Service Pack 4 or higher version

Topology

PROFINET devices in industrial plants are linked via wired or wireless components. An overview of the network components can be found in the "PROFINET System Description" user manual in chapter 3 "Setting up PROFINET".

PROFINET IO supports the established network structures.

- Linear bus topology
- Star topology
- Tree topology

Different topologies can be combined with switches and routers. Switches with up to 10 ports (8 electrical and 2 optical) are available. A redundancy manager closes the open ends of the linear bus topology to form a ring topology.

7.3.3.3 Configuring WinLC RTX as a PN IO controller in STEP 7

This section describes how to configure a WinLC RTX controller as a PN IO controller in STEP 7.

Configuration in the hardware configuration of STEP 7

1. Set up a PROFINET IO system in the STEP 7 hardware configuration as described in Chapter "Configuring the Hardware in STEP 7 (Page 52)".
2. Ensure that you add a PROFINET-capable communications processor on the same IF slot as in the Station Configuration Editor.

   The "Properties" dialog box opens.
3. Set the options of the PROFINET interface.
4. Verify the default IP address, device name and subnet mask.
5. Add the IO devices to the IO system.
6. Configure the IO system in the Topology Editor.

   Further steps are described in the help for STEP 7.

Assigning an IP address and a device name for identification on PROFINET

During communication with PROFINET IO the individual system nodes are not identified by an integer address as with PROFIBUS. As PROFINET is based on the TCP/IP and IT standards, all PROFINET devices are identified and addressed by means of a unique IP address. The IP address is assigned using the STEP 7 software.

The IP address is made up of 4 decimal numbers with a range of values from 0 through 255. The decimal numbers are separated by periods. The IP address is made up of:

- The address of the (subnet) network
- The address of the node (generally called the host or network node)

The IP address is assigned permanently to the device name. Therefore an IO device must have a device name before it can be addressed by an IO controller. This procedure was selected for PROFINET because names are easier to handle than complex IP addresses. Assignment of a device name for a concrete IO device can be compared with the setting of the PROFIBUS address for a DP slave.

Rules for the assignment of device names

You must conform to the following rules when assigning names to PROFINET IO devices:

- The PROFINET IO controller, the PROFINET IO devices, and the Engineering System are located on the same subnet.
- The PROFINET IO controller is interconnected with the Engineering System via the PROFINET IO interface.

PG/PC interface settings

Set up the PG/PC interface as described in Chapter "Connecting STEP 7 to the controller (Page 49)".

7.3.3.4 Loading configuration data in WinLC RTX

Requirement

The following actions must be completed before the configuration data can be loaded in the target device:

- Configuring submodules of the WinLC RTX
- Configuring WinLC RTX as a PN IO controller in STEP 7

Procedure

1. Set the PG/PC interface.
2. Start WinLC RTX.
3. Load the configuration data in WinLC RTX.

Reference

Additional information on the topic:

- Loading configuration data
- PG/PC interface settings

can be found in Chapter "Connecting STEP 7 to the controller (Page 49)".

7.3.3.5 New blocks

Comparison of the Blocks and System and Standard Functions of PROFINET IO and PROFIBUS DP

If a system is migrated from PROFIBUS DP to PROFINET IO, there are some blocks and system and standard functions which are new or which have to be replaced.

For additional information on this topic, refer to chapter 3 "Blocks in PROFINET IO and PROFIBUS DP" in the user manual "PROFINET IO, From PROFIBUS DP to PROFINET IO".

The table below provides an overview of the functions and blocks which have to be replaced by newer ones or which can be emulated in PROFINET IO.

Blocks/functions	PROFINET IO	PROFIBUS DP
SFC5 (query start address of a module)	No Replacement: SFC 70	Yes
SFC 13 (read diagnostic data of a DP slave)	No Replacement: • event-driven: SFB 54 • status-driven: SFB 52	Yes
SFC49 (query the slot at a logical address)	No Replacement: SFC 71	Yes
SFC54 (read default parameters - S7-400 CPU only)	No Replacement: SFB 81	Yes
SFC 55 (write dynamic parameters)	No Emulate using SFB53	Yes
SFC56 (write predefined parameters)	No Emulate using SFB81 and SFB53	Yes
SFC 57 (assign module parameters)	No Emulate using SFB81 and SFB53	Yes
SFC58/59 (write/read record in I/O)	No Replacement: SFB 53/52	Yes should already have been replaced by SFB 53/52 in DPV1
OB83 (hot swapping of modules/submodules)	Yes	Yes
OB86 (rack failure)	New error information	Unchanged

The following SIMATIC system functions are not supported for PROFINET IO:

- SFC11 (synchronize groups of DP slaves)

Note

New system status lists

New system status lists are available to cover PROFINET IO functionality.

A listing of all system status lists is available in Chapter "System Status List (SSL) (Page 266)".

7.3.3.6 Prioritized startup

Prioritized startup describes the PROFINET functionality for the acceleration of IO devices (distributed I/O) in a PROFINET IO system with RT and IRT communication.

The function reduces the time that the correspondingly configured IO devices require in order to return to cyclic user data exchange in the following cases:

- After the supply voltage has returned
- After a station has returned
- After IO Devices have been activated

Note

Firmware update

In prioritized startup it is not possible to perform a firmware update via Micro Memory Card.

Note

Startup times

The startup time depends on the number and type of modules.

Note

Prioritized startup and media redundancy

The inclusion of a IO device with prioritized startup in a ring topology with media redundancy is not possible.

Additional information

For additional information, refer to the STEP 7 online help and the PROFINET System Description (http://support.automation.siemens.com/WW/view/en/19292127) manual.

7.3.3.7 Device replacement without exchangeable media/programming device

IO devices having this function can be replaced in a simple manner:

- A removable medium (such as Micro Memory Card) with stored device name is not required.
- The device name does not have to be assigned with the programming device.

  The loaded IO device is assigned a device name from the IO controller and not from the removable medium or programming device. To do this, the IO controller uses the configured topology and the neighborhood relationships determined by the IO devices. The configured desired topology must agree with the actual topology.

Reset the IO devices, which were already in operation, back to the factory settings before using them again.

Additional information

For additional information, refer to the STEP 7 online help and the PROFINET System Description (http://support.automation.siemens.com/WW/view/en/19292127) manual.

7.3.3.8 Changing IO devices during operation

If the IO controller and the IO devices support this functionality, then a IO device port can be assigned "varying partner ports" from other devices per configuration so that communication can take place with each of the varying IO devices at a particular point in time via this port. A physical connection may only exist between the alternating device and the interchangeable port where the communication should be taking place.

Additional information

For additional information, refer to the STEP 7 online help and the PROFINET System Description (http://support.automation.siemens.com/WW/view/en/19292127) manual.

7.3.3.9 IRT communication

Introduction

IRT is a transmission procedure in which the PROFINET devices are synchronized with very high precision. A Sync-Master specifies the cycle, Sync slaves synchronize to this cycle. Both an IO controller and an IO device can adopt the role of Sync-Master.

IRT Options

WinAC supports both IRT options:

- IRT option "high flexibility"
- IRT option "high performance"

Note

Prerequisite for IRT communication with the IRT option "high performance" is the topology configuration.

Further information

For further information about the topics "IRT" and "Configuring the topology" please refer to the online help for STEP 7 and the PROFINET *system description*.

7.3.3.10 Isochronous mode on PROFINET

What is isochronous mode?

The fast and reliable response time of a system operating in isochronous mode is based on the fact that all data is provided just-in-time. The constant-bus-cycle-time PROFINET IO cycle is the clock generator for it.

The system throughput times are constant through the "Isochronous mode" system characteristic. WinAC RTX is strongly deterministic through the PROFINET IO line.

The application of the synchronous cycle interrupt OBs (OB 61 or OB 62) and the SFCs (SFC126/127) is carried out in the same manner as the use of PROFIBUS.

Requirements

The following requirements must be fulfilled in order to execute the isochronous mode.

- A PC with at least a DualCore processor is required for clock rates of less than 1 ms.
- A communication interface CP1616 is configured.
- The Ethernet interface is configured with an exclusive PCI interrupt (non shared IRQ).

> Note
>
> **Optimum performance utilization**
>
> For optimum performance you should avoid a communications load parallel to PROFINET IO (for example, PROFINET CBA, Open Communication, PROFIBUS, S7 communication, HMI communication).

Devices operated in non-isochronous mode

The following devices cannot be operated in isochronous mode:

- Shared Device
- I-devices at the higher-level IO controller

Additional information

For additional information about minimizing the jitter, refer to the chapter Causes of Jitter (Page 187).

For additional information regarding the use of the synchronous cycle interrupt OBs please refer to Chapter Isochronous Mode for a Constant Bus Cycle (Page 123).

For additional information regarding the ERTEC based Ethernet interfaces please refer to Chapter Supported communication interfaces (Page 125).

For additional information regarding parameter assignment please refer to the PROFINET *System Description*.

7.3.3.11 Shared Device functionality

Shared Device functionality

Numerous IO controllers are often used in larger or widely distributed systems. In this case it is possible that sensors located near each other have to supply data to different IO controllers. In the past this had to be solved through several IO devices that were assigned to different IO controllers. The Shared Device functionality makes it possible to divide an IO device between different IO controllers and thus to save one or more IO devices.

Requirements

The IO controller and the Shared Device have to be located on the same PROFINET line so that the Shared Device function can be used.

Configuration options

The modules of the Shared Device are divided between the individual IO controllers. In the process each module of the Shared Device can be assigned exclusively to an IO controller. Assignment of the individual modules is carried out in HW Config.

Shared Device in the same STEP 7 project:

- Two IO controllers share the submodules of an IO device. The two IO controllers are located in the same STEP 7 project.

Shared Device in different STEP 7 projects:

- Two IO controllers share the submodules of an IO device. The two IO controllers are located in different STEP 7 projects.

Additional information

For further information about the topic of a "Shared Device" refer to the PROFINET system description (http://support.automation.siemens.com/WW/view/en/19292127).

7.3.3.12 Media redundancy

Media redundancy functionality

Media redundancy is the function for ensuring the network and system availability. Redundant transmission links (ring topology) ensure that an alternative communication path is made available if a transmission link fails.

For the IO devices the media redundancy protocol (MRP) can be activated that is part of the PROFINET standardization in accordance with IEC 61158.

Structure of a ring topology

Nodes in a ring topology can be external switches and/or the integrated switches of communications modules.

To set up a ring topology with media redundancy, you bring together the two free ends of a linear bus topology in one device. Closing the linear bus topology to form a ring is achieved with two ports (ring ports: Port 1 or Port 2 of the PROFINET CPUs) of a device in the ring. This device is the redundancy manager. All other devices in the ring are redundancy clients.

The two ring ports of a device are the ports that establish the connection to the two neighboring devices in the ring topology. The ring ports are selected and set in the configuration of the relevant device.

On the module the ring ports are indicated by an "R" after the port number.

Note

If IRT communication is used, media redundancy is not supported.

Further information

For further information about the topic of "Media redundancy" refer to the STEP 7 online help and the PROFINET system description (http://support.automation.siemens.com/WW/view/en/19292127) manual.

7.3.4 PROFINET CBA

7.3.4.1 Using PROFINET CBA

PROFINET CBA supports real-time communication in WinAC RTX.

PROFINET IO and CBA can be operated simultaneously via an Ethernet network. Communication resources must be specified appropriately as they they share the same network. Further information on this can be found in Chapter "Configuring Hardware (Page 140)".

Basic procedure for use in component-based automation

To be able to integrate WinLC RTX in PROFINET communication,
you must perform the following tasks:

1. Set up the Ethernet submodule in the Station Configuration Editor.
2. Configure the hardware HW Config of STEP 7.
3. Create a PROFINET interface DB in STEP 7.
4. Create a PROFINET component in STEP 7.
5. Configure and generate PROFINET communication in SIMATIC iMap.
6. Load the configuration data in the WinLC RTX controller.

Reference

The individual steps will now be briefly described.

They are described in detail in the "SIMATIC iMap" manuals.

7.3.4.2 Possible Configurations

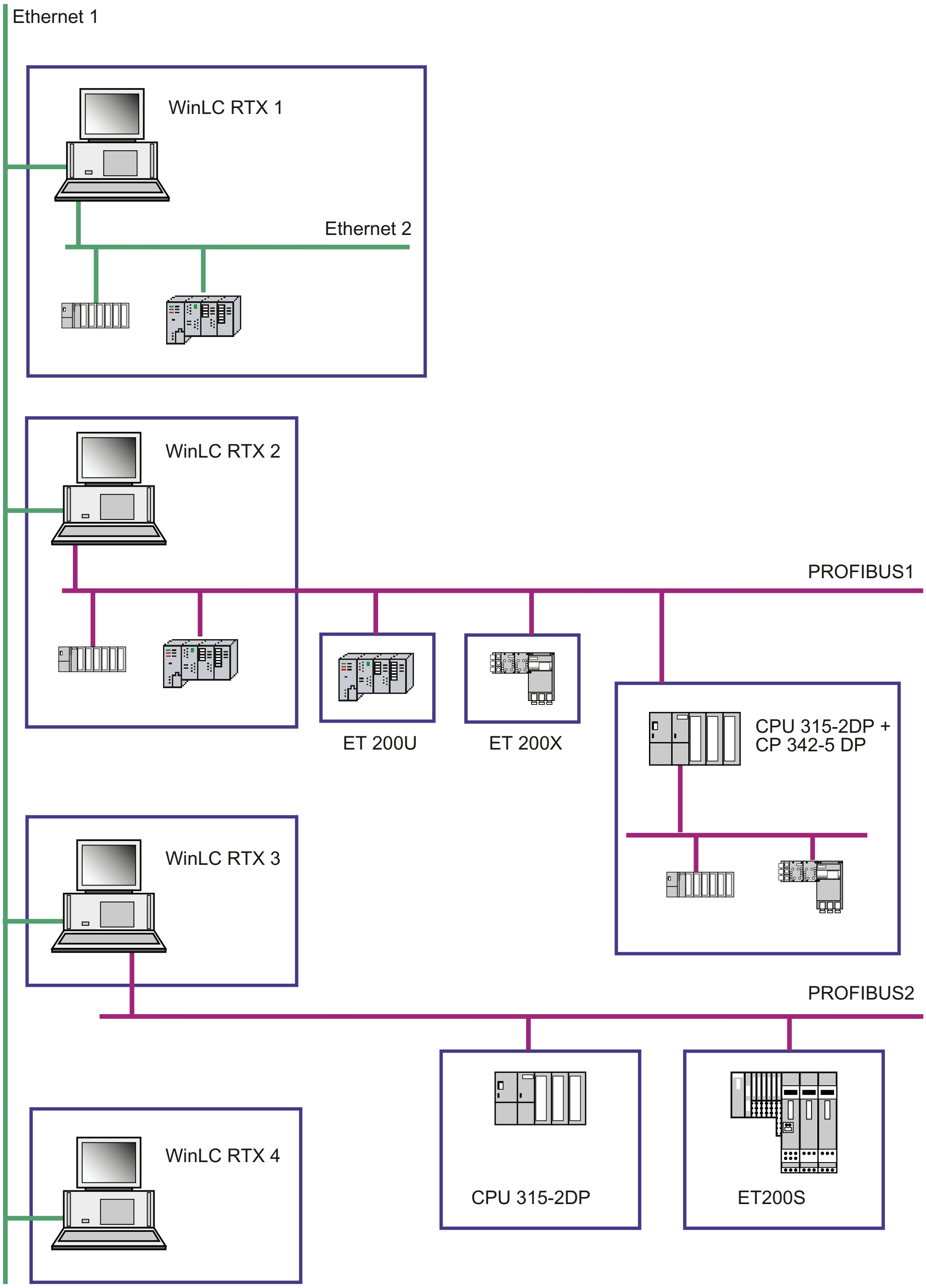
Examples of PROFINET components

There are four ways of carrying out hardware configurations for WinLC RTX a PROFINET components.

- WinLC RTX 1 with proxy functionality as an IO controller
- WinLC RTX 2 with proxy functionality and local DP master system
- WinLC RTX 3 with proxy functionality
- WinLC RTX 4 without proxy functionality

Example Facility Configuration with WinLC RTX

The figure below schematically illustrates an example system containing the four possible configurations for WinLC RTX acting as a PROFINET component. Each frame in the figure designates one PROFINET component.



7.3.4.3 Configuring Hardware

Introduction

The hardware must be configured in the Station Configuration Editor and in HW Config of STEP 7. The station name, index and IF slots must be identical.

The configuration must contain the following modules:

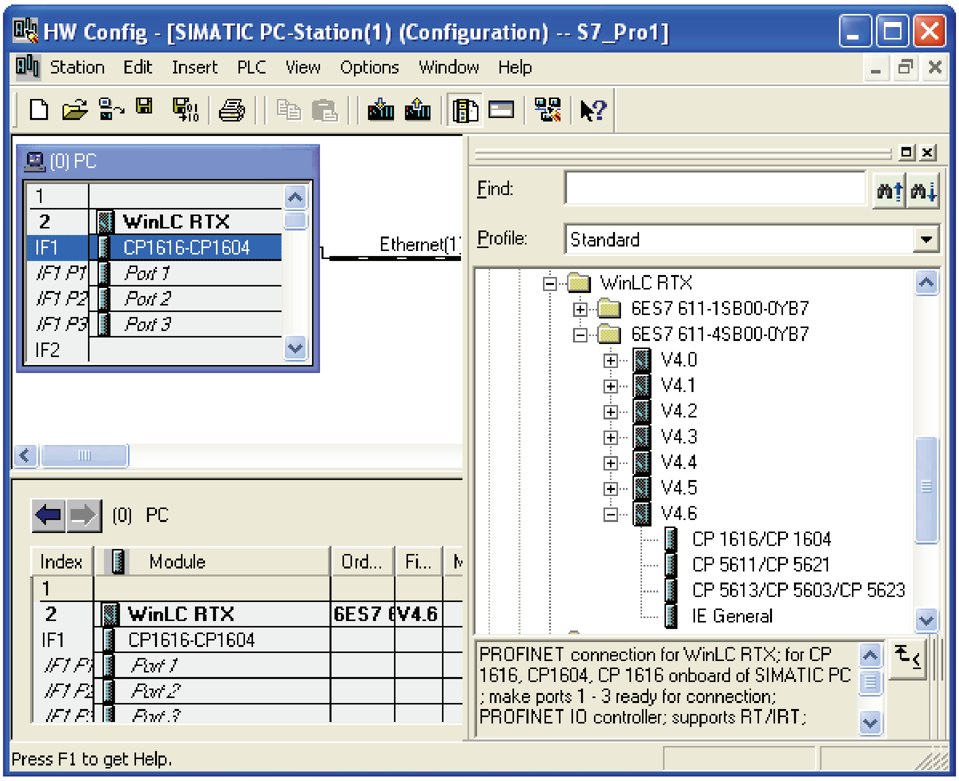
- WinLC RTX V4.4 and higher
- CP 1604/1616 or IE General for Industrial Ethernet – mandatory
- Either a CP 56x1 or CP 56x3 for PROFIBUS DP as a submodule on a IF socket of the PROFIBUS-CP is required for configuration as a DP master with proxy functionality.

Configuring Ethernet submodules in the "WinLC Properties" dialog

1. Configure the WinLC RTX submodules as described in Chapter "Configuring communication interfaces (Page 44)".
2. Ensure that you assign a PROFINET communication interface to an interface slot.

Configuring hardware in HW Config of STEP 7

1. Create a project in SIMATIC Manager and insert a SIMATIC PC station.
   The station name must be identical to that in the Station Configuration Editor.
2. Configure the hardware based on the following diagram:



1	
2	WinLC RTX
IF1	CP1616-CP1604
IF1 P1	Port 1
IF1 P2	Port 2
IF1 P3	Port 3
IF2	

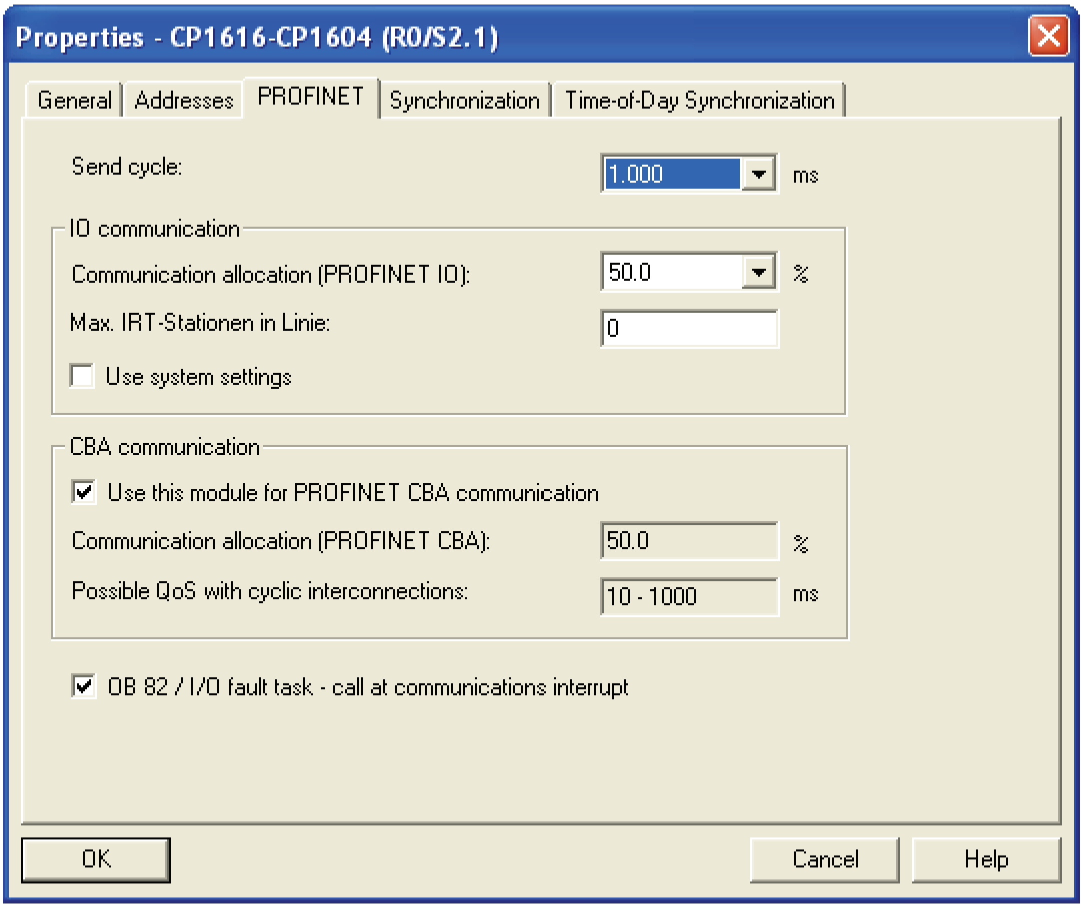
(0) PC

Index	Module	Ord...	Fi...	M
1				
2	WinLC RTX	6ES7 6	V4.6	
IF1	CP1616-CP1604			
IF1 P1	Port 1			
IF1 P2	Port 2			
IF1 P3	Port 3			

- WinLC RTX
  - 6ES7 611-1SB00-0YB7
  - 6ES7 611-4SB00-0YB7
    - V4.0
    - V4.1
    - V4.2
    - V4.3
    - V4.4
    - V4.5
    - V4.6
      - CP 1616/CP 1604
      - CP 5611/CP 5621
      - CP 5613/CP 5603/CP 5623
      - IE General

PROFINET connection for WinLC RTX; for CP 1616, CP1604, CP 1616 onboard of SIMATIC PC ; make ports 1 - 3 ready for connection; PROFINET IO controller; supports RT/IRT;

3. Configure a PROFINET IO system for the PROFINET-CP and link it.
4. In the dialog field "Properties CP 1604-1616" in the field "CBA communication" activate the option "Use this module for PROFINET CBA communication".
5. Assign PROFINET IO and PROFINET CBA to the appropriate volume of communication traffic.



Properties - CP1616-CP1604 (R0/S2.1)

Send cycle: 1.000 ms

IO communication

Communication allocation (PROFINET IO): 50.0 %

Max. IRT-Stationen in Linie: 0

□ Use system settings

CBA communication

☑ Use this module for PROFINET CBA communication

Communication allocation (PROFINET CBA): 50.0 %

Possible QoS with cyclic interconnections: 10 - 1000 ms

☑ OB 82 / I/O fault task - call at communications interrupt

6. Save and compile the HW configuration and close HW Config.

7.3.4.4 Creating the PROFINET interface DB

Prerequisites

Before you can create an interface DB, the hardware configuration of the PC station must be completed.

Procedure

1. In SIMATIC Manager, select the SIMATIC PC station and select **Create PROFINET interface** in the shortcut menu.

   The "New/Open PROFINET Interface" dialog box is opened.
2. Select WinLC RTX in the left window of the "New/Open PROFINET Interface" dialog. Activate the "New" option and confirm by clicking the "OK" button.

   The next dialog box shows the properties of the new block to create.
3. In the "Name and type" field, enter the desired block number, DB100 for example, and select the block type, "Global DB".
4. Click on the "OK" button.

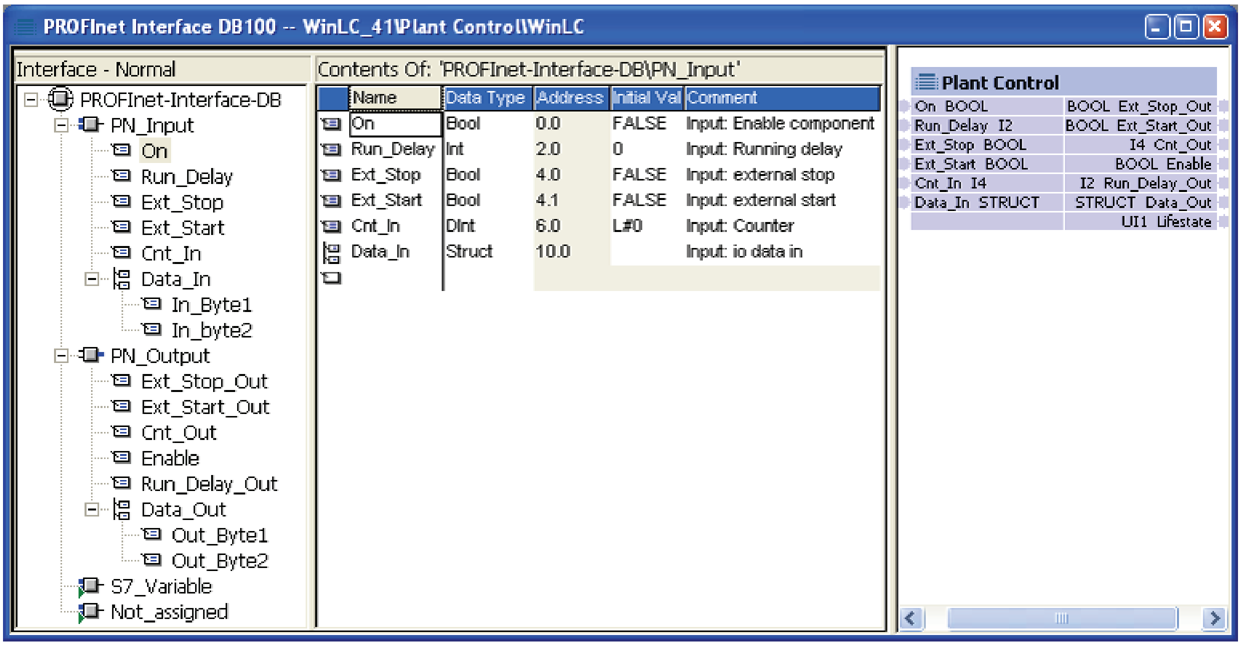
   The interface DB is opened in the PROFINET interface editor.
5. Enter the inputs of the technological function in the PN_Input section and the outputs of the technological function in the PN_Output section, and assign the entries the required properties: name, data type, interconnectable, HMI/MES
6. Save the PROFINET interface DB.

Result

The inputs and outputs for PROFINET communication are defined.

The interface DB is saved in the block folder of the PC station.

The following picture shows an example interface DB in the PROFINET interface editor.



PROFInet Interface DB100 -- WinLC_41\Plant Control\WinLC

Interface - Normal

- PROFInet-Interface-DB
  - PN_Input
    - On
    - Run_Delay
    - Ext_Stop
    - Ext_Start
    - Cnt_In
    - Data_In
      - In_Byte1
      - In_byte2
  - PN_Output
    - Ext_Stop_Out
    - Ext_Start_Out
    - Cnt_Out
    - Enable
    - Run_Delay_Out
    - Data_Out
      - Out_Byte1
      - Out_Byte2
  - S7_Variable
  - Not_assigned

Contents Of: 'PROFInet-Interface-DB\PN_Input'

Name	Data Type	Address	Initial Val	Comment
On	Bool	0.0	FALSE	Input: Enable component
Run_Delay	Int	2.0	0	Input: Running delay
Ext_Stop	Bool	4.0	FALSE	Input: external stop
Ext_Start	Bool	4.1	FALSE	Input: external start
Cnt_In	DInt	6.0	L#0	Input: Counter
Data_In	Struct	10.0		Input: io data in

Plant Control

On BOOL	BOOL Ext_Stop_Out
Run_Delay I2	BOOL Ext_Start_Out
Ext_Stop BOOL	I4 Cnt_Out
Ext_Start BOOL	BOOL Enable
Cnt_In I4	I2 Run_Delay_Out
Data_In STRUCT	STRUCT Data_Out
	UI1 Lifestate

Reference

Additional information about creating an interface DB can be found under "Properties of the PROFINET interface" in the SIMATIC iMap or SIMATIC Manager basic help.

7.3.4.5 Creating PROFINET components

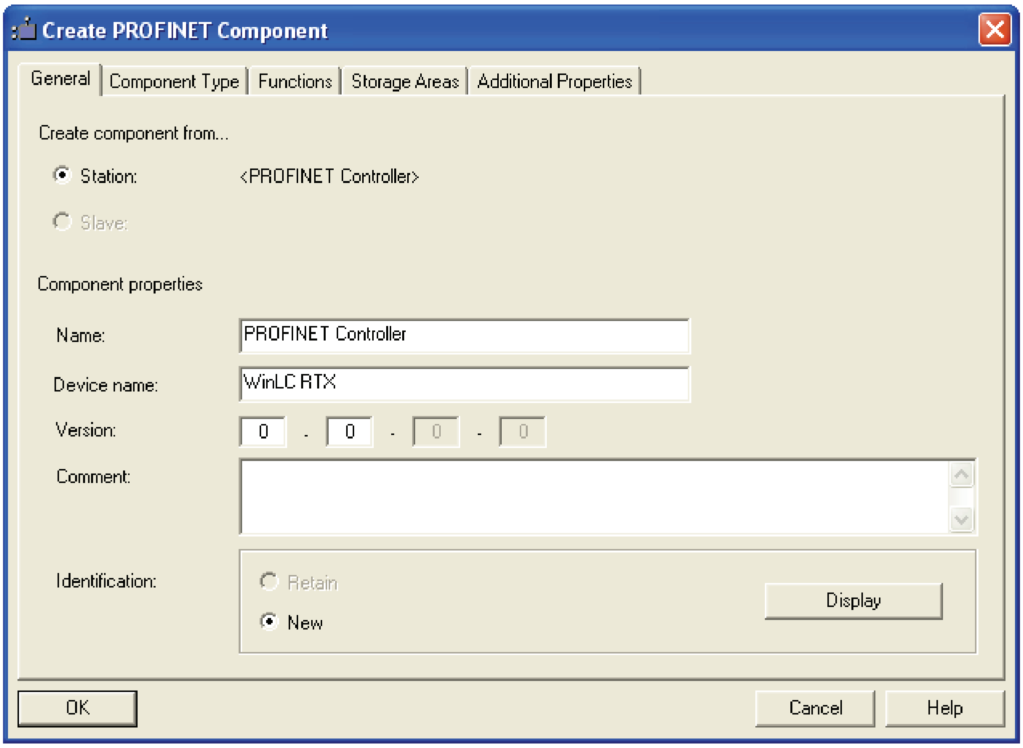
Prerequisites

- Hardware configuration of the PC station is completed.
- The interface DB has been created.

Procedure

1. In SIMATIC Manager, select the SIMATIC PC station and select **Create PROFINET component** in the shortcut menu.

   The "Create PROFINET component" dialog box opens.
2. Activate the option "Identification: New" in the "General" tab and enter a name.



Create PROFINET Component

General | Component Type | Functions | Storage Areas | Additional Properties

Create component from...

● Station: <PROFINET Controller>

○ Slave:

Component properties

Name: PROFINET Controller

Device name: WinLC RTX

Version: 0 . 0 . 0 . 0

Comment:

Identification: ○ Retain ● New

3. In the "Component type" tab, select:
   - Standard component with proxy functionality
   - Update of the PN interface - automatically (at the cycle checkpoint)
4. The "Functions" tab contains information about the technological function or subfunctions of the PROFINET component and which interface blocks are assigned.
5. Enter a storage location in the file system in the "File locations" tab.
6. In the "Additional properties" tab, enter the path of the icon files, and the path of the documentation link. You can also use the included icons (default path: Step7\s7data\s7cbac1x).

Result

The PROFINET component will be stored as an XML file at the specified file location along with the archived component project.

7.3.4.6 Configuring PROFINET communication in SIMATIC iMap.

Requirement

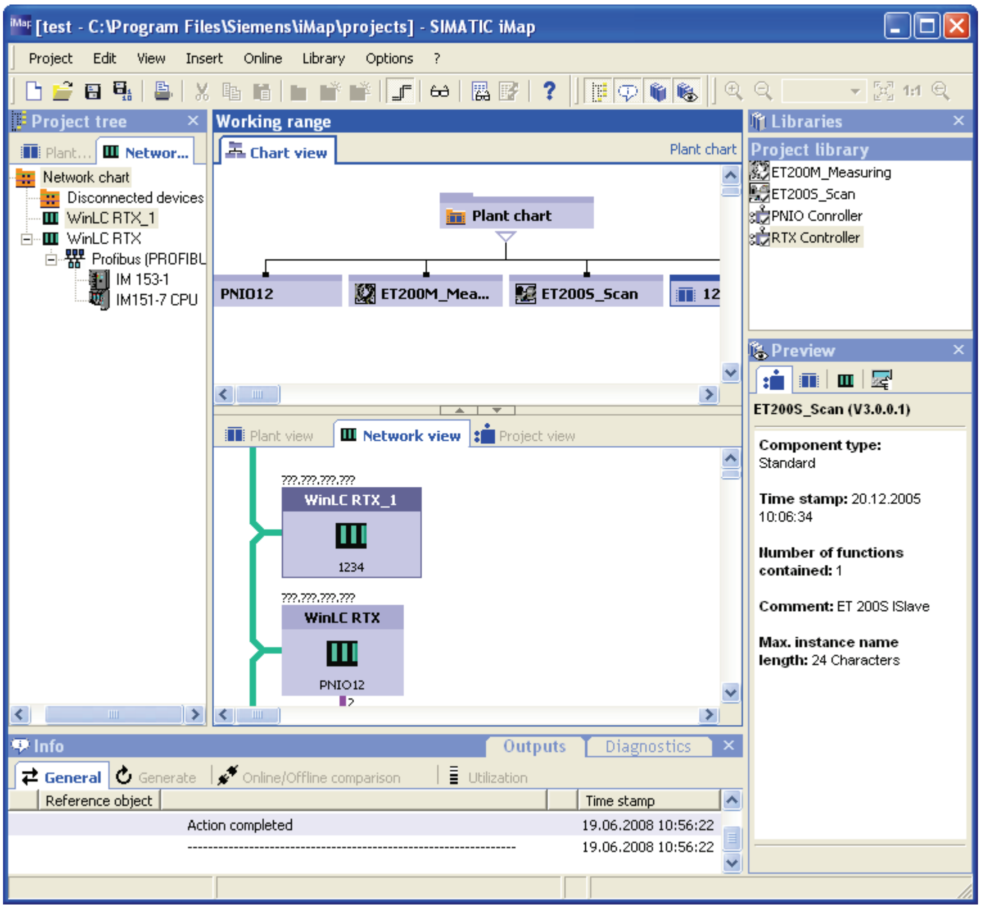
The following actions must have been completed prior to configuration in SIMATIC iMAP:

- Hardware Configuration
- Creating the PROFINET interface DB
- Creating a PROFINET component

Procedure

1. Start SIMATIC iMAP.
2. Import the required components in the project library and move them to the "plant tree" or "plan view" using a drag & drop operation.
3. Configure the communication connection in the "network" view of the work area.
4. In the "network" view of the work areas, assign IP addresses and the subnet mask to all devices.

   The project cannot be generated without assigning an IP address.
5. Interconnect the appropriate inputs and outputs in the "plant view".



ET200S_Scan (V3.0.0.1)

Component type: Standard

Time stamp: 20.12.2005 10:06:34

Number of functions contained: 1

Comment: ET 200S ISlave

Max. instance name length: 24 Characters

Reference object		Time stamp
	Action completed	19.06.2008 10:56:22
	--	19.06.2008 10:56:22

6. Save the project.
7. Generate the project.

7.3.4.7 Loading configuration data in WinLC RTX

Requirement

The following actions must be completed before the configuration data can be loaded in the target device:

- Hardware Configuration
- Creating the PROFINET interface DB
- Creating a PROFINET component
- Configuring PROFINET communication in SIMATIC iMap.

Procedure

1. Set the PG/PC interface.
2. Start WinLC RTX.
3. Select the WinLC RTX target device and select the shortcut menu command **Download selected instances > All**

Result

The configuration data are loaded in WinLC RTX.

WinLC RTX is operational and can be viewed and diagnosed online with SIMATIC iMap.

Reference

Additional information on the topic:

- Loading configuration data
- PG/PC interface settings

can be found in Chapter "Connecting STEP 7 to the controller (Page 49)".

7.3.5 SNMP Communication Service

Availability

WinAC RTX supports the SNMP V1 (MIB-II) network protocol. Applications based on SNMP can be operated on the same network in parallel with PROFINET applications.

Properties

SNMP (Simple Network Management Protocol) is a standard protocol for TCP/IP networks.

Reference

For further information on the SNMP communication service and diagnostics with SNMP, refer to the PROFINET *system description*.

7.4 Using open communication via a PROFINET interface of WinAC RTX

7.4.1 Overview

Introduction

Open User Communication via Industrial Ethernet supports the protocol variants "connection-oriented" and "connectionless" for data communication.

How the function blocks actually function depends on the protocol variant being used.

Note

This functionality is only available via a PROFINET interface of WinAC RTX (as a submodule).

Properties of the communication protocols

The following protocol types exist in data communication:

- Connection-oriented protocols:

  Prior to data transmission, these protocols establish a (logical) connection to the communication peer and close it again, if necessary, after transmission is completed. Connection-oriented protocols are used when security is especially important in data transmission. Usually several logical connections can be established via a physical cable.

  The FBs for open communication via Industrial Ethernet support the following connection-oriented protocols:

  – TCP according to RFC 793 (connection type B#16#11)

  – ISO-on-TCP according to RFC 1006 (connection type B#16#12)

- Connectionless protocols:

  These protocols operate without a connection. This means that no connections to a remote peer are established and terminated. Connectionless protocols transfer the data unacknowledged and thus unsecured to the remote partner.

  For the FBs to open communication by means of Industrial Ethernet, the following connection oriented protocols are supported:

  – UDP according to RFC 768 (connection type B#16#13)

The blocks for Open User Communication via Industrial Ethernet

STEP 7 provides the following FBs and UDTs under "Communication blocks" in the "Standard Library" for data exchange with communication partners:

Block	Name	Description	TCP, ISO-on-TCP	UDP
FB 63	TSEND	Send data to a remote partner via TCP	x	
FB 64	TRCV	Receive data from a remote partner via TCP	x	
FB 65	TCON	Establish connection	x	x
FB 66	TDISCON	Disconnect	x	x
FB 67	TUSEND	Send data to a remote partner via UDP		x
FB 68	TURVC	Receive data from a remote partner via UDP		x
UDT 65	TCON_PAR	Connection-oriented protocols: with the data structure for assigning connection parameters Connectionless protocol: with the data structure for assigning parameters for the local communications access point	x	x
UDT 66	TADDR_PAR	Connectionless protocol: with the data structure for assigning addressing parameters for the remote partner		x
UDT 651	TCP_conn_active	Connection-oriented protocols: with protocol-specific preassignments	x	
UDT 652	TCP_conn_passive	Connection-oriented protocols: With protocol-specific preassignments	x	
UDT 657	UDP_local_open	Connectionless protocol: With protocol-specific preassignments		x
UDT 661	UDP_rem_address and port	Connectionless protocol: With protocol-specific preassignments		x

The use of these blocks enables the controller to speak to any communication partner via any protocol via Ethernet.

Quantity framework

For information on connection parameters and data volumes, refer to the Technical data of WinAC RTX (Page 255), order no. 6ES7671-0RC07-0YA0. These are available at SIMATIC Customer Support.

7.4.2 Use standard FBs and UDTs

Introduction

The blocks and UDTs of the STEP 7 standard library must be used for open communication over Industrial Ethernet.

Procedure

1. Load the standard FBs and UDTs from the STEP 7 standard library to use Open User Communication via Industrial Ethernet.
2. Integrate the standard FBs and UDTs from the STEP 7 standard library in your user program.

   The communication partner needs an appropriate user program in order to be able to communicate with WinAC RTX.
3. Configure the standard FBs with correct parameters.

Additional information

The standard FBs and UDTs are made available in the STEP 7 standard library.

The individual blocks are described in the STEP 7 help.

Wizard for preparation of the connection data (Open Communication Wizard)

The parameters must be specified in a data block for each connection during an open communication. The Open Communication Wizard (OC Wizard) can be used for the configuration of the blocks for open communication. You can easily define an overview of the connection parameters with the help of the OC Wizard.

You will need SIMATIC STEP 7 V5.4 + SP1 or higher in order to use the OC Wizard.

Additional information

Additional information regarding the use of OC Wizard or a current download version and the manual can be found in the Internet (http://support.automation.siemens.com/WW/view/en/25209116).

7.4.3 Communications connection

How can you use open communication?

