SIEMENS

SIMATIC HMI

WinCC V7.2
WinCC: Communication

System Manual

Print of the Online Help

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
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

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, 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 complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of 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.

Table of contents

Process communication

1.1 Communication Basics

Introduction

Communication is defined as the exchange of data between two communication partners.

Communication

Communication partners can be any component of a network that is in a position to communication with others and to exchange data. In the WinCC, these can be central and communication modules in the automation system (AS) as well as communication processors in the PC.

The transferred data between communication partners can serve many different purposes. In the case of WinCC, these may be:

- Controlling a process
- Calling data from a process
- Indicating unexpected states in the process
- Process data archiving
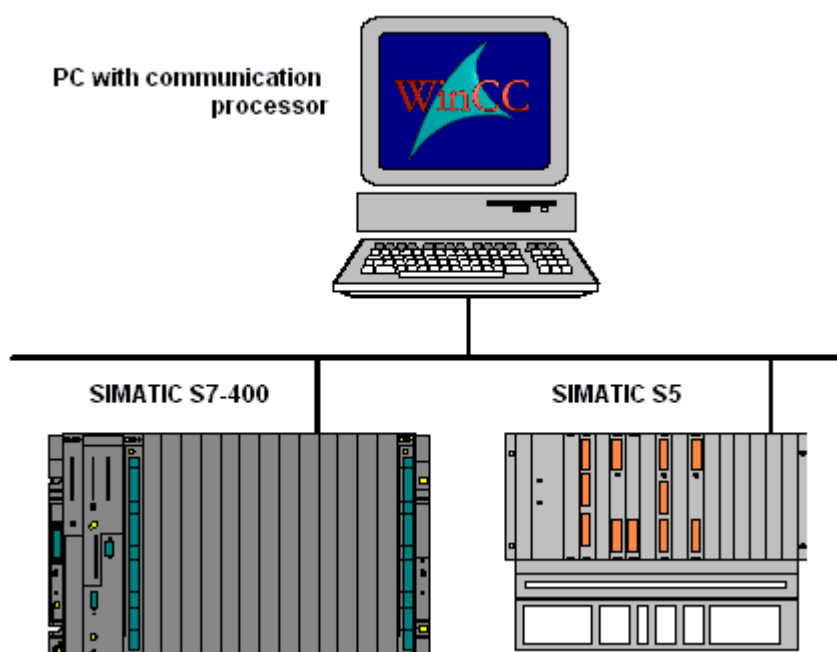
1.2 Basic Rules for Configuring Connections

Acquisition cycle and update time

The acquisition cycles for the tags defined in the configuration software are major factors for the achievable update times.

The update time is the sum of the acquisition cycle, the transmission time and the processing time.

To achieve optimum update times, remember the following points during configuration:

- Optimize the maximum and minimum size of the data areas.

- Define data areas that belong together as belonging together. If you set up one large area instead of multiple small areas, it improves the update time.

- Acquisition cycles that are too small decrease performance. Set the acquisition cycle according to the rate of change of the process values. Take the temperature of an oven for example, it changes much more slowly than the speed of an electrical drive.

- Put the tags of an alarm or a screen in one data area without gaps.

- Changes in the controller can only be detected reliably if these are available for at least one acquisition cycle.

- Set the transmission rate to the highest possible value for error-free transmission.

Images

The refresh rate of screens is determined by the type and volume of data to be visualized.

In the interest of short update times, ensure that you only configure short acquisition times for objects that require fast updates.

Curves

When using bit-triggered curves, if the group bit is set in the "Curve transfer area", all curves for which the bit is set in this area are updated on the WinCC station. It resets the bits in the next cycle.

Only after all bits have been reset in the WinCC station may the group bit be set again in the PLC program.

1.3 WinCC process communication

1.3.1 WinCC process communication

Introduction

You can access process tags (external tags) in an automation system from WinCC. Before you configure the process link in WinCC however, you should use a checklist to check whether the following prerequisites have been met:

● The automation system must be equipped with a communication interface supported by a communication driver in WinCC.

● This interface must be configured in the automation system so that the controller program can access the interface with the communication calls. The configuration parameters for the communication hardware must be known.

● The addresses of the tags that WinCC should access must be known. Note that the addresses depend on the automation system.

● The respective communication hardware (communication processor, standard I/O port COMx, ...) must be installed in the WinCC system. In order to install this hardware, the supplied operating system driver (hardware driver) must also have been installed previously. The settings for the hardware and software of the communication processor must be known.

● Depending on the communication processor used in the WinCC system, more settings may have to be made. When using industrial Ethernet or PROFIBUS for example, a local database must be created. This connection parameter also has to be known.

For operation in runtime, a physical connection must also exist between WinCC and the AS, so that you can access the external tags.

1.3.2 Principle of WinCC communication

Introduction

WinCC manages its tags centrally using so-called tag management. All of data and tags created in the project and stored in the project database are captured and management in runtime by WinCC.
All applications, such as e.g. Graphics Runtime, Alarm Logging Runtime or Tag Logging Runtime (Global Script), must request the data in the for of WinCC tags from tag management.

Communication between WinCC and Automation systems (AS)

Communication in the industrial communication with WinCC means that information is exchanged with tags and process values. To capture the process values, the WinCC communication driver sends request telegrams to the AS. This in turn sends the requested process values back to WinCC in corresponding response telegrams.
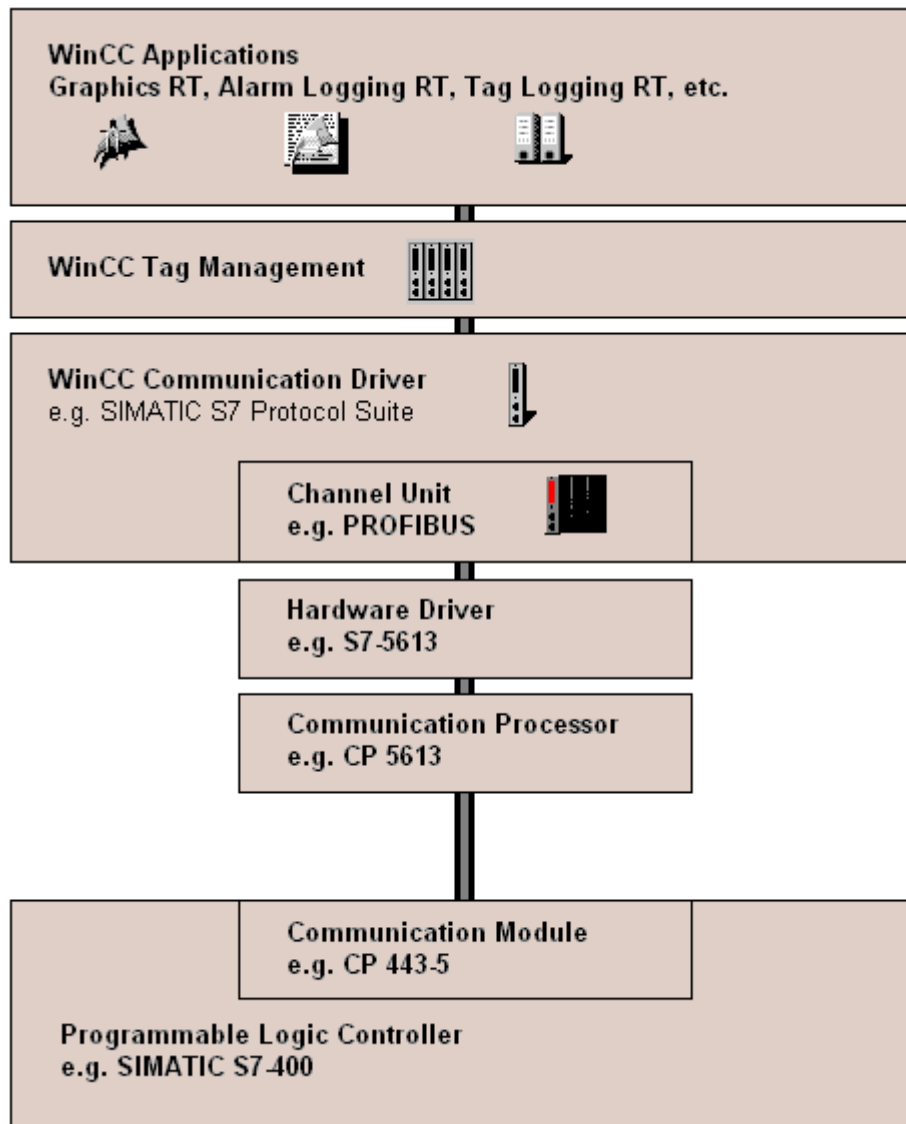
A physical connection must then exist between WinCC and the AS. The properties of this connection, such as e.g. transfer medium and communication network, define the conditions for communication and are required for configuring the communication in WinCC.

Communication driver

A communication driver is a software component that establishes a connection between an AS and the tag management of WinCC, which enables the supply of WinCC tags with process values. In WinCC, there is a number of communication drivers for connecting various AS with different bus systems.

Every communication driver can only be bound into a WinCC project one time.

Communication drivers under WinCC are also called "Channels" and have the file extension "*.chn". All of the communication drivers installed on the computer are located in subdirectory "\bin" in the WinCC installation directory.

A communication driver has different channel units for different communication networks.

Channel Unit

Every channel unit serves as an interface with exactly one underlying hardware driver and therefore to exactly one communication processor in the PC. Every channel unit used must therefore be allocated with the respective communication processor.

For some channel units, an additional configuration is performed in the so-called system parameters. For channel units that work on the transport layer (Layer 4) of the OSI model, the transport parameters are also defined.

Connection (logical)

If WinCC and the AS are correctly connected, physically, then a communication driver and a corresponding channel unit are required in WinCC in order to create or configure a (logical) connection with the AS. The data exchange will take place via this connection in Runtime.
In WinCC, there is a connection of a configured, logical allocation of two communication partners for executing a certain communication service. Every connection has two end points that also contain necessary information for addressing the communication partner and other attributes for establishing the connection.
A connection is configured under a channel unit with your specific connection parameters. A number of connections can also be created under one channel unit, depending on the communication driver.

1.3.3 External tags

1.3.3.1 External tags

Introduction

In order to obtain certain data of an AS, WinCC tags are required. These tags that affect the connection to an AS are designated as external tags. Other tags that have no process connection are designated as internal tags.

Data Type and Type Conversion

In configuring external tags, besides tag names you must define a data type and with some data types a type conversion is required:

The data type determines the data format in WinCC. Along with the type conversion, the conversion from AS format to the WinCC format is defined. The type conversion applies for both transfer directions:

- In the AS: e.g. for certain functions (such as timer values / BCD displays) or with the information to be addressed (e.g. byte, word addresses in data block or I/O area).

- In WinCC: e.g. for analog value processing or calculations.

In practice, normally the AS data format is normally defined. The following possibilities are available for selecting the WinCC format:

- The WinCC data format can match the AS format. This is done by selecting a type conversion that uses the same formats on both sides and considers the leading sign independent of the WinCC data type, e.g. "WordToSignedWord". If this cannot be achieved with the selected data type, it must be changed in WinCC.

- The WinCC format is set up in accordance of the value processing in WinCC.

When selecting the data type and the type conversion, if required, the following points are important:

- Leading sign: Should it be considered for the adaptation? Can negative tag values also occur during operation? (such as e.g. with control differences as percentage)

- Value Range: Are tag values that occur during operation in the value range of both formats or can a possible overflow of the value be expected in WinCC or the AS ? If an overflow occurs, a value cannot be displayed on the other side or it can lead to faults in subsequent processing.

- Different type conversions with the same value range: It is possible that different type conversions of a data type have the same value range. E.g. "ByteToUnsignedDword" and "ByteToUnsignedWord" with value range [0...127]. In this case, check which format is defined for the data from the AS and whether this format wastes resources unnecessarily because of over-dimensioning. (E.g. DWord instead of Word).

If the value range required in the AS is not covered by the selected type conversion, you must change the data type in WinCC.

Note

Communication to the automation system can be disrupted when a process tag is configured incorrectly, for example, due to an address error.

WinCC data types and type conversion

The following table shows which WinCC data types support a type conversion.

Data type	Type conversion
Binary tag	No
Unsigned 8-bit value	Yes
Signed 8-bit value	Yes
Unsigned 16-bit value	Yes
Signed 16-bit value	Yes

Data type	Type conversion
Unsigned 32-bit value	Yes
Signed 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Floating-point number 64-bit IEEE 754	Yes
Text tag 8-bit character set	No
Text tag 16-bit character set	No
Raw data type	No

Note

Not that with a type conversion, the data sent by AS can be interpreted by WinCC within the selected format adaptation. If the data cannot be interpreted by WinCC, a fault entry occurs in the file "WinCC_sys_0x.log" in directory "..\Siemens\WinCC\Diagnose".

Linear scaling of numerical tag types

A linear scaling can be performed for numerical data types. The value range for a variable existing in the process can be represented linearly within a certain value range of a WinCC tag.

The process can e.g. request the definition of a set value in the unit [bar] but this value should be defined by the user in [mbar] in WinCC. With linear scaling, the value range in the process [0...1] can be converted to value range [0...1000] of the WinCC tags.

Length definition for text tags

A length definition is required for tags of data types "Text tag 8-bit character set" and "Text tag 16-bit character set". A text tag that is required to accommodate 10 characters later must be ten long for the "8-bit character set" and 20 long for the "16-bit character set".

Addressing in the automation system

WinCC tags are assigned to a data range in the AS. These must be addressed in the AS in a certain way. The addressing type depends on the type of communication partner.

1.3.3.2   How to Create a New Connection

Introduction

External tags can only be created on the basis of a connection to an AS. If the required connection does not exist, it must be created first.

Requirements

- The required communication processor and the respective hardware driver are installed.
- The desired communication driver is also installed, e.g. "SIMATIC S7 Protocol Suite".

Procedure

1. Expand the view of the communication drivers in "Tag management".

2. Select the desired channel unit, e.g. "PROFIBUS".

3. Select the entry "New Connection..." in the pop-up menu of the channel unit.
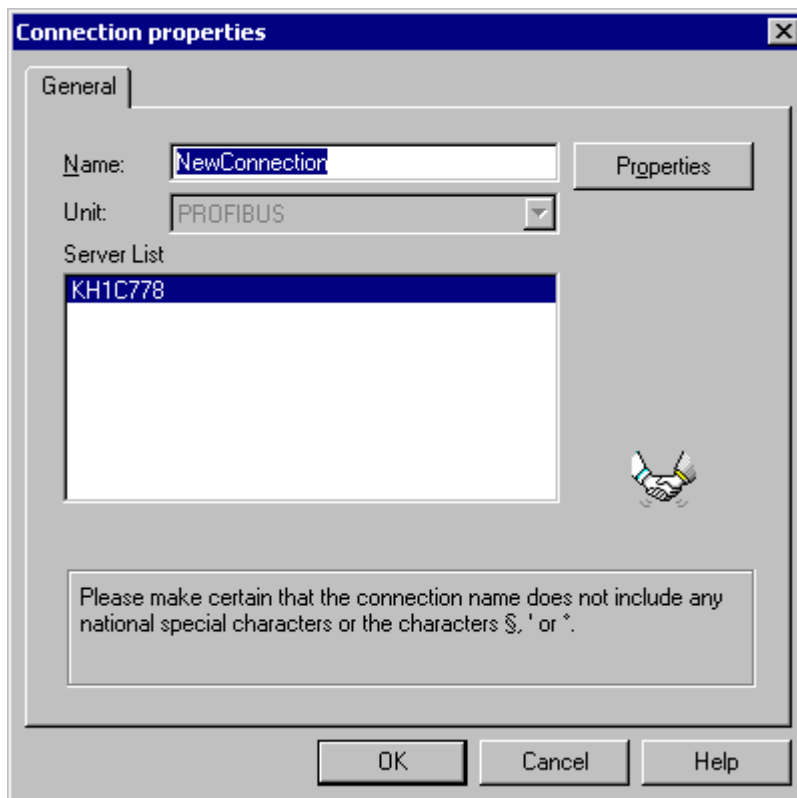
4. Give the connection a unique name within the project in the field "Name" on the General tab.

5. Click the "Properties" button to open the "Connection Parameters" dialog. Define the required parameter for this connection. More information can be found under Help / Documentation for the respective channel.

6. Close all dialogs using the "OK" button.

1.3.3.3   An external tag is configured as follows

Introduction

The procedures for creating a tag is similar for almost all data types.

For some data types however, special settings are required (steps 5 - 7).

Requirements

- The required communication processor and the hardware driver are installed.
- The desired communication driver is installed, e.g. "SIMATIC S7 Protocol Suite".
- A connection is already created based on a channel unit (e.g. "PROFIBUS").

Procedure

1. Choose the "New Tag..." option from the shortcut menu for the desired connection. The "Tag Properties" dialog opens.
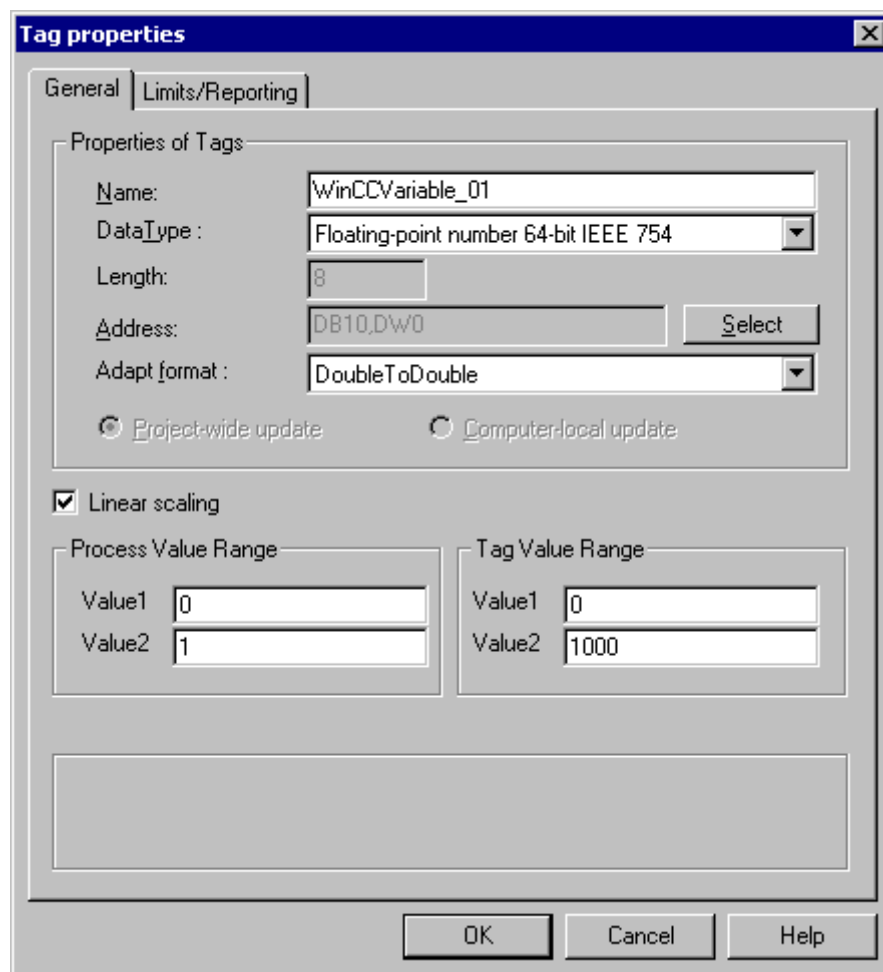
2. Define a variable name that is unique in the WinCC project, e.g. WinCCTag_01", in field "Name" on the General tab.

3. Define the data type for the tag in the "Data type" field, e.g. "Floating Point Number 64 bit IEEE 754".

4. Click on the "Select" button to open dialog "Tag properties" for defining the address range of the tags in the AS.
    With channels that do not support bit-/byte-access with binary or 8 bit tags, first the dialog "Bit-/Byte-Tag" and then dialog "Tag properties" will also be shown.
    More information is located under "Principle of the BinWrite-Mechanism".
    Close the dialog "Bit-/Byte-Variable" or "Tag properties" with the "OK" button.

5. With numerical tags, WinCC suggests a format adaptation in the "Format adaptation" field. Select another format adaptation if necessary. The display is in sequence "X to Y", whereby X = WinCC format and Y = AS format, e.g. "DoubleToDouble".

6. Activate the checkbox "Linear scaling" to scale a numerical tag linearly. Enter the upper and lower limits for the "Process value range" (in AS) and "Tag value range" (in WinCC).

7. The "Length" field is activated for a text tag. Enter the length of the text tag in characters here.

8. Close all dialogs using the "OK" button.

1.3.3.4   Format adaptation sorted by WinCC data type

Introduction

When configuring external tags, another format adaptation must be done for all numeric data types.

The data type determines the data format on the WinCC side. The format adaptation also defines the conversion from WinCC format to the AS format. The definition applies for both transfer directions.

Choose the required WinCC data type in the following selection box. You are then provided with a list of the respective possible format adaptations and value ranges in the table below.

WinCC Data Type

Table 1-1   Signed 8-bit value

Format adaptation "Signed 8-bit value"	Value range
CharToUnsignedByte	0...127
CharToUnsignedWord	0...127
CharToUnsignedDword	0...127
CharToSignedByte	-128...+127 (no conversion)
CharToSignedWord	-128...+127
CharToSignedDword	-128...+127
CharToMSBByte	-128...+127
CharToMSBWord	-128...+127
CharToMSBDword	-128...+127
CharToBCDByte	0...99
CharToBCDWord	0...127
CharToBCDDword	0...127

Format adaptation "Signed 8-bit value"	Value range
CharToSignedBCDByte	-9...+9
CharToSignedBCDWord	-128...+127
CharToSignedBCDDword	-128...+127
CharToExtSignedBCDByte	-79...+79
CharToExtSignedBCDWord	-128...+127
CharToExtSignedBCDDword	-128...+127
CharToAikenByte	0...99
CharToAikenWord	0...127
CharToAikenDword	0...127
CharToSignedAikenByte	-9...+9
CharToSignedAikenWord	-128...+127
CharToSignedAikenDword	-128...+127
CharToExcessByte	0...99
CharToExcessWord	0...127
CharToExcessDword	0...127
CharToSignedExcessByte	-9...+9
CharToSignedExcessWord	-128...+127
CharToSignedExcessDword	-128...+127

Table 1-2   Unsigned 8-bit value

Format adaptation "Unsigned 8-bit value"	Value range
ByteToUnsignedByte	0...255 (no conversion)
ByteToUnsignedWord	0...255
ByteToUnsignedDword	0...255
ByteToSignedByte	0...127
ByteToSignedWord	0...255
ByteToSignedDword	0...255
ByteToBCDByte	0...99
ByteToBCDWord	0...255
ByteToBCDDword	0...255
ByteToAikenByte	0...99
ByteToAikenWord	0...255
ByteToAikenDword	0...255
ByteToExcessByte	0...99
ByteToExcessWord	0...255
ByteToExcessDword	0...255