To allow data to be exchanged with other communication peers, STEP 7 provides FBs and UDTs under "Communication Blocks" in the "Standard Library":

Establishing of a communication connection

- Use with TCP and ISO-on-TCP

  Both communication peers call FB 65 "TCON" to establish the connection. In the parameterization you define which communication peer is the activate and which one is the passive communication end point. To determine the number of possible connections, refer to your PC station's technical specifications.

  The PC station automatically monitors and maintains the active connection.

  If the connection is interrupted, for example due to an open circuit or by the remote communication peer, the active peer tries to reestablish the connection. You do not have to call FB 65 "TCON" again.

  An active connection is terminated by calling the FB 66 "TDISCON" or when the PC station is in STOP mode. To reestablish the connection you have to call FB 65 "TCON" again.
- Use with UDP

  Both communication peers call FB 65 "TCON" to set up their local communication access point. This establishes a connection between the user program and operating system's communication layer. No connection is established to the remote peer.

  The local access point is used to send and receive UDP telegrams.

Terminating a communication connection

- Use with TCP and ISO-on-TCP

  FB 66 "TDISCON" terminates a connection between the PC station and communication peer.
- Use with UDP

  FB 66 "TDISCON" disconnects the local communication access point, i.e. the connection between user program and communication layer of the operating system is interrupted.

Options for terminating the connection

The following events are available for terminating communication connections:

- You program the discontinuation of the connection with the FB 66 "TDISCON".
- The PC station changes from RUN to STOP.
- At Power Off/Power On

Reference

For further information about the blocks described above, refer to the *STEP 7* online help.

7.4.4 Peculiarities with WinAC RTX

Introduction

To assign parameters for the communication connections in TCP, create a DB containing the data structure from UDT 65 "TCON_PAR". This data structure contains the parameters necessary for configuring the connection.

UDT 65

UDT 65 has the name TCON_PAR and is used by FB 65.

For establishing a connection it is important to know to which IF slot your device is assigned. For the values for local_device_id for WinAC RTX depend on the interface slot of the network adapter.

The values of local_device_id for the individual slots are:

- IF1: B#16#01
- IF2: B#16#06
- IF3: B#16#0B
- IF4: B#16#0F

Transmittable data length

The following table describes which data lengths can be transmitted by which protocol variants.

Protocol variant	Data length, max.
TCP according to RFC 793 (connection type B#16#11)	65 534 byte
ISO-on-TCP according to RFC 1006 (connection type B#16#12)	65 534 byte
UDP according to RFC 768 (connection type B#16#13)	1 472 byte

7.5 Web server

7.5.1 Properties of the web server

Activating the Web server

The Web server is activated with its factory settings. It is deactivated in HW Config with its basic configuration. Before using the Web server for the first time you have to activate it in HW Config, as described in Settings in HW Config, "Web" tab (Page 155)

Using the Web server

The Web server allows you to monitor the WinAC RTX via the Internet or via your company's Intranet. This allows evaluation and diagnostics to be carried out remotely.

Messages and status information are visualized on HTML pages.

Web browser

You need a Web browser to access the HTML pages of WinAC RTX.

The following Web browsers are suitable for communication with WinAC RTX:

- Internet Explorer (version 6.0 and higher)
- Mozilla Firefox (V1.5 and higher)
- Opera (version 9.0 and higher)
- Netscape Navigator (version 8.1 and higher)

Reading information via the Web server

The web server can be used to read the following information from WinAC RTX:

- Start page with general information
  - Module name
  - Module type
  - State
  - Mode selector switch setting
  - Order number of the software package
  - Firmware release version
  - Plant identifier
  - Mode
- Content of the diagnostics buffer

- Variable table
  - You can monitor up to 50 variable tables with a of maximum 200 variables. Select the variable tables on the relevant Web site, as described in section Variable tables (Page 181)
- Variable status
  - You can monitor up to 50 variables after specifying their address.
- Module state
  - The status of a station is indicated using symbols and comments.
- Messages (message state ALARM_S, ALARM_SQ, ALARM_D, ALARM_DQ) without option of acknowledgement
- Information about Industrial Ethernet (submodule of WinAC RTX)
  - Ethernet MAC address
  - IP address
  - IP subnet address
  - Default router
  - Auto negotiation mode ON/OFF
  - Number of packets sent/received
  - Number of faulty packets sent/received
  - Transmission mode (10 Mbps or 100 Mbps full duplex)
  - Link status
- Topology of the PROFINET nodes
  - The configured PROFINET nodes of a station are displayed.

Note

Incorrect display

Delete all cookies and temporary Internet files from your PC / programming device if incorrect data is output while you are working with the Web server.

Web Access on WinAC RTX via Web browser

Proceed as follows to access the Web server:

1. Connect your Web browser capable device, e.g. PG/PC or HMI device, via the submodule-type configured PROFINET interface of the WinAC RTX.
2. Open the Web browser (for example, Internet Explorer).

Enter into the Web browser's "address" field the IP address of the submodule-type configured PROFINET interface of the WinAC RTX in the form <http://a.b.c.d/> for example, http://192.168.0.1/
The start page of the PC station appears. From the start page, you can navigate to additional information.

Web access to the PC station via PDA

You can also access the web server using a PDA. You can select a compact view for this. The procedure is as follows:

1. Connect the PDA to the PC station via the PROFINET interface.
2. Open the Web browser (for example, Internet Explorer).

   Enter the IP address of the PC station in the "Address" field of the Web browser in the format <http://a.b.c.d/basic> e.g. http://192.168.0.1/basic

The start page of the PC station opens. From the start page, you can navigate to additional information.

Note

HMI devices operating with Windows CE operating system V 5.x or earlier process PC station information in a browser specially developed for Windows CE. The information appears in a simplified format in this browser. The figures in this manual show the more detailed format.

Security

The Web server by itself does not provide any security functions for access protection and user management. Protect your web-compliant PC stations against unauthorized access by means of a firewall.

7.5.2 Settings in HW Config, "Web" tab

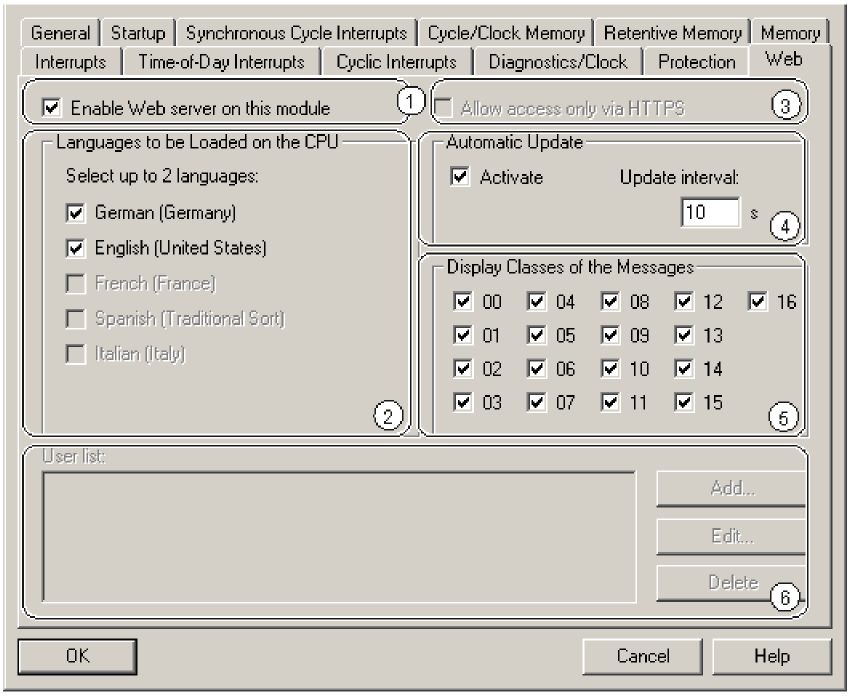
Requirements

- You have opened the WinAC RTX dialog "Properties" in the HW Config.
- PG/PC is connected via the WinAC RTX submodule configured PROFINET interface.

Basic procedure

Proceed as follows in order to use the full functionality of the Web server:

1. Select the "Web" tab.
2. Check the "Activate web server on this module" check box.
3. Select up to two languages to be used to display plaintext messages.
4. Activate the "Activate" option button in the "Automatic update" field.
5. Activate the display classes of the messages.
6. Save and compile the project.
7. Download the configuration to WinAC RTX.



☑ Enable Web server on this module

□ Allow access only via HTTPS

Languages to be Loaded on the CPU

Select up to 2 languages:

☑ German (Germany)

☑ English (United States)

□ French (France)

□ Spanish (Traditional Sort)

□ Italian (Italy)

Automatic Update

☑ Activate

Update interval: 10 s

Display Classes of the Messages

☑ 00 ☑ 04 ☑ 08 ☑ 12 ☑ 16

☑ 01 ☑ 05 ☑ 09 ☑ 13

☑ 02 ☑ 06 ☑ 10 ☑ 14

☑ 03 ☑ 07 ☑ 11 ☑ 15

User list:

Preparation

Proceed in the following manner to edit the properties of WinAC RTX.

1. Double-click on the element "WinAC RTX."

   The "Properties" dialog box is opened.
2. Select the "Web" tab.

You can now enter your personal settings.

① Activate the Web server

The web server is deactivated in the basic configuration in HW Config. Activate the Web server in HW Config.

1. Check the "Activate web server on this module" option button.

② Set the language for Web

Select up to two languages for the Web from the languages installed for display devices.

1. Select up to two languages for the Web.

Note

If you activate the Web server and do not select a language, messages and diagnostic information will be displayed in hexadecimal code.

③ Access via HTTPS

The encryption of the communication between the browser and the Web server via https is not supported by WinAC RTX.

④ Activate automatic update

To enable automatic updates, proceed as follows:

1. Activate the "Activate" option button in the "Automatic update" field.
2. Enter the update interval.

Note

Update interval

The activation interval set in HW Config is the shortest update time.

There may be a significant delay in the refreshing of the Web pages during high CPU loads if WinAC RTX is overloaded during operation, for example, if there is a high number of PROFINET interrupts or if the volume of communication jobs is extensive.

⑤ Message display classes

All message display classes are activated in the basic configuration in HW Config. The messages for the selected display classes are displayed later on the "Messages" web page. Messages for display classes that are not selected are shown as hexadecimal code and not as plain text.

Note

Reducing memory requirements of the Web SDBs

You can reduce memory requirements of the Web SDBs by selecting only the messages to be filled in the Web SDB.

There are two alternatives for configuring the message classes:

- For "Report system error" in HW Config under **Options > Report system error**.
- For block-specific messages in STEP 7

For information about configuring message texts and classes, refer to the online help of STEP 7.

⑥ User list

The assignment of authorizations for specific users is not supported by WinAC RTX.

7.5.3 Language settings

Introduction

The Web server provide information in the following languages:

- German (Germany)
- English (United States)
- French (France)

What you need to display texts in different languages

Language settings to be made in STEP 7 in order to ensure proper output of data in the selected language:

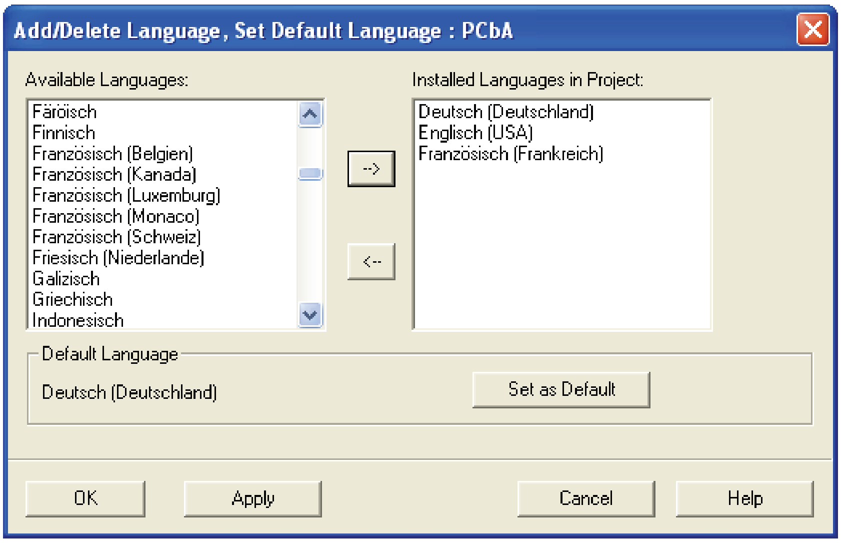
- Set the language for display devices in SIMATIC Manager
- Select the Web language in the WinAC RTX properties dialog.

Setting the language for display devices in SIMATIC Manager

Proceed as follows to select the languages for the display devices in the SIMATIC Manager:

1. Open the SIMATIC Manager.
2. In the "Options" menu, select the "Language for Display Devices" command.

   The language dialog is opened.
3. Select the desired language settings.



Add/Delete Language, Set Default Language : PCbA

Available Languages:

Färöisch
Finnisch
Französisch (Belgien)
Französisch (Kanada)
Französisch (Luxemburg)
Französisch (Monaco)
Französisch (Schweiz)
Friesisch (Niederlande)
Galizisch
Griechisch
Indonesisch

Installed Languages in Project:

Deutsch (Deutschland)
Englisch (USA)
Französisch (Frankreich)

Default Language

Deutsch (Deutschland)

Setting the language for Web

Select up to two languages for the Web from the languages installed for display devices. Proceed as described in the "Settings in HW Config, "Web" tab (Page 155)" chapter.

7.5.4 Updating and saving information

The Web server provides you with several functions which can be controlled via symbols during ongoing operation of WinAC RTX.

Refresh status of the print screen content

Automatic refresh is deactivated in the basic configuration in HW Config.
This means that the screen of the Web server outputs static information.

Refresh the Web pages manually using the <F5> function key or the following icon:

On

Refresh status of the printouts

The created printouts always indicate current information about WinAC RTX. The printed information may therefore be more up to date than the screen contents.

Print Web pages using the following icon:

🖨

Filter settings have no effect on the print-out, The print-out always shows the entire contents of the message buffer.

Deactivating automatic refresh for individual Web pages

To deactivate automatic refresh for a Web page for a short time, select the following icon:

Off

Enable automatic refresh again using the <F5> function key or the following icon:

On

Note

Update time if CPU is heavily loaded

There may be a significant delay in the refreshing of the Web pages during high CPU loads if WinAC RTX is overloaded during operation, for example, if there is a high number of PROFINET interrupts or if the volume of communication jobs is extensive.

Saving messages and entries of the diagnostics buffer

Messages and diagnostics buffer entries can be saved to a csv file. The displayed 250 entries of the diagnostics buffer are saved. Use the following symbol to save the data.

CSV

A dialog box opens in which you can enter the file name and target directory.

To prevent incorrect display of the data in Excel, do not open the csv file with double-click. Import the file in Excel by selecting the "Data" and "Import external data" menu command.

Select the "Separated" file type and "Unicode UTF-8" as file source. Select the comma separator and the text recognition character " .

7.5.5 Web pages

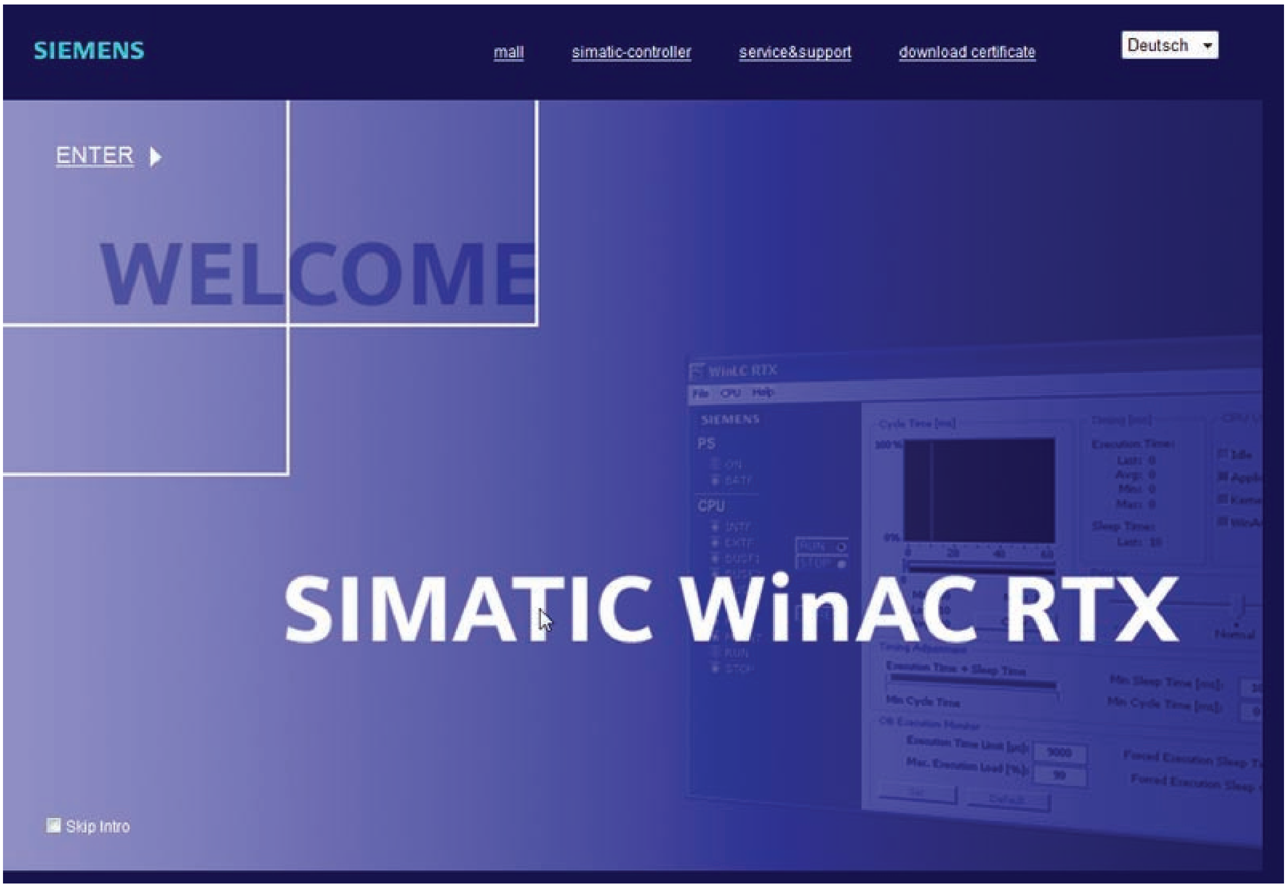
7.5.5.1 Start page with general information

Going online to the Web server

A connection with the Web server is established by entering the IP address of the PROFINET configured submodule interface of the configured WinAC RTX into the address list of the Web browser (for example, http: //192.168.1.158). The connection opens with the "Intro" page.

Intro

The screenshot below shows the first page (Intro) called by the Web server.



Click the ENTER link to go to the Web server pages.

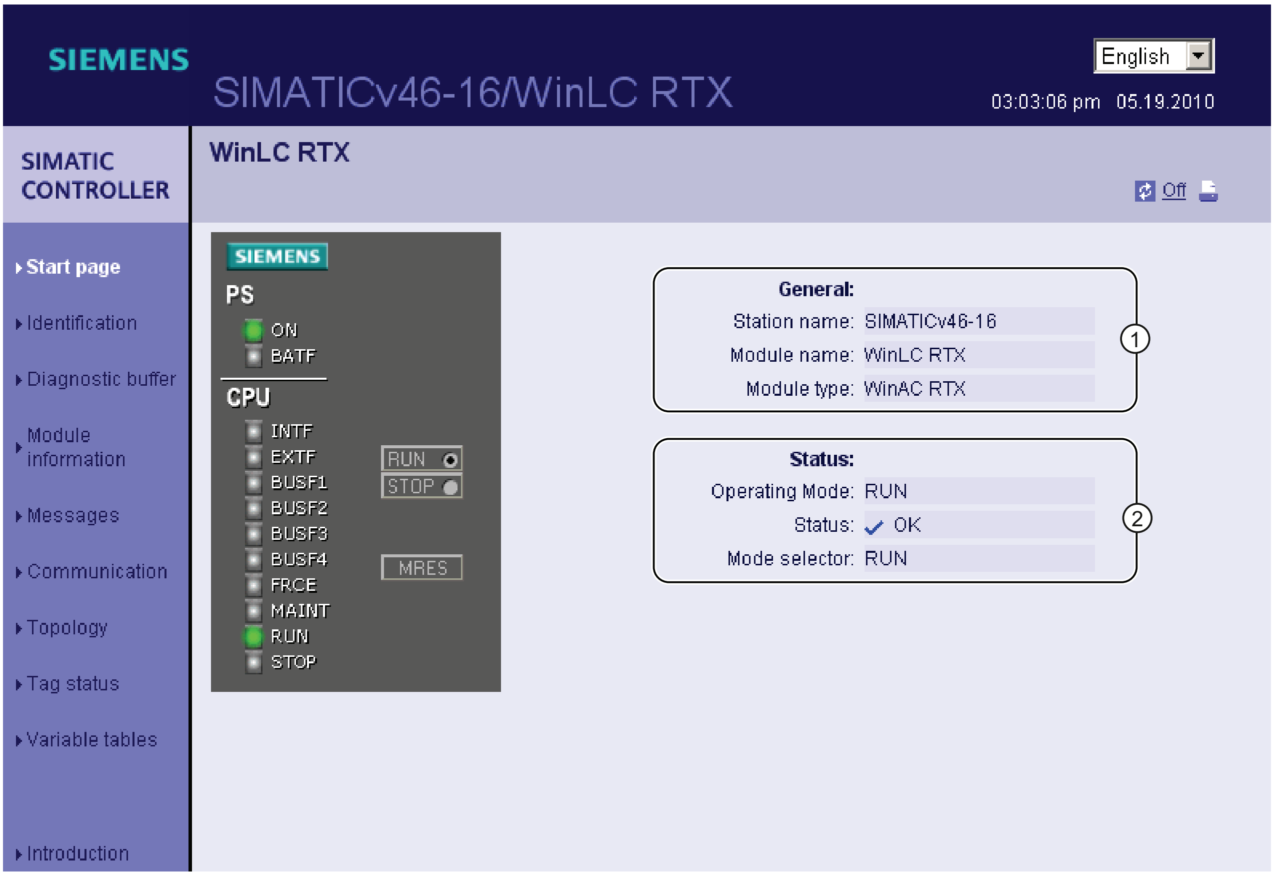
Note

Skipping the Intro Web page

Set the "Skip Intro" check box in order to skip the Intro. The Web server will now directly open its start page. You can undo the "Skip intro" setting by clicking the "Intro" link on the start page.

Start page

The start page displays information as shown in the picture below.



General:

Station name: SIMATICv46-16
Module name: WinLC RTX
Module type: WinAC RTX

Status:

Operating Mode: RUN
Status: ✓ OK
Mode selector: RUN

The image of WinAC RTX with LEDs reflects the current status at the time of the data query.

① "General"

Information about WinAC RTX, and its Web server that is your current connection, is summarized in this group.

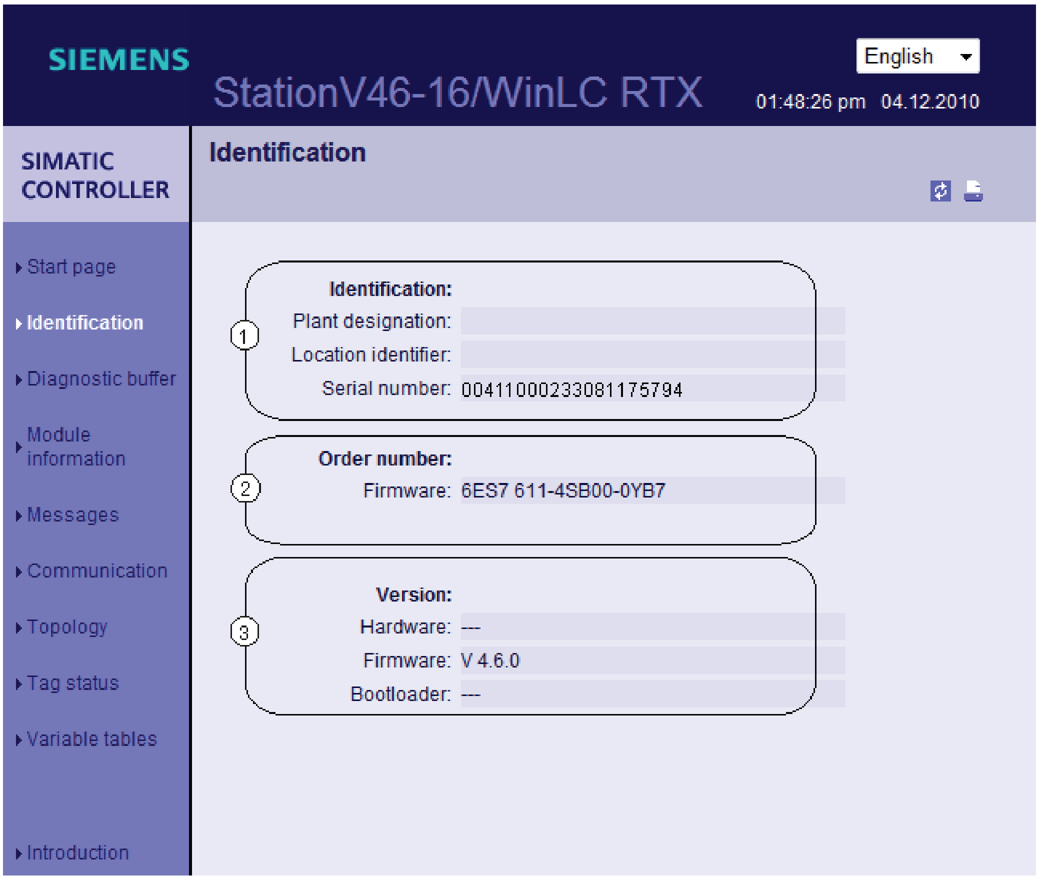
② "Status"

WinAC RTX status information at the time of the query are summarized in the info box "Status."

7.5.5.2 Identification

Characteristics

Characteristics of WinAC RTX can be found on the Web page "Identification."



Identification:
Plant designation:
Location identifier:
Serial number: 00411000233081175794

Order number:
Firmware: 6ES7 611-4SB00-0YB7

Version:
Hardware: ---
Firmware: V 4.6.0
Bootloader: ---

① Identification

The "Identification" field contains the plant and location ID, and the serial number. After you have licensed WinAC RTX (Page 32), the license number is displayed as the serial number. A placeholder with a sequence of zeros is displayed as the serial number instead of the license number if no license key is imported.

Installations and local IDs can be configured in the WinAC RTX properties dialog, tab "General" in HW Config.

② Order number

You can find the order numbers of the software package in the "Order number" field.

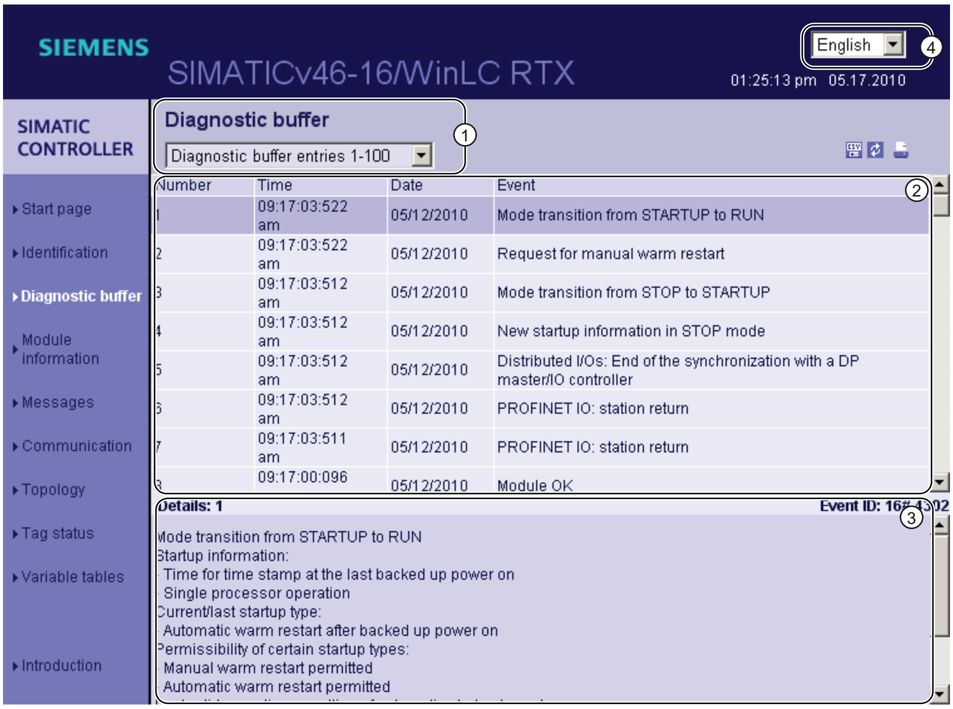
③ Version

You can find the software package and firmware versions in the "Version" field.

7.5.5.3 Diagnostics buffer

Diagnostics buffer

The browser displays the content of the diagnostics buffer on the "Diagnostics buffer" web page.



Number	Time	Date	Event
1	09:17:03:522 am	05/12/2010	Mode transition from STARTUP to RUN
2	09:17:03:522 am	05/12/2010	Request for manual warm restart
3	09:17:03:512 am	05/12/2010	Mode transition from STOP to STARTUP
4	09:17:03:512 am	05/12/2010	New startup information in STOP mode
5	09:17:03:512 am	05/12/2010	Distributed I/Os: End of the synchronization with a DP master/IO controller
6	09:17:03:512 am	05/12/2010	PROFINET IO: station return
7	09:17:03:511 am	05/12/2010	PROFINET IO: station return
8	09:17:00:096	05/12/2010	Module OK

Details: 1 **Event ID: 16#4302**

Mode transition from STARTUP to RUN
Startup information:
- Time for time stamp at the last backed up power on
- Single processor operation
Current/last startup type:
- Automatic warm restart after backed up power on
Permissibility of certain startup types:
- Manual warm restart permitted
- Automatic warm restart permitted

Requirements

The Web server is activated, languages are set, and the project is compiled and downloaded in STEP 7.

① Diagnostics buffer entries 1-250

The diagnostics buffer can save up to 3200 messages. Select an buffer input interval from the list. Each interval comprises 250 entries. Note that the program does not displays all the buffer entries in RUN for reasons that include performance.

② Events

The "Events" fields displays the diagnostics event and the corresponding date and time stamp.

③ Details

This field outputs detailed information about a selected event.

Select the corresponding event from the ② "Events" field.

④ Special features when changing languages

You can change the language, for example, from German to English, by clicking in this field. If you select a language you have not configured the program shows a hexadecimal code instead of plain text information.

7.5.5.4 Module state

Requirements

- You made the following settings in HW Config:
  - Web server activated
  - Language settings
  - "Report system error" generated and activated.
- You compiled the project using STEP 7 HW Config, loaded the SDB container and the user program (particularly the user program blocks generated by "Report system error")
- WinAC RTX is in RUN mode.

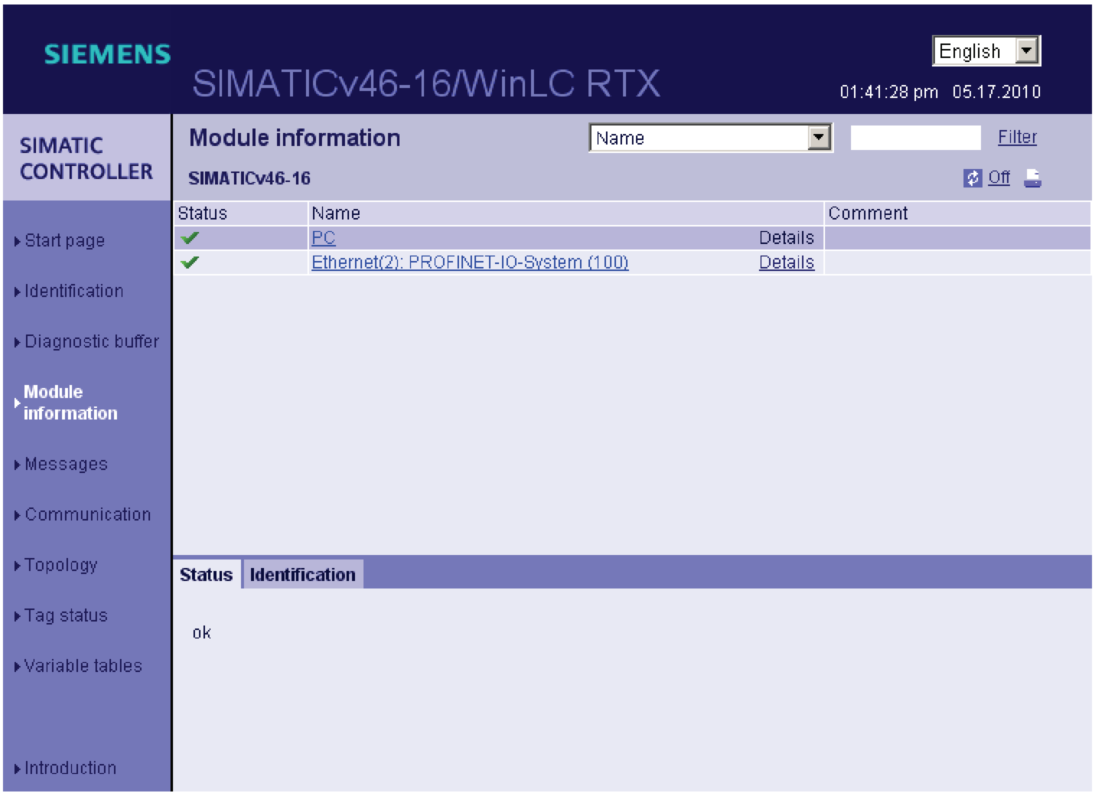
Note

"Report system error"

- **Duration of display:** Depending on the plant configuration, the display "Report system error" may take some time to create the startup evaluation of the state of all the configured I/O modules and I/O systems. There is no concrete display of the status on the "Module state" page during this time. A "?" is displayed in the "Error" column.
- **Dynamic response:** "Report system error" must be called cyclically at least every 100 ms.
  The call may either take place in OB 1, or if the cycle time is more than 100 ms in the cyclic interrupt OB 3x (≤ 100 ms) and in the restart OB 100.
- Diagnostics support: In the "Report system error" dialog box, the "Diagnostics status DB" check box must be selected in the "Diagnostics support" tab and a DB number entered. This check box is normally selected by default for configured Web servers. However, during migration of old projects it may be necessary to select this check box manually.
- **Restart:** Depending on the station setup the module state is indicated after a delay of a few seconds when WinAC RTX is restarted.

Module state

The status of a station is displayed on the "Module state" page using symbols and comments.



Module information

Name

SIMATICv46-16

Status	Name		Comment
✓	PC	Details	
✓	Ethernet(2): PROFINET-IO-System (100)	Details	

ok

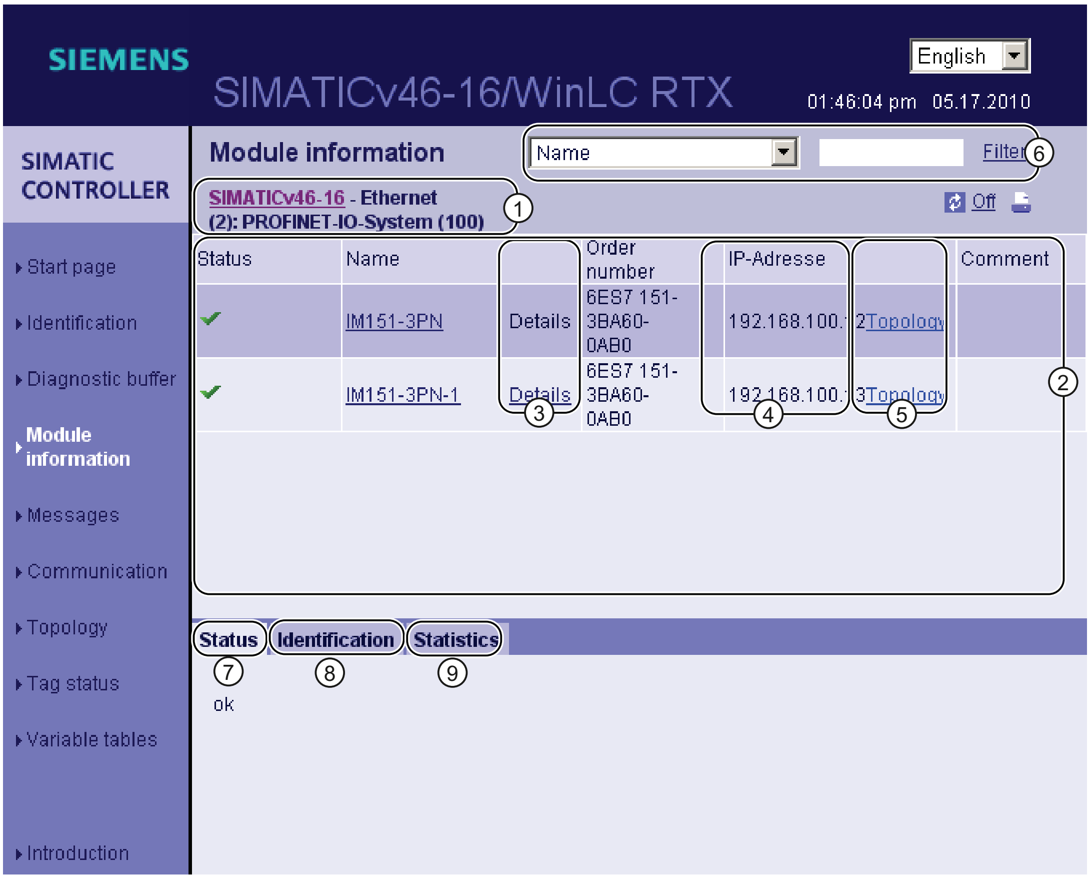
Meaning of symbols

Symbol	Color	Meaning
✓	green	Component OK
✓	gray	Disabled PROFIBUS slaves or PROFINET devices
?	black	Component not available / unable to determine state "Unable to determine state" is, for example, always displayed in the WinAC RTX STOP mode or during the startup evaluation of "Report system error" for all configured I/O modules and I/O systems after a WinAC RTX restart. However, this state can also be displayed temporarily at runtime for all modules in the event of a surge of diagnostics interrupts.
🔧	green	Maintenance required
🔧	yellow	Maintenance demanded
🔧	red	Fault - component failed or faulty
!	-	Fault at lower module level

Navigation to other module levels

The state of individual modules is displayed when you navigate to further module levels:

- Go to the next higher module level using the link in the title line
- Go to the next lower module level using the link in the name



Module information

SIMATICv46-16 - Ethernet (2): PROFINET-IO-System (100)

Status	Name		Order number	IP-Adresse		Comment
✓	IM151-3PN	Details	6ES7 151-3BA60-0AB0	192.168.100.1	2Topology	
✓	IM151-3PN-1	Details	6ES7 151-3BA60-0AB0	192.168.100.1	3Topology	

ok

Requirements

The Web server is activated, languages are set, and the project is compiled and downloaded in STEP 7.

Note

State of the AS-i slaves

The state of AS-i slaves is not displayed on the "Module state" page. Only the state of the AS-i link is displayed.

---

① "Module state"

The table contains information about the rack, the nodes, the DP master system and the individual modules of the station relating to the selected level.

This setup requires prior configuration of the function "Report system error" and the download of the created blocks into WinAC RTX for WinAC RTX and the station.

② "Display of module levels"

The link in the title line takes you to the "module state" of the next higher module level.

③ "Details"

The "Details" link provides you with information on the selected module in the "Status" and "Identification" tabs.

④ "IP address"

You can use the link to access the Web server of the selected, configured devices.

⑤ Topology

The "Topology" and "Module status" Web pages are linked. Click "Topology" of the selected module to automatically jump to this module in the graphic view of the "Topology" web page. The module appears in the visible area of the "Topology" web page and the device head of the selected module flashes for a few seconds.

⑥ "Filter"

You can sort the table by certain criteria.

Use the dropdown list to view only the entries of the selected parameter. Enter the value of the selected parameter in the input box and then click "Filter".

1. Select, for example, the "Name" parameter from the drop-down list box.
2. Click "Filter".

The filter criteria are also retained when you update a page.

⑦ "Status" tab

The tab contains information on the status of the selected module:

⑧ "Identification" tab

The tab contains data on the identification of the selected module.

Note

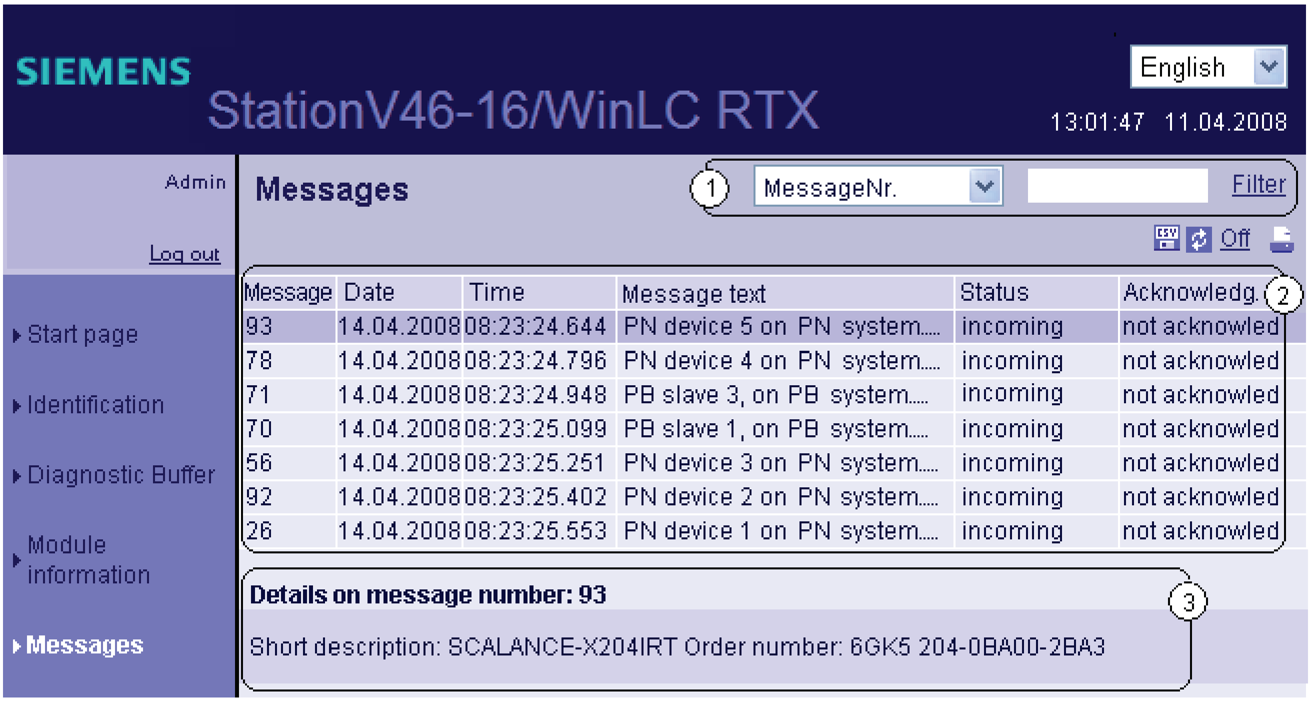
"Identification" tab

This tab displays only the data configured offline, not the online module data.