Table 1-3   Signed 16-bit value

Format adaptation "Signed 16-bit value"	Value range
ShortToUnsignedByte	0...255
ShortToUnsignedWord	0...32767
ShortToUnsignedDword	0...32767

Format adaptation "Signed 16-bit value"	Value range
ShortToSignedByte	-128...+127
ShortToSignedWord	-32768...+32767 (no conversion)
ShortToSignedDword	-32768...+32767
ShortToMSBByte	-127...+127
ShortToMSBWord	-32767...+32767
ShortToMSBDword	-32768...+32767
ShortToBCDByte	0...99
ShortToBCDWord	0...9999
ShortToBCDDword	0...32767
ShortToSignedBCDByte	-9...+9
ShortToSignedBCDWord	-999...+999
ShortToSignedBCDDword	-32768...+32767
ShortToExtSignedBCDByte	-79...+79
ShortToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDDword	-32768...+32767
ShortToAikenByte	0...99
ShortToAikenWord	0...9999
ShortToAikenDword	0...32767
ShortToSignedAikenByte	-9...+9
ShortToSignedAikenWord	-999...+999
ShortToSignedAikenDword	-32768...+32767
ShortToExcessByte	0...99
ShortToExcessWord	0...9999
ShortToExcessDword	0...32767
ShortToSignedExcessByte	-9...+9
ShortToSignedExcessWord	-999...+999
ShortToSignedExcessDword	-32768...+32767

Table 1-4   Unsigned 16-bit value

Format adaptation "Unsigned 16-bit value"	Value range
WordToUnsignedWord	0...65535 (no conversion)
WordToUnsignedByte	0...255
WordToUnsignedDword	0...65535
WordToSignedByte	0...127
WordToSignedWord	0...32767
WordToSignedDword	0...65535
WordToBCDByte	0...99
WordToBCDWord	0...9999
WordToBCDDword	0...65535
WordToAikenByte	0...99
WordToAikenWord	0...9999
WordToAikenDword	0...65535

Format adaptation "Unsigned 16-bit value"	Value range
WordToExcessByte	0...99
WordToExcessWord	0...9999
WordToExcessDword	0...65535
WordToSimaticCounter	0...999
WordToSimaticBCDCounter	0...999

Table 1-5   Signed 32-bit value

Format adaptation "Signed 32-bit value"	Value range
LongToSignedDword	-2147483647...+2147483647 (no conversion)
LongToUnsignedByte	0...255
LongToUnsignedWord	0...65535
LongToUnsignedDword	0...2147483647
LongToSignedByte	-128...+127
LongToSignedWord	-32768...+32767
LongToMSBByte	-127...+127
LongToMSBWord	-32767...+32767
LongToMSBDword	-2147483647...+2147483647
LongToBCDByte	0...99
LongToBCDWord	0...9999
LongToBCDDword	0...99999999
LongToSignedBCDByte	-9...+9
LongToSignedBCDWord	-999...+999
LongToSignedBCDDword	-9999999...+9999999
LongToExtSignedBCDByte	-79..+79
LongToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDDword	-79999999...+79999999
LongToAikenByte	0...99
LongToAikenWord	0...9999
LongToAikenDword	0...99999999
LongToSignedAikenByte	-9...+9
LongToSignedAikenWord	-999...+999
LongToSignedAikenDword	-9999999...+9999999
LongToExcessByte	0...99
LongToExcessWord	0...9999
LongToExcessDword	0...99999999
LongToSignedExcessByte	-9...+9
LongToSignedExcessWord	-999...+999
LongToSignedExcessDword	-9999999...+9999999
LongToSimaticTimer	10...9990000
LongToSimaticBCDTimer	10...9990000

Table 1-6   Unsigned 32-bit value

Format adaptation "Unsigned 32-bit value"	Value range
DwordToUnsignedDword	0...4294967295 (no conversion)
DwordToUnsignedByte	0...255
DwordToUnsignedWord	0...65535
DwordToSignedByte	0...127
DwordToSignedWord	0...32767
DwordToSignedDword	0...2147483647
DwordToBCDByte	0...99
DwordToBCDWord	0...9999
DwordToBCDDword	0...99999999
DwordToAikenByte	0...99
DwordToAikenWord	0...9999
DwordToAikenDword	0...99999999
DwordToExcessByte	0...99
DwordToExcessWord	0...9999
DwordToExcessDword	0...99999999
DwordToSimaticTimer	10...9990000
DwordToSimaticBCDTimer	10...9990000

Table 1-7   Floating-point number 32-bit IEEE 754

Format adaptation "Floating-point number 32-bit IEEE 754"	Value range
FloatToFloat	+-3.402823e+38 (no conversion)
FloatToUnsignedByte	0...255
FloatToUnsignedWord	0...65535
FloatToUnsignedDword	0 to 4.294967e+09
FloatToSignedByte	-128...+127
FloatToSignedWord	-32768...+32767
FloatToSignedDword	-2.147483e+09 to +2.147483e+09
FloatToDouble	+-3.402823e+38
FloatToMSBByte	-127...+127
FloatToMSBWord	-32767...+32767
FloatToMSBDword	-2.147483e+09 to +2.147483e+09
FloatToBCDByte	0...99
FloatToBCDWord	0...9999
FloatToBCDDword	0 to 9.999999e+07
FloatToSignedBCDByte	-9...+9
FloatToSignedBCDWord	-999...+999
FloatToSignedBCDDword	-9999999...+9999999
FloatToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDDword	-7.999999e+07 to +7.999999e+07
FloatToAikenByte	0...99

Format adaptation "Floating-point number 32-bit IEEE 754"	Value range
FloatToAikenWord	0...9999
FloatToAikenDword	0 to 9.999999e+07
FloatToSignedAikenByte	-9...+9
FloatToSignedAikenWord	-999...+999
FloatToSignedAikenDword	-9999999...+9999999
FloatToExcessByte	0...99
FloatToExcessWord	0...9999
FloatToExcessDword	0 to 9.999999e+07
FloatToSignedExcessByte	-9...+9
FloatToSignedExcessWord	-999...+999
FloatToSignedExcessDword	-9999999...+9999999
FloatToS5Timer	10...9990000
FloatToS5Float	+-1.701411e+38

Table 1-8　　Floating-point number 64-bit IEEE 754

Format adaptation "Floating-point number 64-bit IEEE 754"	Value range
DoubleToDouble	+-1.79769313486231e+308 (no conversion)
DoubleToUnsignedByte	0...255
DoubleToUnsignedWord	0...65535
DoubleToUnsignedDword	0...4294967295
DoubleToSignedByte	-128...+127
DoubleToSignedWord	-32768...+32767
DoubleToSignedDword	-2147483647...+2147483647
DoubleToFloat	+-3.402823e+38
DoubleToMSBByte	-127...+127
DoubleToMSBWord	-32767...+32767
DoubleToMSBDword	-2147483647...+2147483647
DoubleToBCDByte	0...99
DoubleToBCDWord	0...9999
DoubleToBCDDword	0...99999999
DoubleToSignedBCDByte	-9...+9
DoubleToSignedBCDWord	-999...+999
DoubleToSignedBCDDword	-9999999...+9999999
DoubleToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDWord	-7999...+7999
DoubleToExtSignedBCDDword	-79999999...+79999999
DoubleToAikenByte	0...99
DoubleToAikenWord	0...9999
DoubleToAikenDword	0...99999999
DoubleToSignedAikenByte	-9...+9
DoubleToSignedAikenWord	-999...+999
DoubleToSignedAikenDword	-9999999...+9999999

Format adaptation "Floating-point number 64-bit IEEE 754"	Value range
DoubleToExcessByte	0...99
DoubleToExcessWord	0...9999
DoubleToExcessDword	0...99999999
DoubleToSignedExcessByte	-9...+9
DoubleToSignedExcessWord	-999...+999
DoubleToSignedExcessDword	-9999999...+9999999
DoubleToS5Timer	10...9990000
DoubleToS5Float	+-1.701411e+38

1.3.3.5   Format adaptation sorted by AS data type

Introduction

When configuring external tags, another format adaptation must be done for all numeric data types.

The data type determines the data format on the WinCC side. The format adaptation also defines the conversion from WinCC format to the AS format. The definition applies for both transfer directions.

Choose the required AS data type in the following selection box. You are then provided with a list of the respective possible format adaptations and respective value ranges in the table below.

AS data type

Format adaptation and value range

Table 1-9   AikenByte

Format adaptation "AikenByte"	Value range
ByteToAikenByte	0...99
CharToAikenByte	0...99
DoubleToAikenByte	0...99
DwordToAikenByte	0...99
FloatToAikenByte	0...99
LongToAikenByte	0...99
ShortToAikenByte	0...99
WordToAikenByte	0...99

Table 1-10   AikenWord

Format adaptation "AikenWord"	Value range
ByteToAikenWord	0...255
CharToAikenWord	0...127
DoubleToAikenWord	0...9999
DwordToAikenWord	0...9999
FloatToAikenWord	0...9999
LongToAikenWord	0...9999
ShortToAikenWord	0...9999
WordToAikenWord	0...9999

Table 1-11   AikenDWord

Format adaptation "AikenDWord"	Value range
ByteToAikenDword	0...255
CharToAikenDword	0...127
DoubleToAikenDword	0...99999999
DwordToAikenDword	0...99999999
FloatToAikenDword	0 to 9.999999e+07
LongToAikenDword	0...99999999
ShortToAikenDword	0...32767
WordToAikenDword	0...65535

Table 1-12   BCDByte

Format adaptation "BCDByte"	Value range
ByteToBCDByte	0...99
CharToBCDByte	0...99
DoubleToBCDByte	0...99
DwordToBCDByte	0...99
FloatToBCDByte	0...99
LongToBCDByte	0...99
ShortToBCDByte	0...99
WordToBCDByte	0...99

Table 1-13   BCDWord

Format adaptation "BCDWord"	Value range
ByteToBCDWord	0...255
CharToBCDWord	0...127
DoubleToBCDWord	0...9999
DwordToBCDWord	0...9999
FloatToBCDWord	0...9999
LongToBCDWord	0...9999

Format adaptation "BCDWord"	Value range
ShortToBCDWord	0...9999
WordToBCDWord	0...9999

Table 1-14   BCDDWord

Format adaptation "BCDDWord"	Value range
ByteToBCDDword	0...255
CharToBCDDword	0...127
DoubleToBCDDword	0...99999999
DwordToBCDDword	0...99999999
FloatToBCDDword	0 to 9.999999e+07
LongToBCDDword	0...99999999
ShortToBCDDword	0...32767
WordToBCDDword	0...65535

Table 1-15   Double

Format adaptation "Double"	Value range
DoubleToDouble	+-1.79769313486231e+308 (no conversion)
FloatToDouble	+-3.402823e+38

Table 1-16   ExcessByte

Format adaptation "ExcessByte"	Value range
ByteToExcessByte	0...99
CharToExcessByte	0...99
DoubleToExcessByte	0...99
DwordToExcessByte	0...99
FloatToExcessByte	0...99
LongToExcessByte	0...99
ShortToExcessByte	0...99
WordToExcessByte	0...99

Table 1-17   ExcessWord

Format adaptation "ExcessWord"	Value range
ByteToExcessWord	0...255
CharToExcessWord	0...127
DoubleToExcessWord	0...9999
DwordToExcessWord	0...9999
FloatToExcessWord	0...9999
LongToExcessWord	0...9999
ShortToExcessWord	0...9999
WordToExcessWord	0...9999