7.5.5.5 Alarms

Messages

The browser displays the content of the message buffer on the "Messages" web page. The messages cannot be acknowledged on the web server.



StationV46-16/WinLC RTX

Messages

Message	Date	Time	Message text	Status	Acknowledg.
93	14.04.2008	08:23:24.644	PN device 5 on PN system....	incoming	not acknowled
78	14.04.2008	08:23:24.796	PN device 4 on PN system....	incoming	not acknowled
71	14.04.2008	08:23:24.948	PB slave 3, on PB system....	incoming	not acknowled
70	14.04.2008	08:23:25.099	PB slave 1, on PB system....	incoming	not acknowled
56	14.04.2008	08:23:25.251	PN device 3 on PN system....	incoming	not acknowled
92	14.04.2008	08:23:25.402	PN device 2 on PN system....	incoming	not acknowled
26	14.04.2008	08:23:25.553	PN device 1 on PN system....	incoming	not acknowled

Details on message number: 93

Short description: SCALANCE-X204IRT Order number: 6GK5 204-0BA00-2BA3

Requirements

The message texts were configured in the user-specific languages. You can find information on configuring message texts in STEP 7 and on the Internet (http://support.automation.siemens.com/WW/view/en/23872245):

① Filter

This functionality allows you to select specific information from this page.

Use the corresponding list to view only the entries of the selected parameter.
Enter the value of the selected parameter in the input box and then click "Filter".

To view all alarms with "incoming" status, for example:

1. Select the "Status" parameter from the list.
2. Enter the "incoming" text in the input box.
3. Click "Filter".

The filter criteria are also retained when you update a page. Filter settings have no effect on the print-out. A printout always contains the entire content of the message buffer.

② Messages

Messages from WinAC RTX are displayed in chronological order with **date** and **time**.

The **message text** parameter is an entry which contains the message texts configured for the corresponding fault definitions.

Sorting

You can also view the parameters in ascending or descending order. Click in the column header of one of the parameters.

- Message number
- Date
- Time
- Message Text
- Status
- Acknowledgment

The messages are returned in chronological order when you click the "Date" entry. Incoming and outgoing events are output at the **Status** parameter.

③ Message number details

You can view detailed message information in this info field. To do this, select a message the details of which you are interested.

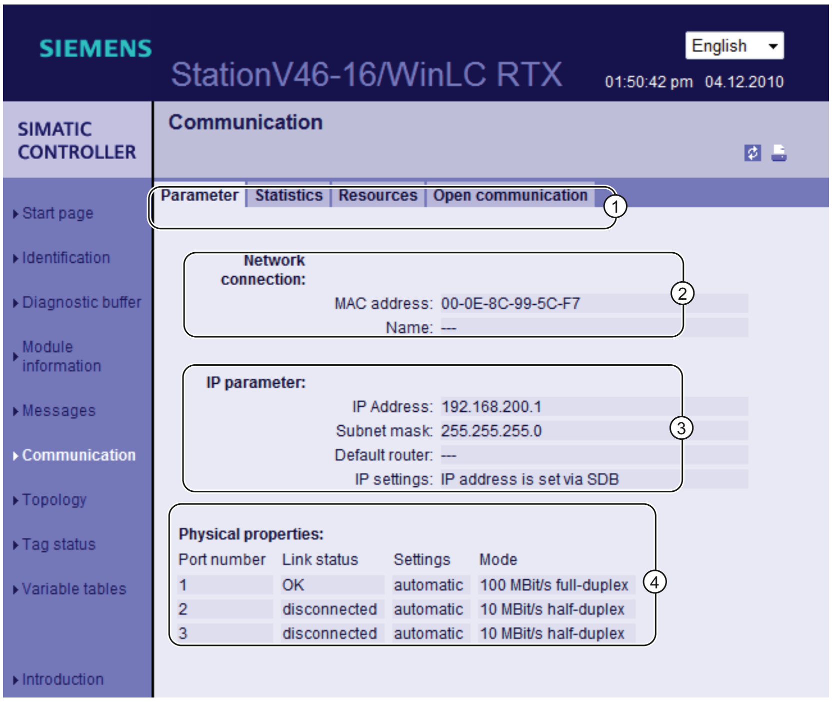
Special features when changing languages

You can change the language, for example, from German to English, by clicking the object in the upper right corner. If you select a language or corresponding message texts you have not configured the program shows a hexadecimal code instead of plain text information.

7.5.5.6 Communication

"Parameters" tab

In the tab ① "Parameter" of this Web page information is summarized about the WinAC RTX integrated PROFINET interface.



Network connection:

MAC address: 00-0E-8C-99-5C-F7
Name: ---

IP parameter:

IP Address: 192.168.200.1
Subnet mask: 255.255.255.0
Default router: ---
IP settings: IP address is set via SDB

Physical properties:

Port number	Link status	Settings	Mode
1	OK	automatic	100 MBit/s full-duplex
2	disconnected	automatic	10 MBit/s half-duplex
3	disconnected	automatic	10 MBit/s half-duplex

② Network connection

This section contains information for identifying the integrated PROFINET interfaces of the corresponding WinAC RTX.

③ IP parameters

Information about the configured IP address and the number of the subnetwork in which the corresponding WinAC RTX is located.

④ Physical properties

You can find the following information in the "Physical properties" field:

- Port number
- Link status
- Settings
- Mode

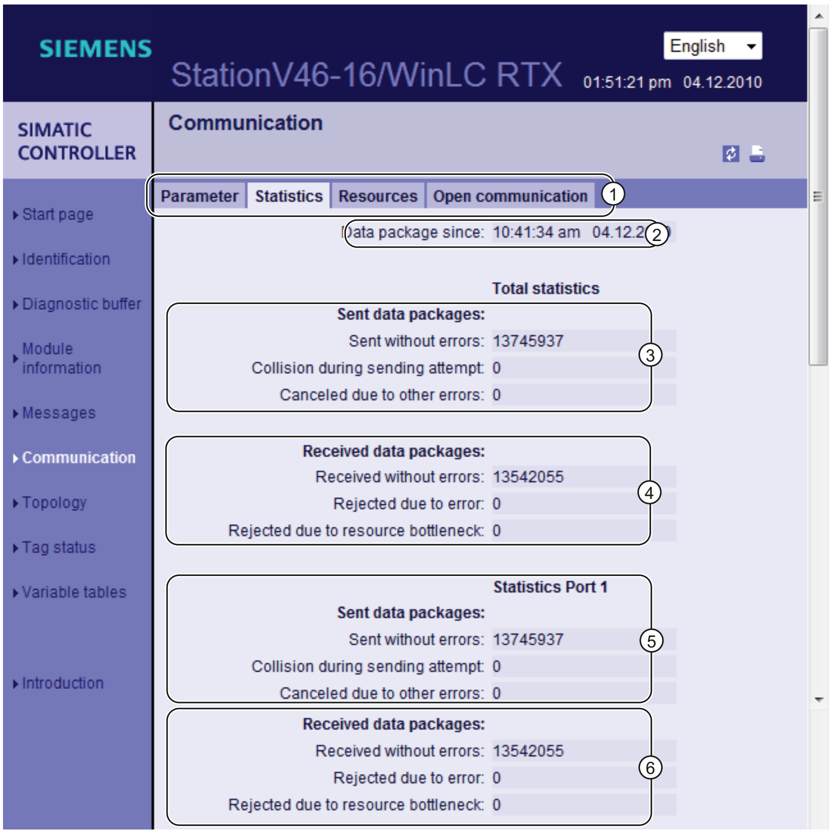
Note

Updating data

The data you see in the HTML browser are only automatically updated if you activated automatic update in HW Config. Otherwise, you can view the current data by updating the view in the HTML browser at regular intervals (Update button).

"Statistics" tab

Information about the quality of the data transmission can be found in the tab ① "Statistics."



Data package since: 10:41:34 am 04.12.2

Total statistics

Sent data packages:
Sent without errors: 13745937
Collision during sending attempt: 0
Canceled due to other errors: 0

Received data packages:
Received without errors: 13542055
Rejected due to error: 0
Rejected due to resource bottleneck: 0

Statistics Port 1

Sent data packages:
Sent without errors: 13745937
Collision during sending attempt: 0
Canceled due to other errors: 0

Received data packages:
Received without errors: 13542055
Rejected due to error: 0
Rejected due to resource bottleneck: 0

② Data packets since

This shows the time at which the first data packet was sent or received after the last POWER ON / memory reset.

③ Overall Statistics - Transmitted Data Packages"

The quality of the data transmission on the transmission line can be determined from the key figures in this info box.

④ Overall Statistics - Received Data Packages"

The quality of the data transmission on the reception line can be determined from the key figures in this info box.

⑤ Statistics Port 1/Port 2 - Transmitted Data Packages"

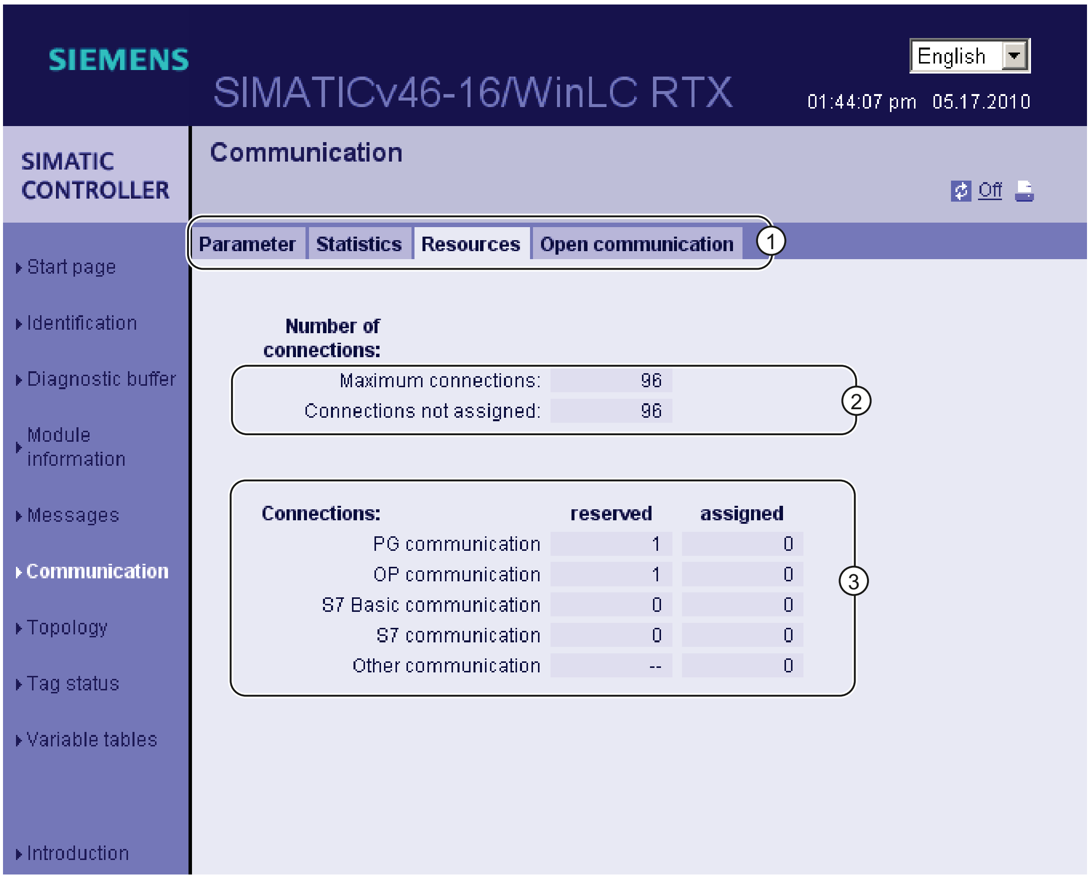
The quality of the data transmission on the transmission line can be determined from the key figures in this info box.

⑥ Statistics Port 1/Port 2 - Received Data Packages"

The quality of the data transmission on the reception line can be determined from the key figures in this info box.

Tab "Resources"

Information about the resources can be found in the tab ① "Resources."



Number of connections:

Maximum connections:	96
Connections not assigned:	96

Connections:	reserved	assigned
PG communication	1	0
OP communication	1	0
S7 Basic communication	0	0
S7 communication	0	0
Other communication	--	0

② Number of connections

This section contains information about the maximum number of connections and the number of connections available.

③ Connections

This section contains detailed information about the individual connections.

7.5.5.7 Topology

Topology of the PROFINET nodes

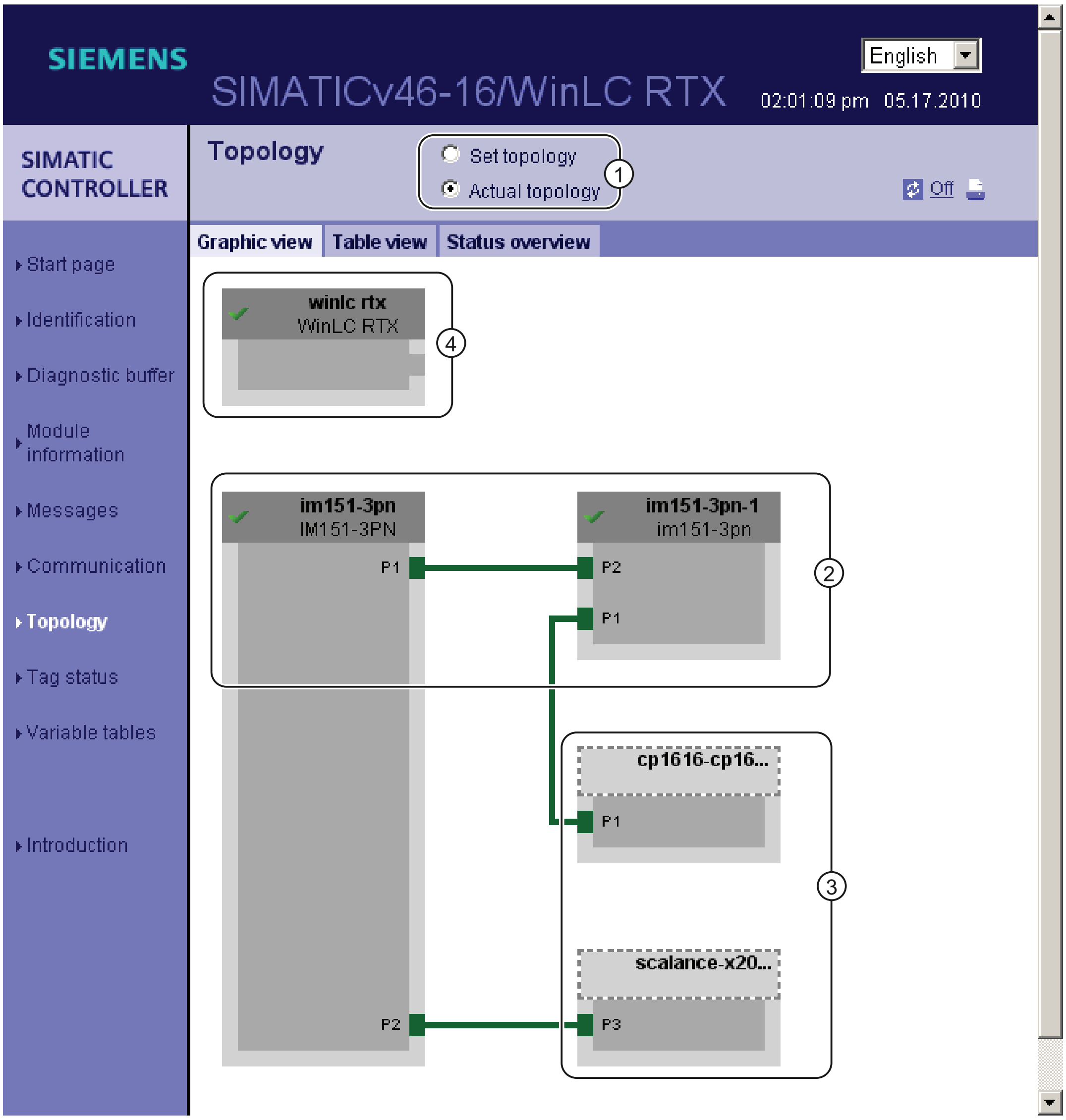
The "Topology" Web page shows the configured and non-configured PROFINET nodes of a station, which are nevertheless accessible through neighbor recognition.

There are three views.

- Graphic view: Two display options are available in this view.
  - Actual topology: Shows the actual topological structure of the configured PROFINET devices and the determined direct neighboring on-configured PROFINET devices (display of the neighbor relationship insofar as this is determinable). New non-configured directly adjacent PROFINET devices are only shown in the actual topology.
  - Saved topology: A saved actual topology can be adopted as setpoint topology, making it easier to diagnose changes in the actual topology. The topological assignment of failed PROFINET devices remains recognizable in this view.
- Tabular view
- Status overview

The views can be printed out. Use your browser's print preview function before printing and correct the format if necessary.

Topology - Graphical view



Requirements

You have activated the web server, set up the language, and compiled the project in HW Config and downloaded it.

Display of the actual topology

The actual topology is always shown in the following scenarios:

- When the "Topology" Web page is called via the navigation bar
- When switching from the "Module status" Web page, overview of PNIO devices, to the "Topology" web page via the "Topology" link

① "Setpoint topology" and "Actual topology"

The "Actual topology" shows the status of the actual device interconnection. You can use the following symbol to save the "Actual topology":

💾

The most recently stored connection status is displayed when switching to the "Setpoint topology" view. If a device should fail in the meantime, the following happens:

- this device remains in the same position in the "Setpoint topology" view.
- In the "Actual topology" view, the device is displayed separately in the bottom area.

In both cases the failed device is indicated by a red bordered device head and red wrench .

🔧

- Non-configured, directly adjacent PROFINET devices that were saved, remain in the same place in the "Saved topology" view when they fail and are displayed with red dashed line device head. Directly adjacent PROFINET devices added are not recognized in the display of the "Saved topology" and are not shown.

Ports which were interchanged are not marked in color.

Note

Retentivity of the "Saved topology" after POWER ON / OFF/ and memory reset

The "Saved topology" is not retentive.

② Configured and accessible PROFINET nodes

Configured and accessible PROFINET nodes are displayed in dark gray. Green connections show through which ports the PROFINET nodes of a station are connected.

② Non-configured and accessible PROFINET devices

PROFINET devices that are not configured but can be accessed directly ("neighboring stations") are displayed in light gray and with a dashed line.

④ Configured nodes without neighbor relationships

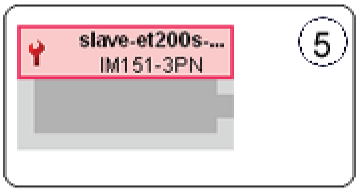
Nodes for which no neighbor relationship can be determined are displayed in dark gray and only with the device number:

- IE/PB links and the PROFIBUS nodes connected to them
- PROFINET devices that do not support LLDP (neighbor recognition)

The PROFINET nodes can be identified by the device numbers in HW Config.

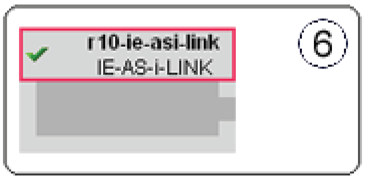
⑤ Configured, but unavailable PROFINET nodes

The configured but inaccessible PROFINET nodes are displayed in the lower section in pink, with a red frame and device number.



⑥ Displaying faulty neighbor relationships

The nodes whose neighbor relationships are incomplete or can only be read out with an error are displayed in light gray with a red frame.



Note

Displaying faulty neighbor relationships

A firmware update of the affected component is required.

Deactivated nodes

Deactivated nodes are shown in light gray.

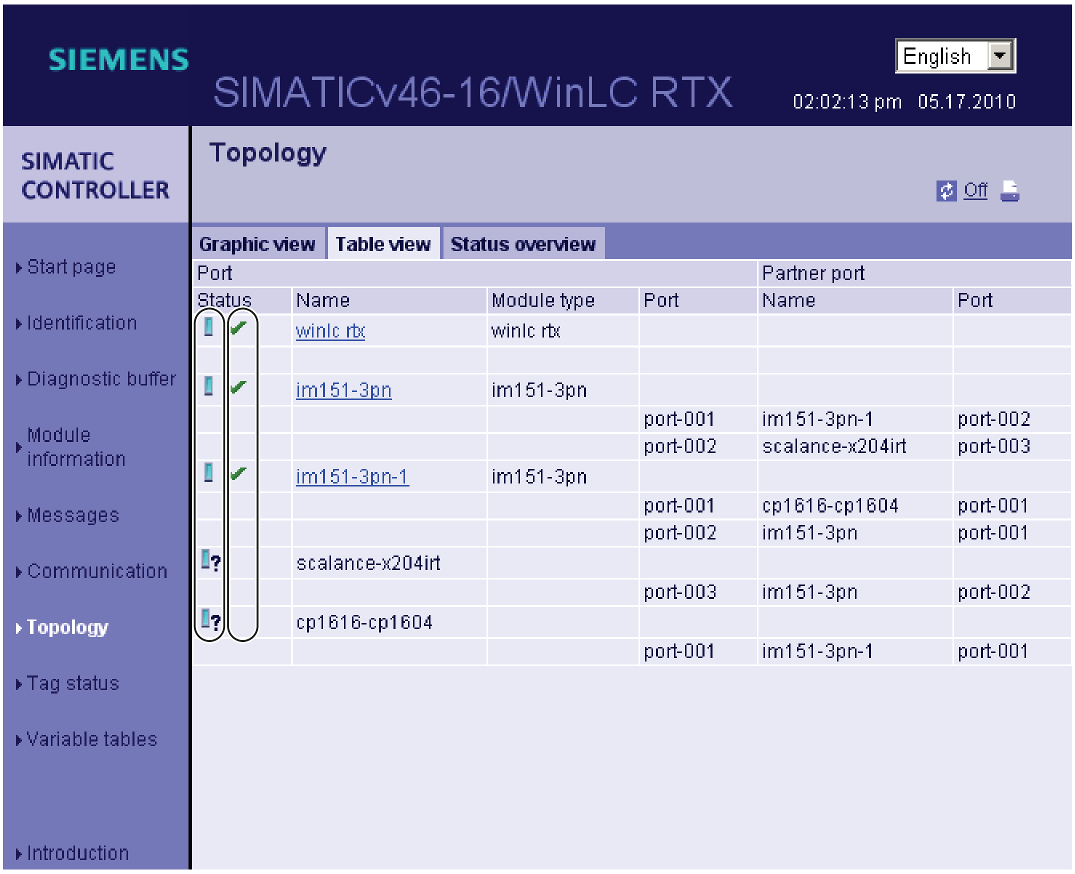
Link between the "Topology" and "Module state" Web pages

The "Topology" and "Module status" Web pages are linked. In the topology view, click the head of the select module to jump automatically to this module in the "Module status" Web page.

See also chapter Module state (Page 164).

Topology - Tabular view

The tabular view always shows the actual topology.



Port					Partner port	
Status		Name	Module type	Port	Name	Port
▮	✓	winlc rtx	winlc rtx			
▮	✓	im151-3pn	im151-3pn			
				port-001	im151-3pn-1	port-002
				port-002	scalance-x204irt	port-003
▮	✓	im151-3pn-1	im151-3pn			
				port-001	cp1616-cp1604	port-001
				port-002	im151-3pn	port-001
▮?		scalance-x204irt				
				port-003	im151-3pn	port-002
▮?		cp1616-cp1604				
				port-001	im151-3pn-1	port-001

Meaning of the symbols relating to the status of the PROFINET nodes

Symbol	Meaning
▮	Configured and available PROFINET nodes
▮?	Non-configured and available PROFINET nodes
▮☒	Configured, but unavailable PROFINET nodes
▮!	Nodes for which neighbor relationships cannot be determined, or for which the neighbor relationship could not be read out completely, or only with errors

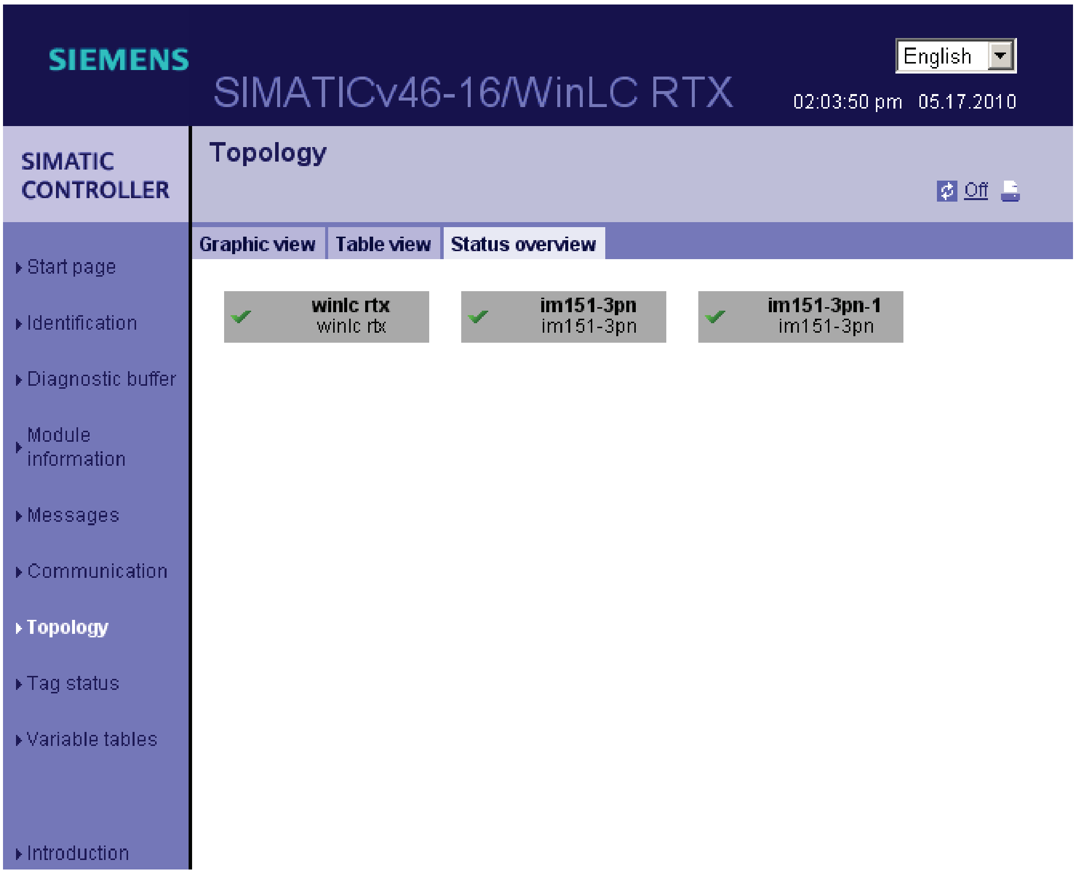
Meaning of the symbols relating to the module state of the PROFINET nodes

Symbol	Color	Meaning
✓	green	Component OK
✓	gray	Disabled PROFIBUS slaves or PROFINET devices Requirements for support: • WinAC RTX 2010 (V4.6) or higher and STEP 7 V5.5 • Activating/deactivating the PROFIBUS slaves and PROFINET IO devices with SFC12 Mode 3/4
?	black	Component not available / unable to determine state "Unable to determine state" is, for example, always displayed in STOP mode of the PC station, or during the startup evaluation of "Report system error" for all configured I/O modules and I/O systems after restart of the PC station. However, this state can also be displayed temporarily at runtime for all modules in the event of a surge of diagnostics interrupts.
🔧	green	Maintenance required
🔧	yellow	Maintenance demanded
🔧	red	Fault - component failed or faulty
❗	-	Fault at lower module level

Topology - Status overview

The "Status overview" shows a clear representation of all PN IO devices / PROFINET devices (without connection relationships) on one page. A quick error diagnostics is possible based on the symbols that show the module states.

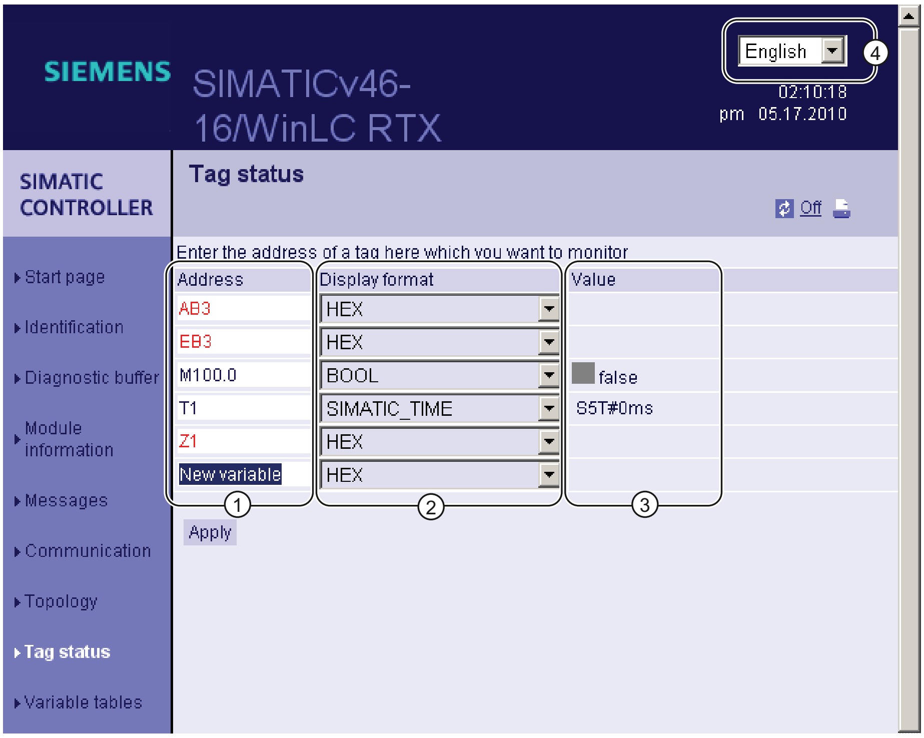
Here, too, there is a linking of the modules to the "Module status" Web page.



7.5.5.8 Variable status

Variable status

The browser outputs the variable status on the Web page of the same name.
You can monitor the status of up to 50 variables.



Enter the address of a tag here which you want to monitor

Address	Display format	Value
AB3	HEX	
EB3	HEX	
M100.0	BOOL	▨ false
T1	SIMATIC_TIME	S5T#0ms
Z1	HEX	
New variable	HEX	

① Address

Enter the address of the operand of which you want to monitor the response in the "Address" text box. Invalid addresses entered are displayed in red font.

To retain these entries, save the variable status Web page in the Favorites list of your browser.

② Display format

Select the display format of a variable using the drop-down list. The program indicates the variable in hex code if it does not support the selected display format.

③ Value

Outputs the value of the corresponding operand in the selected format.

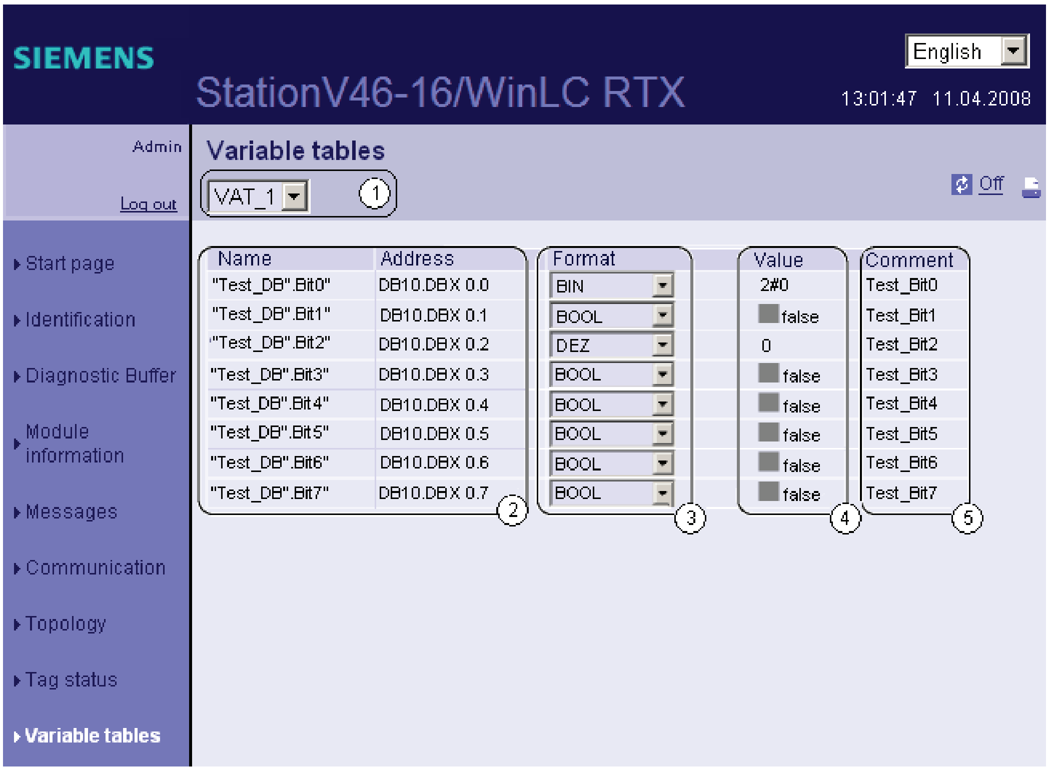
④ Special features when changing languages

You can change the language, for example, from German to English, by clicking the object in the upper right corner. The German mnemonics differ compared to other languages. The syntax of operands you enter may be invalid for this reason when you change languages. For example, ABxy instead of QBxy. The browser outputs a faulty syntax in red font.

7.5.5.9 Variable tables

Variable tables

The browser displays the content of the variable tables on the Web page of the same name. You can monitor up to 50 variable tables with a of maximum 200 variables.



Name	Address	Format	Value	Comment
"Test_DB".Bit0"	DB10.DBX 0.0	BIN	2#0	Test_Bit0
"Test_DB".Bit1"	DB10.DBX 0.1	BOOL	■ false	Test_Bit1
"Test_DB".Bit2"	DB10.DBX 0.2	DEZ	0	Test_Bit2
"Test_DB".Bit3"	DB10.DBX 0.3	BOOL	■ false	Test_Bit3
"Test_DB".Bit4"	DB10.DBX 0.4	BOOL	■ false	Test_Bit4
"Test_DB".Bit5"	DB10.DBX 0.5	BOOL	■ false	Test_Bit5
"Test_DB".Bit6"	DB10.DBX 0.6	BOOL	■ false	Test_Bit6
"Test_DB".Bit7"	DB10.DBX 0.7	BOOL	■ false	Test_Bit7

① Selection

Select one of the configured variable tables from this drop-down list.

② Name and address

This field displays the operand's name and address.

③ Format

Select the display format of the corresponding operand using the drop-down lists. The drop-down list outputs a selection of all valid display formats.

④ Value

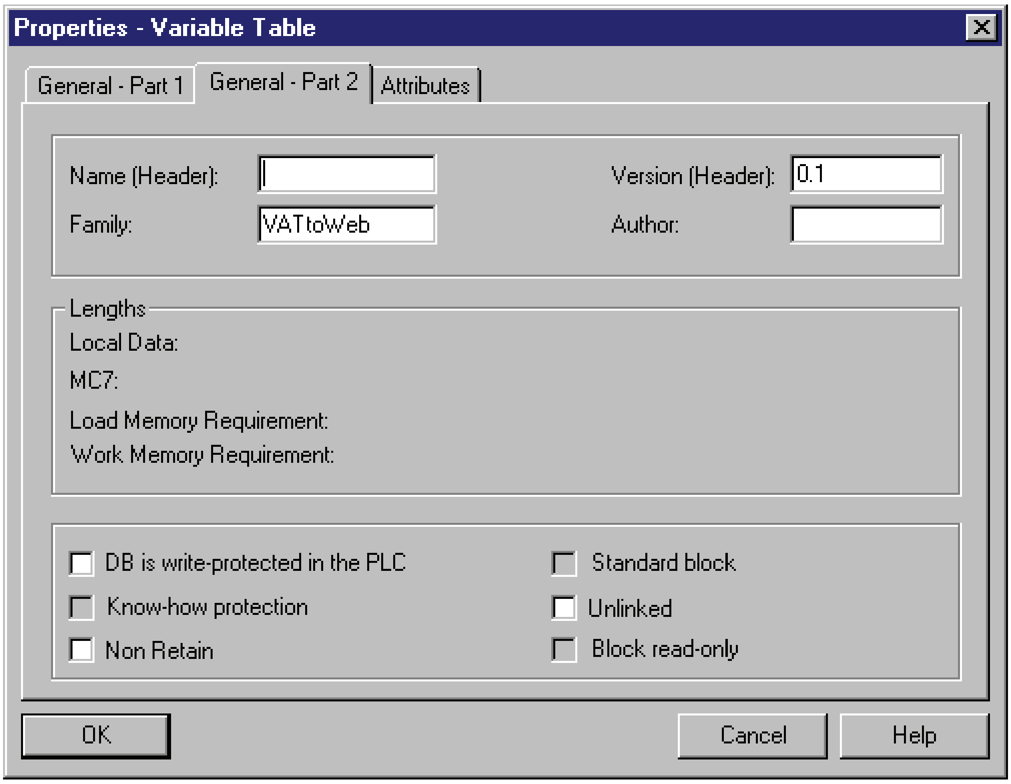
This column shows the values in the corresponding display format.

⑤ Comment

The program outputs the comment you configured in order to highlight the meaning of an operand.

Creating a variable table for the Web server

1. Generate a variable table in STEP 7.
2. Open the properties dialog of the variable table and select the "General - Part 2" tab.
3. Activate the "Web server" check box. As an alternative, you can enter the ID "VATtoWEB" in the "Family" field.



Name (Header):

Version (Header): 0.1

Family: VATtoWeb

Author:

Lengths

Local Data:

MC7:

Load Memory Requirement:

Work Memory Requirement:

□ DB is write-protected in the PLC

□ Know-how protection

□ Non Retain

□ Standard block

□ Unlinked

□ Block read-only

4. Save and compile the project and download the configuration data to the PC station.

Tuning the controller performance 8

8.1 Scan Cycle for a PC-Based Controller

During one scan cycle, the controller updates the outputs, reads the inputs, executes the STEP 7 user program, performs communication tasks and provides time for other applications to run. The following parameters affect the scan cycle:

- The execution time (in milliseconds) is the actual amount of time used by the controller to update the I/O and to execute the STEP 7 user program.
- The cycle time (in milliseconds) is the number of milliseconds from the start of one scan cycle to the start of the next scan cycle. This value must be greater than the execution time of the scan cycle to provide execution time for any application that has a lower priority than WinLC RTX.
- The sleep time (in milliseconds) determines how much time is available during the free cycle (execution cycle for OB 1) to allow higher priority OBs and other applications to use the resources of the computer.

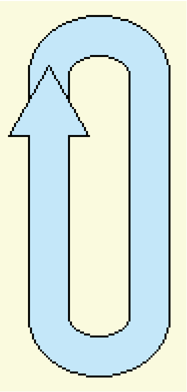
The priority of the controller application also affects the scan cycle by determining when the controller runs or is interrupted by other Windows applications. You must ensure that the sleep time occurs at least every 50 milliseconds in order for other Windows applications, such as moving the mouse, to operate smoothly.

The tuning panel (Page 79) allows you to tune and test the performance of the controller by adjusting the parameters that affect the scan cycle (minimum cycle time, minimum sleep time and priority) without affecting the system configuration in the controller. After testing tuning parameters (Page 200), you use STEP 7 to configure the minimum cycle time for the controller when you create the system (hardware) configuration.

Tasks performed during the scan cycle

After you have used STEP 7 to create and download your control program to the controller, the controller starts executing the control program when you set the controller to RUN mode. Like any other S7 PLC, the controller executes your STEP 7 user program in a continuously repeated scan cycle.

In one scan cycle, the controller performs the following tasks:



1 The controller writes the signal statuses of the process image output table (Q memory area) assigned to OB 1 to the I/O module outputs.

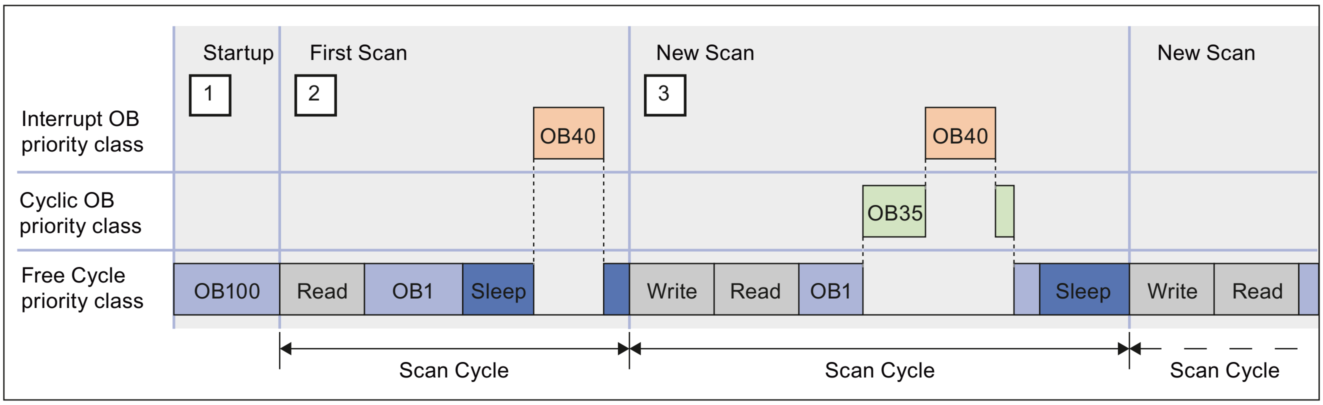
2 The controller reads the signal statuses of the I/O module inputs and stores them in the process-image input table (I memory area) assigned to OB 1.