Table 1-18   ExcessDWord

Format adaptation "ExcessDWord"	Value range
ByteToExcessDword	0...255
CharToExcessDword	0...127
DoubleToExcessDword	0...99999999
DwordToExcessDword	0...99999999
FloatToExcessDword	0 to 9.999999e+07
LongToExcessDword	0...99999999
ShortToExcessDword	0...32767
WordToExcessDword	0...65535

Table 1-19   ExtSignedBCDByte

Format adaptation "ExtSignedBCDByte"	Value range
CharToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDByte	-79...+79
LongToExtSignedBCDByte	-79..+79
ShortToExtSignedBCDByte	-79...+79

Table 1-20   ExtSignedBCDWord

Format adaptation "ExtSignedBCDWord"	Value range
CharToExtSignedBCDWord	-128...+127
DoubleToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDWord	-7999...+7999

Table 1-21   ExtSignedBCDDWord

Format adaptation "ExtSignedBCDDWord"	Value range
CharToExtSignedBCDDword	-128...+127
DoubleToExtSignedBCDDword	-79999999...+79999999
FloatToExtSignedBCDDword	-7.999999e+07 to +7.999999e+07
LongToExtSignedBCDDword	-79999999...+79999999
ShortToExtSignedBCDDword	-32768...+32767

Table 1-22   Float

Format adaptation "Float"	Value range
DoubleToFloat	+-3.402823e+38
FloatToFloat	+-3.402823e+38 (no conversion)

Table 1-23   MSBByte

Format adaptation "MSBByte"	Value range
CharToMSBByte	-128...+127
DoubleToMSBByte	-127...+127
FloatToMSBByte	-127...+127
LongToMSBByte	-127...+127
ShortToMSBByte	-127...+127

Table 1-24   MSBWord

Format adaptation "MSBWord"	Value range
CharToMSBWord	-128...+127
DoubleToMSBWord	-32767...+32767
FloatToMSBWord	-32767...+32767
LongToMSBWord	-32767...+32767
ShortToMSBWord	-32767...+32767

Table 1-25   MSBDWord

Format adaptation "MSBDWord"	Value range
CharToMSBDword	-128...+127
DoubleToMSBDword	-2147483647...+2147483647
FloatToMSBDword	-2.147483e+09 to +2.147483e+09
LongToMSBDword	-2147483647...+2147483647
ShortToMSBDword	-32768...+32767

Table 1-26   S5Float

Format adaptation "S5Float"	Value range
DoubleToS5Float	+-1.701411e+38
FloatToS5Float	+-1.701411e+38

Table 1-27   S5Timer

Format adaptation "S5Timer"	Value range
DoubleToS5Timer	10...9990000
FloatToS5Timer	10...9990000

Table 1-28   SignedByte

Format adaptation "SignedByte"	Value range
ByteToSignedByte	0...127
CharToSignedByte	-128...+127 (no conversion)
DoubleToSignedByte	-128...+127
DwordToSignedByte	0...127
FloatToSignedByte	-128...+127

Format adaptation "SignedByte"	Value range
LongToSignedByte	-128...+127
ShortToSignedByte	-128...+127
WordToSignedByte	0...127

Table 1-29   SignedWord

Format adaptation "SignedWord"	Value range
ByteToSignedWord	0...255
CharToSignedWord	-128...+127
DoubleToSignedWord	-32768...+32767
DwordToSignedWord	0...32767
FloatToSignedWord	-32768...+32767
LongToSignedWord	-32768...+32767
ShortToSignedWord	-32768...+32767 (no conversion)
WordToSignedWord	0...32767

Table 1-30   SignedDWord

Format adaptation "SignedDWord"	Value range
ByteToSignedDword	0...255
CharToSignedDword	-128...+127
DoubleToSignedDword	-2147483647...+2147483647
DwordToSignedDword	0...2147483647
FloatToSignedDword	-2.147483e+09 to +2.147483e+09
LongToSignedDword	-2147483647...+2147483647 (no conversion)
ShortToSignedDword	-32768...+32767
WordToSignedDword	0...65535

Table 1-31   SignedAikenByte

Format adaptation "SignedAikenByte"	Value range
CharToSignedAikenByte	-9...+9
DoubleToSignedAikenByte	-9...+9
FloatToSignedAikenByte	-9...+9
LongToSignedAikenByte	-9...+9
ShortToSignedAikenByte	-9...+9

Table 1-32   SignedAikenWord

Format adaptation "SignedAikenWord"	Value range
CharToSignedAikenWord	-128...+127
DoubleToSignedAikenWord	-999...+999
FloatToSignedAikenWord	-999...+999
LongToSignedAikenWord	-999...+999
ShortToSignedAikenWord	-999...+999

Table 1-33   SignedAikenDWord

Format adaptation "SignedAikenDWord"	Value range
CharToSignedAikenDword	-128...+127
DoubleToSignedAikenDword	-9999999...+9999999
FloatToSignedAikenDword	-9999999...+9999999
LongToSignedAikenDword	-9999999...+9999999
ShortToSignedAikenDword	-32768...+32767

Table 1-34   SignedBCDByte

Format adaptation "SignedBCDByte"	Value range
CharToSignedBCDByte	-9...+9
DoubleToSignedBCDByte	-9...+9
FloatToSignedBCDByte	-9...+9
LongToSignedBCDByte	-9...+9
ShortToSignedBCDByte	-9...+9

Table 1-35   SignedBCDWord

Format adaptation "SignedBCDWord"	Value range
CharToSignedBCDWord	-128...+127
DoubleToSignedBCDWord	-999...+999
FloatToSignedBCDWord	-999...+999
LongToSignedBCDWord	-999...+999
ShortToSignedBCDWord	-999...+999

Table 1-36   SignedBCDDWord

Format adaptation "SignedBCDDWord"	Value range
CharToSignedBCDDword	-128...+127
DoubleToSignedBCDDword	-9999999...+9999999
FloatToSignedBCDDword	-9999999...+9999999
LongToSignedBCDDword	-9999999...+9999999
ShortToSignedBCDDword	-32768...+32767

Table 1-37   SignedExcessByte

Format adaptation "SignedExcessByte"	Value range
CharToSignedExcessByte	-9...+9
DoubleToSignedExcessByte	-9...+9
FloatToSignedExcessByte	-9...+9
LongToSignedExcessByte	-9...+9
ShortToSignedExcessByte	-9...+9

Table 1-38   SignedExcessWord

Format adaptation "SignedExcessWord"	Value range
CharToSignedExcessWord	-128...+127
DoubleToSignedExcessWord	-999...+999
FloatToSignedExcessWord	-999...+999
LongToSignedExcessWord	-999...+999
ShortToSignedExcessWord	-999...+999

Table 1-39   SignedExcessDWord

Format adaptation "SignedExcessDWord"	Value range
CharToSignedExcessDword	-128...+127
DoubleToSignedExcessDword	-9999999...+9999999
FloatToSignedExcessDword	-9999999...+9999999
LongToSignedExcessDword	-9999999...+9999999
ShortToSignedExcessDword	-32768...+32767

Table 1-40   SimaticCounter

Format adaptation "SimaticCounter"	Value range
WordToSimaticCounter	0...999

Table 1-41   SimaticBCDCounter

Format adaptation "SimaticBCDCounter"	Value range
WordToSimaticBCDCounter	0...999

Table 1-42   SimaticTimer

Format adaptation "SimaticTimer"	Value range
DwordToSimaticTimer	10...9990000
LongToSimaticTimer	10...9990000

Table 1-43   SimaticBCDTimer

Format adaptation "SimaticBCDTimer"	Value range
DwordToSimaticBCDTimer	10...9990000
LongToSimaticBCDTimer	10...9990000

Table 1-44   UnsignedByte

Format adaptation "UnsignedByte"	Value range
ByteToUnsignedByte	0...255 (no conversion)
CharToUnsignedByte	0...127
DoubleToUnsignedByte	0...255
DwordToUnsignedByte	0...255

Format adaptation "UnsignedByte"	Value range
FloatToUnsignedByte	0...255
LongToUnsignedByte	0...255
ShortToUnsignedByte	0...255
WordToUnsignedByte	0...255

Table 1-45   UnsignedWord

Format adaptation "UnsignedWord"	Value range
ByteToUnsignedWord	0...255
CharToUnsignedWord	0...127
DoubleToUnsignedWord	0...65535
DwordToUnsignedWord	0...65535
FloatToUnsignedWord	0...65535
LongToUnsignedWord	0...65535
ShortToUnsignedWord	0...32767
WordToUnsignedWord	0...65535 (no conversion)

Table 1-46   UnsignedDWord

Format adaptation "UnsignedDWord"	Value range
ByteToUnsignedDword	0...255
CharToUnsignedDword	0...127
DoubleToUnsignedDword	0...4294967295
DwordToUnsignedDword	0...4294967295 (no conversion)
FloatToUnsignedDword	0 to 4.294967e+09
LongToUnsignedDword	0...2147483647
ShortToUnsignedDword	0...32767
WordToUnsignedDword	0...65535

1.3.3.6   Principle of the BinWrite-Mechanism

Introduction

In WinCC, not all communication drivers and their channel units support the direct bit-wise or byte-wise access (short: Bit-/Byte-access) to address ranges in a connected automation system. Instead, they use the BinWrite mechanism.

Bit-/Byte-access

With channel units of communication drivers with bit-/byte-access, the desired bit or byte can be read and written directly.

In the following figure, a bit x is allocated the value = 1 via direct bit-/byte-access.

Example:
Write bit x = 1
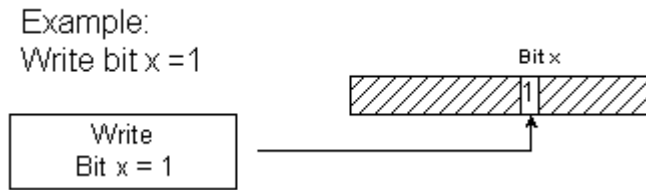
BinWrite Mechanism

The following communication drivers do not support bit-/byte-access and instead use the BinWrite mechanism for the respective channel units:

- Modbus Serial

- SIMATIC S5 Ethernet Layer 4

- SIMATIC S5 Programmers Port AS511

- SIMATIC S5 Serial 3964R

- SIMATIC TI Ethernet Layer 4

- SIMATIC TI Serial

To write a bit or byte, the channel unit first reads the entire data word with the BinWrite mechanism. The data to be addressed is then changed in the word that is read. Then, instead of just the changed bit or byte, the entire (!) word is written back.