3 The controller executes the STEP 7 user program in OB 1.

4 OB 1 waits until the minimum sleep time and minimum cycle time requirements are met before starting the next scan cycle. Other OBs can execute at this time.

Because the PC-based controller shares the resources of your computer with other programs (including the operating system), you must ensure that the controller provides sufficient time for other Windows applications to be processed. If the actual execution time of the scan cycle is less than the minimum cycle time that you configured with STEP 7, the controller suspends the free cycle (OB 1) until the minimum cycle time is reached before starting the next scan cycle. This waiting period or sleep time, allows other applications to use the resources of the computer.

The following illustration provides an overview of the tasks that are performed by the controller during various scan cycles.



1 Startup

On a transition from STOP to RUN mode, the controller loads the system configuration, sets the I/O to the default states and executes the startup OB (OB 100 or OB 102).

The startup cycle is not affected by the minimum cycle time and minimum sleep time or watchdog parameters; however, it is affected by the maximum execution time.

2 First scan cycle

An OB with a higher priority class can interrupt the free cycle at any time, even during the sleep time.

In the example above, the controller handles a hardware (I/O) interrupt that occurs during the sleep time by executing OB 40. After OB 40 has finished, the controller waits for the minimum cycle time to expire before starting the next scan cycle.

Note: It is possible for the controller to use all of the sleep time for processing higher-priority OBs. In this case, other Windows applications may not have sufficient time to run. Refer to the techniques for managing sleep time listed below.

3 New scan cycle

In the example above, the controller suspends the execution of OB 1 to execute a cyclic OB (OB 35), which has a higher S7 priority than OB 1. The controller also suspends the execution of OB 35 to handle another I/O interrupt (OB 40).

After OB 40 finishes, the controller resumes the execution of OB 35, and after OB 35 finishes, the controller resumes the execution of OB 1.

The length of the scan cycle is determined by the execution time of all OBs executed during the scan cycle, the minimum cycle time and the minimum sleep time. If the execution time is less than the minimum cycle time that was configured in the system configuration, the controller suspends the free cycle until the minimum sleep time is met. During the sleep time, the computer runs any interrupt OBs and other Windows applications.

⚠ WARNING

Variation in the execution time or response time of the STEP 7 user program could potentially create a situation where the equipment or application being controlled can operate erratically and possibly cause damage to equipment or injury to personnel.

If the controller does not provide sufficient sleep time for other applications to run, the computer can become unresponsive to operator input or the controller and other applications can operate incorrectly. In addition, the execution of the STEP 7 user program can experience non-deterministic behavior (jitter) such that execution times can vary and start events can be delayed.

Always provide an external emergency stop circuit. In addition, always tune the sleep time and manage the performance of the controller so that your STEP 7 user program executes consistently.

Methods for managing the performance of WinLC RTX

While executing the STEP 7 user program, WinLC RTX can experience a variation in the process execution time or response time that causes the cycle times to vary or to exhibit non-deterministic behavior ("jitter (Page 187)"). You can use the following methods to manage the performance of WinLC RTX:

- Adjusting the priority of the controller (Page 193): Affects the execution of WinLC RTX in relation to other RTX processes executing on your computer
- Adjusting the minimum sleep time and minimum cycle time parameters (Page 200): Affects the execution of the free cycle or OB 1 (OB priority class 1)
- Inserting sleep time into the STEP 7 user program (SFC 47 "WAIT") (Page 203): Affects the execution of the priority class of the OB that calls SFC 47 (and any lower priority class)
- Adjusting the sleep-monitoring algorithm of the execution monitor (Page 204): Affects the execution of all OB priority classes (if the other mechanisms do not meet the requirements for sleep time)

WinLC RTX provides a tuning panel (Page 79) for monitoring the performance and for modifying the parameters that affect the scan cycle.

8.2 Causes of Jitter

Because the PC-based controller has to share the computer with running processes, "jitters" may result during the execution of the control program if a process having a higher priority or an active process utilizes the CPU or the system resources of the computer. Jitter is a deviation in the execution time or the response time of the process so that cycle times change and non-deterministic behavior occurs.

Jitter occurs when there is a delay starting or ending an OB. For example, The execution time can vary by a few milliseconds from scan cycle to scan cycle or the start of an interrupt OB can be delayed. In some control applications, these variations cause no problems in the operation of the controller; in an extremely time-critical process, however, even a jitter of 1 ms may be significant.

The following settings for WinLC RTX can result in jitter during the execution of the control program:

- Priority setting for concurrent RTX applications
- Priorities between the WinLC RTX threads
- Sleep time of the execution monitor

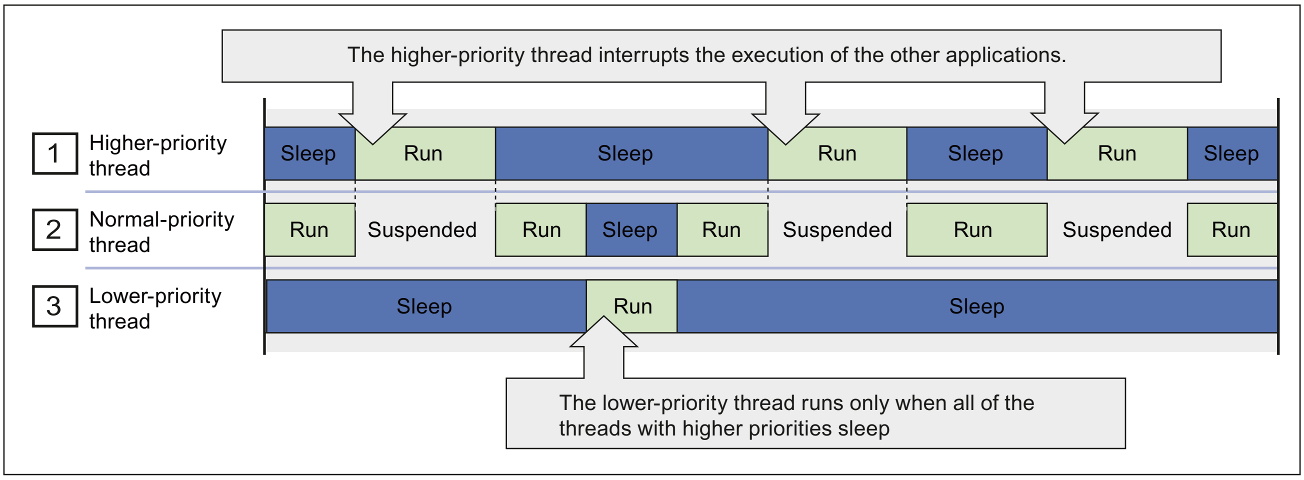
The tuning panel (Page 79) of WinLC RTX provides various tools for reducing jitter in the control program.

Jitter can also be caused by sources other than WinLC RTX:

- Jitter can be caused by the design of your control program. Differing logic branches in the control program can, for example, trigger deviations in the execution time.
- Jitter can be caused by the hardware of the computer. Jitter can, for example, be triggered by an operation with a long DMA cycle, such as a video card which uses the PCI bus. Jitter can also be caused by a driver, for example by a driver for the CD or floppy disk drive. Jitter triggered by hardware cannot be managed by software. IntervalZero provides an application that helps you assess whether PC hardware is suitable for the use of RTX extensions.
- Jitter can be triggered by an application that was created using the WinAC RTX Open Development Kit (ODK), for example, when a synchronous process takes too long for the execution. For additional information refer to the documentation for WinAC RTX ODK.

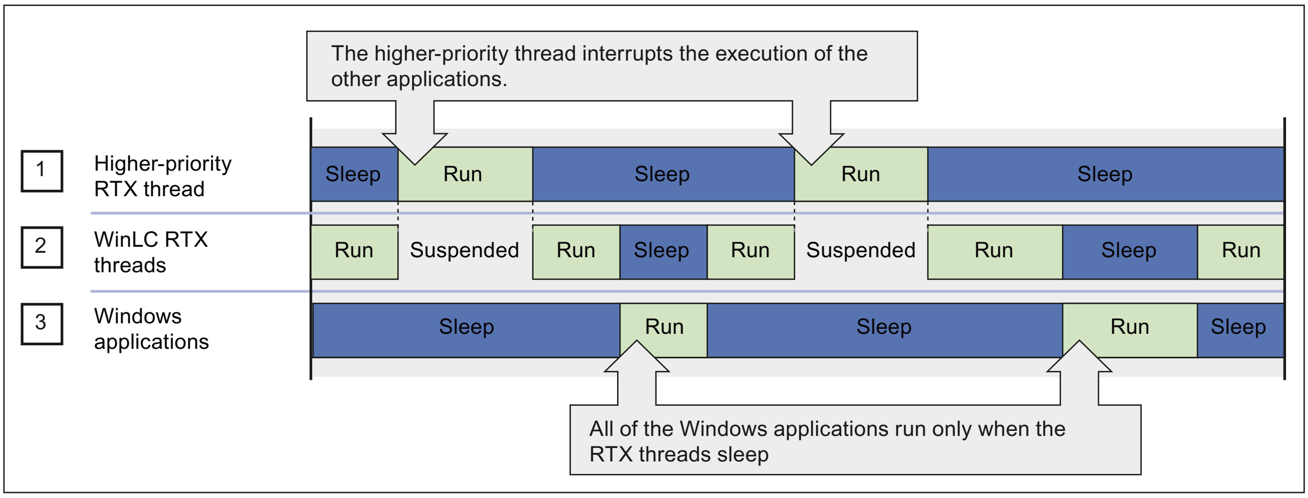
Priority settings for concurrent RTX applications can cause jitter.

Each RTX application that runs on your computer has one or more threads (or tasks) and each thread has a priority. The RTX system executes the RTX application threads with the highest priority first. Lower-priority threads are only executed when all of the higher priority threads are suspended (for example, to wait for some other activity to finish or to remain idle for a specified time). Threads with higher priorities interrupt and suspend the operations of other threads that have lower priorities. After the higher-priority thread finishes, the lower-priority thread resumes its operation.

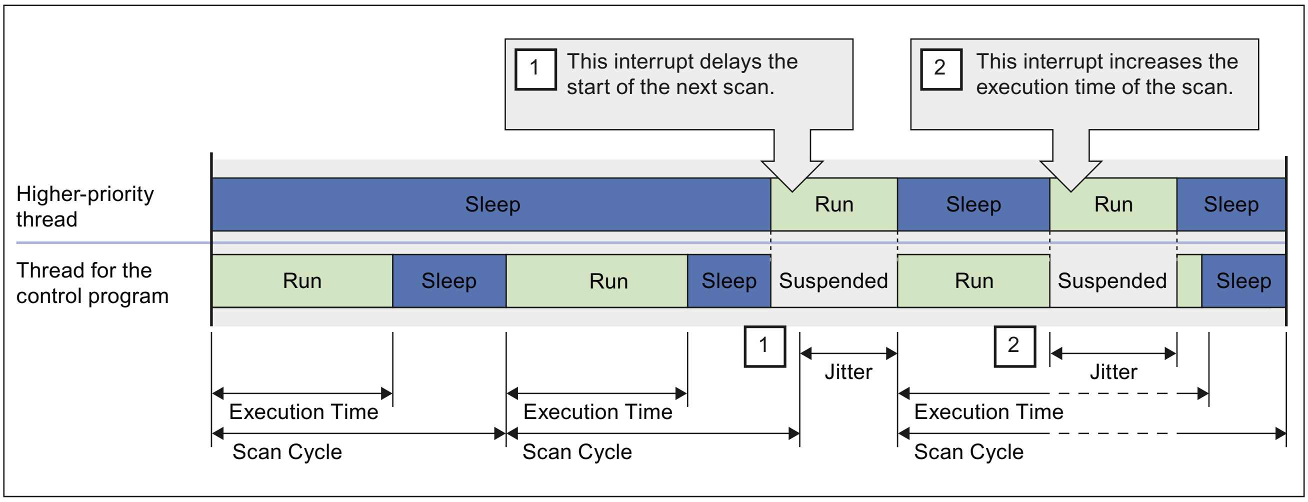

The higher-priority thread interrupts the execution of the other applications.
1 Higher-priority thread
Sleep
Run
Sleep
Run
Sleep
Run
Sleep
2 Normal-priority thread
Run
Suspended
Run
Sleep
Run
Suspended
Run
Suspended
Run
3 Lower-priority thread
Sleep
Run
Sleep
The lower-priority thread runs only when all of the threads with higher priorities sleep

WinLC RTX is operated in a real-time system (RTSS) that has a higher priority area than typical Windows priorities. All threads of WinLC RTX will be executed with a higher priority than threads for Windows applications. Windows applications cannot cause any jitter in WinLC RTX, but a different RTX thread with a higher RTSS priority than WinLC RTX can cause jitter.

You must also ensure that WinLC RTX and all other RTX applications make sufficient sleep time available so that Windows applications can be run.


The higher-priority thread interrupts the execution of the other applications.
1 Higher-priority RTX thread
Sleep
Run
Sleep
Run
Sleep
2 WinLC RTX threads
Run
Suspended
Run
Sleep
Run
Suspended
Run
Sleep
Run
3 Windows applications
Sleep
Run
Sleep
Run
Sleep
All of the Windows applications run only when the RTX threads sleep

Jitter can occur when a process with a higher RTSS priority interrupts and stops the execution of the controller. As you can see in the following figure, there are usually two types of jitter.



1 The threads with a higher priority can cause jitter by delaying the start of an OB. This can delay the start of the free cycle (OB 1) or the start of an interrupt OB (for example, OB 35 or OB 40).

2 The application with a higher priority can cause jitter because the execution time will be extended for a single scan cycle.

You can increase or decrease the priority of WinLC RTX threads in the tuning panel (Page 79). The higher the priority of the WinLC RTX threads in relation to the threads of other RTX applications, the less jitter will occur. However, you must ensure that WinLC RTX makes sufficient sleep time available so that other RTX and Windows applications can be run.

The tuning panel provides information that you can use to monitor jitter in the scan cycle.

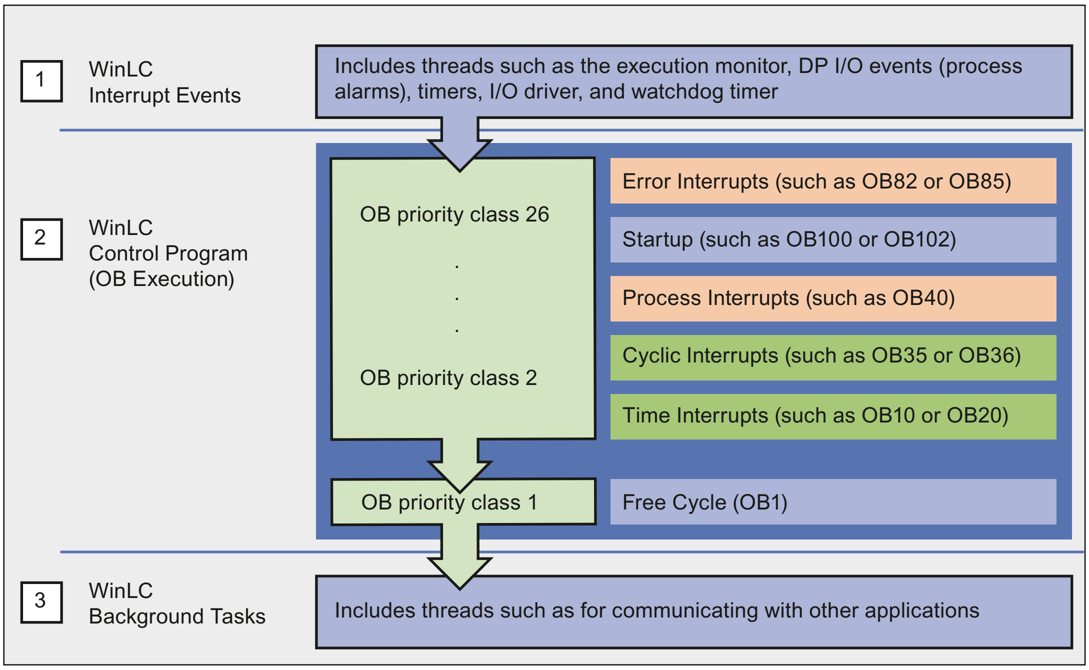
For detailed information on the priorities refer to the following help topic:

- Changing the priority (Page 193)
- Priority of the real time system (Page 193)

Priorities of the WinLC RTX threads can cause jitter

In addition to the thread that executes the OBs of the control program, WinLC RTX uses other threads, including some having a higher priority than the thread for the execution of the OBs. Some examples of threads with a higher priority are execution monitoring, the start event for an OB, the monitoring events, the times, communication interfaces and I/O events. All these threads with a higher priority can cause jitter in the execution of the control program.

The relative priorities (priority classes) of the OBs in the control program itself can also cause jitter. An error OB, for example, delays or interrupts the execution of all OBs with lower priorities.


1
WinLC
Interrupt Events
Includes threads such as the execution monitor, DP I/O events (process alarms), timers, I/O driver, and watchdog timer
2
WinLC
Control Program
(OB Execution)
OB priority class 26
.
.
.
OB priority class 2
Error Interrupts (such as OB82 or OB85)
Startup (such as OB100 or OB102)
Process Interrupts (such as OB40)
Cyclic Interrupts (such as OB35 or OB36)
Time Interrupts (such as OB10 or OB20)
OB priority class 1
Free Cycle (OB1)
3
WinLC
Background Tasks
Includes threads such as for communicating with other applications

1 The threads of the interrupt event have a higher priority than the thread for execution of the control program. These threads can cause jitter by interrupting the control program.

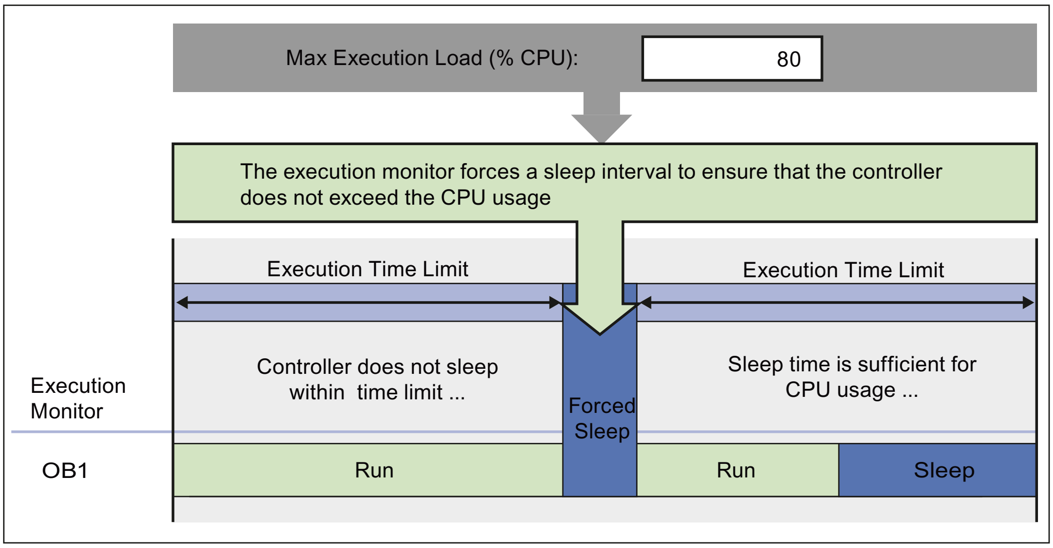
2 The thread for the execution of OBs encompasses the various priority classes of the OBs in the control program. The interrupt OBs can cause jitter not only by interrupting the free cycle (OB 1), but also by interrupting other interrupt OBs with a lower priority class.

3 The background tasks for WinLC RTX encompass the threads for communication with other applications, such as STEP 7. The thread for the execution of OBs and the threads with a higher priority influence the execution of these tasks.

The sleep time forced by the execution monitor can cause jitter.

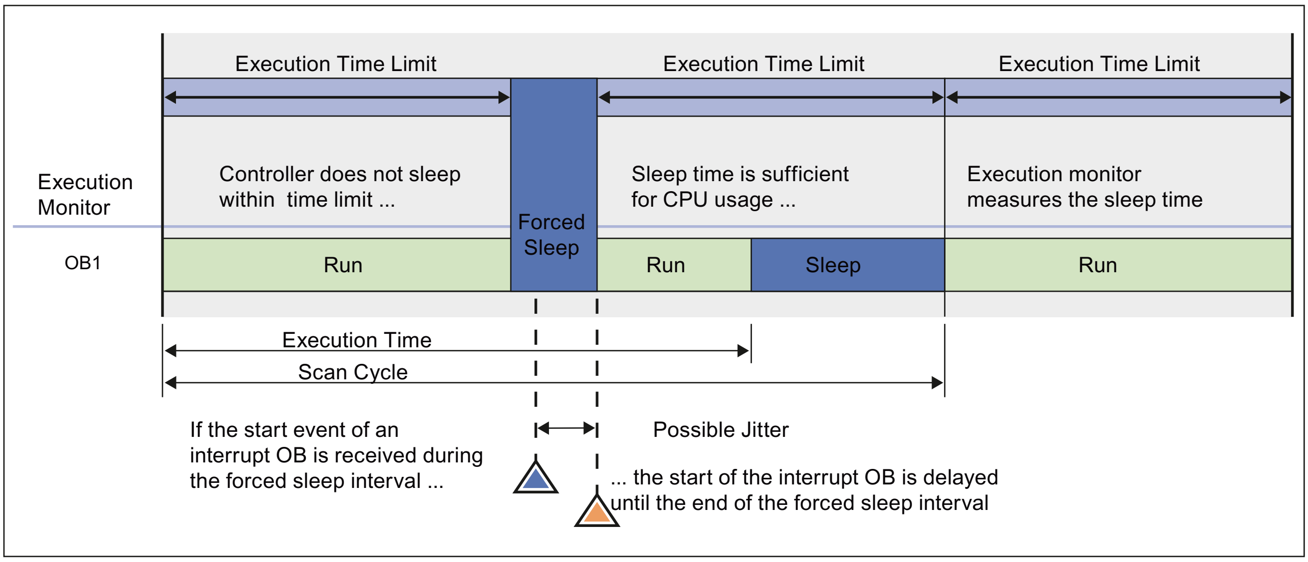
WinLC RTX has to be idle regularly (release the CPU) so that other applications can be executed. The free cycle includes a sleep time that follows the execution of OB 1. However, this sleep time can be interrupted by OBs with a higher priority. A scan cycle with a relatively long execution time can force other applications to wait too long for access to the CPU.

To ensure that the controller does not exceed a specified percentage of the CPU load, an execution monitor measures the sleep time within a fixed maximum execution time. If the controller is not idle for the specified period within the maximum execution time, then the execution monitor (Page 204) will force a sleep time.


Max Execution Load (% CPU):
80
The execution monitor forces a sleep interval to ensure that the controller does not exceed the CPU usage
Execution Time Limit
Execution Time Limit
Controller does not sleep within time limit ...
Sleep time is sufficient for CPU usage ...
Forced Sleep
Execution Monitor
OB1
Run
Run
Sleep

Because the execution monitor runs in a higher priority class than any OB, the controller cannot interrupt the forced sleep time. This can delay the start of an interrupt OB, for example OB 35, until the end of the forced sleep time. This delay during the execution of an interrupt OB results in jitter.

As a general rule to reduce jitter your control program should always be designed to keep the execution time of the OBs with higher priority as short as possible.


Execution Time Limit
Execution Time Limit
Execution Time Limit
Execution Monitor
Controller does not sleep within time limit ...
Forced Sleep
Sleep time is sufficient for CPU usage ...
Execution monitor measures the sleep time
OB1
Run
Run
Sleep
Run
Execution Time
Scan Cycle
If the start event of an interrupt OB is received during the forced sleep interval ...
Possible Jitter
... the start of the interrupt OB is delayed until the end of the forced sleep interval

WinLC RTX provides various options for managing the sleep time to avoid interruption-free forced sleep time:

- You can increase the parameter for the minimum sleep time (Page 200) to manage the sleep time for the free cycle (priority class 1 or OB 1).
- You can call SFC 47 ("WAIT") (Page 203) to insert an additional interruption-free sleep time into the control program and to manage the sleep time for an application-defined priority class (priority classes 2 to 24).
- You can change the sleep time monitoring algorithm for the execution monitor (Page 204) to manage the sleep time with a higher priority class than any OB.

8.3 Adjusting the priority of the controller

If other RTSS applications are executing on your computer in addition to WinLC RTX, you can adjust the priority of the controller to improve performance. If no other RTSS applications are installed, you do not need to adjust the controller priority. The priority of the controller determines how WinLC RTX runs in relation to the other RTSS applications that are running on the computer.

Adjusting the priority of the controller can reduce or increase the amount of jitter in the scan time. The tuning panel allows you to change the priority of the controller application. When you use the tuning panel to change the priority, the controller automatically ensures that its interrupt activities, such as those that schedule interrupt OBs, are also set to an appropriate priority.

A PC-based controller must maintain the essential features of a SIMATIC S7 PLC; however, the PC-based controller must also allow the other applications to run on the computer.

Procedure

To change the priority, follow these steps:

1. In the tuning panel, use the Priority slider to choose a priority based on the priority levels (Page 193) of your operating system. The new priority is displayed as you move the slider.
2. Click "Set" to set the priority to the new value.

8.3.1 Real-Time Subsystem Priorities

WinLC RTX provides real-time priorities for the most demanding control projects that are absolutely time-critical. Because WinLC RTX competes only with other applications in the real-time subsystem, the controller provides the most deterministic behavior, with a possibility for reducing jitter in the scan cycle to less than 500 microseconds.

Because the controller runs with an RTSS priority above the Windows priorities, the sleep time for the STEP 7 user program determines the amount of time for other Windows activities and applications. Provide sleep time (Page 195) that allows other application to run. Use the tuning panel to monitor the variation in scan times that occurs as the controller executes your STEP 7 user program.

Although the RTSS environment allows priorities from 1 to 127, WinLC RTX only runs up to priority 62. Another RTSS application thread could have a higher or lower priority than WinLC RTX.

The controller application installs with a default RTX priority of 50, which typically delivers satisfactory performance. If the controller competes with other RTSS applications for the computer resources, set the priority for the controller application to run either above or below the priority of the other RTSS applications.

8.3.2 Threads and priorities

The operating system of the computer uses a concept of execution threads (or tasks) to execute the applications. Each application has one or more threads, and each thread has a priority. The operating system executes the threads with the highest priority first. Lower-priority threads are only executed when all of the higher priority threads are suspended (for example, to wait for some other activity to complete or to "sleep" for a specified time). Threads with higher priorities interrupt and suspend the operations of other threads that have lower priorities. After the higher-priority thread finishes, the lower-priority thread resumes its operation.

WinLC RTX does not control priorities in user-specific software, such as asynchronous threads or other applications in the same environment.

Note

The CCX interface of the WinAC Open Development Kit (ODK) provides an ODK_CreateThread function. Calling the ODK_CreateThread function creates asynchronous threads with priorities that are adjusted when you change the priority of the controller.

If you do **not** use the ODK_CreateThread function to create threads (for example, if you use a Windows API call to create or call a thread), changing the priority of the controller does **not** adjust the priority of those threads.

Refer to the documentation of the WinAC Open Development Kit (ODK) for more information.

8.4 Managing the sleep time

8.4.1 Sleep Management Techniques

The controller provides other applications with additional computer resources during the sleep time. By managing the sleep time you can set the performance of the controller to ensure that all of the computer applications are executed at acceptable performance levels. You can manage the sleep times for the controller in many different ways:

- Changing the parameters for the minimum sleep time (Page 200): The minimum sleep time determines the sleep time that is added during the execution of the free cycle (OB 1). This sleep time only affects the OB priority class 1.
- Changing the parameters for the minimum cycle time (Page 200): The minimum cycle time specifies the minimum number of milliseconds that must elapse between the beginning of a free cycle and the beginning of the next free cycle.
- Calling SFC 47 in the STEP 7 User Program (Page 203): SFC 47 adds a sleep time into the execution of the STEP 7 user program. This sleep time affects the OB priority classes from 2 - 24.
- changing the execution monitor (Page 204): The execution monitor uses a sleep-monitoring algorithm (according to the parameters of the maximum execution time and the maximum execution load) to force a sleep time. The execution monitor runs asynchronous to the scan cycle. This sleep time affects all of the OB priority classes.

Managing the sleep time of the controller

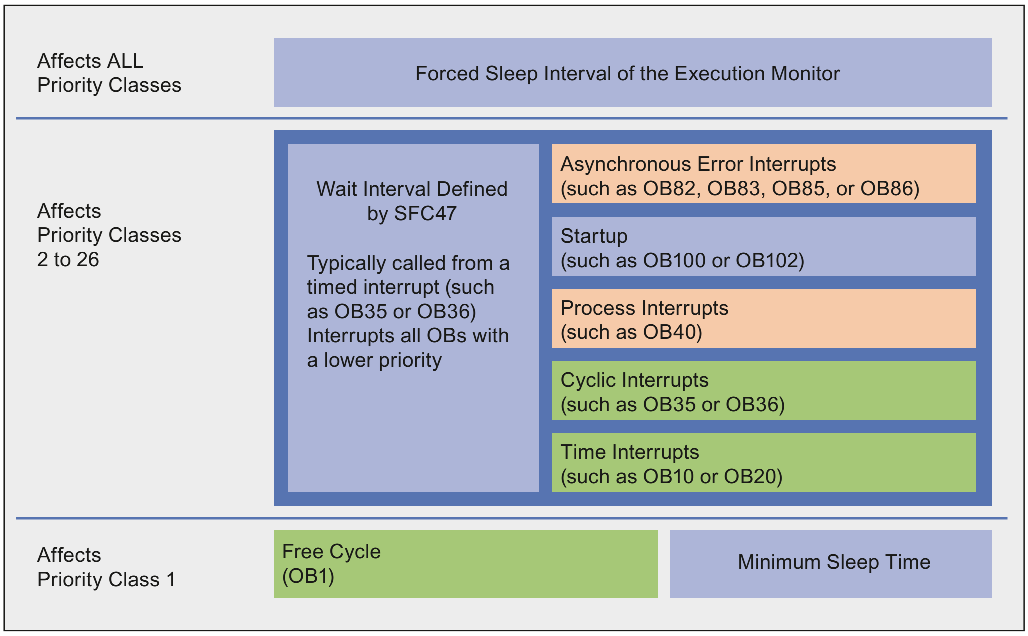
You must ensure that the controller has adequate sleep time for executing other applications since your controller shares the resources of your computer with other applications.

NOTICE

Adjusting the maximum value allowed by your controller application for the minimum sleep time is the most efficient method of ensuring that other applications will have adequate time for execution. The other methods for managing the sleep time provide the other applications with adequate time for execution; however, they may also reduce the performance of the controller.

The controller provides the following methods for managing the sleep time:

- The controller provides an execution monitor that forces the maximum execution load for the resources of the computer. The execution monitor measures the sleep time of the controller within a maximum execution time which is independent of the execution time of the cycle. If necessary, the execution monitor forces a sleep time in order to obtain the specified execution load. This forced sleep time interrupts the execution of OBs and can also delay the start of an interrupt OB.
- The controller provides a minimum sleep time which it adds to the end of a free cycle. This sleep time occurs after the execution of OB 1. The Minimum sleep time only affects priority class 1. An OB in a higher priority class can interrupt this sleep time.
  The controller does not change the minimum sleep time to compensate for the execution time of the interrupt OB. However, each forced sleep time (generated by the execution monitor) is subtracted from the sleep time that was generated by the minimum sleep time.
- The controller supports SFC 47 ("WAIT"), which adds a specific sleep time for the priority class of the OB, which accesses SFC 47. This sleep time affects OBs of the same or lower priority class in comparison to the OB that calls-up SFC 47; however, an OB with a higher priority class can interrupt this sleep time. You can create sleep time with SFC 47 that can also be interrupted to enable the controller to prevent jitters when critical application interrupts are processed.


Affects ALL Priority Classes
Forced Sleep Interval of the Execution Monitor
Affects Priority Classes 2 to 26
Wait Interval Defined by SFC47
Typically called from a timed interrupt (such as OB35 or OB36) Interrupts all OBs with a lower priority
Asynchronous Error Interrupts (such as OB82, OB83, OB85, or OB86)
Startup (such as OB100 or OB102)
Process Interrupts (such as OB40)
Cyclic Interrupts (such as OB35 or OB36)
Time Interrupts (such as OB10 or OB20)
Affects Priority Class 1
Free Cycle (OB1)
Minimum Sleep Time

8.4.2 Tuning the Scan Cycle

Procedure

As you test the performance of the controller during the development phase of your project, consider the following strategy for adjusting the sleep time:

1. Set the minimum sleep time parameter to 0 and run the STEP 7 user program.
   This allows you to determine whether there is unacceptable jitter in the scan cycle.
2. To reduce any unacceptable jitter, first use the tuning panel to increase the minimum sleep time and observe the effect on cycle time and CPU usage.
3. If the amount of jitter is still unacceptable, review the sections of the STEP 7 user program that are being affected by the jitter. If possible, have your STEP 7 user program call SFC 47 to add sleep time.
4. To further reduce any jitter, increase the execution time limit to the maximum possible execution time for your control program.

Result

The tuning techniques in most cases achieve an acceptable or imperceptible jitter time.

If the sleep management techniques do not provide adequate improvement in reducing jitter, consider increasing the priority of the controller (Page 193). (The priority of the controller is not the same as the priority class of an OB.)

See also

Tuning panel (Page 79)

Adjusting the Minimum Sleep Time and Cycle Time (Page 200)

Adjusting the sleep-monitoring algorithm of the execution monitor (Page 204)

Using SFC 47 to Add Sleep Time in the STEP 7 User Program (Page 203)

8.4.3 Example: Using the Execution Monitor Alone

To help explain the tools for managing the sleep time of the controller, the following example shows how the execution monitor (Page 204) alone generates sleep time. You do not specify a minimum sleep time (Page 200) in this example.

A second example, shows how adding a minimum sleep time to the free cycle affects the execution of the free cycle.

The following example describes the execution of a STEP 7 user program that uses OB 1 to start a 1-second timer, and then check the timer after an elapsed time of 1 second (1000 ms). The tuning panel of controller has been configured with the following parameters:

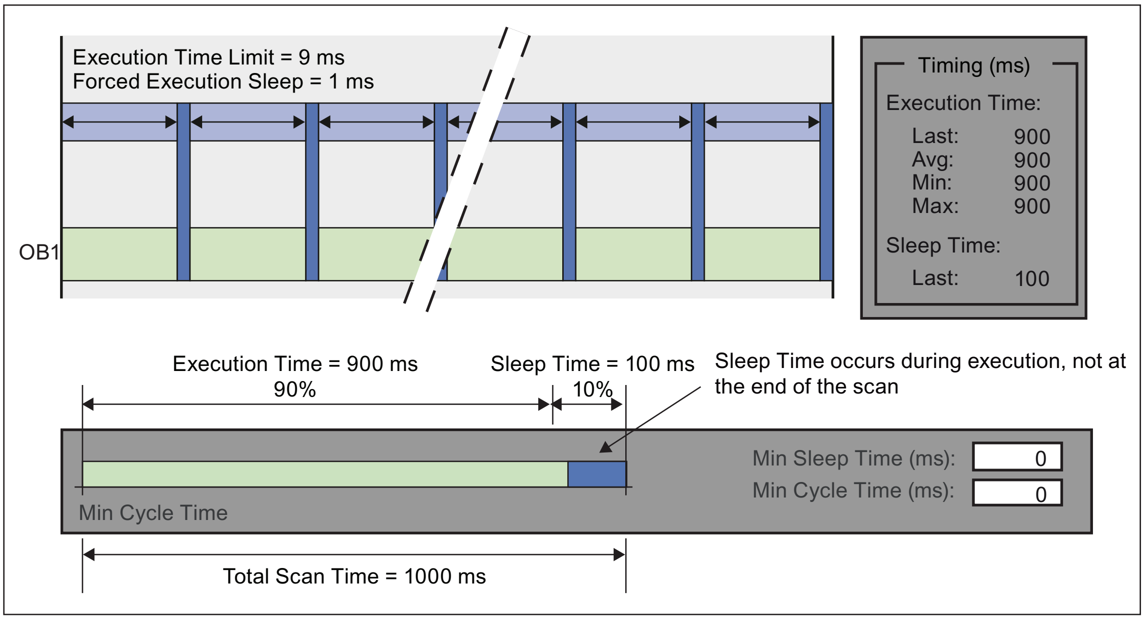
Parameter	Value
Execution Time	OB 1 takes 900 ms to execute.
Minimum Sleep Time	0 ms
Minimum Cycle Time	0 ms
Maximum Execution Load	90% (uses the default wake/sleep algorithm)
Execution Time Limit	9 ms (uses the default value)
Forced Execution Sleep	1 ms (uses the default value)

Sleep Time Generated by the Execution Monitor (Minimum Scan Time = 0)

If you set the minimum sleep time parameter to 0, the controller uses the execution monitor alone to provide sleep time. The figure shows the operation of the execution monitor, using the default values.

The execution monitor suspends the execution of OB 1 for 1 ms after every 9 ms of execution by default in order to enforce a limit of 90% execution load (CPU usage). For every 1 second of elapsed clock time, the default execution time for OB 1 is 900 ms, with forced sleep intervals totaling 100 ms.

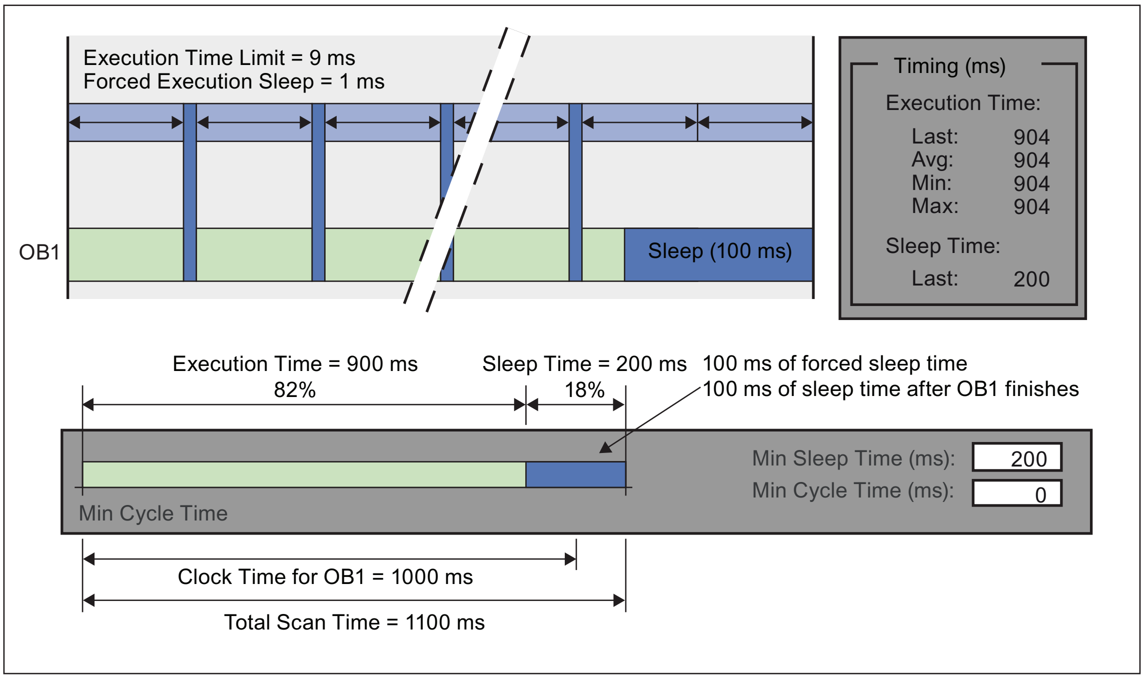
Notice that the sleep time occursat intervals within the execution of OB 1.



8.4.4 Example: Effect of Adding a Minimum Sleep Time for the Free Cycle

This figure shows how changing the minimum sleep time from 0 to 200 affects the execution of OB 1. The execution monitor still forces 100 ms of sleep time to occur during the execution of OB 1. With the minimum scan time parameter set to 200 ms, the controller then sleeps for only another 100 ms, for a combined total of 200 ms, before starting the next free cycle.

The total scan time increases to approximately 1100 ms: the execution time (900 ms) for OB 1, the forced sleep time (100 ms), and the sleep time at the end of the scan cycle (100 ms).



8.4.5 Adjusting the Minimum Sleep Time and Cycle Time

The tuning panel provides the following parameters, which you can use to manage the sleep time of the free cycle (priority class 1 or OB 1):

- The minimum cycle time (in milliseconds) specifies the minimum number of milliseconds that must elapse between the beginning of one free cycle and the beginning of the next free cycle. The value must be greater than the execution time to permit a sleep time to occur during the free cycle. You set up the minimum cycle time for the controller in STEP 7 when you create the hardware configuration (system configuration). You can change the minimum cycle time in the tuning panel, but the changes will be discarded when you close the controller. You must enter changes in STEP 7 to save them permanently.
- The minimum sleep time (in milliseconds) specifies how much sleep time is available during the free cycle (OB 1) to allow higher priority OBs and other applications to use the resources of the computer. The controller automatically saves any changes to the minimum sleep time that you make in the tuning panel. You do not have to be working in STEP 7 in order to permanently save changes to the minimum sleep time.

The execution of the free cycle is influenced by the values for minimum sleep time and minimum cycle time.

- The minimum cycle time in itself produces a fixed scan cycle with variable sleep time (provided the minimum cycle time is long enough to encompass the execution time and the sleep time).
- The minimum sleep time in itself produces a fixed sleep time with variable cycle time, depending on the length of the execution time.

The value for minimum sleep time ensures that a sleep time of the configured length occurs in each free cycle, even if the value for minimum cycle time is too small. The controller relinquishes control of the CPU during sleep time. This sleep time is the either the configured minimum sleep time or another sleep time calculated using the minimum cycle time parameter, whichever is greater.