In the following figure, a bit x is allocated the value = 1 via the BinWrite mechanism.
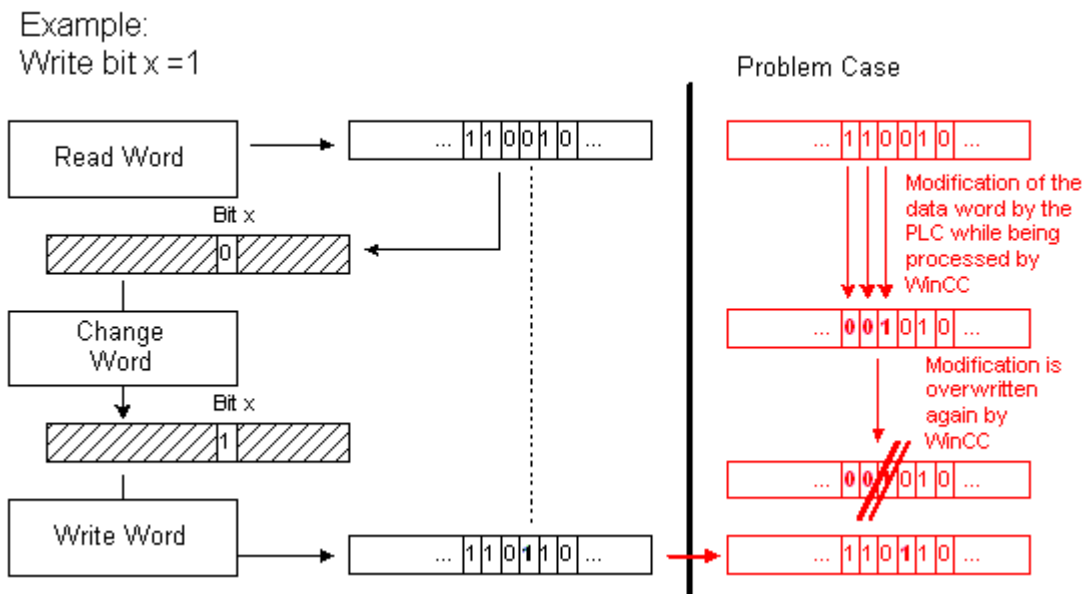
Note

If a data word changes in an AS at the same time as this data word was read via the BinWrite mechanism in the WinCC (see figure "Problem case"), then the change is lost in the AS, as soon as WinCC writes the data word back.

1.3.3.7   How to Configure a Tag with "BinWrite"

Introduction

If you want to configure a "Binary tag" for the channel unit of a communication driver, which does not support bit-/byte-access, you have to activate and configure the BinWrite mechanism using a dialog, which otherwise does not exist.

Requirements

- The required communication processor and the hardware driver are installed.
- The desired communication driver, which does not support bit-/byte-access however, is installed, e.g. "SIMATIC S5 Ethernet Layer 4".
- A connection has already been created based on its channel units.

Procedure

1. Select entry "New tag..." in the context menu of the connection that you want to create a new tag for. The "Create new tag" dialog is displayed.

2. Enter a tag name that is unique within the project in the "Name" field on the General tab. Select "Binary tag", "Unsigned 8 bit value" or "Signed 8 bit value" as data type.

3. Click the "Select" button to open dialog "Bit-/Byte tag". (Setting the address range in the AS is done with this button for channels with bit-/byte access.)

Bit-/Byte-Variable

Access Bits/Bytes

☑ Access a   Byte   in a: Word ▼

Byte number: Byte 1 ▼

Select

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

OK   Cancel   Apply
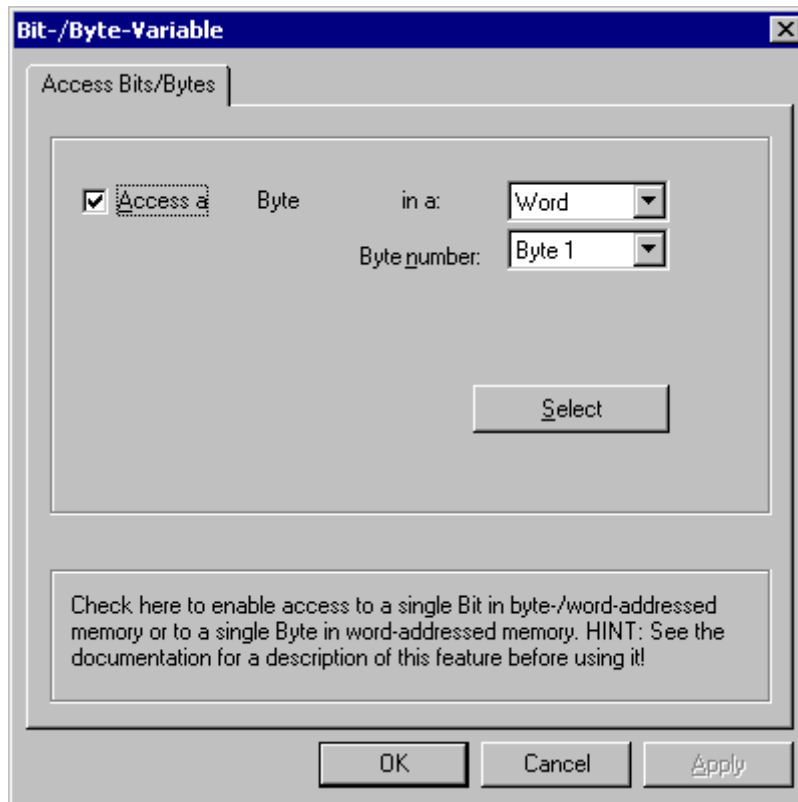
4. Activate the checkbox "Access to one bit" or "Access to one byte" and make the normal settings. The display in this dialog depends on the data type selected in step 2.

5. Close the dialog by clicking "OK".

Note

The dialog is also opened if you neglect to activate the BinWrite mechanism for a new "Binary tag" or 8 bit tag. If you want to close the "Create new tag" window again immediately with "OK", you will get error message "The address parameter is invalid". Acknowledge the dialog with "YES". This opens the dialog "Bit-/Byte tag" and you can repeat the settings this way.

1.3.4 Port Addresses for Coupling via Ethernet

Port Addresses

⚠ CAUTION
Ethernet communication
When using Ethernet-based communication, the end users is responsible for the security of his/her own data network. If targeted attacks lead to an overload of the device for instance, the functionality is no longer guaranteed.

When connected via Ethernet, you may require information on the port addresses. This information is required for configuring a firewall or a router. The port addresses that WinCC applications use as defaults are shown in the table.

	Port address TCP/IP	Port address UDP
S7 Communication	102	
HTTP (communication; transfer)	80	
HTTPS (communication; transfer)	443	
WebServices (SOAP)	80 HTTP 443 HTTPS	
OPC-XML (CE as OPC Master)	80 HTTP 443 HTTPS	
SendEmail	25	
Transfer (via Ethernet; CE-Stub; PC Loader; PC)	2308 alternative 50523	
Logging (via Ethernet) CSV File	139, 445	137, 138
Modbus Ethernet	502	
Allen-Bradley Ethernet CIP	44818	
Allen-Bradley Ethernet CSP2	2222	

Allen Bradley - Ethernet IP

<div align="right">2</div>

2.1 WinCC Channel "Allen Bradley - Ethernet IP"

Introduction

The channel "Allen Bradley - Ethernet IP" is used for linking to Allen-Bradley automation systems. The communication is handled with the Ethernet IP protocol.

Depending on the communication hardware used, the system supports connections via the following channel units:

- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x
- Allen Bradley E/IP ControlLogix

2.2 Channel Unit Assignment

Introduction

The channel unit must be selected for the channel in order to create a connection from WinCC to an existing or planned network.

Channel Unit Assignment

The following table shows an allocation of the channel units of channel "Allen Bradley - Ethernet IP" to the network and automation system (AS).

Channel Unit of the Channel	Communication Network	AS
Allen Bradley E/IP PLC5	Ethernet IP	PLC-5 with Ethernet Port
Allen Bradley E/IP SLC50x	Ethernet IP	SLC 500 with Ethernet Port, e.g. SLC 5/05
Allen Bradley E/IP ControlLogix	Ethernet IP	ControlLogix 5500

2.3   Supported Data Types

Introduction

Define the required tags for a logical connection. The following data types are supported by the "Allen Bradley - Ethernet IP" channel:

- Binary Tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag 8-bit character set
- Text tag 16-bit character set

2.4 Configuring the Channel

2.4.1 Configuring the Channel "Allen Bradley - Ethernet IP"

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS). This section shows how the "Allen Bradley - Ethernet IP" channel is configured.

When implementing the TCP/IP protocol, you must define the IP address of the AS for the logic connection. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range of 0-255.

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol. The check-back message can take up to a minute.

Connectable controllers

Connections can be implemented for the following Allen-Bradley PLCs:

- Allen-Bradley ControlLogix 5500
- Allen-Bradley CompactLogix 5300
- PLC-5 with Ethernet Port
- SLC 500 with Ethernet Port, e.g. SLC 5/05
- MicroLogix

Released communication types

The following types of communication are system-tested and released for the "Allen Bradley - Ethernet IP" channel:

- Point-to-point connection:
- Multiple point connection from the WinCC station with an optional amount of controllers.

Online Configuration

The online configuration of the "Allen Bradley - Ethernet IP" channel is not supported.

2.4.2   How to configure a "Allen Bradley E/IP PLC5" channel unit connection

Introduction

The following steps are required for configuring a connection of the "Allen Bradley E/IP PLC5" channel unit:

1. Configuring a connection
2. Configuring Tags

Requirements

● The communication driver for channel "Allen Bradley - Ethernet IP" is installed and integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the "Allen Bradley E/IP PLC5" channel unit. The "Connection Properties" dialog is opened.

2. Enter a connection name in the "General" tab.

3. Click on the "Properties" button to open the "Allen Bradley E/IP Connection Parameters" dialog.

4. Enter the IP address of the Ethernet/IP module of the controller. Port 44818 is permanently set by default for Ethernet/IP devices.

5. Define the CIP path from the Ethernet module to the controller in the "Communication path" field. This setting creates a logical interconnection of the Ethernet module with the PLC, irrespective of their location in different CIP networks.

6. Close both of the dialogs by clicking the "OK" button.

2.4.3   How to configure an "Allen Bradley E/IP SLC50x" channel unit connection

Introduction

The following steps are required for configuring a connection of the "Allen Bradley E/IP SLC50x" channel unit:

1. Configuring a connection
2. Configuring Tags

Requirements

- The communication driver for channel "Allen Bradley - Ethernet IP" is installed and integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the "Allen Bradley E/IP SLC50x" channel unit. The "Connection Properties" dialog is opened.
2. Enter a connection name in the "General" tab.
3. Click on the "Properties" button to open the "Allen Bradley E/IP Connection Parameters" dialog.

4. Enter the IP address of the Ethernet/IP module of the controller. Port 44818 is permanently set by default for Ethernet/IP devices.
5. Define the CIP path from the Ethernet module to the controller in the "Communication path" field. This setting creates a logical interconnection of the Ethernet module with the PLC, irrespective of their location in different CIP networks.
6. Close both of the dialogs by clicking the "OK" button.

2.4.4 How to configure an "Allen Bradley E/IP ControlLogix" channel unit connection

Introduction

The following steps are required for configuring a connection of the "Allen Bradley E/IP ControlLogix" channel unit:

1. Configuring a connection
2. Configuring Tags

Requirements

- The communication driver for channel "Allen Bradley - Ethernet IP" is installed and integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the "Allen Bradley E/IP ControlLogix" channel unit. The "Connection Properties" dialog is opened.

2. Enter a connection name in the "General" tab.

3. Click on the "Properties" button to open the "Allen Bradley E/IP Connection Parameters" dialog.

4. Enter the IP address of the Ethernet/IP module of the controller. Port 44818 is permanently set by default for Ethernet/IP devices.

5. Define the CIP path from the Ethernet module to the controller in the "Communication path" field. This setting creates a logical interconnection of the Ethernet module with the PLC, irrespective of their location in different CIP networks.

6. Close both of the dialogs by clicking the "OK" button.

2.4.5 Examples: Communication path

Example 1:

Connection with a PLC in the same Allen-Bradley rack.

1,0

Number	Meaning
1	Stands for a backplane connection.
0	Stands for a CPU slot number.

Example 2:

Connection with a PLC in remote Allen-Bradley racks. Two Allen-Bradley racks are networked on Ethernet.

1,2,2,190.130.3.101,1,5

Number	Meaning
1	Backplane connection
2	Stands for the CPU slot number of the second Ethernet module.
2	Stands for an Ethernet connection.
190.130.3.101	IP address of a remote AB rack on the network – in particular the third Ethernet module
1	Backplane connection
5	Slot number of the CPU

2.4.6   Configuring the tags

2.4.6.1   Configuring the tags

Introduction

For a connection between WinCC and the automation system (AS) via channel "Allen Bradley - Ethernet IP", tags of different data types can be created in WinCC. The permitted data types are listed in this section.

Tag Updating

If the tags are retrieved simultaneously in a picture from a PLC, the "Allen Bradley - Ethernet IP" channel attempts to optimize the update. This can only be accomplished under the following conditions however:

* - the tags are in the same address range.

* - the tags are as close to one another as possible within the address range.

If you do not observe these recommendations, it can lead to noticeable differences in the picture refresh with large amounts of tags. The acquisition cycles may not be maintained under certain circumstances.

The best performance for the connection is achieved if you observe the following rules when configuring the tags:

- Update of maximum 2000 tags simultaneously.
- Combine the tags in the least possible space, best in only one address range.

Permitted data types

The selection of data types listed below can be used to configure tags.

Basic data types

Data type	Bit address space
Bool	-
SInt	0-7
USInt	0-7
Int	0-15
UInt	0-15
DInt	0-31
UDInt	0-31
Real	-
String	-

Arrays

Address	Permitted data types
Array	SInt, USInt, Int, UInt, DInt, UDInt, Real

2.4.6.2 Addressing

Addressing

A tag is uniquely referenced in WinCC by means of an address in the controller. The address must correspond with the tag name in the PLC. The tag address is defined by a string with a length of up to 128 characters.

Using characters for addressing

Valid characters for tag addressing:

- Letters (a to z, A to Z)
- Numbers (0 to 9)
- Underscore (_)

The tag address consists of tag name and other character strings used to specify the tag in the PLC.

Tag name properties:

- The tag name may begin but not end with an underscore character.

- Strings with successive underscore and space characters are invalid.

- The address may not exceed a length of 128 characters.

Note

The characters reserved for tag addressing may not be used in program/tag names or at any other address instance.

The reserved characters are listed below:

Reserved character	Function
.	Element delimiter
:	Definition of a program tag
,	Delimiter for addressing multi-dimensional arrays
/	Reserved for bit addressing.
[ ]	Addressing of array elements or arrays

Controller and program tags

The "Allen-Bradley E/IP ControlLogix" allows addressing of PLC tags (global project tags) and/or program tags (global program tags). Program tags are declared via the program names in the controller and the actual tag names. Controller variables are addressed by their names.

Note
Addressing errors

Addressing errors are generated when the tag name and data type are inconsistent.

The tag name defined in the address field in WinCC must correspond with the tag name in the controller. The data type of tags in WinCC and in the controller must correspond.

Note

You cannot address module-specific Tags, such as data at input and output modules, directly. Use an Alias tag in the controller instead.

Example: Local:3:O.data cannot be addressed in WinCC

If, for Local:3:O in the controller, the alias "MyOut" is defined, you can address with WinCC via MyOut.Data.

2.4.6.3 Addressing syntax

Notation of addresses

The following tables define the possibilities for writing individual addressing.

Table 2-1   Access to arrays, basic data types and structure elements

Data types	Type	Address
Basic data types	PLC tag	Tag name
	Program tag	Programname:tagname
Arrays	PLC tag	Array tag
	Program tag	Program name: array tag
Bits	PLC tag	Tagname/bitnumber
	Program tag	Programname:tagname/bitnumber
Structure elements	PLC tag	Structure tag. Structure element
	Program tag	Program name: structure tag. structure element

Note

Bit addressing with the data types Bool, Real and String is not permitted and will cause an addressing fault.

Description of the syntax

Syntax description:

```
(Programname:)tagname([x(,y)(,z)]){.tagname([x(,y)(,z)])}(/
bitnumber)
```

- The "( )" defines an optional, single instance of an expression.
- The "{ }" defines an optional expression with multiple single instances.

The address string length may not exceed 128 characters.

2.4.6.4 Addressing Types

Array elements

Elements of one-dimensional, two-dimensional and three-dimensional arrays in the PLC are indexed by setting an index and the corresponding notation in the tag editor. Array addressing starts at element "0", with arrays of all basic types being valid for element addressing. Read/write operations are only carried out at the addressed element, and not for the entire array.

Bits and bit tags

Bit access is allowed to all basic data types with the exception of Bool, Real and String. Bit addressing is also allowed at array/structure elements. Data type Bool is defined in WinCC for addressing bits and bit tags in the basic data types.

One-place bit numbers will be address with "/x" or "/0x" (x = bit number). Bit numbers are defined by up to two digits.

Note

With the "Bool" data type in the data types SInt, Int and DInt, after changing the specified bit the complete tag is then written in the PLC again. In the meantime, no check is made as to whether other bits in the tag have since changed. Therefore, the PLC may have only read access to the specified tag.

Structures

User-defined data types are created by means of structures. These structures group tags of different data types. Structures may consist of basic types, arrays and of other structures. In WinCC, only basic data types are addressed as structure elements and not entire structures.

Structure elements

Structure elements are addressed by means of the name of the structure and of the required structure element. This addressing is separated by point. In addition to basic data types, the structure elements may represent arrays or other structures. Only one-dimensional arrays may be used as a structure element.

Note

The nesting depth of structures is only limited by the maximum length of 128 characters for the address.

2.4.6.5   Examples for Addressing

Example of a table for addressing

The following table shows basic addressing variations for control variables. Other addressing variants are possible by means of combination.

Type	Type	Address
General	PLC tag	Tag name
	Program tag	Program:tagname
Array	Access to an element of a 2-dimensional array	Arraytag[Dim1,Dim2]

Type	Type	Address
	Element of a structure array (1-dimensional)	Arraytag[Dim1].structureelement
	Bit in element of a basic type array (2-dimensional)	Arraytag[Dim1,Dim2]/Bit
Structure	Array in structure	Structuretag.arraytag
	Bit in element of an array in a sub-structure	Structuretag.structure2.arraytag [element]/bit

Note

Program tags are addressed by leading the address with the program name derived from the PLC with colon delimiter.

Example: Programname:arraytag[Dim1,Dim2]

Access to array elements

Type	Address
PLC tag	Arraytag[Dim1]
	Arraytag[Dim1,Dim2]
	Arraytag[Dim1,Dim2,Dim3]
Program tag	Programname:arraytag[Dim1]
	Programname:arraytag[Dim1,Dim2]
	Programname:arraytag[Dim1,Dim2,Dim3]

2.4.6.6   How to configure a tag for the Allen Bradley E/IP ControlLogix channel unit

Introduction

This section shows how you configure a tag for channel unit "Allen Bradley E/IP ControlLogix" in the automation system (AS) address range.

Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.

- A connection must be created in the "Allen Bradley E/IP ControlLogix" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Ethernet IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Select the desired data type in the "Data type" field.

3. The "Select" button opens the dialog "Allen Bradley ControlLogix Variable".
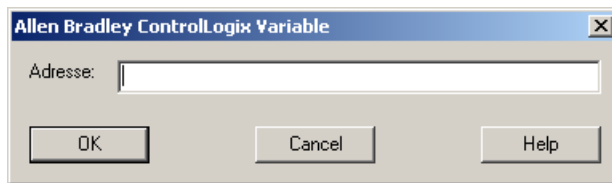
4. Enter the address of the tags in AS in the "Address" field.

    Note

    Addressing errors

    Addressing errors are generated when the tag name and data type are inconsistent.

    The tag name defined in the address field in WinCC must correspond with the tag name in the controller. The data type of tags in WinCC must correspond with the data types in the controller.

    ---

5. Close both of the dialogs by clicking the "OK" button.

2.4.6.7 How to configure a tag with bit by bit access for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

Requirements

*   The channel "Allen Bradley - Ethernet IP" must be integrated in the project.

*   A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Ethernet IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Select the desired data type in the "Data type" field.

3. The "Select" button opens the dialog "Allen Bradley PLC/SLC Tag".

4. Select an address range in the "File" field. Entries N, R, C, T, B, S, I, O, D, A, ST are available.

5. Enter the "File Number" if it is different from the suggested file number.

6. Enter the "Element".

7. Depending on the setting in the "File" field, define the "Bit" to address or select a value for "Bit (octal)" or "Sub".

8. Close both of the dialogs by clicking the "OK" button.

2.4.6.8 How to configure a tag with byte by byte access for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for byte by byte access for the address area in the automation system (AS).

Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.
- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Ethernet IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Select the desired data type in the "Data type" field.

3. The "Select" button opens the dialog "Allen Bradley PLC/SLC Tag".

4. Select the A or ST address range in the "File" field.

5. Enter the "File Number" if it is different from the suggested file number.

6. Enter the "Element".

7. Close both of the dialogs by clicking the "OK" button.

2.4.6.9 How to configure a tag with word by word access for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for word by word access for the address area in the automation system (AS).

Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.

- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Ethernet IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Select the desired data type in the "Data type" field.

3. The "Select" button opens the dialog "Allen Bradley PLC/SLC Tag".

4. Select the address range in the "File" field. Entries N, R, C, T, B, S, I, O, D, A and ST are available.

5. Enter the "File Number" if it is different from the suggested file number. The File number cannot be changed for the setting "S".

6. Enter the "Element".

7. Select one of the values for the "Sub" field if it is displayed. This depends on the setting made in the "File field.

8. Close both of the dialogs by clicking the "OK" button.

2.4.6.10   How to configure a text tag for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for word by word access for the address area in the automation system (AS).

Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.

- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Ethernet IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Select the desired data type in the "Data type" field.

3. The "Select" button opens the dialog "Allen Bradley PLC/SLC Tag".

4. Select the A or ST address range in the "File" field.

5. Enter the "File Number" if it is different from the suggested file number. The File number cannot be changed for the setting "S".

6. Enter the "Element".

7. Select one of the values for the "Sub" field if it is displayed. This depends on the setting made in the "File field.

8. Close both of the dialogs by clicking the "OK" button.

Mitsubishi Ethernet

3

3.1 WinCC channel "Mitsubishi Ethernet"

Introduction

The "Mitsubishi Ethernet" channel is intended for communication between a WinCC station and Mitsubishi controllers of the FX3U and Q series. The communication takes place via the MELSEC communication protocol (MC protocol).

Channel units

The "Mitsubishi Ethernet" channel comes equipped with the "Mitsubishi FX3U Series" and "Mitsubishi Q Series" channel units.

3.2 Supported data types

Introduction

Define the required tags for a logical connection between WinCC and a connected controller.

The following data types are supported by the "Mitsubishi Ethernet" channel:

- Binary Tag
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag 8-bit character set
- Text tag 16-bit character set
- Raw Data Tag

3.3 Configuring the Channel

3.3.1 Configuring the "Mitsubishi Ethernet" channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS). This section illustrates how to configure the "Mitsubishi Ethernet" channel.

You can establish a connection to an AS with either TCP/IP or UDP/IP protocol. Therefore, you have to enter the IP address and IP port number for the AS when configuring a logical connection for the channel "Mitsubishi Ethernet" and select either the UDP or TCP/IP protocol. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range 0 to 255.

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol. The feedback can take longer and is dependent on the operating system.

Mitsubishi controllers that are supported

You can configure logical connections for the following Mitsubishi controllers:

- MELSEC FX3U series
- MELSEC System Q

When you configure the connections and tags, the procedure is identical for both controller families. The configuration differs only in relation to the used address type of the specific controller family. In addition, the routing of information is only supported by MELSEC System Q models.