⚠ WARNING

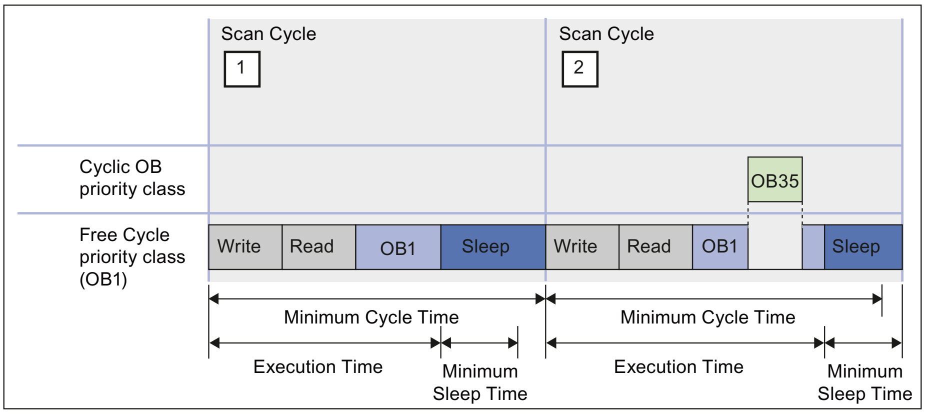
If you set a value for minimum cycle time that is greater than the watchdog, WinLC goes to STOP mode at the end of the watchdog interval of the first scan cycle.

If the controller switches unexpectedly to STOP mode, this can result in injuries to persons or property damage.

Do not set a minimum cycle time that is longer than the cycle watchdog set in STEP 7 HW Config.

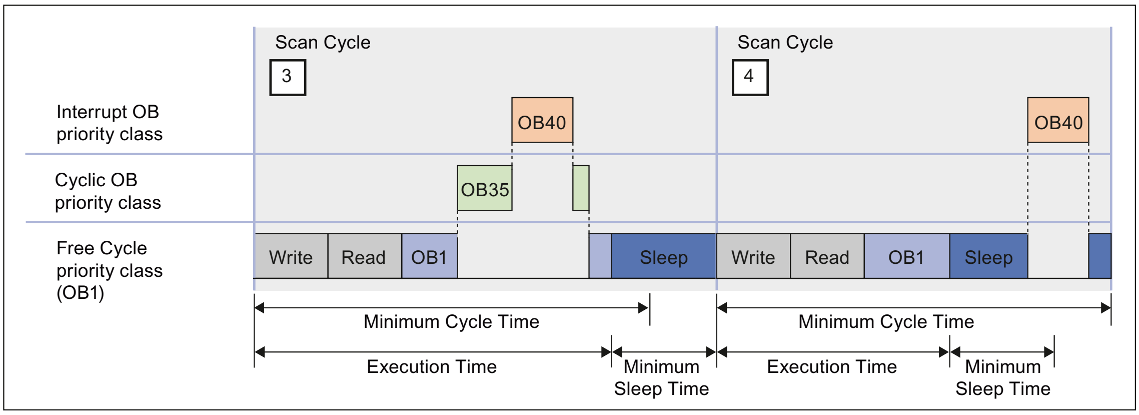
Parameters that affect the sleep time of the free cycle

The following figures explain the relationship between the parameters for execution time, minimum sleep time and minimum cycle time.



1 In the first example scan cycle in the figure above, the sum of execution time plus minimum sleep time is less than the minimum cycle time. In this case, the controller increases the sleep time until the minimum cycle time is reached.

2 In the second example in the figure above, execution of OB 35 increases the execution time and the sum of execution time and minimum sleep time is greater than the minimum cycle time. In this case, the controller waits until the end of the minimum sleep time before starting the next scan cycle.



3 In the third example in the figure above, the controller executes a cyclic interrupt (OB 35) and an I/O interrupt (OB 40). The execution time exceeds the minimum cycle time and the controller waits until the end of the minimum sleep time before executing the next scan cycle.

4 In the fourth example in the figure above, the controller executes OB 40 during the sleep time as soon as OB 1 ends. In this case, the controller waits until the end of the minimum cycle time before starting the next scan cycle.

Because execution of OB 40 does not reset the counter for the minimum sleep time, it is possible that the controller may not have enough sleep time to process other Windows applications. You must then take other measures to ensure that the controller has sufficient sleep time.

Tips

You can use the following methods to change the controller performance using the parameters for minimum sleep time and minimum cycle time.

- You can test the values for minimum cycle time in the tuning panel. Once you have determined the optimum value for minimum cycle time, update the system configuration for the controller in STEP 7 and then download the configuration.

  ### Note

  When the controller changes from STOP to RUN mode, all values that were entered in the tuning panel are deleted and the minimum cycle time is reset to the value that was saved in the system configuration.

- To ensure that the controller executes a fixed scan cycle, use the parameter for the minimum cycle time.
- To ensure that sleep time is always available even if the execution time is changed, set the the minimum cycle time to the value "0" (default value) and change the minimum sleep time as needed. Changing the minimum sleep time is especially useful during development of the STEP 7 user program.

If you tune the controller operation, you need to ensure that the time for completing the scan cycle can be extended in the following situations:

- The controller is processing other OBs (such as OB 40 and OB 35) with a higher priority than OB 1.
- You are monitoring and testing the STEP 7 user program.
- You are displaying the status of the STEP 7 user program in a variable table (VAT) in STEP 7.
- An application with a higher priority is running on your computer.
- The controller is connected to an HMI interface, for example, WinCC.

Additional methods for managing sleep time

- Addition of sleep time in the STEP 7 user program with SFC 47 (Page 203)
- Changing the sleep time monitoring algorithm of the execution monitor (Page 204)

8.4.6 Using SFC 47 to Add Sleep Time in the STEP 7 User Program

SFC 47 (WAIT) inserts sleep time into the execution of the STEP 7 user program, allowing you to manage the sleep time for a control program by inserting the sleep time in a specific priority class. When the STEP 7 user program calls SFC 47, the controller suspends the execution of the OB for a specified number of microseconds and sleeps. During this sleep period, the controller can interrupt this sleep period to execute an interrupt OB. Because an OB with a higher priority class can interrupt the sleep time, higher priority OBs execute with less chance of jitter.

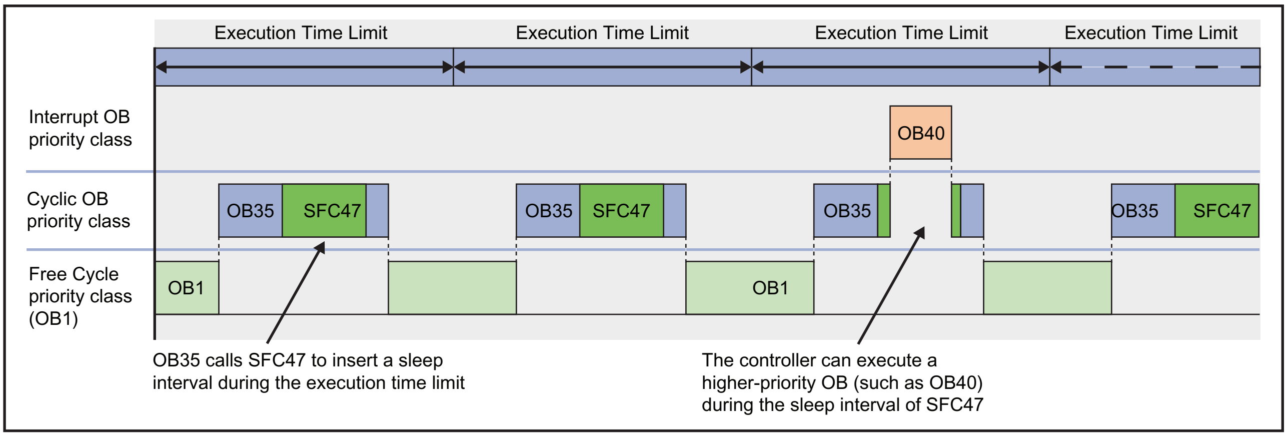
Procedure

To use SFC 47 to add sleep time, follow these steps:

1. Call SFC 47 from an OB in your STEP 7 user program. Typically, you call SFC 47 from a cyclic OB (such as OB 35) that starts within the execution time limit of the execution monitor.
2. To provide greater control over when the sleep time occurs, you can define which OBs are affected by setting the priority class of the OB that calls SFC 47.
   For more information, refer to the example: Avoiding Jitter in the Start Time of an OB (Page 209)

As shown in the following figure, you can use SFC 47 to insert a sleep interval that can satisfy the execution monitor and still allow the controller to handle an interrupt OB. By using a cyclic OB (such as OB 35) to call SFC 47, you can ensure that the sleep interval occurs within the execution time limit of the execution monitor.

The sleep time parameter is rounded up to the nearest multiple of the HAL timer (Page 257) period defined in the RTX Properties dialog. For example, if the HAL timer period is 500 microseconds (the default), and the sleep time parameter is 1200 microseconds, WinLC RTX rounds up the sleep time to 1500 microseconds.


Execution Time Limit
Execution Time Limit
Execution Time Limit
Execution Time Limit
Interrupt OB priority class
Cyclic OB priority class
Free Cycle priority class (OB1)
OB40
OB35
SFC47
OB35
SFC47
OB35
OB35
SFC47
OB1
OB1
OB35 calls SFC47 to insert a sleep interval during the execution time limit
The controller can execute a higher-priority OB (such as OB40) during the sleep interval of SFC47

Additional Methods for Managing the Sleep Time

- Adjusting the Minimum Sleep Time and Cycle Time (Page 200)
- Adjusting the Sleep-Monitoring Algorithm of the Execution Monitor (Page 204)

8.4.7 Adjusting the sleep-monitoring algorithm of the execution monitor

The execution monitor uses a sleep-monitoring algorithm to ensure that the controller does not exceed a configurable maximum execution load for the CPU usage within a monitor interval.

The monitoring interval is calculated as the amount of time such that the maximum load percentage of the monitor interval equals the entered execution time limit. The execution monitor calculates the forced execution sleep time as the difference between the monitor interval and the execution time limit.

The execution monitor determines whether to insert a forced execution sleep time if the OB execution exceeds the execution time limit.

If there is sufficient sleep time within the monitor interval, the execution monitor does not affect the execution of the program. Otherwise, the execution monitor forces a sleep interval. The default execution load is 90%, and the default execution time limit is 9 ms. For the default settings, the execution monitor calculates a monitor interval of 10 ms and a forced sleep time of 1 ms.

The execution monitor runs asynchronous to the scan cycle and measures the amount of sleep time that occurs within the monitor interval and enforces a minimum sleep interval.

- If the scan cycle (execution time plus sleep time) is shorter than the monitor interval and the sleep time is greater than or equal to the forced sleep value: The execution monitor does not force a sleep interval.
- If the scan cycle is longer than the monitor interval: The execution monitor forces the controller to sleep for the required amount of time. Because the execution monitor runs in a higher priority class than any OB, the controller cannot interrupt the forced sleep interval. This could delay the start of an interrupt OB, such as OB 35 or OB 40.

Use the tuning panel to configure the parameters for the sleep-monitoring algorithm of the execution monitor.

For more information, see the example: Avoiding jitter in the start time of an OB (Page 209)

This section contains the following information:

- Operation of the execution monitor
- Parameters of the sleep-monitoring algorithm
- Configuring the parameters of the sleep-monitoring algorithm
- Situations that cause the execution monitor to force a sleep interval
- Situations that prevent the execution monitor from providing sufficient sleep time

Operation of the execution monitor

In addition to the sleep time that is added to the scan cycle (based on the minimum sleep time and minimum cycle time parameters), the execution monitor uses a sleep-monitoring algorithm that is based on a maximum execution load (percentage of CPU usage). For the default execution load (90% CPU usage), the execution monitor measures the length of time that the controller sleeps during the monitor interval of 10 ms and ensures that the controller sleeps for at least 1 ms.

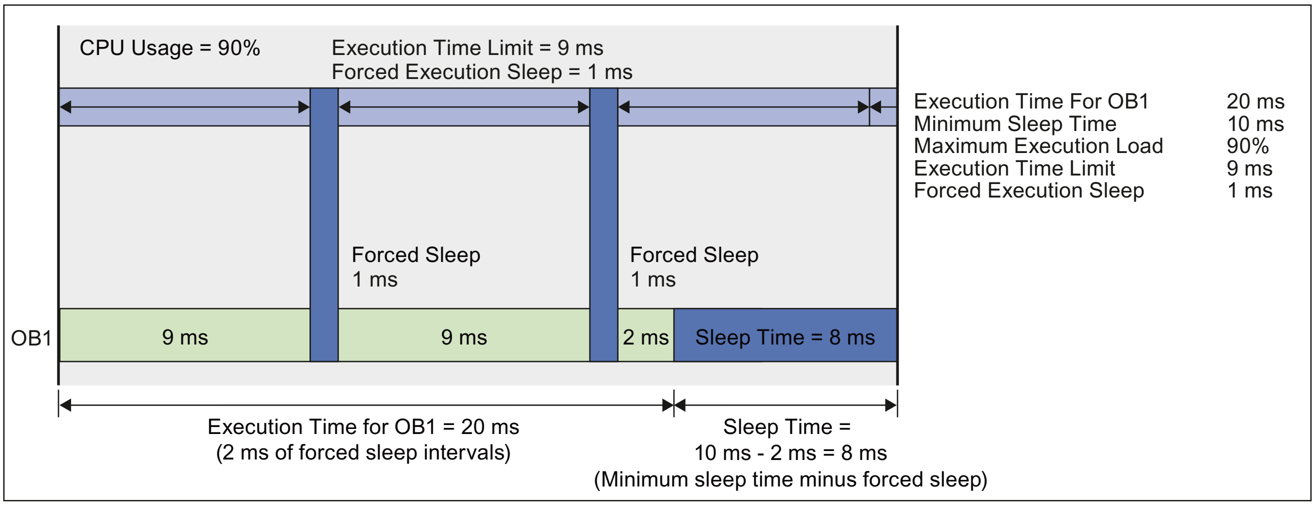
By measuring the sleep time, the execution monitor ensures that the controller allows the other applications to access the computer resources while the controller sleeps. The execution monitor also provides the safety net in cases where there are programming errors (for example, an infinite loop in OB 100) that are not handled with other mechanisms.

The difference between the forced sleep time and the minimum sleep time is that the controller can interrupt the minimum sleep time to handle interrupts (such as OB 35 or OB 40), but cannot interrupt the forced execution sleep time.

When the execution monitor forces a sleep interval, the following actions occur:

- The controller immediately suspends the execution of the OB for the forced sleep interval. By forcing a sleep interval, the execution monitor increases the actual time between starting and finishing the OB being executed.
- The controller cannot respond to the start event for any interrupt OB until the end of the forced sleep time. Delaying the start of the OB (for example, OB 35 or OB 40) until the end of the forced sleep time creates jitter or latency in the actual start time of the OB.

The following illustration shows how the execution monitor might affect a control program. Because the execution time for OB 1 in this example is greater than the execution time limit, the execution monitor inserts a 1-ms sleep interval after the first two monitor intervals. However, the execution monitor does not insert a forced sleep interval in the third monitor interval because the controller sleeps longer than the required forced sleep time as required by the configured minimum sleep time.


CPU Usage = 90%
Execution Time Limit = 9 ms
Forced Execution Sleep = 1 ms
Forced Sleep
1 ms
Forced Sleep
1 ms
OB1
9 ms
9 ms
2 ms
Sleep Time = 8 ms
Execution Time for OB1 = 20 ms
(2 ms of forced sleep intervals)
Sleep Time =
10 ms - 2 ms = 8 ms
(Minimum sleep time minus forced sleep)

Execution Time For OB1	20 ms
Minimum Sleep Time	10 ms
Maximum Execution Load	90%
Execution Time Limit	9 ms
Forced Execution Sleep	1 ms

Note

The execution monitor runs asynchronous to the scan cycle. The example above shows the execution monitor measuring time from the beginning of the scan cycle, but because the execution monitor runs asynchronous to the control program, the beginning of the execution time limit of the execution monitor does not necessarily coincide with the beginning of the scan cycle.

Parameters of the sleep-monitoring algorithm

Table 8- 1 The sleep-monitoring algorithm of the execution monitor uses the following parameters:

Parameters	Description
Execution time limit	This value defines the maximum time (in microseconds) that the execution monitor allows for OB execution before exceeding the configured maximum execution load (CPU usage) of the monitor interval. To determine the CPU load caused by the execution of the control program, the execution monitor measures the time that the controller sleeps during the monitor interval. If the controller does not use a sufficient amount of sleep time (indicating that the CPU load exceeds the maximum execution load), the execution monitor forces the controller to sleep for the remainder of the required forced execution sleep time. The default value is 9000 microseconds (9 ms). **Note:** If you set this value greater than approximately 50000 (50 ms), you may observe jitter in Windows applications and in response to the mouse or keyboard. Test that the execution time limit you choose is appropriate for your application.
Maximum execution load	This value defines the maximum percentage of CPU usage that is allowed for the controller to execute OBs during each monitor interval. The default value is 90%.
Forced execution sleep time	This read-only field shows how much sleep time (in microseconds) the execution monitor requires during the monitor interval to satisfy the requirement for the maximum execution load. The execution monitor subtracts any controller sleep time that occurs during a monitor interval from the forced execution sleep time to determine how much sleep time (if any) to force. The forced execution sleep time is a calculated number based on the execution time limit and the maximum execution load. The execution monitor corrects this value as required, depending on the capability of the operating system configuration to have timers operate at the specified intervals. For example, if the HAL timer period (Page 257) (in the "RTX Properties" dialog box) is set to 500 microseconds, you cannot set the forced execution sleep time to 1200 microseconds. The value would be rounded up to 1500 microseconds. The default value is 1000 microseconds (1 ms).

The execution monitor uses the execution time limit and the maximum execution load to calculate the forced execution sleep. For example, the execution monitor uses the 90% usage rate and the 9-ms execution time limit to calculate a 1-ms sleep time. In this case, the monitor interval is 10 ms such that 90% of the monitor interval corresponds to the entered execution time limit (9 ms).

During the monitor interval, the execution monitor measures the actual amount of time that no OBs are executing (the sleep time), and performs the following actions:

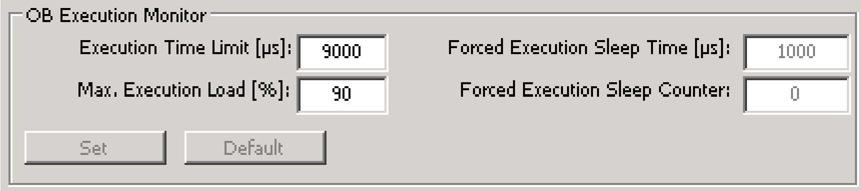
- If the controller sleeps **longer** than the sleep time (forced execution sleep time), then the execution monitor restarts another monitor interval equal to the execution time limit and does not affect the control program.
- If the controller sleeps **less** than the sleep time (forced execution sleep time), then the execution monitor blocks the execution of any OBs for the **remainder** of the sleep time.

Any control program sleep time imposed because of the sleep-monitoring algorithm is subtracted from the sleep time configured for the end of the free cycle as defined by the minimum sleep time parameter.

The default value for the "Execution Time Limit" interval is 9000 microseconds (or 9 milliseconds) and the default value for the "Forced Execution Sleep" interval is 1000 microseconds (or 1 millisecond). This ratio ensures that the control program execution cannot use more than 90% of the CPU time in any of the worst case situations described above.

Configuring the parameters of the sleep-monitoring algorithm

The parameters of the sleep-monitoring algorithm of the execution monitor are configurable in the tuning panel.



OB Execution Monitor

Execution Time Limit [µs]: 9000

Max. Execution Load [%]: 90

Forced Execution Sleep Time [µs]: 1000

Forced Execution Sleep Counter: 0

To change the sleep-monitoring parameters, follow these steps:

Enter values in the Execution Time Limit and the Max. Execution Load fields.
You can change one of the fields or both.

Click "Set" to set the parameters.

To restore the default sleep-monitoring parameters, follow these steps:

1. Click "Default" to display the default parameters.
2. Click "Set" to set the default parameters.

Changes to the sleep-monitoring parameters take effect when the controller is in RUN mode.

Situations that cause the execution monitor to force a sleep interval

Table 8- 2 The controller must relinquish control of the CPU long enough to satisfy the maximum execution load. Typically, the sleep time that is added to the end of the scan cycle allows sufficient time for the operating system to process the other Windows applications. However, some situations may require that the execution monitor force a sleep time.

Condition	Description
Execution time for the control program exceeds the execution time limit.	The configured minimum sleep time for the free cycle begins after OB 1 finishes. If the execution time is longer than the execution time limit, the execution monitor forces a sleep interval because the controller did not sleep the required amount within the monitor interval.
Minimum sleep time is insufficient for the maximum execution load.	Even when the scan cycle is less than the execution time, the minimum sleep time may not provide enough sleep time. In this case, the controller would exceed the maximum execution load. The execution monitor forces an additional sleep interval to ensure that the operating system can run the other applications.
Interrupt OBs reduce the sleep time.	To process an interrupt OB (such as OB 35, OB 40, or OB 85), the controller can interrupt the sleep time for the scan cycle. This reduces the time that the controller actually sleeps and can cause the controller to exceed the maximum execution load, which affects the performance of the other Windows applications. By forcing a sleep interval, the execution monitor ensures that the other Windows applications can be processed.

Situations that prevent the execution monitor from providing sufficient sleep time

Table 8- 3 In some cases, a high execution time limit can prevent the execution monitor from managing the sleep time of the control program adequately. Under the following conditions, the control program utilizes too much CPU time, which can result in jitter in Windows response time to the mouse, keyboard, or other applications. For either case, the problem can be resolved by lowering the execution time limit.

Condition	Description
Execution time for the startup OB (OB 100 or OB 102) and the configured execution time limit exceed approximately 50 ms.	During startup, the controller turns the watchdog timer off and cannot handle a program error, such as a loop in the logic of the OB or an excessively long initialization routine. Because the scan cycle does not provide any sleep time for the startup OB (such as OB 100), the execution monitor cannot relinquish CPU time for other applications. If the startup OB executes for more than approximately 50 ms, jitter can occur in Windows response time to the mouse, keyboard, or other applications.
Execution time for the control program and the configured execution time limit exceed approximately 50 ms.	Whenever the operating system has to wait more than approximately 50 ms to process the other Windows applications, the performance of those applications can be noticeably affected. This can be a problem for an OB 1 with a long execution time, especially if other OBs (such as OB 35 or OB 40) extend the execution of OB 1. Because the sleep time is added at the end of the scan cycle, and the execution time limit is set to a high value, the sleep intervals are then spaced too far apart for the other Windows applications to perform naturally.

Additional methods for managing sleep time

- Adjusting the minimum sleep time and minimum cycle time parameters (Page 200)
- Inserting sleep time into the control program (SFC 47 "WAIT") (Page 203)

8.4.8 Example: Avoiding Jitter in the Start Time of an OB

The following example discusses two possible solutions for a program that experiences jitter in the start of a cyclic interrupt (OB 32 to OB 36):

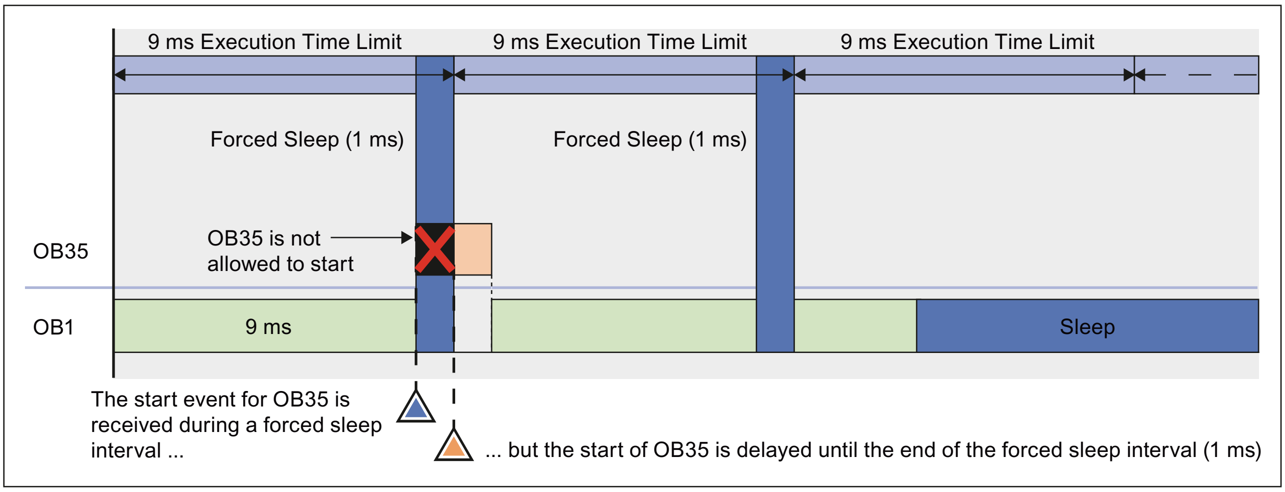
- Inserting a sleep interval into the execution of your STEP 7 user program. For this solution, you call SFC 47 ("WAIT") and specify the length of time to sleep.
  The controller can interrupt this sleep interval to process other OBs.
- Changing the sleep-monitoring algorithm of the execution monitor.
  For this solution, you use the tuning panel to change the execution time limit.

Scenario

Table 8- 4 In the example, a STEP 7 user program consists of OB 1 and OB 35. OB 1 takes 20 ms to execute, and OB 35 starts every 100 ms and takes 1 ms to execute. The controller has been configured with the following parameters:

Parameter	Value
Execution Time for the STEP 7 user program	OB 1: 20 ms, and OB 35: 1 ms
Minimum Sleep Time	10 ms (uses the default value)
Minimum Cycle Time	0 ms (uses the default value)
Maximum Execution Load	90% (uses the default wake/sleep algorithm)
Execution Time Limit	9 ms (uses the default value)
Forced Execution Sleep	1 ms (uses the default value)

The sleep time (10 ms) is added to the scan cycle after OB 1 has finished. However, because the execution time for OB 1 (20 ms) exceeds the execution time limit (9 ms), the controller exceeds the configured maximum execution load (90%) by not sleeping during the execution time limit. Therefore, the sleep-monitoring algorithm forces the controller to sleep for 1 ms after every 9 ms that OB 1 executes. As shown in the following figure, this forced sleep can cause a variance or jitter of up to 1 ms between the time that the start event and the time that the controller starts to execute OB 35. This jitter happens because all controller operations are suspended during a forced sleep interval. Similarly, OB 35 could be suspended for 1 millisecond if the end of the execution time limit interval occurs while OB 35 is executing.


9 ms Execution Time Limit
9 ms Execution Time Limit
9 ms Execution Time Limit
Forced Sleep (1 ms)
Forced Sleep (1 ms)
OB35 is not allowed to start
OB35
OB1
9 ms
Sleep
The start event for OB35 is received during a forced sleep interval ...
... but the start of OB35 is delayed until the end of the forced sleep interval (1 ms)

For many applications, a 1-ms jitter might be acceptable. However, you have several options for removing this jitter:

- You can modify the STEP 7 user program to call SFC 47 and insert sleep time that can be interrupted by OB 35.
- You can adjust the parameters for the sleep-monitoring algorithm to avoid the jitter caused by the execution monitor.

Solution 1: Insert a sleep interval into the execution of your STEP 7 user program

You could avoid the forced sleep interval by using SFC 47 to add a periodic sleep interval that occurs within the execution time limit (for this example, 9 ms). This sleep interval not only ensures that the sleep-monitoring algorithm does not force the controller to sleep, but also allows the controller to suspend this sleep interval and execute any OB that has a higher priority than the OB that called SFC 47.

For this example, you can use SFC 47 to remove the jitter in OB 35:

- By ensuring that SFC 47 executes at a specified time. The STEP 7 user program calls SFC 47 from an OB (such as OB 36) that has a priority greater than OB 1.
- By ensuring that OB 35 executes as scheduled. You configure OB 36 to have a lower priority than OB 35.
- By ensuring a sufficient sleep interval during the execution time limit. You configure SFC 47 to wait for 3 ms, which ensures a sleep interval of at least 2 ms.

To maintain a 50% ratio for CPU usage (20 ms execution time for OB 1 with a 10 ms minimum sleep time), configure OB 36 to run every 6 ms (so that OB 1 executes for 6 ms, then sleeps for 3 ms). You can then change the minimum sleep time to 0 ms, unless you want to decrease the ratio for CPU usage.

To create an OB 36 that calls SFC 47 to create a 3 ms sleep interval, follow these steps:

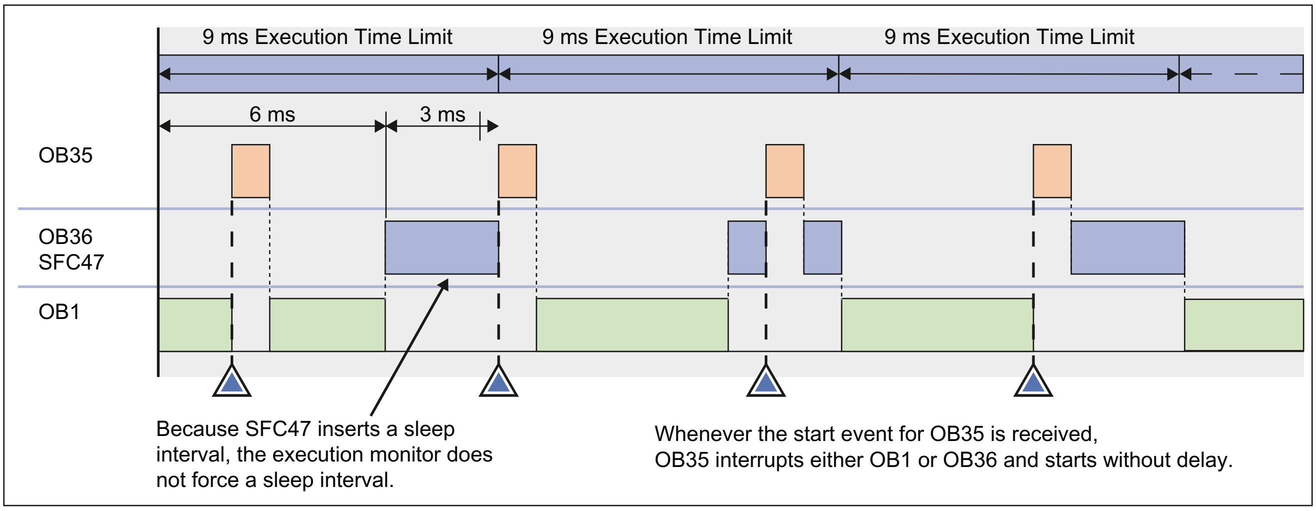
1. From the STEP 7 Program Editor, create an OB 36 for your STEP 7 user program.
2. Enter the following program:

```
CALL "WAIT"    // SFC 47 wait function
WT: 3000       // 3000 microseconds or 3 milliseconds
```

To configure the priority level time and execution time for OB 36, follow these steps:

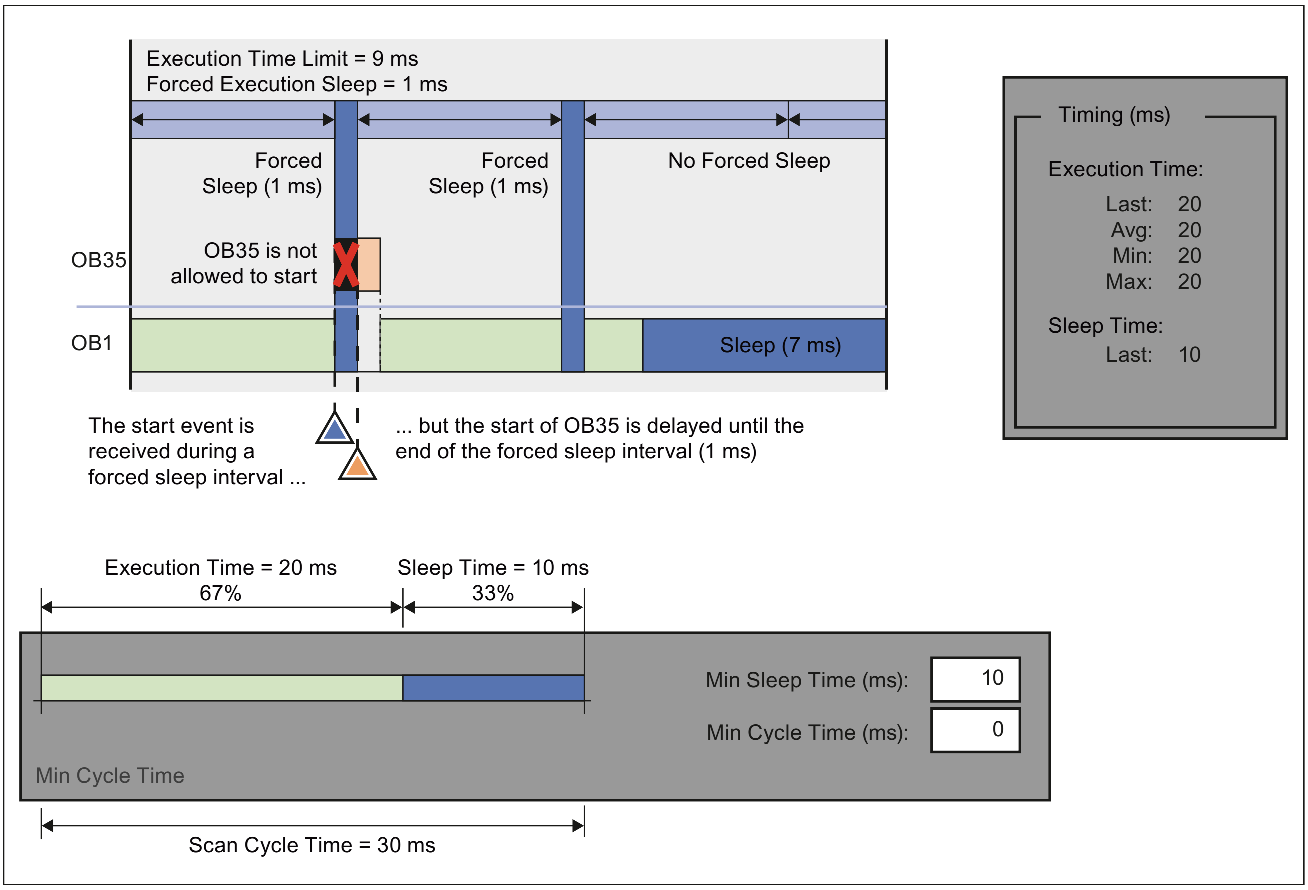
1. Using the STEP 7 Hardware Configuration tool, open the WinLC Properties dialog and select the Cyclic Interrupt tab.
2. Set the priority for OB 36 to 2 (or any other priority lower than the priority for OB 35).
3. Configure OB 36 to execute every 6 ms (by entering 6 in the Execution field).

The following figure shows how SFC 47 affects the execution of the STEP 7 user program. Because OB 36 ensures that the controller sleeps at least 1 ms within the 90% wake interval, the execution monitor does not insert a forced sleep interval. Therefore, OB 35 executes without any delay or jitter.


9 ms Execution Time Limit
9 ms Execution Time Limit
9 ms Execution Time Limit
6 ms
3 ms
OB35
OB36
SFC47
OB1
Because SFC47 inserts a sleep interval, the execution monitor does not force a sleep interval.
Whenever the start event for OB35 is received, OB35 interrupts either OB1 or OB36 and starts without delay.

Solution 2: Change the sleep-monitoring algorithm to eliminate the forced sleep interval

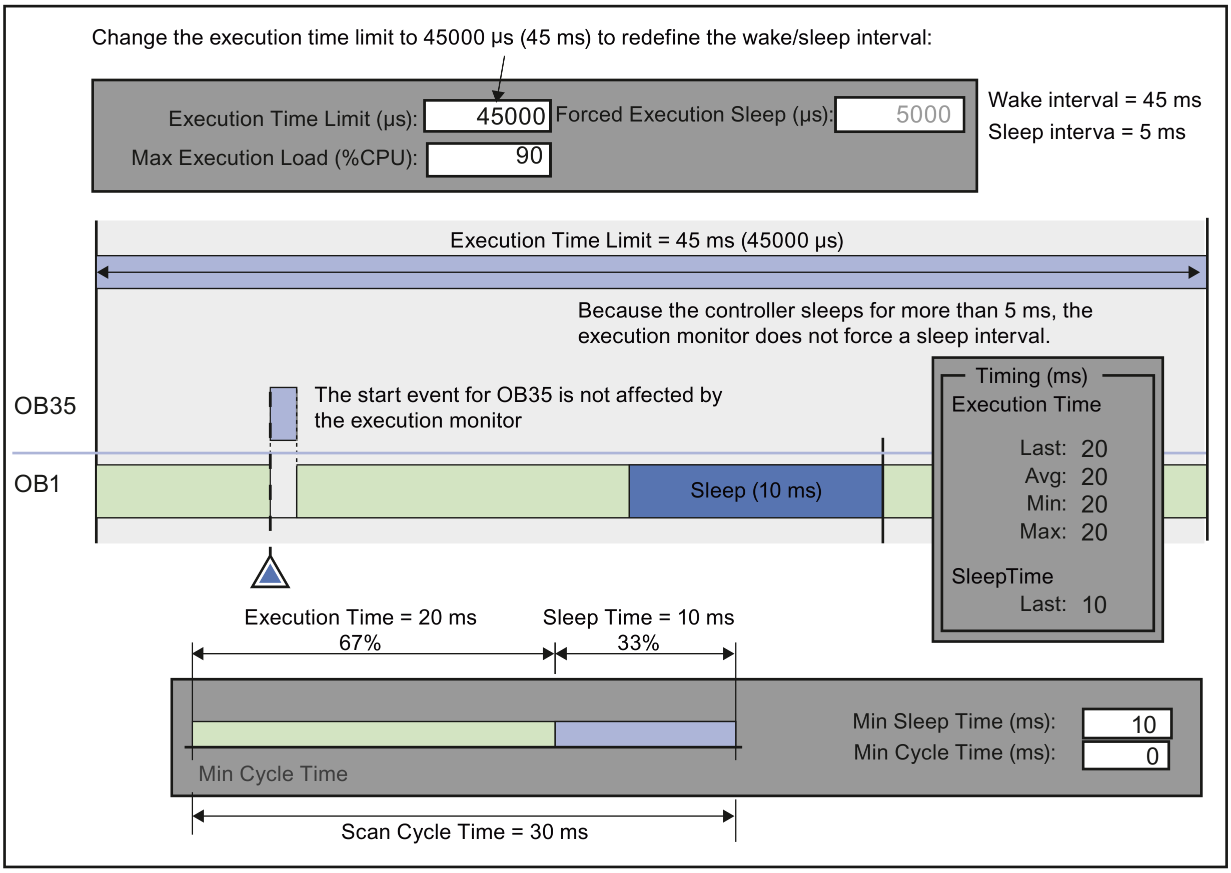
The following figure shows the jitter in the start time of OB 35 and also shows the values displayed by the tuning panel. Notice that the tuning panel shows only the information about OB 1. The tuning panel does not display information about OB 35. For this example, the execution time for OB 1 is 20 ms. With the minimum sleep time of 10 ms, the total free cycle time is 30 ms. OB 35 and other interrupt OBs can make the total scan time more than this, depending on how fast the interrupt OBs execute.



By changing the parameters of the sleep-monitoring algorithm, you can configure the execution monitor to use the minimum sleep time in the free cycle. For example: if the longest total scan time for this example is less than 45 ms, change the execution time limit to 45000 microseconds (45 ms):

1. Open the tuning panel.
2. Change the execution time limit to 45000 (microseconds). For this example, do not change the value for the maximum execution load.
3. Apply the new value.

The following illustration shows the effect of the changed execution time limit.


Change the execution time limit to 45000 µs (45 ms) to redefine the wake/sleep interval:
Execution Time Limit (µs): 45000
Forced Execution Sleep (µs): 5000
Max Execution Load (%CPU): 90
Wake interval = 45 ms
Sleep interva = 5 ms
Execution Time Limit = 45 ms (45000 µs)
Because the controller sleeps for more than 5 ms, the execution monitor does not force a sleep interval.
OB35
The start event for OB35 is not affected by the execution monitor
OB1
Sleep (10 ms)
Timing (ms)
Execution Time
Last: 20
Avg: 20
Min: 20
Max: 20
SleepTime
Last: 10
Execution Time = 20 ms
67%
Sleep Time = 10 ms
33%
Min Cycle Time
Min Sleep Time (ms): 10
Min Cycle Time (ms): 0
Scan Cycle Time = 30 ms

Connecting the controller to the SIMATIC NET OPC Server 9

9.1 Connecting the Controller to the SIMATIC NET OPC Server

WinAC RTX can read and write data via the network using a SIMATIC NET OPC server. You can use the following tools to configure the OPC connection:

- OPC Scout for configuring the connection to the SIMATIC NET OPC server
- STEP 7 (HW Config and NetPro) for configuring the WinAC RTX controller
- Station Configuration Editor for configuring the PC station

SIMATIC NET must be installed in order to configure the OPC Server connection.

Note

The critical step most frequently overlooked is Step 3: Configuring the S7 connection for the OPC server (Page 221) in NetPro. After adding the connection for the OPC server, you must set the "S7 connection" connection type and enter a local ID for the connection.

Task summary

Step 1 (Page 216): Station Configuration Editor (SIMATIC NET)
Add the OPC server to the PC station

Step 2 (Page 218): HW Config (STEP 7)
Add the OPC server to the STEP 7 HW Config

Step 3 (Page 221): NetPro (STEP 7)
Add S7 connection for the OPC server for configuring WinLC RTX

Step 4 (Page 224): SIMATIC Manager (STEP 7)
Load the configuration into the WinLC controller

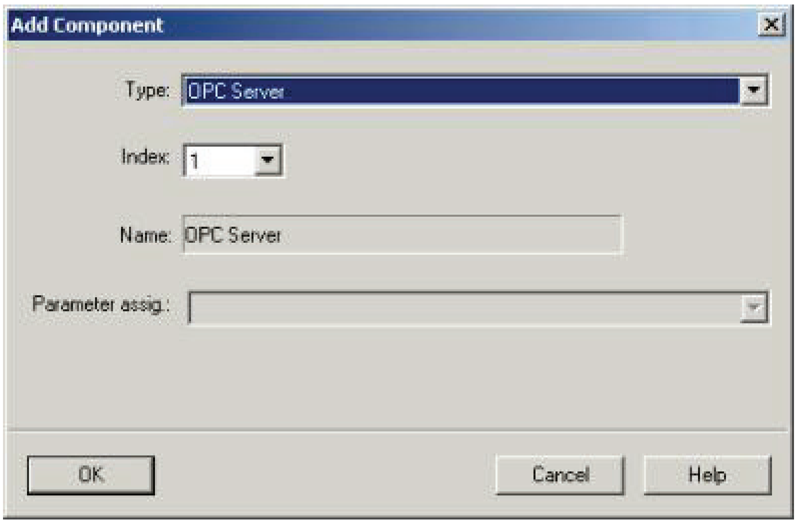
Step 5 (Page 225): OPC Scout (SIMATIC NET)
Connect the controller to the OPC server

9.2 Step 1: Add the OPC Server to the PC Station

Tool: Station Configuration Editor (SIMATIC NET)

To configure the OPC server in the PC station, follow these steps:

1. Open the Station Configuration Editor and select an index.
2. Right-click to display the "Add" button. Click the "Add" button. This opens the "Add Component" dialog box.
3. Select "OPC server" from the drop-down list of component types:



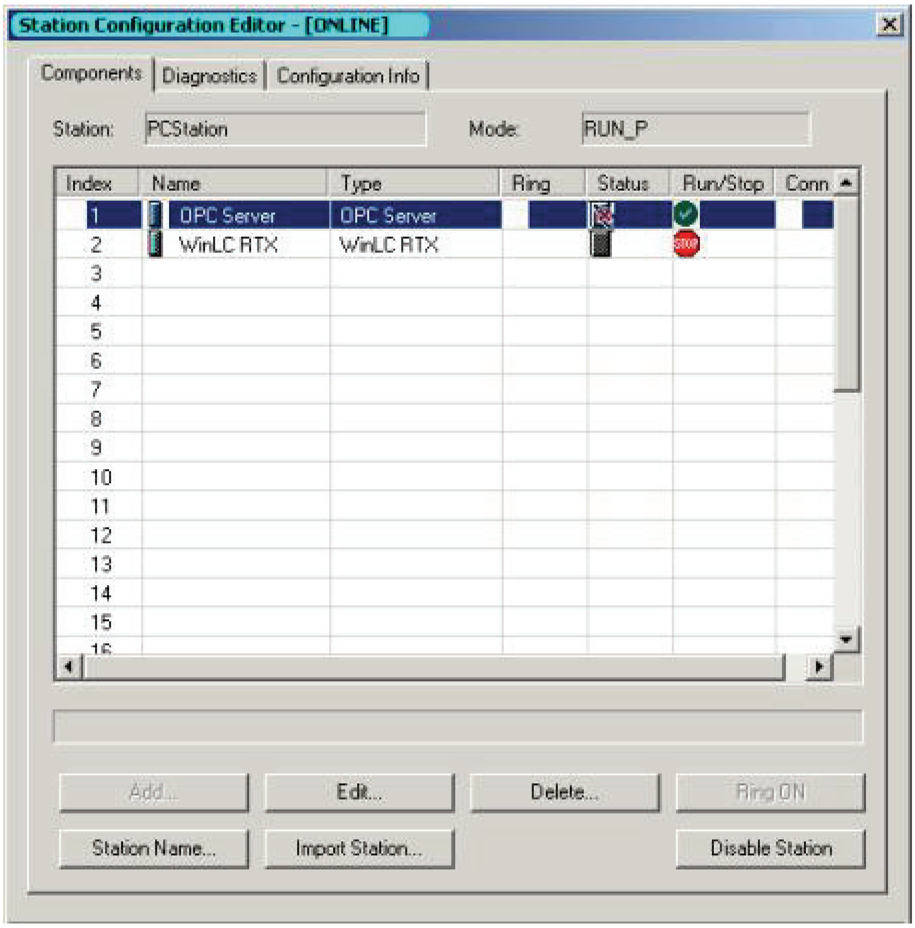
Type: OPC Server

Index: 1

Name: OPC Server

Parameter assig.:

4. Click "OK" to add the OPC server to the station configuration. The Station Configuration Editor displays the OPC server in the index selected.
   (For this example, the OPC server is configured for Index 1.)
5. Click OK to save the PC station configuration and to close the Station Configuration Editor.



Station: PCStation

Mode: RUN_P

Index	Name	Type	Ring	Status	Run/Stop	Conn
1	OPC Server	OPC Server			✓	
2	WinLC RTX	WinLC RTX			STOP	
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

9.3 Step 2: Add the OPC Server to the Hardware Configuration

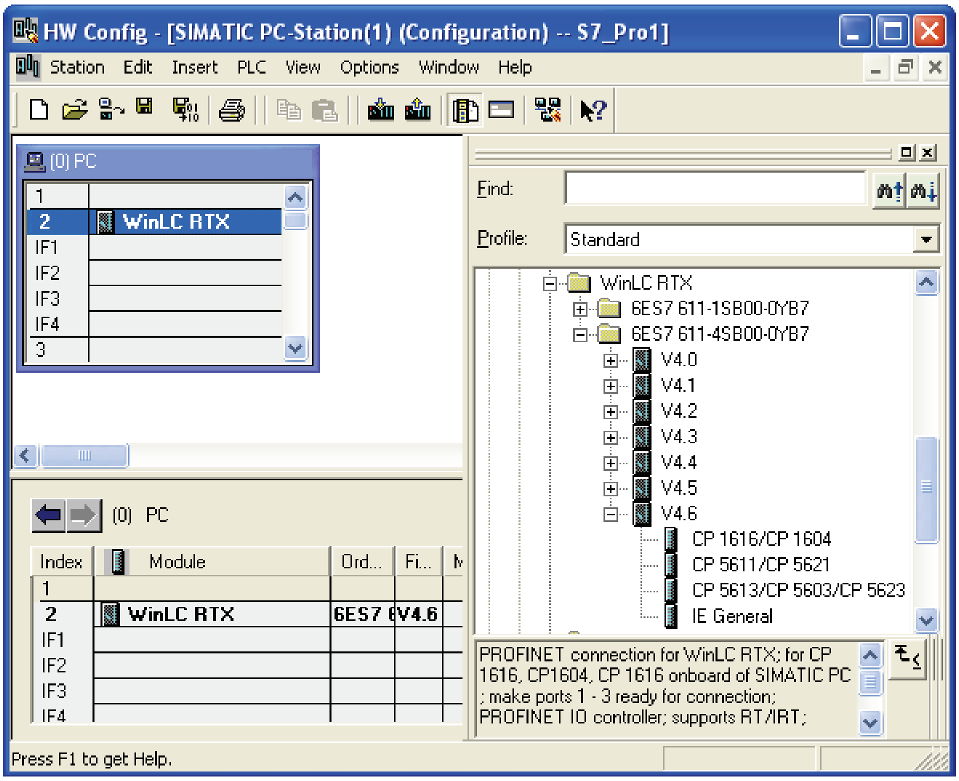
Tool: HW Config (STEP 7)

Task summary

- Create a STEP 7 project for a PC station.
- Insert the OPC server into the hardware configuration.
- Configure the OPC server.

Creating the STEP 7 project

1. Open STEP 7 and create a project (for example, OPCProject).
2. Insert a SIMATIC PC station with the same name as entered in the Station Configuration Editor.
3. Double-click the Configuration icon for the PC station to open the STEP 7 HW Config utility.
4. Insert the WinAC RTX controller in the same index as configured in the Station Configuration Editor (Page 216).


HW Config - [SIMATIC PC-Station(1) (Configuration) -- S7_Pro1]
Station Edit Insert PLC View Options Window Help
(0) PC
Find:
Profile:
(0) PC

1	
2	WinLC RTX
IF1	
IF2	
IF3	
IF4	
3	

Profile: Standard

```
WinLC RTX
  6ES7 611-1SB00-0YB7
  6ES7 611-4SB00-0YB7
    V4.0
    V4.1
    V4.2
    V4.3
    V4.4
    V4.5
    V4.6
      CP 1616/CP 1604
      CP 5611/CP 5621
      CP 5613/CP 5603/CP 5623
      IE General
```

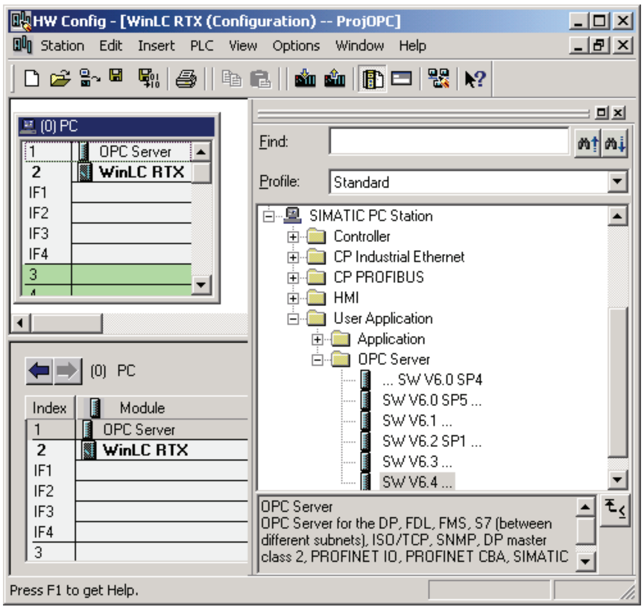
Index	Module	Ord...	Fi...	M
1				
2	WinLC RTX	6ES7 6	V4.6	
IF1				
IF2				
IF3				
IF4				

PROFINET connection for WinLC RTX; for CP 1616, CP1604, CP 1616 onboard of SIMATIC PC ; make ports 1 - 3 ready for connection; PROFINET IO controller; supports RT/IRT;

Press F1 to get Help.

Adding the OPC server to the hardware configuration

1. Expand the "User Application" folder in the hardware catalog.
2. Expand the "OPC Server" folder and select the following component: SW V6.4
3. Use a drag-and-drop operation to move the "SW V6.4" component to the same index as configured in the Station Configuration Editor (Page 216). (For this example, the OPC server is configured for Index 1.)



(0) PC

1	OPC Server
2	WinLC RTX
IF1	
IF2	
IF3	
IF4	
3	

(0) PC

Index	Module
1	OPC Server
2	WinLC RTX
IF1	
IF2	
IF3	
IF4	
3	

Find:

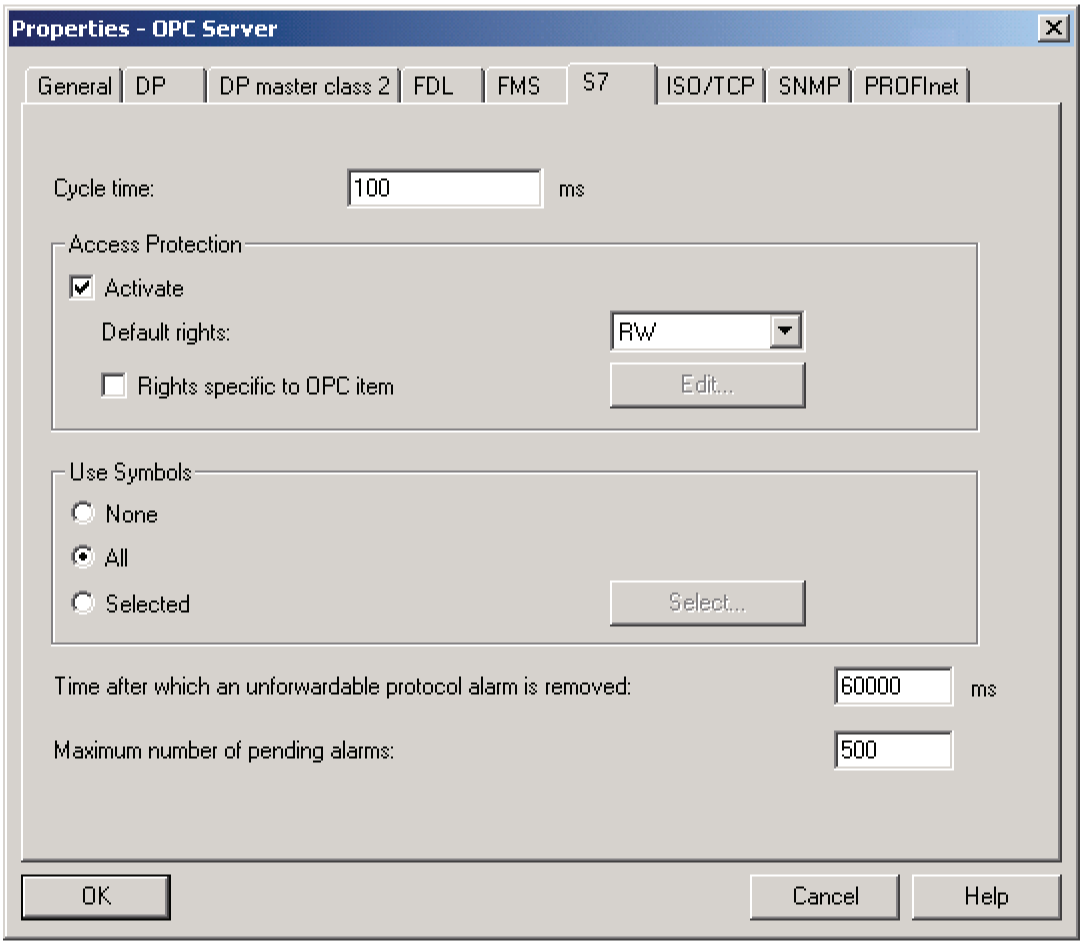
Profile: Standard

OPC Server
OPC Server for the DP, FDL, FMS, S7 (between different subnets), ISO/TCP, SNMP, DP master class 2, PROFINET IO, PROFINET CBA, SIMATIC

Press F1 to get Help.

Configuring the OPC server

1. Double-click the OPC Server entry (Index 1) to open the "Properties" dialog.
2. Open the "S7" tab and select the "Activate" option (under "Access Protection").
3. To use the STEP 7 symbols (Page 225) for accessing controller data from the OPC server, select the option "All" (or "Selected", to specify specific entries in the symbol table) in the "Use Symbols" area.
4. Click “OK” to close the “Properties” dialog.
5. Click the "Save and Compile" button to create the hardware configuration for the PC station.



Properties - OPC Server

Cycle time: 100 ms

Access Protection

☑ Activate

Default rights: RW

☐ Rights specific to OPC item

Use Symbols

○ None

● All

○ Selected

Time after which an unforwardable protocol alarm is removed: 60000 ms

Maximum number of pending alarms: 500

Once you have compiled the configuration in the STEP 7 project, you can close HW Config and return to SIMATIC Manager.

9.4 Step 3: Add an S7 Connection for the OPC Server in NetPro

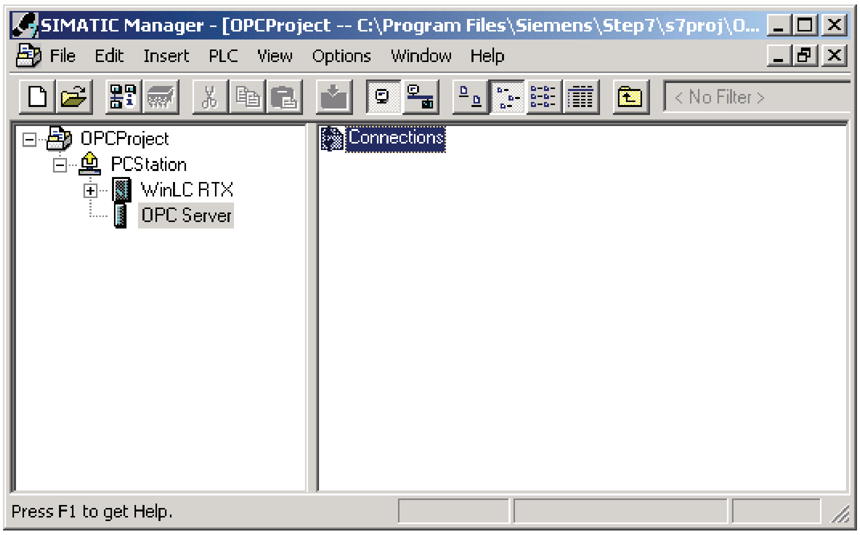
Tool: NetPro (STEP 7)

Task summary

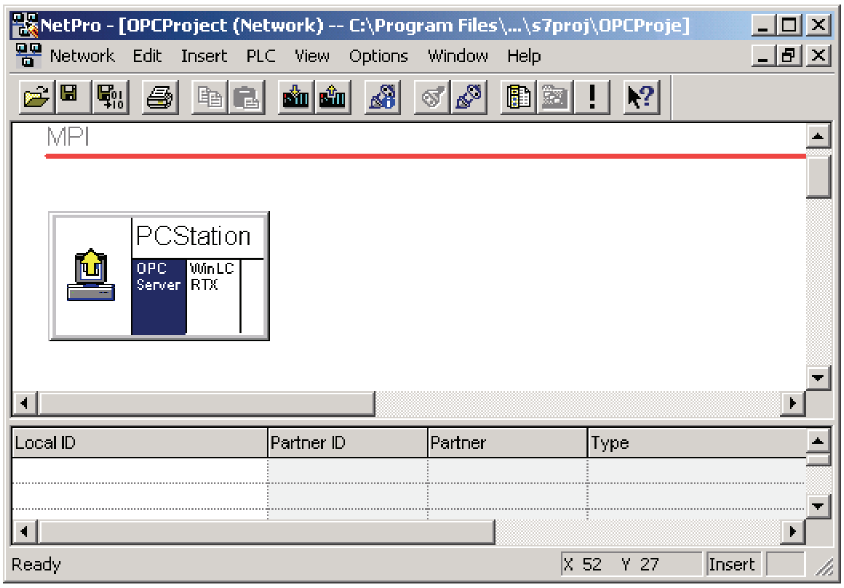
- Configuring an S7 connection for the OPC server in the PC station configuration.
- Assigning a Local ID for the OPC server connection

Configuring an OPC server connection in NetPro

1. Search for the OPC server in the SIMATIC Manager and double-click the "Connections" icon to open NetPro.

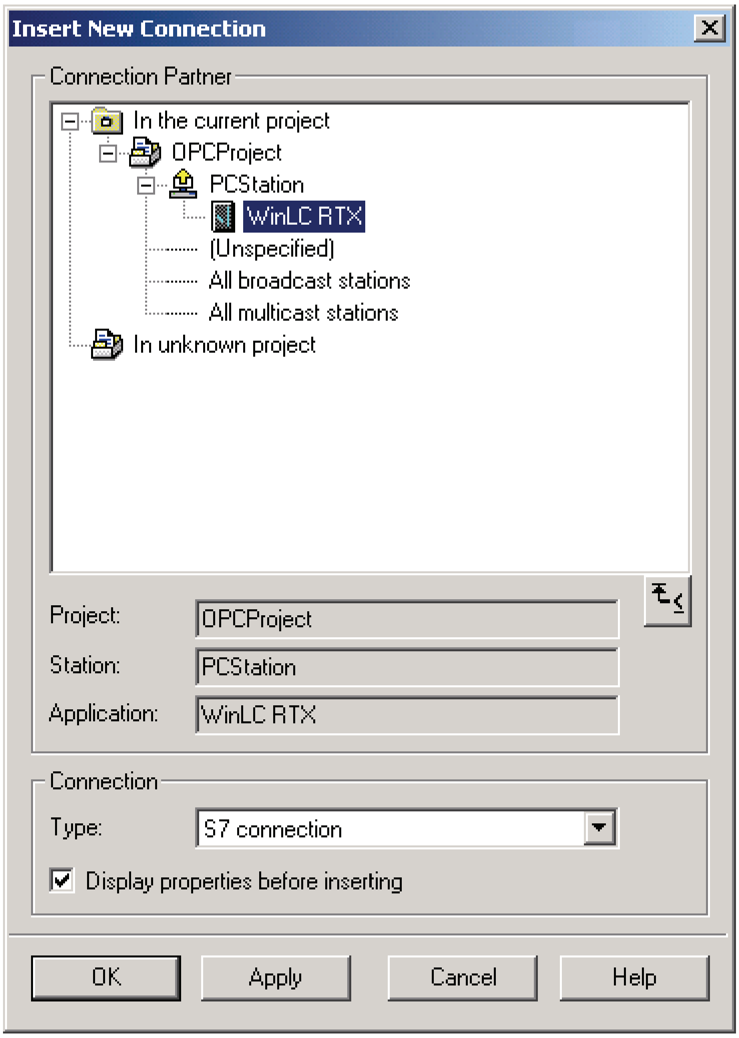

SIMATIC Manager - [OPCProject -- C:\Program Files\Siemens\Step7\s7proj\O...
File Edit Insert PLC View Options Window Help
< No Filter >
OPCProject
PCStation
WinLC RTX
OPC Server
Connections
Press F1 to get Help.

2. Select the OPC server in the PC station.


NetPro - [OPCProject (Network) -- C:\Program Files\...\s7proj\OPCProje]
Network Edit Insert PLC View Options Window Help
MPI
PCStation
OPC Server
WinLC RTX
Ready
X 52 Y 27
Insert

Local ID	Partner ID	Partner	Type

3. Right-click the OPC server to display the context menu. Select the **Add new connection** menu command to open the "Add new connection" dialog box.



Project: OPCProject

Station: PCStation

Application: WinLC RTX

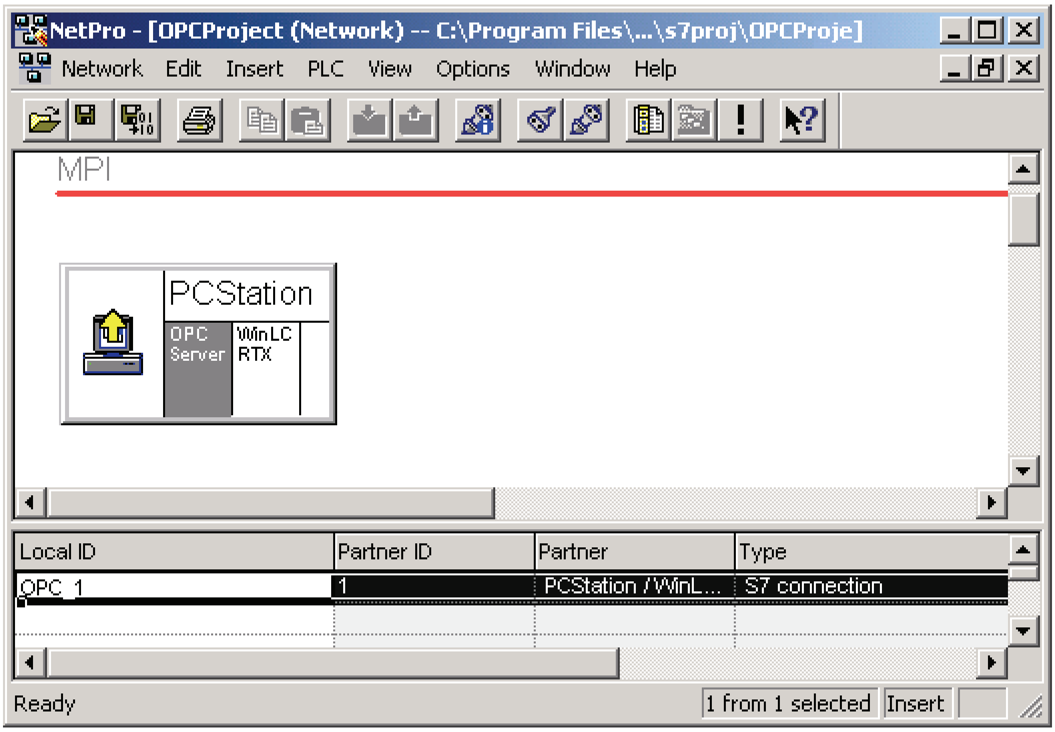
Type: S7 connection

☑ Display properties before inserting

4. Set the connection type to "S7 connection" and click "OK" to add the S7 connection for the OPC server. The "Properties" dialog box for the S7 connection opens automatically.

Assigning a Local ID for the OPC server connection

1. In the "Properties" dialog, enter the Local ID for the S7 connection (such as OPC_1).
2. Click "OK" to add the S7 connection to NetPro.
3. Click the "Save and Compile" button to save and compile your changes into the STEP 7 project.



Local ID	Partner ID	Partner	Type
OPC_1	1	PCStation / WinL...	S7 connection

Once you have compiled the S7 connection for the OPC server in the STEP 7 project, you can close NetPro and return to the SIMATIC Manager.

9.5 Step 4: Download the Configuration to the Controller

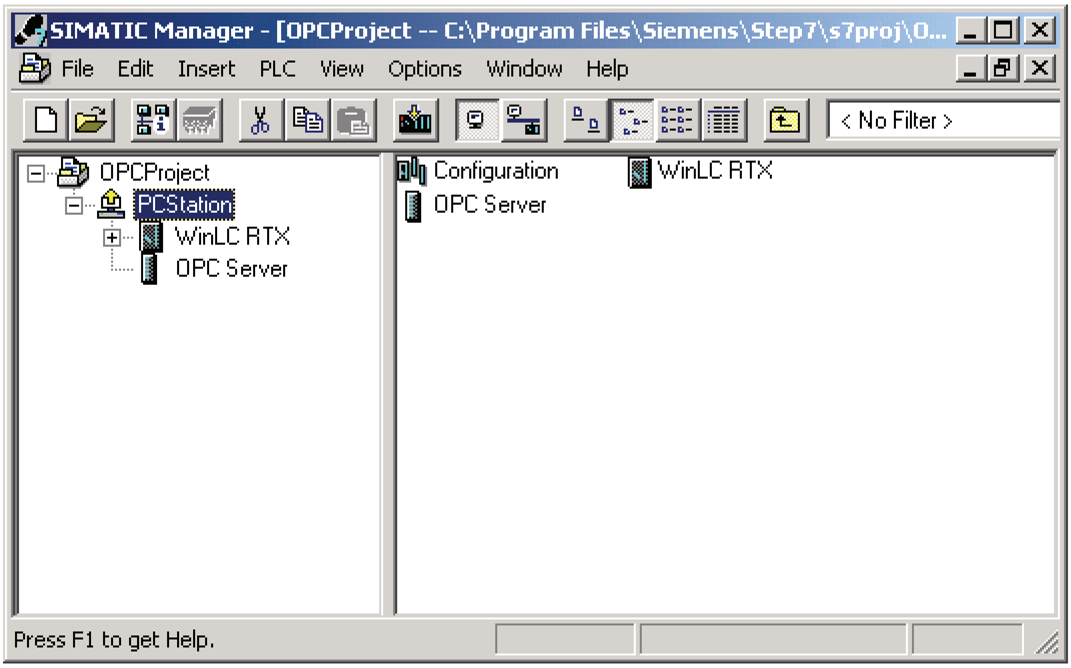
Tool: SIMATIC Manager (STEP 7)

Note

The controller must be running in order to download the configuration from STEP 7.

To download the configuration, follow the steps described below:

1. If the controller is not running, start the controller.
2. In SIMATIC Manager, select the symbol of the PC station.
3. Select the **Target system > Load** menu command or click the "Load" button in the toolbar.



SIMATIC Manager - [OPCProject -- C:\Program Files\Siemens\Step7\s7proj\O...
File Edit Insert PLC View Options Window Help
< No Filter >
OPCProject
PCStation
WinLC RTX
OPC Server
Configuration
WinLC RTX
OPC Server
Press F1 to get Help.

9.6 Step 5: Connect the Controller to the OPC server

Tool: OPC Scout

Task summary

- Adding a connection in an OPC project for the SIMATIC NET OPC server.
- Define the objects to be accessed using the OPC server.

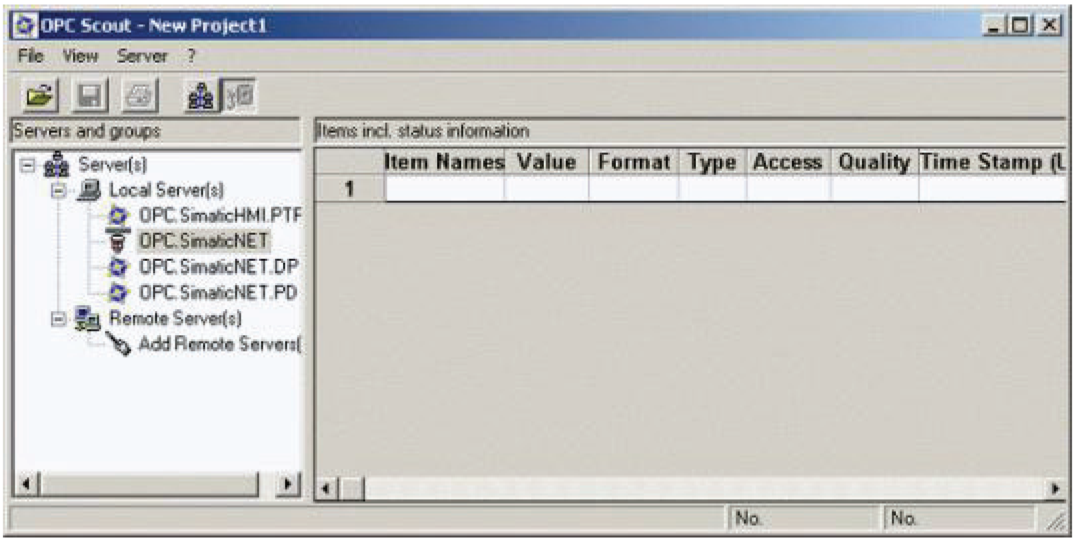
Creating an OPC project

Select the **Start > SIMATIC > SIMATIC NET > OPC SCOUT** menu command to create a new project in the OPC Scout.

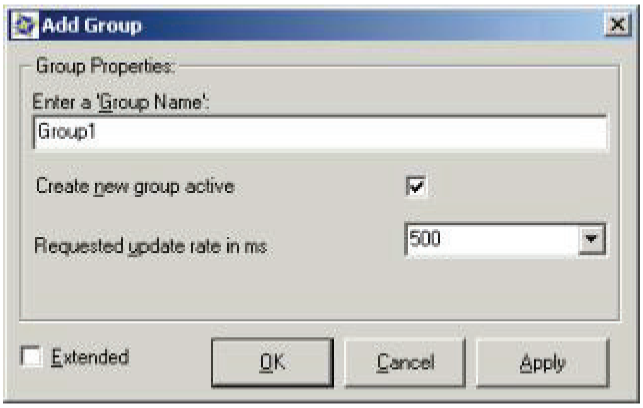
Adding a connection (group) for the OPC server at an OPC project

To add a connection to the SIMATIC NET OPC Server, follow these steps:

1. Expand the "Local Server(s)" directory in the "Server and Groups" for the project.
2. Double-click the "OPC.SimaticNet" element to add a connection (or group) for the SIMATIC NET OPC server.



3. Enter the group name for the connection in the "Add group" dialog box (e.g. Group1).



Add Group

Group Properties:

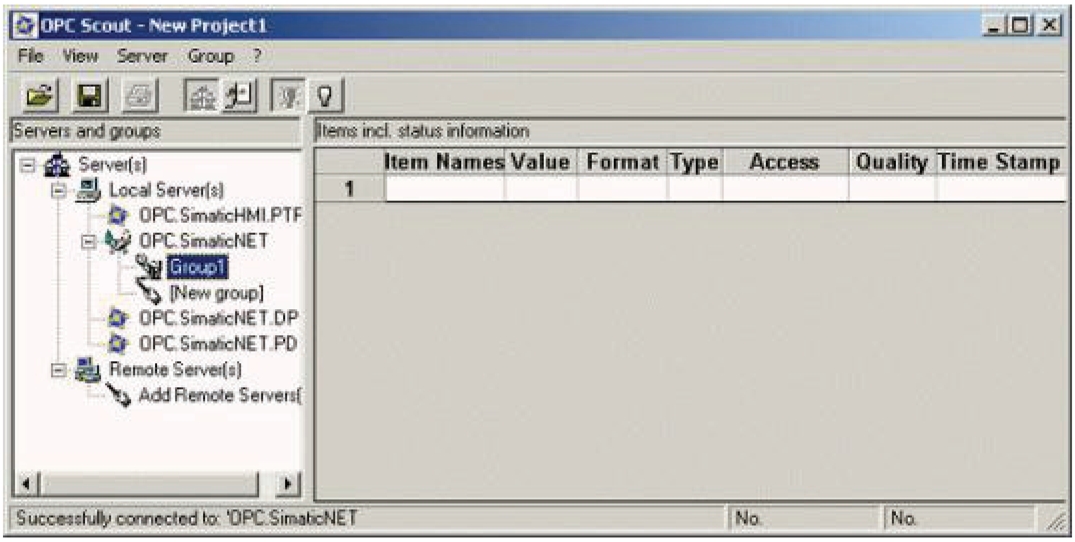
Enter a 'Group Name': Group1

Create new group active ☑

Requested update rate in ms 500

□ Extended

4. Click "OK" to add the group to the OPC server. OPC Scout adds the connection (Group1) to the OPC server.



OPC Scout - New Project1

Servers and groups

- Server(s)
  - Local Server(s)
    - OPC.SimaticHMI.PTF
    - OPC.SimaticNET
      - Group1
      - [New group]
    - OPC.SimaticNET.DP
    - OPC.SimaticNET.PD
  - Remote Server(s)
    - Add Remote Servers(

Items incl. status information

	Item Names	Value	Format	Type	Access	Quality	Time Stamp
1							

Successfully connected to: 'OPC.SimaticNET | No. | No.

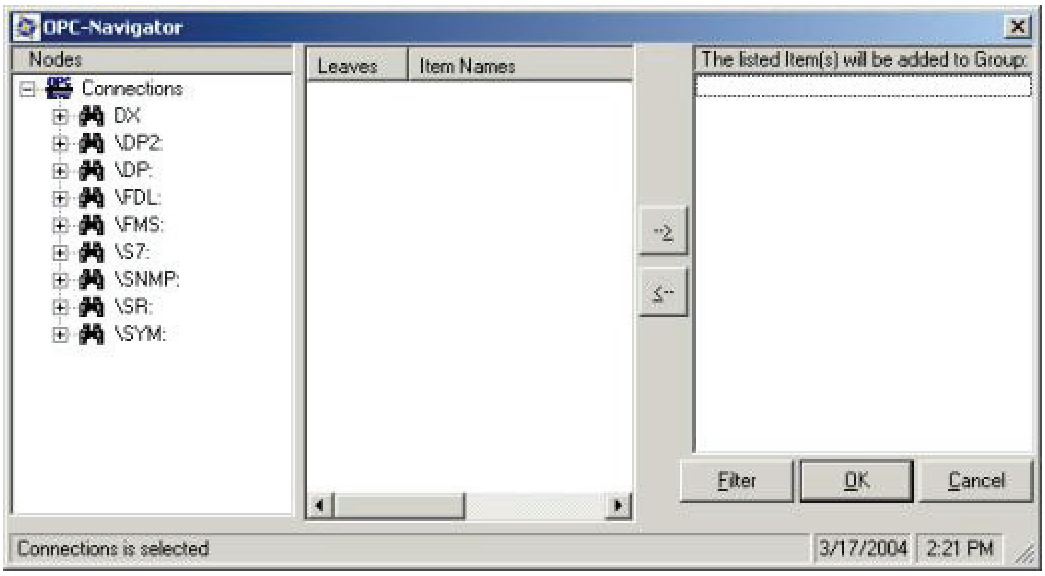
Configuring the items to be accessed (using absolute addressing)

Note

This procedure describes how you use the absolute addressing for the configuration of the OPC server. You can also use the STEP 7 symbol table to connect the OPC server. This procedure is described in "Configuring the Items to be Accessed
(Using the STEP 7 Symbol Table)".

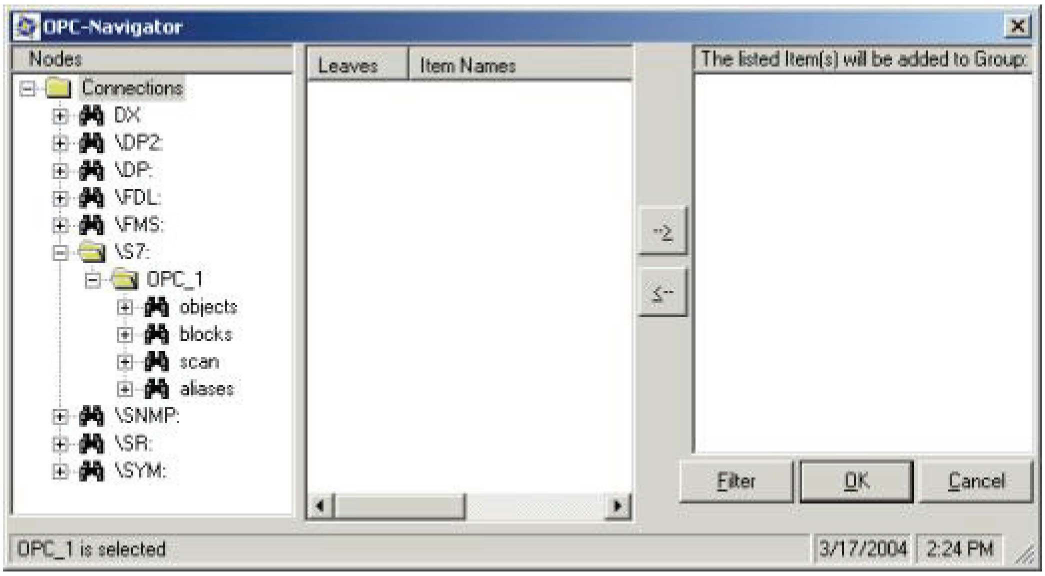
Use the following procedure to configure the OPC server to use an absolute address for accessing data in the controller:

1. Open the OPC Navigator by double-clicking the connection (Group1) for the OPC server.



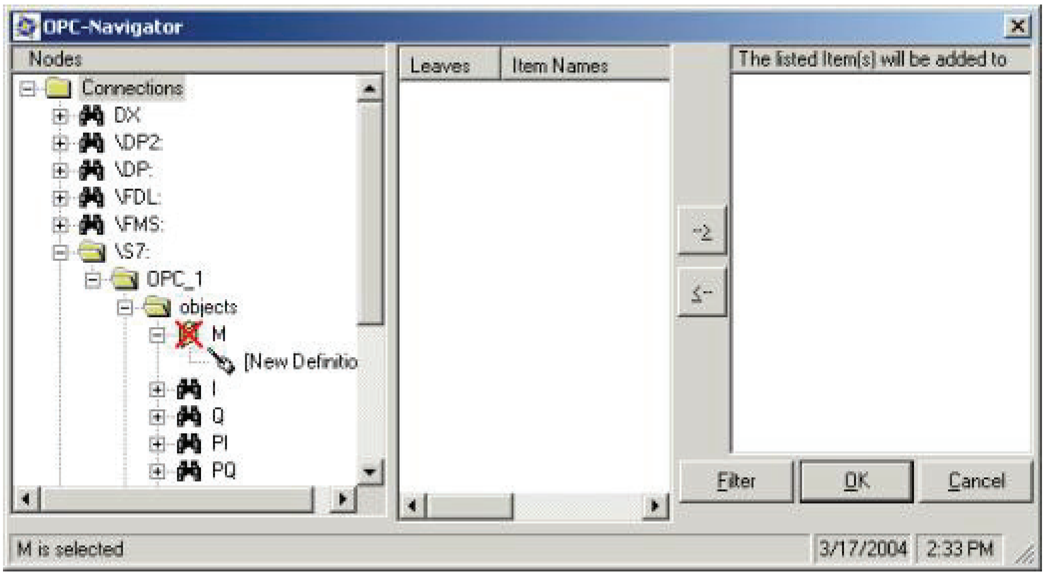
Connections is selected | 3/17/2004 | 2:21 PM

2. To add an item to be accessed, expand the \S7: folder and select OPC_1.



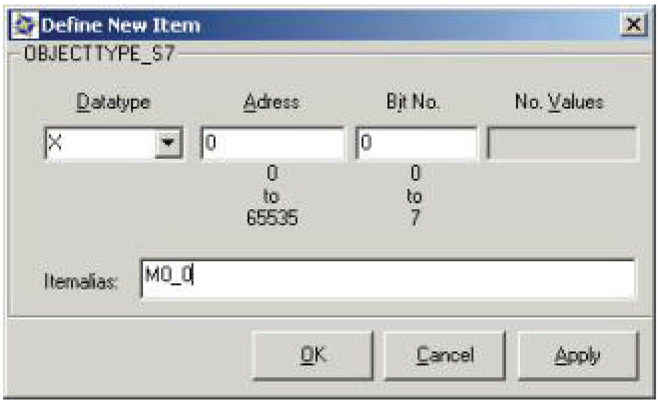
OPC_1 is selected | 3/17/2004 | 2:24 PM

3. To configure access to M0.0, expand the "Objects" folder and expand the "M" folder (for the bit memory area).



4. Double-click the "New Definition" icon to open the "Define New Item" dialog.

5. To define a connection for M0.0, select X (for bit) in the "Data type" pulldown listbox and enter the byte address (0) and the bit number (0). (You can also enter an alias for the item.)