Online Configuration

The "Mitsubishi Ethernet channel supports the online configuration of tags and connections.

3.3.2 How to configure a "Mitsubishi FX3U Series" channel unit connection

Introduction

The following steps are required for configuring the "Mitsubishi Ethernet" channel.

1. Configuring a connection
2. Configuring tags

This section shows you how to configure the connection for the "Mitsubishi FX3U Series" channel unit.

Requirements

- The communication driver for the "Mitsubishi Ethernet" channel is installed and integrated into the project.

  Note

  When you use Mitsubishi controllers, you will also have to configure the connections in the controller. To do this, follow the operating instructions provided by the manufacturer.

Procedure

1. Open the menu structure for the "Mitsubishi Ethernet" communication driver in the "Tag Management" editor of the WinCC Explorer.

2. Select the entry "New Connection" in the shortcut menu of the "Mitsubishi FX3U Series" channel unit. The "Connection Properties" dialog is opened.

3. Enter the name of the connection.

4. Click the "Properties" button to open the "Connection properties" dialog.

5. Enter the IP address of the controller.

6. Enter the port you want to use for the TCP/IP connection.
   The valid range of values is from 0 to 65535.

7. Select the protocol to be used, "TCP" or "UDP".

   Note

   We recommend that you use the default protocol "UDP". The timeout behavior is better with this protocol. Only use TCP if you are not able to use UDP.

8. Enter the PC number.
    If you do not want to enter a PC number, you must enter the value 255 or 0.

9. To establish the connection, select "Establish connection".

10. Close each dialog box by clicking "OK."

3.3.3 How to configure a "Mitsubishi Q Series" channel unit connection

Introduction

The following steps are required for configuring the "Mitsubishi Ethernet" channel.

1. Configuring a connection.

2. Configuring tags.

This section shows you how to configure the connection for the "Mitsubishi Q Series" channel unit.

Requirements

*   The communication driver for the "Mitsubishi Ethernet" channel is installed and integrated into the project.

    Note

    When you use Mitsubishi controllers, you will also have to configure the connections in the controller. To do this, follow the operating instructions provided by the manufacturer.

Procedure

1. Open the menu structure for the "Mitsubishi Ethernet" communication driver in the "Tag Management" editor of the WinCC Explorer.

2. Select the entry "New Connection" in the shortcut menu of the "Mitsubishi Q Series" channel unit. The "Connection Properties" dialog is opened.

3. Enter the name of the connection.

4. Click the "Properties" button to open the "Connection properties" dialog.

5. Enter the IP address of the controller.

6. Enter the port you want to use for the TCP/IP connection.
   The valid range of values is from 0 to 65535.

7. Select the protocol to be used, "TCP" or "UDP".

   Note

   We recommend that you use the default protocol "UDP". The timeout behavior is better with this protocol. Only use TCP if you are not able to use UDP.

8. Enter the network number.
   The default setting is 0.

9. Enter the PC number.
   If you do not want to enter a PC number, you must enter the value 255 or 0.

10. To establish the connection, select "Establish connection".

11. Close each dialog box by clicking "OK."

3.3.4 Configuring the tags

3.3.4.1 Configuring the tags

Introduction

Tags of different data types can be created in WinCC for a connection between WinCC and the automation system via the "Mitsubishi Ethernet" channel.

The following sections describe how to configure the tags.

Address types, address ranges and data types

The table lists the address types, address ranges and data types that can be used when configuring tags and structured tags. In addition, it is also specified which automation system (AS) the respective address type supports. "FX" here stands for the MELSEC FX3 U series and "Q" for the MELSEC Q system.

Note

If you want to use a specific WinCC data type, you have to consider the selected address type; for example, address type "D" does not support the WinCC data type "Binary".

Address type	Code	Address range	Display	Data type	AS that are supported
Input relays	X	000000...001FFF	Hex. (FX:Octal)	Bit	FX, Q
Output relays	Y	000000...001FFF	Hex. (FX: Octal)	Bit	FX, Q
Bit memories	M	0...8191	Decimal	Bit	FX, Q
Latch relay	L	0...8191	Decimal	Bit	Q
Error bit memory	F	0...2047	Decimal	Bit	Q
Edge relay	V	0...2047	Decimal	Bit	Q
Link relay	B	000000...001FFF	Hex.	Bit	Q
Data register	D	0...12287	Decimal	Word	FX, Q
Link register	W	000000...001FFF	Hex.	Word	Q
Timer (contact)	TS	0...2047	Decimal	Bit	FX, Q
Timer (OUT coil)	TC	0...2047	Decimal	Bit	Q
Timer (current value)	TN	0...2047	Decimal	Word	FX, Q
Retentive timer (contact)	SS	0...2047	Decimal	Bit	Q
Retentive timer (coil)	SC	0...2047	Decimal	Bit	Q
Retentive timer (value)	SN	0...2047	Decimal	Word	Q
Counter / Contact	CS	0...1023	Decimal	Bit	FX, Q
Counter / Coil	CC	0...1023	Decimal	Bit	Q
Counter / current value	CN	0...1023	Decimal	Word	FX, Q
Expansion register	R	0...65535	Decimal	16-bit	FX, Q
Special link relay	SB	000000...0007FF	Hex.	Bit	Q
Special link register	SW	000000...0007FF	Hex.	Word	Q
Step flags	S	0...8191	Decimal	Bit	Q
Input relay (direct)	DX	000000...001FFF	Hex.	Bit	Q
Output relay (direct)	DY	000000...001FFF	Hex.	Bit	Q
Index register	Z	0...15	Decimal	Word	Q
Diagnostic bit memory	SM	0...2047	Decimal	Bit	Q
Diagnostic register	SD	0...2047	Decimal	Word	Q

3.3.4.2 How to configure a tag

Introduction

This section shows you how to configure a tag access for the address range in the automation system (AS).

Requirements

- The "Mitsubishi Ethernet" channel is integrated in the project.
- A connection is established in the "Mitsubishi FX3U Series" or "Mitsubishi Q Series" channel unit.

Procedure

1. Select the "New Tag" entry in the shortcut menu of the Mitsubishi Ethernet connection. The "Tag properties" dialog box opens.

2. Enter the tag name.

3. Set the required data type.

4. Click the "Select" button to open the "Address properties" dialog.
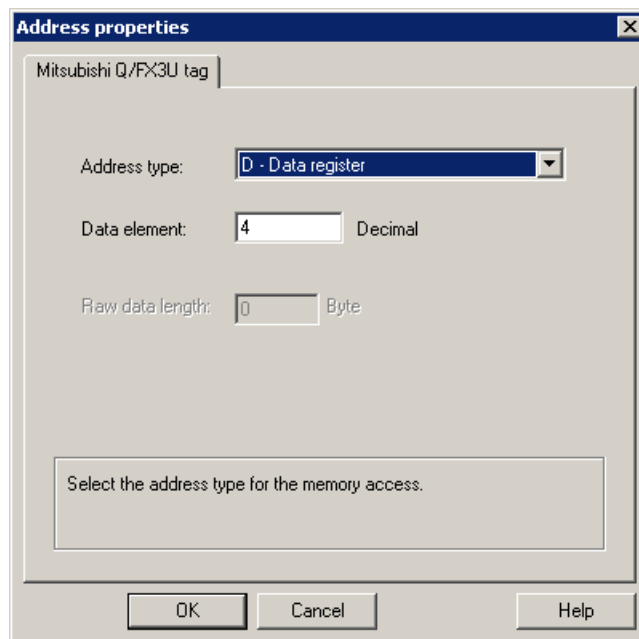
5. Set the address type.

6. Enter the corresponding data element. The value depends on the configuration of the controller.

7. If necessary, enter the raw data length.

8. Close the "Address properties" dialog by clicking "OK".
   The address of the tag is displayed in the "Address" field of the "Tag properties" dialog.
   The address is adjusted to the data format of the AS.

9. Close the "Tag properties" dialog by clicking "OK".

Modbus TCPIP

<div style="text-align: right; font-size: 3em;">4</div>

4.1 "Modbus TCP/IP" channel

Introduction

The "Modbus TCPIP" channel is for communication between a WinCC station and PLCs that support Modbus via Ethernet. The communication is handled with the Modbus TCP/IP protocol.

Channel units

The "Modbus TCPIP" channel comes with the "Modbus TCP/IP Unit #1" channel unit.

4.2 Supported Data Types

Introduction

Define the required tags for a logical connection with a connected controller. The following data types are supported by the "Modbus TCPIP" channel:

- Binary tag
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Text tag 8-bit character set
- Text tag 16-bit character set

4.3   Configuring the Channel

4.3.1   Configuring the "Modbus TCPIP" Channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS). This section describes the communication with the "Modbus TCP/IP Unit #1" channel unit. All connection-specific parameters are defined during the setup.

When implementing the TCP/IP protocol, you must define the IP address of the AS for the logic connection. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range of 0-255.

> **Note**
>
> **Timeout Behavior**
>
> Interrupted connections are not detected immediately when using the TCP/IP protocol. The check-back message can take up to a minute.

Enabled Communication Methods with Modbus TCPIP

The following types of communication have been system-tested and approved:

- Point-to-point communication

- Multiple point connection of the WinCC station with an optional amount of controllers.

> **Note**
>
> Integrating the WinCC station via a bridge in a Modbus network is not possible because the WinCC station works as a Modbus Master.

Online Configuration

The Online configuration is not supported.

4.3.2   How to configure a connection

Introduction

The following steps are required for configuring the channel "Modbus TCPIP":

1. Configuring a connection

2. Configuring Tags

Requirements

- The communication driver for channel "Modbus TCPIP" is installed and integrated into the project.

Procedure

1. Open the menu structure for the "Modbus TCPIP" communication driver in the "Tag Management" editor of the WinCC Explorer.

2. Select the entry "New Connection" in the pop-up menu of the "Modbus TCP/IP Unit #1" channel unit. The "Connection Properties" dialog is opened.

3. Enter the name of the connection.

4. Click on the "Properties" button to open the "Modbus TCPIP Properties" dialog.
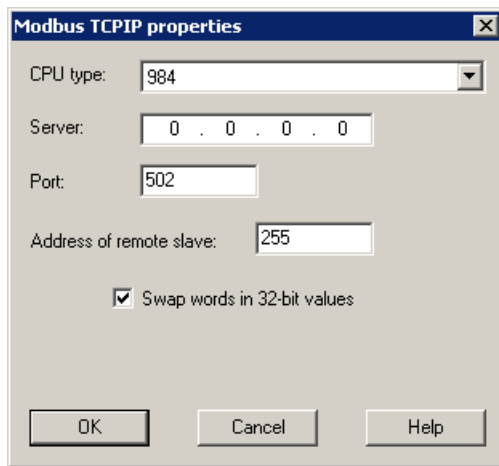
5. Select the connected Modicon controller under "CPU Type". The following CPUs are available for selection:

   – 984
     Use this CPU type for the CPU 984 (except for CPU 984A, 984B and 984X).

   – Modicon Compact, Modicon Quantum, Modicon Momentum

   – Modicon Premium, Modicon Micro

6. Enter the IP address of the controller in the "Server" field.

7. Enter the port to be used for the TCP/IP connection in the "Port" field. The default port for Modbus TCP/IP connections is 502.

8. If you are using a bridge, enter the slave address of the remote controller in the "Address of remote slave" field.
   If no bridge is used, you must enter the default value of 255 or 0 as the address.

   Note

   Integrating the WinCC station via a bridge in a Modbus network is not possible because the WinCC station works as a Modbus Master.