Define New Item

OBJECTTYPE_S7

Datatype: X

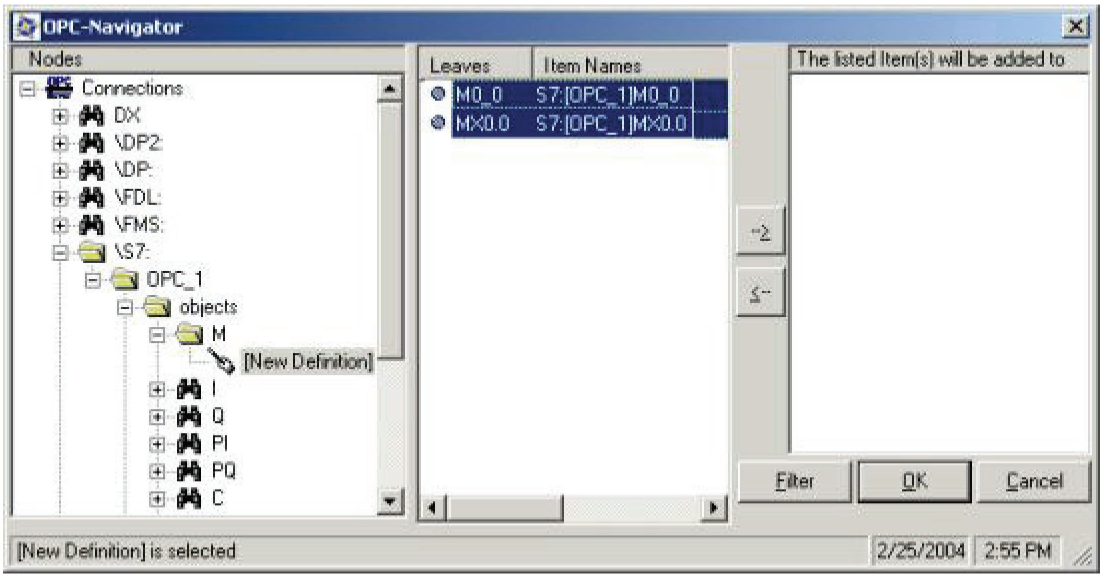
Adress: 0 (0 to 65535)

Bit No.: 0 (0 to 7)

No. Values:

Itemalias: M0_0

6. Click "OK" to define an object for M0.0.



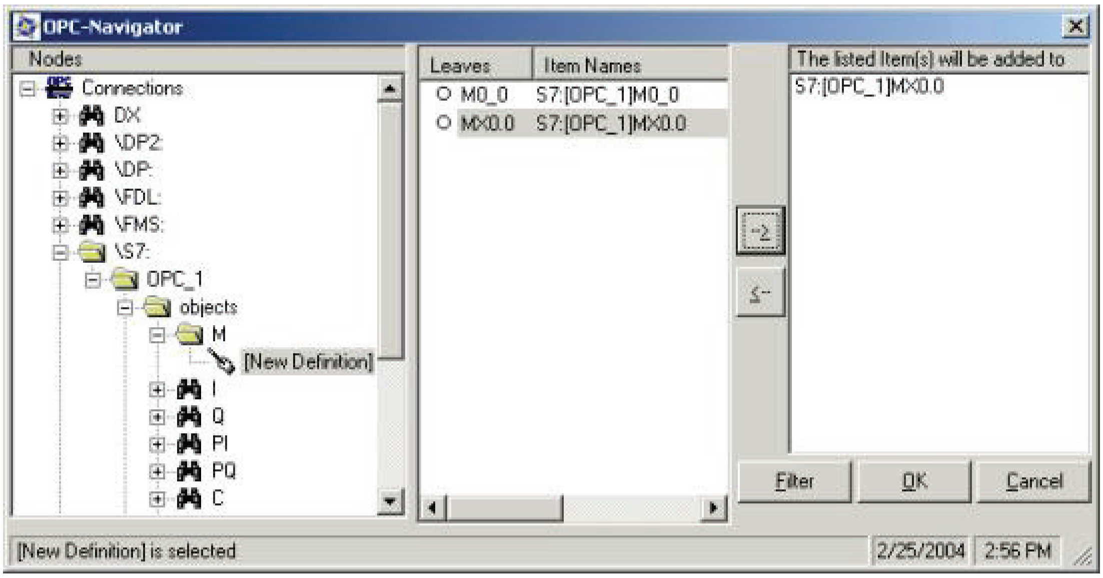
Leaves	Item Names
M0_0	S7:[OPC_1]M0_0
MX0.0	S7:[OPC_1]MX0.0

The listed Item(s) will be added to

[New Definition] is selected 2/25/2004 2:55 PM

7. Select the MX0.0 entry and click the Add arrow (-->) to enter the following syntax that defines a connection for MX0.0: S7:[OPC_1]MX0.0

8. Select the entry (S7:[OPC_1]MX0.0) and click OK to add the connection for MX0.0 to Group1.



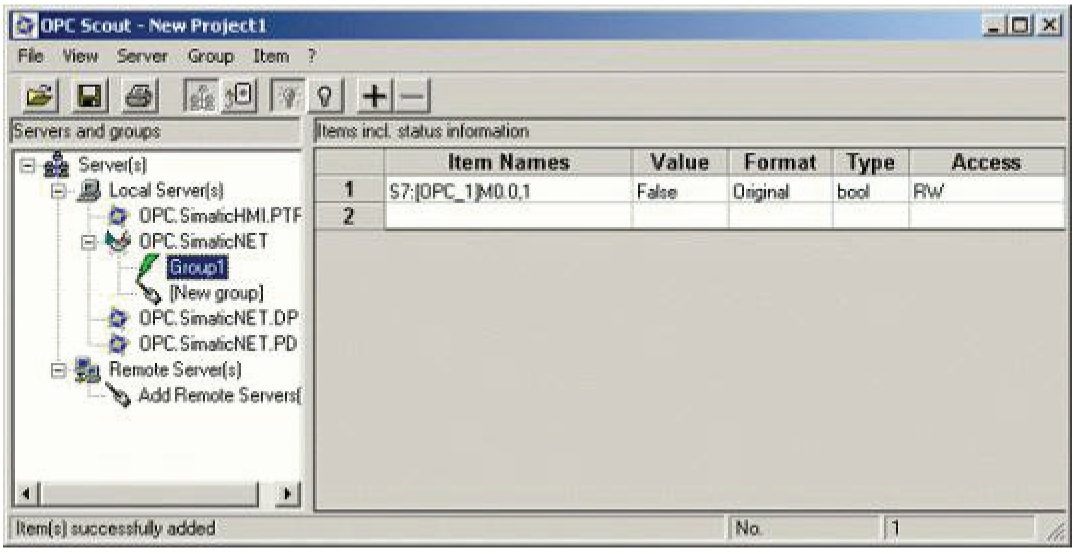
Leaves	Item Names
M0_0	S7:[OPC_1]M0_0
MX0.0	S7:[OPC_1]MX0.0

The listed Item(s) will be added to
S7:[OPC_1]MX0.0

[New Definition] is selected 2/25/2004 2:56 PM

Result

After adding the item to Group1, OPC Scout displays the name and other parameters for the item. You can now use any of the methods supported by SIMATIC NET OPC Server.



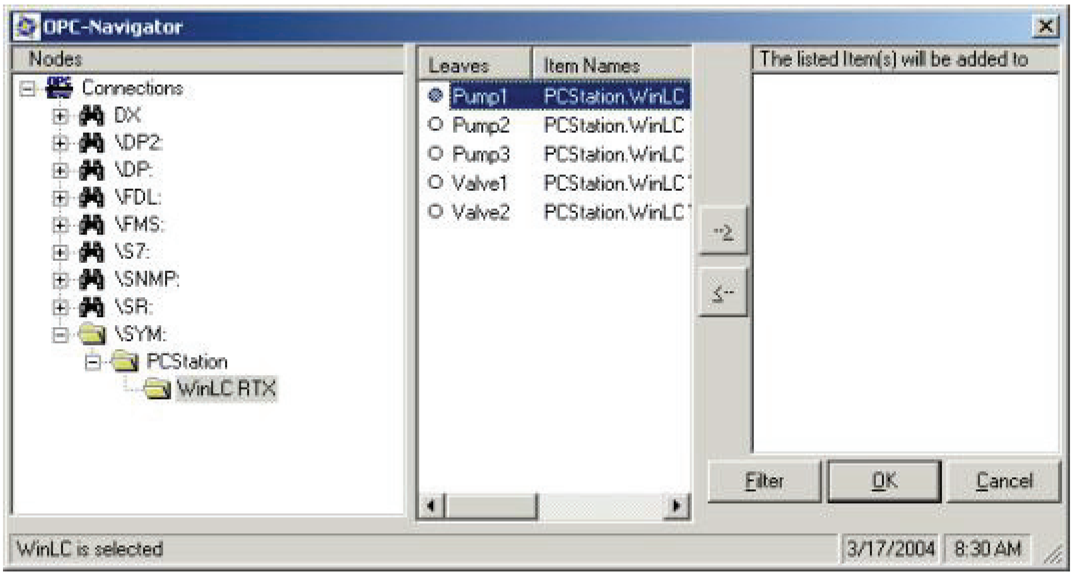
	Item Names	Value	Format	Type	Access
1	S7:[OPC_1]M0.0,1	False	Original	bool	RW
2					

Item(s) successfully added | No. | 1

Configuring the items to be accessed (using the STEP 7 symbol table)

If you have created an icon table for the STEP 7 program that you have loaded into the target system, you can use the icons to connect the OPC server with the data in the controller. Proceed as follows to configure the items to be accessed by means of the STEP 7 symbol table:

1. Open the OPC Navigator by double-clicking the connection (Group1) for the OPC server.
2. Search the controller folder to display the symbols that have been loaded into the controller.

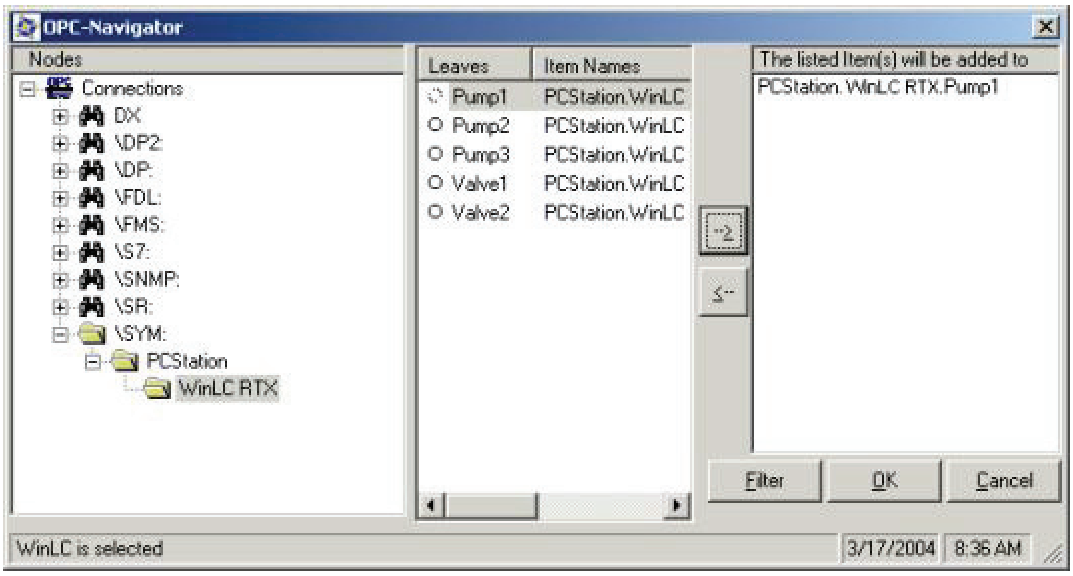


Leaves	Item Names
● Pump1	PCStation.WinLC
○ Pump2	PCStation.WinLC
○ Pump3	PCStation.WinLC
○ Valve1	PCStation.WinLC
○ Valve2	PCStation.WinLC

The listed Item(s) will be added to

WinLC is selected | 3/17/2004 | 8:30 AM

3. After selecting the symbols for the data to be connected to the OPC server, click the "Add" (-->) button.



Leaves	Item Names
○ Pump1	PCStation.WinLC
○ Pump2	PCStation.WinLC
○ Pump3	PCStation.WinLC
○ Valve1	PCStation.WinLC
○ Valve2	PCStation.WinLC

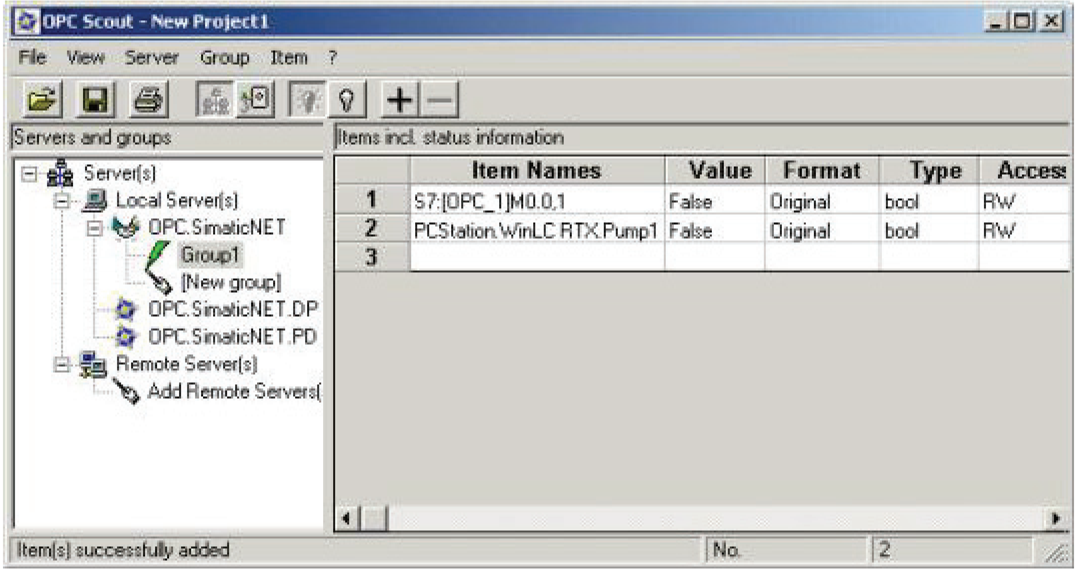
The listed Item(s) will be added to
PCStation.WinLC RTX.Pump1

WinLC is selected | 3/17/2004 | 8:36 AM

4. Click the "OK" button to add the symbol to Group1.

Result

After the object has been added to the group, OPC Scout shows the symbol name and other parameters for the STEP 7 symbol.



	Item Names	Value	Format	Type	Acces
1	S7:[OPC_1]M0.0,1	False	Original	bool	RW
2	PCStation.WinLC RTX.Pump1	False	Original	bool	RW
3					

Item(s) successfully added | No. | 2

Operating WinAC RTX with S7-modular Embedded Controller 10

10.1 Overview

Introduction

S7-modular Embedded Controller (S7-mEC) is a PC with S7-300 design that runs on a preinstalled Windows XP Embedded operating system.

S7-mEC can be mounted in horizontal position onto a mounting rail. It can be expanded into a modular S7-mEC system by installing S7-300 modules and expansion modules.
A flash memory is used to backup the data.

S7-mEC is an open platform for user specific applications and provides access to S7-300 I/O.

EC31-RTX

EC31-RTX is the controller module within the S7-mEC, pre-installed with the operating system Windows Embedded Standard 2009, WinAC RTX 2010 and SIMATIC NET 2008 SP2 (with SIMATIC NET Softnet-S7 Lean V7.1).

Compared to WinAC RTX on other platforms, there are certain differences in the handling of EC31 with WinAC RTX:

- Layout and appearance of the controller panel
- Operation
- Hardware configuration

Note

New System status lists

The system status lists have been enhanced by S7-mEC functionality.

A listing of all system status lists is available in Chapter "System Status List (SSL) (Page 266)".

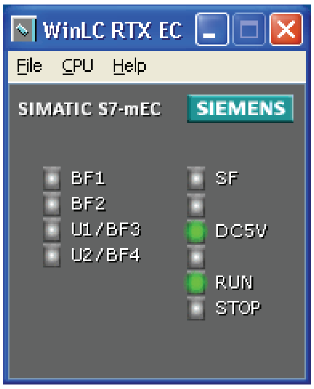
Reference

Additional information about WinAC RTX within S7-mEC in contained in the manual S7-modular Embedded Controller EC31-HMI/RTX (http://support.automation.siemens.com/WW/view/en/28593853/133300).

10.2 Controller Panel

Status and error displays of the panel

The picture below shows the WinAC RTX panel when used within S7-mEC:



The panel of the Embedded Controller is equipped with the following LEDs:

LED designation	Color	Meaning
BF1	red	Flashes on bus error at interface X1 PN LAN P1 or X1 PN LAN P2
BF2	red	Flashes on bus error at submodule on slot IF2
U1/BF3	red	Flashes on bus error at submodule on slot IF3
U2/BF4	red	Flashes on bus error at submodule on slot IF4
SF	red	Group errors
5 VDC	green	5 V supply for the backplane bus (is switched off after you shut down the Controller by means of "Shutdown").
RUN	green	WinLC on the Embedded Controller is in RUN.
STOP	Yellow	WinLC on the Embedded Controller is in STOP.

LEDs BF1 to BF4 correspond to the IF slots of WinLC RTX. These are preconfigured in the component configurator.

Notes on operation

Note

The RUN and STOP mode selector positions show the selected operating mode. The RUN and STOP indicators show the actual operating mode of the EC31-RTX.

You cannot change the status of the controller by clicking the status indicators.

When the RUN LED flashes and the STOP LED is lit, this signals a transition from STOP to RUN mode. The STOP LED goes dark to indicate that the transition to RUN mode is complete.

10.3 Operation

Mode selector switch

You cannot change the operating state using the panel. You must use the mode selector on the module to switch the EC31-RTX/EC31-HMI/RTX to RUN, STOP, or MRES mode.

Memory reset (MRES)

You can reset the memory of the EC31-RTX/EC31-HMI/RTX (same as a CPU S7-300) as follows:

- With the mode selector (see *S7-modular Embedded Controller EC31-HMI/RTX* operating instructions, Chapter "Operating EC31-RTX/EC31-HMI/RTX")
- Via STEP 7 (see STEP 7 help, "Memory reset of CPUs/FMs")

Data memory

- Flash memory

  The EC31 is equipped with a flash memory that can be used for storing data and to archive configuration data. This memory is split up into two default partitions.
- NVRAM

  The EC31 is equipped with an NVRAM area. This 512 KB memory is located on a separate chip.

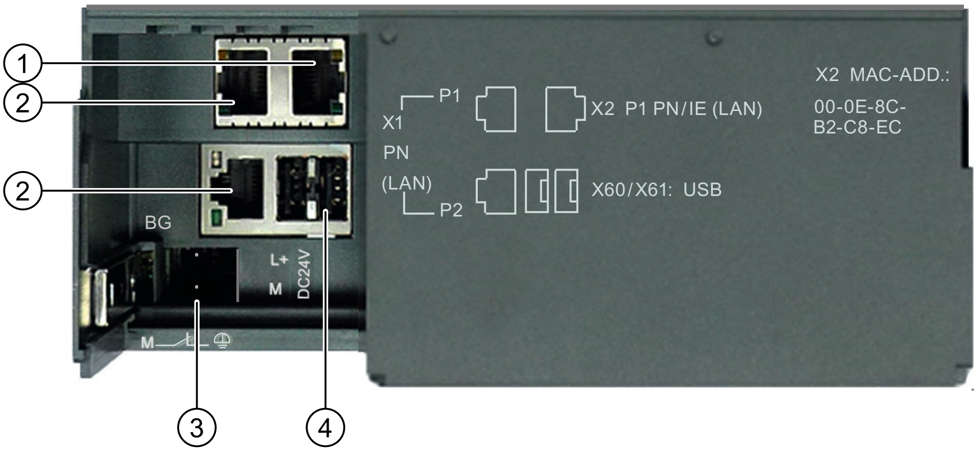
10.4 Hardware configuration

Hardware configuration

For use of WinAC RTX on the Embedded Controller, centralized I/O can also be connected. The connection is implemented by means of backplane bus similar to the setup for an S7-300 CPU.

Interfaces of the EC31-RTX

The picture below shows the interfaces and connections of the EC31-RTX.



Number	Element
1	Interface X2 PN (LAN) for Ethernet
2	Interface X1 PN (LAN) P1/P2 for PROFINET (as WinAC submodule)
3	Power supply connection
4	USB 2.0 ports

EC31-RTX comes with two Industrial Ethernet interfaces:

- X1 PN (LAN) P1/X1 PN LAN P2

  The interface is used as PROFINET interface.

- X2 PN (LAN)

  The interface is assigned to the PC station in index 3 at IE_General and is preconfigured for Industrial Ethernet communication. It supports PROFINET basic communication.

Communication

Ethernet/PROFINET: The communication interface X1 PN (LAN) is already preconfigured (ERTEC 400 on Slot IF1 as a submodule of WinLC RTX).

PROFIBUS DP: You have the option to use the CP 5603 communication processor in the EM PCI-104 expansion module. The CP 5603 can be found in the hardware catalog under "SIMATIC PC Station > Controller > S7-modular Embedded Controller > EC31 ...". You can insert the CP 5603 into one of the free slots IF2, IF3, or IF4 of the PC station.

MPI: WinAC RTX within S7-mEC does not support the MPI communication interface.

For further information, refer to Chapter "Pre-configured system" in the S7-modular Embedded Controller EC31-HMI/RTX (http://support.automation.siemens.com/WW/view/en/28593853/133300) operating instructions.

Archiving

Configuration data can be archived in *.wld files for reuse and transfer. For more information, read the section Archiving and restoring STEP 7 user programs (Page 85).

NOTICE
The wld files of WinAC RTX and WinAC RTX within S7-mEC are incompatible.

WinAC RTX F 11

11.1 Introduction

Importance of this section

This section contains important information about WinAC RTX F 2010, version V4.6 (menu command **Start > SIMATIC > Product Info > Installed Software**), order number 6ES7671-1RC08-0YA0 (full version) and 6ES7671-1RC08-0YE0 (upgrade). It is to be considered as a separate component with overriding importance if there is doubt in connection with other information provided in manuals and catalogues.

Validity

This section is valid for the software package Windows Automation Center Failsafe with real-time extensions (WinAC RTX F 2010, version V4.6), which includes the following components:

- Windows Logic Controller RTX F (WinLC RTX F V4.6)
- IntervalZero RTX 2009
- WinAC Time Synchronization V4.2
- Automation License Manager V5.0 SP1
- SIMATIC NET V7.1 SP2 and SIMATIC NET V8.0 including license for Softnet S7 Lean V8.0

In this section we will provide you with a description of the specifications of WinAC RTX F 2010 as compared to WinAC RTX 2010 with the order number 6ES7671-0RC08-0YA0.

WinAC RTX F 2010 software package

WinAC RTX F 2010 is an F-capable software PLC, that runs on a standard computer (PC) with real-time expansions. WinAC RTX F provides the same functionality as WinAC RTX and as a F-CPU (F-capable CPU).

WinAC RTX F is approved for use in F-systems S7 Distributed Safety. This means that a safety program created with S7 Distributed Safety can be executed on the PC with WinAC RTX F.

Additional required documentation

You require the following documentation when using WinAC RTX F 2010. The current product information is only a supplement to these manuals.

- S7 Distributed Safety, configuring and programming (http://support.automation.siemens.com/WW/view/en/22099875) Programming and Operating Manual, 10/2007 edition

Path information for Windows 7

The path specifications relevant for operating systems mentioned in this section refer to Windows XP. This section lists the most important path information also for Windows 7.

- **Checking the WinAC RTX F 2010 software version**:

  Start > All Programs > Siemens Automation > SIMATIC > Product Info > Installed Software
- **Readme file**:

  Start > All Programs > Siemens Automation > SIMATIC > Product Info > German
- **Assignment of IRQ 8 to the real-time clock**:

  Start > Control Panel > System and Safety > System > Device Manager
- **Procedure for testing the F-capability of the hardware**:

  Start > All Programs > Siemens Automation > SIMATIC > PC Based Control > Evaluation Tool

Reading the Readme file

You can find important current product information about the supplied software in the Readme file (for example, which Windows versions are supported). You can view the readme file during the setup program or open it later on with the menu command **Start > SIMATIC > Product Info > German** .

Important Note for Maintaining Operational Safety of Your System

Note

The operators of systems with safety-related characteristics must adhere to operational safety requirements. The supplier is also obliged to comply with special product monitoring measures. To keep you informed, a special newsletter is therefore available containing information on product developments and properties that are important (or potentially important) for operating systems where safety is an issue. To ensure that you are always up-to-date and able to make changes to your system, it is necessary that you subscribe to the appropriate newsletter. To do this, got to theInternet (http://www.siemens.com/automation/csi_en_WW/news).

There, you can register for the following newsletters:

- Embedded/PC-based Automation
- SIMATIC S7-300/S7-300F
- SIMATIC S7-400/S7-400H/S7-400F/FH
- Distributed I/O
- SIMATIC Industrial Software

Select the "Updates" check box for each newsletter.

11.2 Selecting suitable hardware

Hardware requirements

The same system requirements apply to WinAC RTX F as for WinAC RTX with the following list of specifications. The system requirements are described in Chapter System Requirements (Page 21).

Using systems with Programmable Interrupt Controller (PIC)

Older PC systems with programmable interrupt controllers (PIC) are not suitable for WinAC RTX F.

Using removable media (for example, Micro Memory Card, Flash Card or hard disk)

The us of removable storage devices may result in longer waiting times during a memory reset of the WinAC RTX F ("Memory reset" see Chapter Resetting the memory (Page 76)).

Use the corresponding high-performance memory media.

F-capability of the hardware

The hardware used must be compliant for a safety program created with S7 Distributed Safety. You can find a complete, up-to-date list of the hardware tested for F-capability in the Internet at ID 35924276 (http://support.automation.siemens.com/WW/view/en/10805639/133100).

⚠ WARNING

Use a hardware that meets the environment and application conditions, as well as the electromagnetic compatibility.

Using System Management Interrupts (SMI)

A System Management Interrupt (SMI) can influence the real-time behavior of PC systems and result in a stop of the F controller (F-capable controller). In case of an SMI, the PC hardware will switch to System Management Mode (SMM) which executes special system functions. There will be a delay in the cyclic behavior of the real-time operating system that WinAC RTX cannot prevent and that may affect availability.

SMIs may be triggered by monitor switch, volume control and control of brightness via special function keys as they are found on notebooks.

We recommend that you operate systems that are not subject to a significant impact on real-time behavior through SMIs (see ID 35924276 (http://support.automation.siemens.com/WW/view/en/10805639/133100)).

Type tests

WinAC RTX F based F controllers respond fail-safe even with increased electromagnetic interference. Special type tests for functional safety are therefore not required for these F controllers – contrary to F I/O devices.

Testing the F capability of the hardware

⚠ WARNING

If you use F-capable hardware that is not included in the above list, you have use a special software to test the F capability.

The test software is also installed when WinAC RTX F is installed on your PC.

For series PCs with identical hardware (motherboard), you only have to perform the test once as an example.

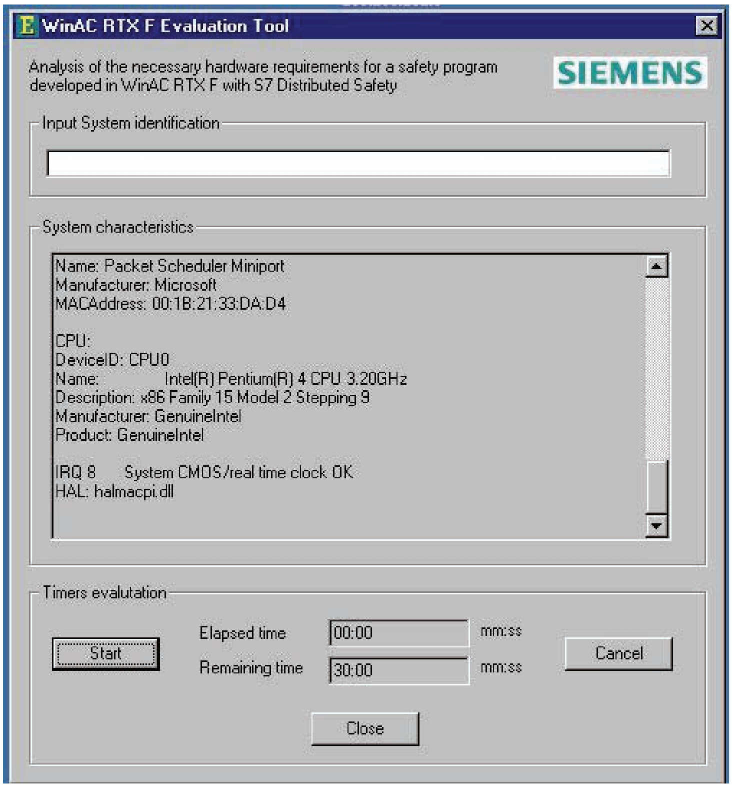
Procedure for testing the F capability of the hardware

Requirement: The software WinAC RTX F 2010, version V4.6 must be installed on the PC.

To test the F capability, follow these steps:

1. Call-up the software to test the F-capability under WinAC RTX F via **START > SIMATIC > PC Based Control > Evaluation Tool.**

   The software opens with the following dialog box:



WinAC RTX F Evaluation Tool

Analysis of the necessary hardware requirements for a safety program developed in WinAC RTX F with S7 Distributed Safety

Input System identification

System characteristics

```
Name: Packet Scheduler Miniport
Manufacturer: Microsoft
MACAddress: 00:1B:21:33:DA:D4

CPU:
DeviceID: CPU0
Name:           Intel(R) Pentium(R) 4 CPU 3.20GHz
Description: x86 Family 15 Model 2 Stepping 9
Manufacturer: GenuineIntel
Product: GenuineIntel

IRQ 8     System CMOS/real time clock OK
HAL: halmacpi.dll
```

Timers evalutation

Elapsed time: 00:00 mm:ss

Remaining time: 30:00 mm:ss

2. Enter an appropriate system designation for your hardware.
3. Start the test with the "Start" button.

Result: Data for identifying your hardware (for example, MAC address, CPU ID) are displayed in the "System Information" field. After the expiry of the period for the test, you receive a message indicating whether the F capability of the hardware was determined or not.

The data for the identification of the hardware and the result of the test are offered for saving as a text file.

Note

Note that if you save the file in the installation path of WinAC RTX F, the file will be deleted when if you uninstall WinAC RTX F.

Assignment of IRQ 8 to the real-time clock

The permanent assignment of the IRQ 8 (interrupt request 8) to the real-time clock (system CMOS/real-time clock) is required for the operation of WinAC RTX F. If this is not satisfied, for example by assigning IRQ 8 to the high precision event timer (HPET), then the system is not suited for operation with WinAC RTX F.

You can check the assignment of the IRQ 8 prior to the installation of WinAC RTX F in Windows XP as follows:

1. Select START > Settings > Control Panel > System > Hardware tab > Device Manager.
2. In View select "Resources by type".
3. Under "Interrupt request (IRQ)" check the assignment of IRQ 8 to the real-time clock.

PROFIBUS and PROFINET conformity

a complete and current list of hardware tested for WinAC RTX F based on PROFIBUS and PROFINET conformity can be found in the Internet at ID 35923715 (http://support.automation.siemens.com/WW/view/en/10805639/133100).

Using the yellow TÜV sticker

The yellow stickers contained in the delivery package with the TÜV SÜD mark of conformity for functional safety may only be used in connection with WinAC RTX F for labeling hardware on which WinAC RTX F and the corresponding safety-related software system (S7 Distributed Safety) is installed.

11.3 Installing/removing the WinAC RTX F 2010 software package

Installing WinAC RTX F software

To install WinAC RTX F, following the WinAC RTX installation instructions.
Your PC must satisfy certain system requirements for the installation.

The installation and the system requirements are described in the Chapters Installation (Page 27) and System Requirements (Page 21).

Software requirements

Table 11- 1 Software requirements

You want to use WinAC RTX F 2010 for	You then require
Applications of safety technology	STEP 7 version 5.5 or higher + HSP212, 135, 178 for WinAC RTX F 2010 on embedded controller + HSP211 for WinAC RTX F 2010 on PC station Optional package S7 Distributed Safety V5.4 + SP5 or later S7 F Configuration Pack V5.5 + SP6 + HF1
Standard applications	STEP 7 version 5.5 or higher + HSP212, 135, 178 for WinAC RTX F 2010 on embedded controller + HSP211 for WinAC RTX F 2010 on PC station

PC with multiple Windows installations

⚠ WARNING

Only one WinAC RTX F may be installed on a PC with several Windows installations (for example via boot manager).

11.4 Technical data

11.4.1 Quantity structure and technical specifications

The technical data for the WinAC RTX F 2010 with order number 6ES7671-1RC08-0YA0 are available in SIMATIC customer support (http://support.automation.siemens.com/WW/view/en/6ES7671-1RC08-0YA0).

PROFIsafe mode

WinAC RTX F supports PROFIsafe V1 and PROFIsafe V2-MODE.

F local data parameters

You can configure the local data for each priority class for the WinAC RTX F. Therefore, assign the largest possible area of local data for the priority classes, in which the safety program (the F-CALL blocks) are called (for example OB 35) (also refer to the S7 Distributed Safety, configuring and programming (http://support.automation.siemens.com/WW/view/en/22099875) manual).

Probabilities of failure

The following table shows the failure probabilities for a WinAC RTX F based controller with S7 Distributed Safety:

Table 11- 2 Failure probability values WinAC RTX F

Operation in Low Demand Mode Low demand mode (average probability of failure on demand)	**Operation in High Demand or Continuous Mode** High demand/continuous mode (probability of a dangerous failure per hour)	**Proof-test interval**
< 1E-04	< 3E-09	10 years

Safety-oriented communication

With WinAC RTX F you have the same safety-related communication as with other F-CPUs (for example, CPU 416F-3 PN/DP) in F-systems with S7 Distributed Safety.

The configuration and programming of the communication are described in the S7 Distributed Safety, configuring and programming (http://support.automation.siemens.com/WW/view/en/22099875) manual.

Start-up protection for inconsistent safety program

In connection with the safety program, WinAC RTX F supports the recognition of an inconsistent safety program. During startup WinAC RTX F recognizes an inconsistent safety program, the F controller goes to STOP and the following diagnostic event is entered in the diagnostic buffer of the F controller:

- "Inconsistent safety program"

Restrictions with SFC 22 "CREAT_DB", SFC 23 "DEL_DB", SFC 82 "CREA_DBL" and SFC 85 "CREA_DB"

F-DBs cannot be generated or deleted.

Restrictions with SFC 83 "READ_DBL" and SFC 84 "WRIT_DBL"

The destination address may not point to an F-DB.

Restriction in the configuration of the retentive behavior of the data block.

The configuration of data block retentivity is not supported for F-DBs. This means that the actual values of the DB are non-retentive at POWER ON/OFF and restart (STOP-RUN) of the CPU. The F-DBs receive the initial values from the load memory. The "Non-retain" ("non-retentive") check box must be selected and grayed out in the block properties of the F-DBs.

Calculation of the runtimes of the F-run-time groups

The Excel file S7fcotia.xls (http://support.automation.siemens.com/WW/view/en/25412441) for S7 Distributed Safety is available in the Internet for sample hardware to support the approximate calculation of the runtimes of the F-runtime groups of the safety program with WinAC RTX F.

Operation of Embedded Controller with isolation module 6ES7195-7KF00-0XA0

Note

Unlike other modules, it is not permitted to plug the isolation module (order no. 6ES7195-7KF00-0XA0, production version 01 and 02) into the same racks as the F controller. This restriction no longer applies to version 03 or later of the isolation module.

Ensuring availability

Information about ensuring availability can be found in Chapter Tuning the controller performance (Page 183).

Interference of the IRQ 8 by other applications

The IRQ 8 (interrupt request 8), which is clocked via the real-time clock (system CMOS/realtime clock), is required for operation of the WinAC RTX F. Therefore, avoid software that affects the IRQ 8 (for example, multimedia applications or kernel debuggers) and do not change the device resources assignment of the IRQ 8 to the real-time clock.

Restrictions during a Windows crash (blue screen)

The IRQ8 may be impaired and the WinAC RTX F stopped if Windows crashes (blue screen).

Setting the HAL timer period

The HAL timer period sets a number of microseconds as the basis for RTX timers. The default value is 100 microseconds. (also see Chapter 12.3.1). A setting of this value larger than 200 microseconds is not permitted and can stop the WinAC RTX F.

Disable the power-saving functions of the computer

To ensure the real-time capability of the F controller, the power saving functions of the computer will have to remain disabled.

The "General" tab of the "WinAC Properties" dialog contains an option (Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology) that is used to disable changing of the processor clock rate.

The option Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology is activated by default. The changing of the processor cycle clock is therefore disabled by default. If you use a SIMATIC PC, you can check whether the power saving function is deactivated in the BIOS. If this is the case, the option Disable Intel (R) "SpeedStep" and AMD "Cool'n'Quiet"(TM) Technology can be deactivated .

WinAC RTX F is configured so that a constantly high processor load of 100% is displayed in the Tuning Panel and Task Manager by the manual inhibition of the power saving function. The processor utilization is only simulated artificially. It does not affect the performance of other applications on the PC.

Changes to WinLC RTX options take effect only after a restart of WinLC RTX.

Support of dedicated mode

WinAC RTX F only supports the dedicated mode from IntervalZero RTX on multi-core systems.

Note

For availability reasons, it is prohibited to use the shared mode.

Operation on virtual machines

⚠ WARNING

Operation on a virtual machine is not permitted for WinAC RTX F.

Booting via network drive

⚠ WARNING

Booting via a network drive is not permitted for WinAC RTX F and must be switched-off in the BIOS.

11.5 Downloading and commissioning the safety program

Downloading the safety program to WinAC RTX F

⚠ WARNING

To ensure that the F controller does not contain an "old" safety program, proceed as follows when transferring the safety program to the F controller with a PG/PC:

1. Execute a memory reset of the WinAC RTX F
   (see Chapter Resetting the memory (Page 76)).
2. Download the configuration from HW Config to WinAC RTX F
   (see Chapter Working in STEP 7 (Page 103)).
3. In the "Safety program" dialog, load the safety program the WinAC RTX F
   (see S7 Distributed Safety, configuring and programming
   (http://support.automation.siemens.com/WW/view/en/22099875) manual).

   If the function test of the safety program does not take place in the target F controller, you will also have to include steps 4. and 5.:
4. Perform a program identification (i.e. check whether the complete signatures of all F blocks are identical with the F attribute of the block container online and offline
   (see S7 Distributed Safety, configuring and programming
   (http://support.automation.siemens.com/WW/view/en/22099875) manual).
5. Perform the startup of the F system (see S7 Distributed Safety, configuring and programming (http://support.automation.siemens.com/WW/view/en/22099875) manual).

Make sure that WinAC RTX F is not closed (for example, by POWER OFF/POWER ON or booting) between the online program identification and the startup of the F system.

Rules for inserting removable media (for example, Micro Memory Card, Flash Card or hard disk)

⚠ WARNING

You must use an access protection to limit access to the WinAC RTX F to persons authorized to insert removable media.

You must ensure that the correct safety program is located on the inserted removable media, either by online program identification or by other measures
(such as unique identification of the removable media).

11.6 Archiving and restoring safety programs

Archiving and restoring with WinAC RTX

Via the WinAC RTX control panel, you use the archive command to save the configuration and the STEP 7 user program to an archive file (*.wld). You can quickly restore the configuration and the STEP 7 user program for the controller with the aid of the archive file.

Special consideration relating to archiving F data blocks

The current values for F-DBs are archived from the load memory, unlike data blocks whose values from the work memory are archived.

Protection of the archiving/restoring of safety programs

⚠ WARNING

You must use an access protection to limit access to the WinAC RTX F to persons who are authorized to archive and restore, for example by corresponding password protection for the controller panel of WinAC RTX F.

You can use either online program identification or the unique designation of the archive file * wld to determine whether the correct safety program exists in the archive file.

Creating archive files for WinAC RTX F

You can create an archive file as in the standard, via STEP 7 or via the Controller Panel of WinAC RTX F.

⚠ WARNING

You must comply with the following points when creating an archive file:

- It is not permitted to archive safety programs in an existing archive file.
  Therefore, create a new archive file.
- Creating an archive file with STEP 7: Based on the collective signature in S7 Distributed Safety in the "Safety program" dialog box, make sure that it is the safety program that is to be archived.
- Create an archive file via the Controller Panel. To ensure that the correct safety program is available in WinAC RTX F, perform an online program identification.
- Assign a unique name for the archive file *.wld.

Restoring only possible via the Controller Panel of WinAC RTX F

⚠WARNING
The restoring of a safety program is only permitted via the Controller Panel function "Restoring an S7 user program" and not via a function created with ODK (Open Development Kit) and CMI (Controller Management Interface).

Message during restoring a safety program

⚠WARNING
The successful restoring of safety program is indicated by a corresponding message. If this message does not appear, the restoring has failed. Perform the restoring again.

Further information

Additional information about archiving and restoring configurations and STEP 7 user programs can be found in Chapter Archiving and restoring STEP 7 user programs (Page 85).

11.6.1 Special considerations relating to the "Data Storage" function

⚠ WARNING

You must use an access protection to limit the access to WinAC RTX F to persons authorized to handle WinAC RTX F data, for example by limiting access to the PC.

Changing the program and configuration path ("Data Storage" WinAC function)

⚠ WARNING

When you change the program and configuration path you must comply with the following procedure to ensure that no safety program remains in the previous path:

1. Archive the safety program and the configuration (see chapter Archiving and restoring safety programs (Page 250)).

   Alternatively, you can store the safety program and the configuration in STEP 7.
2. Execute a memory reset of the WinAC RTX F (see Chapter Resetting the memory (Page 76)).
3. Change the program and configuration path via the control panel of the WinAC RTX F in the tab "Data Storage" (see Chapter "Data Storage" tab (Page 66)).
4. Restore the safety program and the configuration (see chapter Archiving and restoring safety programs (Page 250)).

   Alternatively, you can load the safety program and the configuration from STEP 7 in the WinAC RTX F (see chapter Downloading and commissioning the safety program (Page 249)).

It is not permitted to overwrite data in the storage path for "Program and configuration"

⚠ WARNING

Data that has been saved in the storage path for "Program and configuration" may not be overwritten (see Chapter "Data Storage" tab (Page 66)).

Determine the storage path for "Program and configuration" as follows:

1. In the WinAC RTX F Controller Panel, select the **CPU > Tools > Options** menu command and open the "Data Storage" tab in the "Options" dialog box.
2. Determine the path set for "Program and configuration".

Disable extended write filter (EWF, FBWF) for data in the storage path for "Program and configuration"

⚠ WARNING

It is not permitted to enable extended write filter (EWF, FBWF) for data in the storage path for "Program and configuration". Prevent activation of the write filter by implementing access protection for WinAC RTX F.

Further information

Additional information about the function "Data storage" and upgraded writing filters can be found in Chapter "Data Storage" tab (Page 66).

11.6.2 Special considerations in creating and importing images

Creating an image

⚠ WARNING

You must comply with the following points when creating an image with a safety program:

- Use access protection to limit access to WinAC RTX F to persons who are authorized to create images.
- Before creating the image, you must use program identification to ensure that the correct safety program is available in WinAC RTX F.
- Images with safety programs must be created on an empty data carrier (deleted or formatted) or an existing image must be explicitly deleted.
- After creating the image, remove the data carrier containing the image.

Import image

⚠ WARNING

You must comply with the following points when importing an image with a safety program:

- You must use an access protection to limit access to the WinAC RTX F to persons authorized to import images.
- Make sure that there is no "old" safety program in the F controller. Therefore, it is important to execute a memory reset prior to the import (see Chapter Resetting the memory (Page 76).
- You must ensure that the correct safety program is located on the image, either by online program identification or unique identification of the data carrier.
- After importing the image, remove the data carrier containing the image.

Further information

For further information on images, refer to the documentation to your PC.

Reference information 12

12.1 Technical Data

Order number

WinLC RTX V4.6 is a component of the package WinAC RTX 2010: 6ES7671-0RC08-0YA0.

Technical specifications

All of the technical specifications regarding WinAC RTX 2010 with the order number 6ES7671-0RC08-0YA0 can be found under SIMATIC customer support (http://support.automation.siemens.com/WW/view/en/6ES7671-0RC08-0YA0).

12.2 Changing the Virtual Memory Paging Configuration

Procedure

To change the virtual memory paging configuration, follow these steps:

1. Select System from the Windows Control Panel.
2. From the Advanced tab of the System Properties dialog, click the Settings button for Performance.
3. From the Advanced tab of the Performance Options dialog, click the Change button for Virtual memory.
4. Make any changes you need, and click OK on the dialogs to complete your configuration.

Result

The virtual paging memory configuration has your new settings.

12.3 Troubleshooting

12.3.1 Particular configuration consideration when using "WinAC Time Synchronization"

Using "WinAC Time Synchronization"

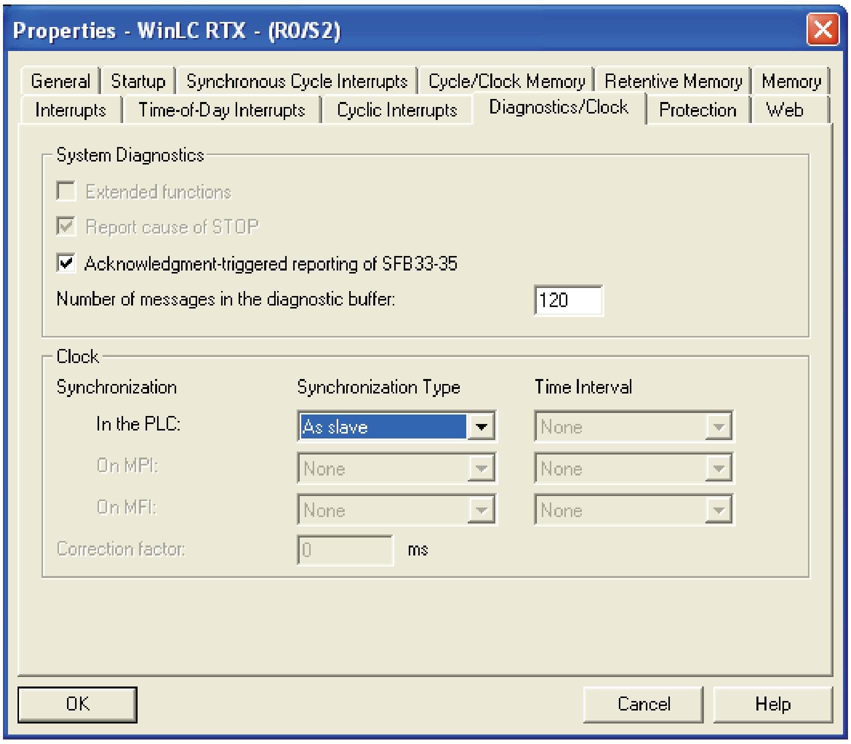
WinAC Time Synchronization can be selected when starting the setup for installation. To do so, insert the WinAC RTX Installation DVD. Follow the instructions of the setup program.

After the installation has been completed successfully, you will also have to activate the utilization of the time synchronization in the configuration.

Basic procedure

Proceed as follows to use WinAC Time Synchronization:

1. Open the SIMATIC Manager.
2. Create a project in HW Config.
3. Open the properties of WinLC RTX.



System Diagnostics

□ Extended functions

☑ Report cause of STOP

☑ Acknowledgment-triggered reporting of SFB33-35

Number of messages in the diagnostic buffer: 120

Clock

Synchronization	Synchronization Type	Time Interval
In the PLC:	As slave	None
On MPI:	None	None
On MFI:	None	None

Correction factor: 0 ms

4. Select the "Diagnostics/Clock" tab.
5. Select the Synchronization type "As slave" under "Synchronization in AS" in the "Clock" field.
6. Confirm with OK.

   The synchronization is active.

12.3.2 Relevant information regarding IntervalZero RTX

The RTX real-time extensions provide the determinism and performance of a real-time operating system within the Windows XP environment. Not all computer configurations (hardware and software) support the installation and operation of IntervalZero RTX. When testing the operation of IntervalZero RTX and WinLC RTX on your computer, check the following items:

- RTX is installed and running. You must have administrator rights (ADMIN) for the computer. Your computer must meet the hardware and software requirements described in the RTX Runtime Release Notes. In addition, the HAL (Hardware Abstraction Layer) installed must be one that RTX supports.
- RTX uses a free interrupt for operation of a communication interface in interrupt mode (differs according to manufacturer). If a free interrupt is not available, the communication interface operates only in polled mode and not in interrupt mode.
- RTX is able to operate without interference from hardware components installed in the computer. Some components (such as the video card) can cause problems that affect the performance of real-time control with IntervalZero RTX.

Setting the HAL timer period

The HAL timer period sets a number of microseconds as the basis for RTX timers. The default value is 100 microseconds. WinLC RTX uses the RTX timers for starting certain OBs, for SFC 47 (WAIT), and for other internal events. Changing the HAL timer period may provide more deterministic behavior for some applications that require accuracy of less than 1 ms. However, decreasing the HAL timer period also increases the CPU load, with no benefit for most applications.

NOTICE
Changing the HAL timer period to a value lower than the default value can increase the load on the CPU of your computer. This increased CPU usage could affect the operation of your application. If you change the HAL timer period, always test your application to ensure that the increased CPU load does not adversely affect the operation of WinLC RTX.

To change the value for the HAL timer period, follow these steps:

1. Use the **Start** menu to open the Windows Control Panel.
2. Double-click the RTX Properties icon to display the "RTX Properties" dialog.
3. Click the "Setting" tab to display the parameters for the HAL timer.
4. Adjust the value for the HAL timer period (in microseconds) and click OK.

Operating a communication interface in interrupt mode

On some computers, RTX allows a free interrupt for a communication interface (varies for different computer manufacturers). If a free interrupt is not available, CP cards (including integrated PROFIBUS or PROFINET communication interfaces on Siemens PCs) operate only in polled mode and not in interrupt mode, which can affect the performance of the CP.

Additional information on this can be found in Improving the performance of a communication interface (Page 262).

Using the RTX Platform Evaluator to check performance

On some computers, some components of the computer (such as a video card) can cause problems with Interval Zero RTX that affect the performance of real-time control.

You can use the RTX Platform Evaluator utility to determine if your computer has any hardware installed (such as a video card) that may introduce jitter or latencies.

The RTX Platform Evaluator is not included with WinAC RTX. Contact Interval Zero to obtain the RTX Platform Evaluator and information about how to install and use it.

Changing the HAL type for the computer

CAUTION
Changing the HAL type can create a situation where the computer cannot be booted. You must then recover by using an Emergency Repair disk. Changing the HAL type changes the entry in the Windows registry. Errors in the registry can keep the computer from rebooting. Before you make any changes to the Windows registry (such as changing the HAL type), always create an Emergency Repair disk. Select the **Start > Programs > Accessories > System Tools > Backup** menu command to create an Emergency Repair disk.

12.3.3 Using several Ethernet interfaces

Please observe the following description of issues when using two or more identical Ethernet interfaces on a PC.

Note

Configuration of more than two Ethernet interfaces

If you require two or more Ethernet interfaces for Windows communication, you have to configure a CP 1616 for the PROFINET communication.

To do so you need to first remove any CP 1616 NDIS drivers that may already be installed on your PC. Then assign the communication interface CP 1616 as a submodule to WinAC RTX.

You can now configure the interfaces remaining under Windows.

Display of the assignment of available Ethernet interfaces in the submodule dialog

One of the Ethernet interfaces of your PC is already configured as "IE General" in the component configurer of the PC station. The other interfaces are erroneously displayed in the submodule dialog of the WinAC RTX with the assignment "Index n."

Remedy:

Only rely on the displayed PCI resources when assigning an Ethernet interface to WinAC RTX as a submodule. First assign the required interface for WinAC RTX as a submodule and then configure the interface for "IE General" in the component configurer.

Changing the interfaces displayed in the component configurer

One of the Ethernet interfaces of your PC is already configured as "IE General" in the component configurer of the PC station, e.g. on index 3. If you assign this interface as a submodule in the "Submodule" tab in the properties dialog box of the WinAC RTX then the "IE General" entry will remain in the component configurer but will no longer have a function.

Remedy:

Remove the IE General in the component configurer in order to have a functional Ethernet interface in the component configurer again. Thereafter, reinsert the remaining Intel interface as IE General. Now you can work with both interfaces.

Setting the IP address of the Ethernet interface remaining under Windows

It may appear that the other Ethernet interfaces under Windows can no longer be reconfigured after you have configured one of the Ethernet interfaces of WinLC RTX as a IF submodule. In this case the option "Retrieve IP address automatically" (DHCP) will always be displayed as the active option in the dialog box "Properties of Internet Protocol (TCP/IP)" in the properties of this interface.

Note

Path to the dialog "Properties of the Internet Protocol (TCP/IP)"

The dialog "Properties of Internet Protocol (TCP/IP)" can be accessed via the path **Start > Settings> Network Connections**. Select a "LAN connection" and click-on the command "Properties" in the context menu. Double-click on the element "Internet Protocol (TCP/IP)" in the dialog "Properties of the LAN connection."

Assign the Ethernet interface parameters by using the cmd commands in the command box

Remedy 1: Activate automatic assignment (DHCP)

- netsh interface ip set address name="LAN connection" source=dhcp
- netsh interface ip set dns name="LAN connection" source=dhcp

The name of the Ethernet interface is referred to with "LAN connection." It can be found under the dialog **Start > Settings> Network Connections**. This name must be set in quotation marks.

Remedy 2: Activate assign a permanent IP address

- netsh interface ip set address name="LAN connection" source=static addr=192.168.1.1 mask=255.255.255.0

Check: Enter the cmd command "ipconfig -all" in the command box in order to check the current or changed settings.

12.3.4 Troubleshooting Network Problems

The controller panel disposes of two status displays EXTF and BUSF that you can use for error diagnostics in the I/O communications network. The table below describes the activity of the EXTF and BUSF indicators based on the particular error. In addition, possible remedies are provided.

EXTF	BUSF	Description	Remedy
Off	Off	No configuration	Ensure that the submodule configuration has been entered into your STEP 7 project. Download the project's System Data container to the controller.
		Normal operation	The configured DP slaves or PROFINET IO devices are responding. No action is required.
On	Flashes	Station failure	Check whether the bus cable is connected to the WinLC RTX (the CP card) and whether all the segments are connected correctly to nodes supplied with voltage. Check whether the bus is interrupted.
		At least one of the DP slaves or one of the PROFINET IO devices could not be accessed.	Wait for completion of the power-on cycle. If the display still flashes check the DP slaves or PROFINET IO devices and evaluate the diagnostic data.
—	On	Bus fault (hardware failure)	Check the bus cable or Ethernet cable for a short-circuit, or a wire break wire or interrupted connection.
On	Off	Diagnostic errors	Indicates that a fault condition was not deleted or that one of the following conditions has occurred: • An I/O module with diagnostics capability has started OB 82. • A submodule configuration does not correspond to the configuration loaded from STEP 7, for example, if one configuration is for CP 5613 and the other is for CP 56x1 or CP 5623/24.
On	On	CP fault	Indicates that either no configured CP was found or that it was defective.

In addition to these displays you can use the hardware diagnostics feature of STEP 7 to determine which nodes are experiencing problems and to determine the cause of the problem.

12.3.5 Improving the Performance of a Communication Interface

To use a PROFIBUS DP interface in isochronous mode, the DP interface must operate in interrupt mode. For maximum performance of a PROFINET IO interface, the interface must also operate in interrupt mode.

Note

You can use isochronous mode in more than one PROFIBUS DP subnets with WinLC RTX. Assign an exclusive interrupt to the CP 56x1 and Intel PRO/1000 communication interfaces. Do not use this assigned interrupt with any other device of the Windows operating system (for example, a video card).

Tool: You use the Windows Device Manager.

WinLC RTX accesses communication interfaces in either interrupt mode or polled mode. Interrupt mode provides improved performance over polled mode.

In order for WinLC RTX to use interrupt mode for accessing a communication interface, you must configure your computer so that the communication interface does not share an IRQ (interrupt request) with a Windows-controlled device.

Checking IRQ assignments

Use the following procedure to determine whether the IRQ assignment for a communication interface is shared with an IRQ assignment for a Windows-controlled device:

1. Right-click the "My Computer" icon and select the "Manage" menu command.
2. Select the Device Manager, and then select the **View > Resources by Type** menu command.
3. Open the "Interrupt request (IRQ)" folder. The numerical values shown beside each entry indicate the IRQ assignment.
4. Locate the entry for the communication card in the device list. If the IRQ assigned to this entry is assigned to any other device, the card is sharing an interrupt with that device. If this other device is Windows-controlled, the communication card will be operated in polled mode if it is configured as a submodule of WinLC RTX. Otherwise, the communication card operates in interrupt mode.

Determining whether a device is Windows-controlled or RTX-controlled

To determine whether a device is Windows-controlled (as opposed to RTX-controlled), follow these steps:

1. Right-click the device entry for the communication card in the Device Manager list and select "Properties".
2. Select the "General" tab on the Properties dialog box and check the device type value. If "RTX Driver" is displayed, the device is RTX-controlled. Otherwise, it is Windows-controlled.

If a CP 56x1 or Intel PRO/1000 communication interface shares an IRQ number with a device controlled by Windows then carry out one of the following actions to change the system configuration for your computer and to assign a different IRQ number to the communication interface:

- Use the BIOS setup utility for your computer to manipulate IRQ assignments and remove the IRQ conflict.
- Install the communication card in a different PCI expansion slot of your computer. Because the PCI slots are often assigned different IRQ numbers, installing the card in a different slot might eliminate the conflict. However, changing the slot can also result in a new conflict.
- If the IRQ conflict is due to a built-in device (for example, an Ethernet or SCSI controller), consider using the BIOS setup utility to disable the conflicting built-in device, if possible. In this case, you might have to use an equivalent expansion card to replace the functionality of the disabled device.

Using these methods can be a slow process, and you might not find a solution that assigns a suitable IRQ number to the communication interface. If no configuration can be found that eliminates the IRQ conflict, you must either select a different PC platform or you must use the polled mode of operation for the communication interface.

For multiple cards, repeat this process as necessary to resolve all interrupt conflicts.

12.3.6 Responding to Diagnostic Events

If an error is detected by the controller, the error condition is logged in the diagnostic buffer (Page 82) as a diagnostic event. The diagnostic events that typically occur with the distributed I/O can cause the controller to execute the following OBs:

- OB 40 responds to hardware interrupts (process alarms) generated by an I/O module with configured interrupt capability.
- OB 82 responds to diagnostic interrupts generated by an I/O module with configured diagnostic interrupt capability.
- OB 83 responds to the removal / insertion of modules at a DP slave or PROFINET IO device (e.g. ET 200M) that was set up for the removal / insertion of modules.
- OB 85 responds to a priority class error. There are multiple causes for OB 85 relating to the DP system. If the controller attempts to copy the inputs (or outputs) of a module during the I/O cycle without the module being ready to operate, OB 85 is executed.
- OB 86 responds to a station failure or some other interruption of the physical network (such as a short circuit).
- OB 122 responds to an I/O access error by the user program. If OB 122 is not programmed, the WinLC controller goes to STOP mode.

You can use SFC 39 to SFC 42 to disable, delay, or re-enable any of these OBs. If an OB that is not loaded in the controller was called, the controller goes to STOP mode.

The local variables of these OBs contain startup information indicating the cause for executing the OB. The program for the OB can use this information for the response to the event. You can also use SFC 13 to read the diagnostic information from a DP slave.
Or you can SFB 52 or SFB 54 to read the diagnostic information from a PROFINET IO device.

Detailed information about the use of OBs, SFC 13, SFC 52 and SFB 54 can be found in the STEP 7 online help or in the *System Software for S7-300/400 System and Standard Functions* reference manual. If you want to open this manual on a PC on which STEP 7 is installed, select **Start > SIMATIC > Documentation > English** and double-click "STEP 7 - System and Standard Functions for S7-300 and S7-400".

12.3.7 Cross-Module Access Errors

Unlike hardware PLCs, PC-based controllers do not allow a Load (L) or Transfer (T) instruction to access bytes of more than one module. Consider a configuration of two output modules, each containing five bytes. Module 1 is addressed from 10 to 14, and Module 2 is addressed from 15 to 19. OB 1 contains the instructions shown below:

```
L 5
T PAW 14
```

In this example, OB 122 is called because of an attempt to access bytes across a module boundary. A word instruction at address 14 attempts to access address 14 and 15, which is prevented because the addresses are not in the same module.

12.3.8 Correcting invalid Characters prior to STEP 7 V5.3 SP1

You can use STEP 7 to create a name for the controller and to download the configuration with the new name to the controller. However, some characters that might have been used in controller names in versions of STEP 7 prior to V5.3 SP1 are invalid. Change these controller names to valid controller names prior to downloading.

CAUTION
Prior to STEP 7 V5.3 SP1, using an invalid character in the controller name creates an instance of the controller that cannot be restarted. Downloading a configuration that uses an invalid character in the controller name creates an invalid instance of the controller. This invalid instance will continue to run and will remain connected to STEP 7 until you shut down the controller. However, the desktop icon and the Start menu command will be removed. Without the desktop icon or Start menu command, you cannot restart the controller after it has been shut down. Avoid the use of the invalid characters in controller names.

Invalid Charaters

The following table describes invalid characters for controller names prior to STEP 7 V5.3 SP1 or SP2:

Character	Name
/	Forward slash (Problematic in versions prior to STEP 7 V5.3 SP1)
.	Period (Problematic in versions prior to STEP 7 V5.3 SP2)
-	Hyphen (also called a dash or a minus sign) (Problematic in versions prior to STEP 7 V5.3 SP1) You cannot create a name that begins with a hyphen (-). You can, however, use a hyphen within the name of the controller. **Valid:** Pump-1: Using a hyphen in the middle of the name is valid. Pump1-: Using a hyphen at the end of the name is valid. **Invalid:** -Pump1: Using a hyphen at the beginning of the name is invalid. -: Using a hyphen as a one-character name is invalid.

Procedure

If you inadvertently downloaded a name that contains an invalid character, follow these steps to correct the problem:

1. Using the STEP 7 Hardware Configuration application, rename the controller to the previous valid name (the name prior to downloading the invalid name).
2. Download the configuration with the previous valid name to the PC station (even if the controller is not running).

Result

After downloading the valid name for the controller, the desktop icon and the Start menu command reappear. You can now rename the controller to a new name that does not use invalid characters.

12.4 System Status List (SSL)

12.4.1 Using SFC 51 to Read the SSL

STEP 7 stores write-protected information about the controller in the system status list (SSL) in the form of partial lists which can be accessed via the SSL ID. WinLC RTX supports an extensive part of the SSL IDs of STEP 7.

You use SFC 51 (RDSYSST) to access the entries in the SSL. You supply the input parameters SSL_ID and Index to access the data records in the sublist. The SFC 51 outputs a header consisting of two words and a sublist or a part of a sublist. The header provides the following information for the sublist:

- The first word defines the length (in bytes) of a data record of the sublist.
- The second word defines the number of data records in the sublist.

The requested information follows the header. The size of the sublist in bytes is data record length multiplied by the number of data records.

Note

The SSL_ID and Index values are represented as hexadecimal (16#) numbers.

Detailed information about the system status list can be found in the STEP 7 online help or in the *System Software for S7-300/400 System and Standard Functions* reference manual. If you wish to call up this manual on a PC on which STEP 7 is installed, select the **Start > SIMATIC > Documentation** menu command. Select the desired language and then double-click "STEP 7 - System and Standard Functions for S7-300 and S7-400".

Some system status lists are only available if you configured at least one WinLC RTX submodule.

See also

SSL_ID 0x11 (Module Identification) (Page 267)

SSL_ID 0x12 (CPU Characteristics) (Page 268)

SSL_ID 0x13 (Memory Areas) (Page 268)

SSL_ID 0x14 (System Areas) (Page 268)

SSL_ID 0x15 (Block Types) (Page 268)

SSL_ID 0x19 (Local Module LED Status) (Page 269)

SSL_ID 0x1C (Component Identification) (Page 269)

SSL_ID 0x22 (Interrupt Status) (Page 269)

SSL_ID 0x25 (Process Image Partitions) (Page 270)

SSL_ID 0x32 (Communications Status) (Page 270)

SSL_ID 0x74 (LED Status) (Page 271)

SSL_ID 0x90 (DP Master System) (Page 271)

SSL_ID 0x91 (Module Status) (Page 272)

SSL_ID 0x92 (Rack and Station Status) (Page 272)

SSL_ID 0x94 (Station status) (Page 273)

SSL_ID 0x95 (Expanded DP master or PN IO system status) (Page 273)

SSL_ID 0x96 (Additional PN IO or DP status information) (Page 274)

SSL_ID 0x9C (tool changer information - PROFINET IO) (Page 274)

SSL_ID 0xA0 (Diagnostic Buffer) (Page 274)

SSL_ID 00B1, 00B2, 00B3, and 00B4 (module diagnostics) (Page 275)

12.4.2 Descriptions of SSL IDs

12.4.2.1 SSL_ID 0x11 (Module Identification)

0111 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0111	Specific information for a module	0001: Order number, module type, and version0007: Firmware version

12.4.2.2 SSL_ID 0x12 (CPU Characteristics)

0012, 0112, 0F12 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0012	All characteristics for a module	MC7 processing unit, time system, system response, and MC7 language description
0112	One specific group of characteristics	0000: MC7 processing unit 0100: Time system 0200: System response 0300: MC7 language description
0F12	Header information only	

12.4.2.3 SSL_ID 0x13 (Memory Areas)

0113 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0113	Specific memory area	0001: User memory 0002: Load memory integrated 0003: Load memory inserted 0004: Maximum insertable Load memory 0005: Backup memory 0006: Peer-to-peer memory (shadow memory)

12.4.2.4 SSL_ID 0x14 (System Areas)

0014, 0F14 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0014	All system memory areas for a module	Size and other parameters for each area of system memory
0F14	Header information only	

12.4.2.5 SSL_ID 0x15 (Block Types)

0015 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0015	All block types for a module	Maximum number and size for each type of block

12.4.2.6 SSL_ID 0x19 (Local Module LED Status)

0019, 0F19 (hexadecimal)

Note

SSL_ID 0x19 supports local, non-redundant CPUs. You can use SSL_ID 0x19 with a redundant H CPU only when the H CPU is in a non-redundant operating mode.
Use SSL_ID 0x74 (Page 271) to access information for a redundant H CPU.

SSL_ID	Sublist	Index and Contents of the Record
0019	All of the LEDs for the local module	Status for all of the LEDs
0F19	Header information only	

12.4.2.7 SSL_ID 0x1C (Component Identification)

001C, 011C, 0F1C (hexadecimal)

SSL_ID	Sublist	Index and contents of the data record
001C	All information of a component	Controller name, module name, module variable, copyright, serial number, project ID, module type and manufacturer information
011C	Specific element of the component	0001: Name of the controller 0002: Name of the module 0003: Module tag 0004: Copyright entry 0005: The number "Zero" is output 0007: Module type 0009: Manufacturer and profile identification 000B: Location designation (LID) of a module
0F1C	Header information only	

12.4.2.8 SSL_ID 0x22 (Interrupt Status)

0222 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0222	Start event for a specific OB	OB number: Start event and time for the requested OB

Note

For a list of the OBs supported by WinLC RTX, refer to the following topics: Logic Blocks Supported by WinLC RTX (Page 104) and Organization Blocks (OBs) (Page 105).

12.4.2.9 SSL_ID 0x25 (Process Image Partitions)

0025, 0125, 0225, 0F25 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0025	All process image partitions	Process image partitions for all of the OBs that have been downloaded to the module
0125	Process image partition for a specific OB	Partition number: OB configured for that partition
0225	OBs assigned for a specific process image partition	OB number: Partition assigned for that OB
0F25	Header information only	

12.4.2.10 SSL_ID 0x32 (Communications Status)

0132, 0232 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0132	Specific set of parameters	0001: Number and type of connections 0002: Connections configured 0003: Operator interface 0004: Protection level and mode switch selection 0005: Diagnostics 0006: Peer-to-peer status data 0008: Time system 000A: Baud rate
0232	Parameters for a redundant system (H CPU)	0004: Protection level and mode switch selection

12.4.2.11 SSL_ID 0x74 (LED Status)

0174 (hexadecimal)

Note

Use SSL_ID 0x74 to access information about LEDs for any module, including a redundant H CPU module. See also SSL_ID 0x19 (Page 269).

SSL_ID	Sublist	Index and Contents of the Record
0174	Specific LED	0002: INTF (Internal failure) 0003: EXTF (External failure) 0004: RUN (Run) 0005: STOP (Stop) 0006: FRCE (Force) 0008: BATF (Battery failure) 000B: BUSF1 (submodule 1 fault) 000C: BUSF2 (submodule 2 fault) 0012: BUSF3 (submodule 3 fault) 0013: BUSF4 (submodule 4 fault) 0021: MAINT (maintenance is required)

12.4.2.12 SSL_ID 0x90 (DP Master System)

0090, 0190, 0F90 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0090	All DP masters configured on the network and downloaded to the module	DP master identifier, address, and attributes for all DP masters
0190	Specific DP master	DP master identifier: Address and attributes
0F90	Header information only	

12.4.2.13 SSL_ID 0x91 (Module Status)

0591, 0991, 0C91, 0D91, 0E91 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0591	Module status information of all submodules of the host module	Irrelevant
0991	Module status information of all submodules of the host module in the rack specified	Rack or DP master system ID
0C91	Specific module, identified by the logical base address	Logical base address: Features and parameters of the specified DP interface module or PROFINET interface module
0D91	Specific station, identified either by rack/station, or one of these means: • PROFIBUS DP: DP master identifier, or DP master identifier with station number • PROFINET IO station number and last two places in the PNIO subsystem ID	Station identifier: Features and parameters for all the modules of the specified station (DP or PROFINET)
0E91	Module status information of all assigned modules	Irrelevant
0F91	Header information only	

12.4.2.14 SSL_ID 0x92 (Rack and Station Status)

0092, 0192, 0292, 0692 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
0092	Expected status of the stations of a DP master	0: Local DP master DP master identifier: Specific DP master
0192	Configuration and activation status for the stations of a DP master	0: Local DP master DP master identifier: Specific DP master
0292	Actual status for the stations of a DP master	0: Local DP master DP master identifier: Specific DP master
0692	OK state for the stations of a DP master	0: Local DP master DP master identifier: Specific DP master

12.4.2.15 SSL_ID 0x94 (Station status)

0094, 0194, 0294, 0694, 0794, 0F94 (hexadecimal)

SSL-ID	Sublist	Index and contents of the data record
0094	Specified status of stations of a central rack, PN IO, or DP subsystem	0: central module 1 to 31: distributed module on PROFIBUS DP 100 to 115: distributed module on PROFINET IO
0194	Activation status of a station	1 to 31: distributed module on PROFIBUS DP 100 to 115: distributed module on PROFINET IO
0294	Actual status of a station	0: Central module 1 to 31: distributed module on PROFIBUS DP 100 to 115: distributed module on PROFINET IO
0694	All faulty stations of a central rack, PN IO, or DP system	0: Central module 1 to 31: distributed module on PROFIBUS DP 100 to 115: distributed module on PROFINET IO
0794	All faulty stations of a central rack, PN IO, or DP system; using additional bits to determine whether a station is faulty	0: Central module 1 to 31: distributed module on PROFIBUS DP 100 to 115: distributed module on PROFINET IO
0F94	Header information only	

12.4.2.16 SSL_ID 0x95 (Expanded DP master or PN IO system status)

0195, 0F95 (hexadecimal)

SSL-ID	Sublist	Index and contents of the data record
0195	Specific DP master or PN IO system	DP master identifier: Properties of the stations of the specified DP master (such as DP mode, equidistant mode and cycle, clock synchronization, and transmission rate) PN IO system identifier: Properties of the stations of the specified IO controller (such as rack number, slot of the IO Controller, type of IO Controller, and logical start address)
0F95	Header information only	

12.4.2.17 SSL_ID 0x96 (Additional PN IO or DP status information)

0696, 0C96 (hexadecimal)

SSL-ID	Sublist	Index and contents of the data record
0696	Additional information for all configured submodules of a module	Logical base address of a module (PNIO modules only)
0C96	Additional information for a submodule	Logical base address of a module

12.4.2.18 SSL_ID 0x9C (tool changer information - PROFINET IO)

009C, 019C, 029C, 039C, 0F9C (hexadecimal)

SSL ID	Sublist	Index and contents of the data record
009C	Information about all the tool changers and their tools at a PROFINET IO system	PROFINET IO system ID
019C	Information about all the tool changers at a PROFINET IO system	PROFINET IO system ID
029C	Information about a tool changer and its tools	Logical address of the tool changer
039C	Information about a tool and its IO devices	Logical address of a IO device of the tool
0F9C	Header information only	

12.4.2.19 SSL_ID 0xA0 (Diagnostic Buffer)

00A0, 01A0, 0FA0 (hexadecimal)

SSL_ID	Sublist	Index and Contents of the Record
00A0	All of the entries in the diagnostics buffer	Event information for every event listed in the diagnostics buffer
01A0	Most recent entries in the diagnostics buffer	Number: Event information for the specified number of entries in the diagnostics buffer
0FA0	Header information only	

12.4.2.20 SSL_ID 00B1, 00B2, 00B3, and 00B4 (module diagnostics)

00B1, 00B2, 00B4, 00B4 (hexadecimal)

Note: The information is based on the associated module type.

SSL-ID	Sublist	Index and contents of the data record
00B1	Diagnostic information (4 bytes) for a specific module, identified by the logical base address	Logical base address: First 4 bytes of the diagnostic information
00B2	All of the diagnostic information for a specific module, identified by its rack and slot (S7-mEC modules only)	Rack and slot: Complete diagnostic information
00B3	All of the diagnostic information for a specific module, identified by the logical base address	Logical base address: Complete diagnostic information
00B4	Specific DP slave, identified by the configured diagnostic address	Diagnostic address: Standard diagnostic information for a DP station

Note: 00B2 is only available for S7-mEC.

Glossary

Backplane bus

For hardware controllers such as the S7-300 or S7-400, the backplane bus is the printed circuit board inside the rack into which modules are inserted. (See also: "What is a PC station?")

Blue screen

Termination of the Windows operating system, resulting in a display on the monitor of the fatal error on a blue background. A blue screen is also known as a Windows Stop Error.

Cold restart

The controller executes OB 102 before the start of the free scan cycle (OB 1). Like a warm restart, a cold restart resets the peripheral inputs (PI) and changes the peripheral outputs (PQ) to a pre-defined safe state (default is 0). However, a cold restart does not save the retentive memory (M, T, C, or DB), but sets these areas to their default settings.

Communication interface

CP card, Siemens PC built-in PROFIBUS interface, or Industrial Ethernet interface that WinLC RTX uses for communication.

Control program

The control program is the application program created with STEP 7 and downloaded to the controller for execution. The control program includes all organization blocks (such as OB 1 or OB 35) and the other logic blocks that they call, including functions (FCs), system functions (SFCs), function blocks (FBs), and system function blocks (SFBs).

CP card

Communications processor: (See also: "What is a communication interface?")

Cycle time

The cycle time is the time required to execute the complete scan cycle, which includes the execution of OB 1 and the minimum idle time.

Deterministic behavior

Predictability of execution time and response time.

ERTEC

Enhanced Real-Time Ethernet Controller

Execution load

The percentage of CPU time used by the controller.

Execution monitor

The execution monitor of the controller measures the time that the controller sleeps and ensures that the controller does not exceed the maximum execution load. The execution monitor uses the maximum execution load and the execution time limit to calculate the forced idle time.

Execution time

The execution time is the actual time the controller takes to complete one pass through the instructions of the control program. This includes executing OB 1 and updating the I/O.

Execution time limit

The execution time limit defines the maximum amount of time allowed for the controller to execute the control program. The execution monitor uses this value and the maximum execution load to calculate the forced idle time.

Forced execution sleep time

This read-only field shows how much idle time (in microseconds) is required during the monitor interval to meet the maximum execution load requirement.

Free cycle

The free scan cycle consists of the basic tasks for priority class 1: writing to the outputs, reading the inputs, executing OB 1, and completing the idle time requirement before triggering the next free scan cycle. The controller executes these tasks at the base, or lowest, internal priority level for executing the OBs. (Priority level in this context refers to OB priority classes, not the operating system priority level.)

Idle time

The idle time is the difference between the execution time of the free scan cycle and the total scan time. The idle time measures the time between the completion of OB 1 and the start of the next scan cycle, and ensures that the next scan cycle does not start until the end of the idle time. However, if the start event for an interrupt OB (such as OB 40) occurs during the idle time, the controller executes that OB.

IF slot

Interface slot. One of four slots allocated for communication interfaces configured as submodules of the controller. (See also: "What is an IF slot?")

Index

A numbered slot in the PC Station, or virtual rack that represents a PC-based automation system. The controller occupies one index. Other components can occupy other index slots. (See also: "What is an index?")

Industrial Ethernet

Physical communication layer that supports communication with STEP 7, S7 CPUs, PGs, OPs, S7 applications, and PROFINET IO.

IO-Link

IO-Link is a concept for uniform linking of sensors and switching devices to the control level by means of a point-to-point connection.

Isochronous mode

Configuration of the DP cycle that yields a constant bus cycle time. (See also: "Isochronous Mode for a Constant Bus Cycle")

Jitter

Jitter is the difference between the actual cycle time and the configured minimum cycle time.

Load memory

Memory area (RAM) allocated for all of the blocks downloaded from STEP 7 excluding the symbol table and comments.

Maximum execution load

The maximum execution load is the maximum percentage of CPU usage that is allocated for the controller. The execution monitoring uses this value and the execution time limit to calculate the forced idle time.

Memory size

Memory area (RAM) allocated for the blocks used at runtime.

Minimum cycle time

The minimum cycle time is the minimum number of milliseconds from the start of one cycle to the start of the next cycle. You enter a value for the minimum cycle time when you use STEP 7 to configure the system data for the controller. You can use the tuning panel to adjust this value as you test the performance of the controller. After you have tuned the performance of the controller, use STEP 7 to enter the optimum cycle time value and download the new system data. Any value for the cycle time that you enter in the tuning panel is overwritten by the value in the system data when the controller changes from STOP mode to RUN mode.

Minimum idle time

The minimum idle time is the specific amount of time that the controller must wait before starting the next scan cycle. You use the tuning panel to configure this parameter.
The controller uses the minimum idle time and the minimum cycle time parameters to calculate the start of the next scan cycle.

Monitor interval

The length of time used by the execution monitor in determining whether to add a forced idle time. The monitor interval is the sum of the execution time limit and the forced idle time that is calculated based on the maximum execution load percentage.

MPI

Multipoint interface: physical communication layer that can be used for S7 communication with STEP 7, S7-CPUs, and S7 applications.

MRP

Media Redundancy Protocol: Protocol for fault-tolerant networks

Non-deterministic behavior

Lack of predictability of execution time and response time associated with jitter.
(See also: "What Causes Jitter?")

NVRAM

Non-Volatile Random Access Memory: Non-volatile memory area

OP

Operator panel.

OPC

OLE for Process Control - OLE-based industry standard that enables vendor-independent access to industrial communication networks. OPC (OLE for Process Control) defines a standard communication interface for automation technology. OPC allows the user to access OLE (Object Linking and Embedding).

OLE is the component model of Microsoft. Components are the software objects or applications that provide their functionality for other applications. Communication via the OPC interface is based on COM/DCOM. In this case, the object is the process image.

OPC server

The OPC server provides the OPC client with numerous communication functions via industrial networks.

Organization block (OB)

Organization blocks (OBs) represent the interface between the operating system and the control program. Called by the operating system, they control cyclic and interrupt-driven program execution, startup behavior of the controller, and error handling.

PC station

Representation of a software-based virtual rack that defines a PC-based automation system. (See also: "What is a PC station?")

PDM

Process Device Manager: Tool for configuration, parameter assignment, commissioning, diagnostics and maintenance of intelligent field devices and field components

PG

Programming device

PG/OP communication

Communication between WinLC RTX and other S7 applications, such as programming devices, operator panels, and S7 controllers. WinLC RTX supports PROFIBUS and Industrial Ethernet for PG/OP communication.

PLC

Programmable logic controller - electronic control system. The PLC functions are stored in a program on the control device. The device configuration and wiring are therefore independent of the controller functions. The PLC is configured similar to a computer.
It consists of a CPU with memory, input and output devices, and an internal bus system.
The I/O and the programming language are oriented to control engineering requirements.

Priority

The priority of an application determines the order in which the operating system executes or interrupts an application in relation to the other applications that are running on the computer. An application with a higher priority interrupts the execution of an application with a lower priority. After the application with the higher priority finishes, the application with the lower priority resumes. A higher number indicates a higher priority.

Priority class

The priority class determines the order in which the controller executes the individual sections of the control program. Organization blocks (OBs) are subdivided into priority classes. Higher priority OBs interrupt lower priority OBs. The free scan cycle (OB 1) has the lowest priority. You can use STEP 7 to change the priority class for an OB. A higher number indicates a higher priority class.

PROFIBUS

Physical communications layer that can be used for PROFIBUS DP communication with I/O or S7 communication with STEP 7, S7 CPUs, and S7 applications.

PROFIBUS DP

Communications network protocol used to communicate to DP I/O.

PROFINET CBA

PROFINET Component-Based Automation

PROFINET IO

Communications network protocol used to communicate with PROFINET IO devices.

Restart method

The restart method determines which startup OB is executed whenever the controller changes from STOP mode to RUN mode. The startup OB allows you to initialize your control program and variables. The two restart methods are Cold Restart (OB 102) and Warm Restart (OB 100).

RTX

Real-time extensions: IntervalZero real-time extensions to the Windows Operating system extensions that allow processes to run in a real-time environment providing more deterministic execution and protection from Windows operating system crashes.

S7 communication

Communication between controllers on the network, hardware or software, using the S7 communication functions. (See also: "Communication Blocks")

S7 routing

Communication between S7 controllers, S7 applications, or PC stations across subnets through one or more network nodes acting as routers, configured with NetPro.

Scan cycle

The cycle includes writing to the outputs, reading the inputs, executing OB 1 and all other OBs, and satisfying the idle time requirement.

SNMP

Simple Network Management Protocol: Standard protocol for TCP/IP networks

SSL

System status list - STEP 7 saves write-protected information about the controller in the system status list (SSL), which is a set of sublists (can be accessed via SSL_ID).

Station Configuration Editor

Tool, accessible from the taskbar, for configuring the PC Station: for WinLC RTX this includes WinLC Properties, submodule assignments, and submodule diagnostics for some communication interfaces.

STEP 7 user program

Application program created with STEP 7 and downloaded to the controller for execution. It includes all organization blocks (such as OB 1 or OB 35) and the other logic blocks that they call, including functions (FCs), system functions (SFCs), function blocks (FBs), and system function blocks (SFBs).

Submodule

Communication interface in the PC that is designated for exclusive use by WinLC RTX. (See also: "What is a submodule?")

System function (SFC)

An SFC is a preprogrammed function that is integrated as a part of the operating system of the controller and is not downloaded as part of the control program. You can call the SFC in your control program. Like a function (FC), an SFC is a block "without memory".

System function block (SFB)

An SFB is a function block that is integrated as a part of the operating system of the controller and is not downloaded as part of the control program. Like a function block (FB), an SFB is a block "with memory". You must also create an instance data block (DB) for the SFB. The instance DB is then downloaded to the controller as part of the control program.

TCI

Tool Calling Interface: Integration of tools for configuration of distributed devices

TCP

Transmission Control Protocol: Enables transmission of data packets ("messages") if both nodes support RFC 1006.

Time synchronization

The ability to broadcast a system standard time from a single source to all devices within the system so that they can set their own clocks to the standard time.

Time synchronization service (WinAC time synchronization)

Software component of WinAC RTX that provides the capability to synchronize time between components in the PC Station. (See the documentation for the WinAC Time Synchronization Service.)

Virtual backplane bus

For PC-based controllers, the virtual backplane bus is a software-based, virtual "rack" that enables communication between the controller and other PC station components.
(See also: "What is a PC station?")

Waiting time

The wait time, or idle time, is the time that the controller is not using the CPU.
During this time, the operating system can run other applications.

Warm restart

The controller executes OB 100 before the start of the free scan cycle (OB 1). A warm restart resets the peripheral inputs (PI) and changes the peripheral outputs (PQ) to a pre-defined safe state (default is 0). The warm restart also saves the current value for the retentive memory areas for the memory bits (M), timers (T), counters (C), and data blocks (DBs).

Web server

The Web server is used to monitor the WinAC RTX via the Internet or via your company Intranet. This approach lets you carry out evaluations and diagnostics even at great distances. Messages and status information are visualized on HTML pages.

WinAC

Windows Automation Center: WinAC is a software package that includes various components, such as a PC-based controller and real-time extension. WinAC is the integration platform for all parts of the automation solution that run on the PC.

Windows Stop Error

Termination of the Windows operating system, resulting in a display on the monitor of the fatal error on a blue background. A Windows Stop Error is also known as a "blue screen".

WinLC

Windows Logic Controller: WinLC represents the software solution of a CPU - a PC-based controller. You can use WinLC to represent the functions of a CPU on a PC.

Index