9. To swap the order of the 16-bit register in 32-bit process values, select "Swap words in 32-bit values".

10. Close both of the dialogs by clicking the "OK" button.

4.3.3 Configuring the tags

4.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the automation system (AS) via channel "Modbus TCPIP", tags of different data types can be created in WinCC.

The following sections describe how to configure the tags. The addressing of the data range in the AS and the data type of the WinCC tags are different.

Tag Updating with the Modbus TCP/IP protocol

If the tags are retrieved simultaneously in a picture from a controller, the Modbus TCP/IP channel attempts to optimize the update. This can only be accomplished under the following conditions however:

● The tags are in the same address range.

● The tags are as close to one another as possible within the address range.

If you do not observe these recommendations, it can lead to noticeable differences in the picture refresh with large amounts of tags. The acquisition cycles may not be maintained under certain circumstances.

The best performance for the connection is achieved if you observe the following rules when configuring the tags:

● Update of maximum 2000 tags simultaneously.

● Combine the tags in the least possible space, best in only one address range.

Data Types and Address Ranges in the Controller

The table lists the data types and address ranges that can be used when configuring tags and structured tags.

Name	Area with CPU Premium/Micro	Area with CPU 984, Compact, Quantum, Momentum	data type
Coil (discrete output)	%M [1]	0x	Bit
Discrete input	(%I) – not realized by Premium/Micro	1x	Bit

Name	Area with CPU Premium/Micro	Area with CPU 984, Compact, Quantum, Momentum	data type
Input register	(%IW) – not realized by Premium/Micro	3x	Bit, +/- Int, Int
Holding register (output)	%MW	4x	Bit [2], +/- Int, Int, +/- Double, Double, Float, ASCII
Extended memory (Only available with the "Quantum/Momentum" CPU)	--	6x	Bit [2], +/- Int, Int, +/- Double, Double, Float, ASCII

[1]   Due to a system characteristics of the external controller the last x bits on the end of the address area cannot be accessed.

[2]   In the case of write accesses note:
       With the "bit" data type in the "4x", "6x" and "%MW" areas, after changing the specified bit the entire word is written back to the controller. There is no check to determine whether any other bits in the word have changed. As a result, the controller only has read access to the specified word.

The standard bit counting method (16 LSB - 1 MSB) used with controllers of the 984, Compact, Quantum and Momentum series will only be used for these CPUs in the "Tags" editor for the data type "bit". Bit positions have the following allocations:

	Left byte								Right byte							
Counting with tags	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When entering bit numbers in other locations in WinCC, the bit allocation of WinCC applies (0 LSB - 15 MSB):

How the bit positions are counted	Left byte								Right byte							
In the WinCC you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This bit counting method also applies to Modicon Premium and Modicon Micro controllers

Format for "Signed"

The placeholder "+/-" stands for the data types "Signed Int" and "Signed Double".

See also

4.3.3.2   How to Configure a Tag with Bit by Bit Access

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

Requirements

- The channel "Modbus TCPIP" must be integrated in the project.
- A connection must be created in the "Modbus TCP/IP Unit #1" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Modbus TCP/IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Select "Binary tag" in the "Data Type" field.

3. Click on the "Select" button to open the "Modbus TCP/IP Tag Properties" dialog. Define the data area in the AS that holds the data in the "Area" field. You can select from areas 0x, 1x, 3x, 4x and 6x.
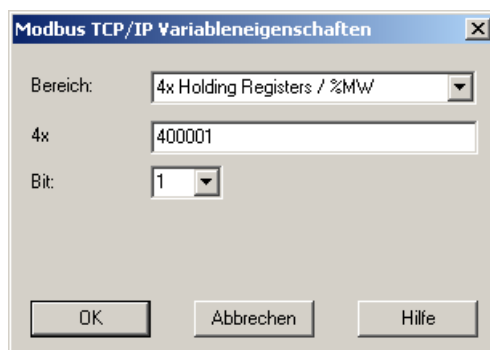
4. Enter the address of the tags in the respective address field, "4x" for instance. The value depends on the configuration of the controller.

5. Enter the address of the bit in the "Bit" field if necessary. Whether an entry is possible depends on your selection in the "Area from" field.

6. Select a value for "File" if you have set the value "6x Extended Memory" in the "Area" field.

7. Close both of the dialogs by clicking the "OK" button.

Note

After closing the "Modbus TCP/IP Tag Properties" dialog, the internal address of the tags in the controller is shown in field "Address" of the "Tag Properties" dialog. This address can differ from the entered address because it is adapted to the AS data format.

See also

How to Configure a Text Tag (Page 77)

4.3.3.3 How to Configure a Tag with Word by Word Access

Introduction

This section shows you how to configure a tag for word by word access for the address area in the automation system (AS).

Requirements

● The channel "Modbus TCPIP" must be integrated in the project.

● A connection must be created in the "Modbus TCP/IP Unit #1" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Modbus TCP/IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. Set the data type to "Unsigned 16-bit value" in the "Data Type" field.

3. Click on the "Select" button to open the "Modbus TCP/IP Tag Properties" dialog. Define the data area in the AS that holds the data in the "Area" field. You can select from the three areas 3x, 4x and 6x.
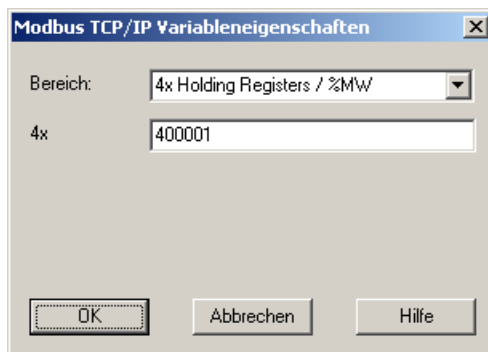
4. Enter the address of the tags in the respective address field, "4x" for instance. The value depends on the configuration of the controller.

5. Close both of the dialogs by clicking the "OK" button.

Note

After closing the "Modbus TCP/IP Tag Properties" dialog, the internal address of the tags in the controller is shown in field "Address" of the "Tag Properties" dialog. This address can differ from the entered address because it is adapted to the AS data format.

4.3.3.4 How to Configure a Text Tag

Introduction

This section shows you how to configure a tag for word by word access for the address area in the automation system (AS).

Requirements

- The channel "Modbus TCPIP" must be integrated in the project.
- A connection must be created in the "Modbus TCP/IP Unit #1" channel unit.

Procedure

1. Select "New Tag" from the shortcut menu of the Modbus TCP/IP connection. The "Tag properties" dialog box opens.

2. Enter the name of the tag in the "Name" field. In the "Data Type" field, set "Text tag 8-bit character set" as the data type.

3. Click on the "Select" button to open the "Modbus TCP/IP Tag Properties" dialog.
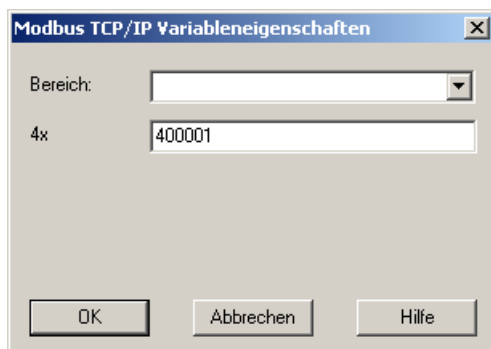
4. Enter the address of the tags in address field "4x". The value depends on the configuration of the controller.

5. Close both of the dialogs by clicking the "OK" button.

OPC Channel

<div style="text-align: right; font-size: 3em;">5</div>

5.1 WinCC OPC Channel

Introduction

WinCC can be used as both an OPC server and as an OPC client. The OPC channel is the OPC client application of WinCC.

The OPC communication driver can be used as OPC DA client, OPC XML client, and OPC UA client.

The following OPC components are installed automatically:

- OPC communication driver
- OPC Item Manager

Possible Applications

WinCC as an OPC DA client

If WinCC is used as an OPC DA client, the OPC channel must be added to the WinCC project. A connection for data exchange is created in the WinCC project of the WinCC OPC DA client; this is used to handle access to the WinCC tags of the OPC DA server.

To simplify the process, the OPC Item Manager is used. A WinCC OPC DA client can access multiple OPC DA servers. This requires that a connection be created for each OPC server. In this way, the WinCC OPC DA client can be used as a central operation and monitoring station.
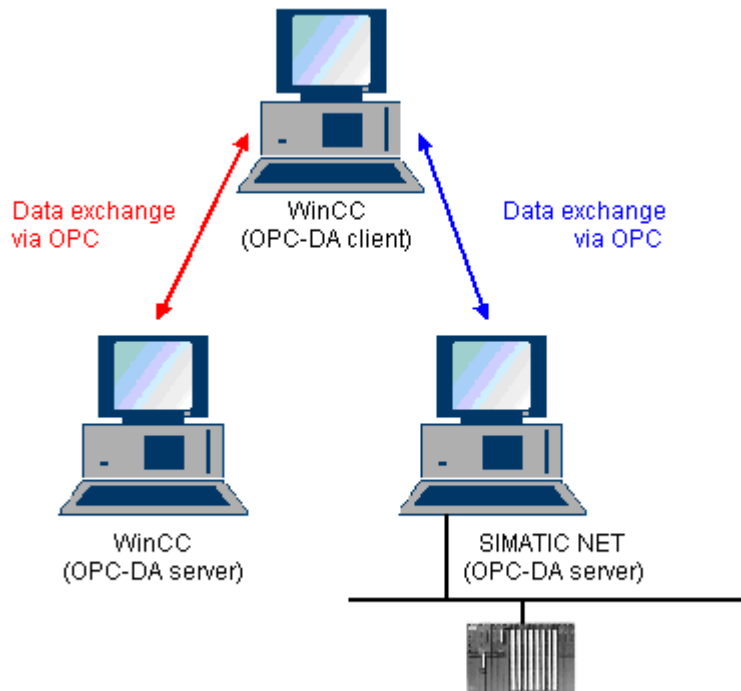
Note

The WinCC OPC channel establishes connections only to OPC servers which have the status "OPC_STATUS_RUNNING".

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

5.2   OPC Item Manager

Introduction

A connection and a WinCC tag are configured in the WinCC project of the WinCC OPC client to enable access to tags of an OPC server. The OPC Item Manager simplifies this process for you. The OPC Item Manager is automatically installed with WinCC.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Requirements

The following requirements must be met in order to use the OPC Item Manager for configuration:

- The OPC server is an OPC DA server, an OPC XML server, or an OPC UA server
- A tag is already configured on the OPC server.
- If WinCC is to be used as the OPC server the WinCC project of the WinCC OPC server must be enabled. If this is not the case, the OPC Item Manager cannot access the WinCC OPC server.
- It must be possible to access the computer of the OPC servers via the IP address or HTTP.
- The OPC server must support the browser functionality. If that is not the case, access to the tag of the OPC server must be configured manually.

Note

If you change language in the WinCC Explorer while the OPC Item Manager is open, no tags are displayed when you click the "Browse Server" button. Exit the OPC Item Manager before changing language.

Tasks of the OPC Item Manager

The OPC Item Manager assumes the following tasks:

- Select OPC server
- Creating a connection
- Tag selection
- Adding a tag

Selecting the OPC server

OPC DA server

The OPC Item Manager can be used to determine the name of the OPC DA server in the network. These OPC DA servers can run on the same computer or on different computers in the connected network environment. For further details, refer to "WinCC OPC DA client ".

OPC XML Server

The OPC Item Manager cannot be used to determine the name the OPC XML server. The OPC XML server can be added using the "OPC Web Server" button. For further details, refer to "WinCC OPC XML client ".

OPC UA Server

The OPC Item Manager cannot be used to determine the name the OPC UA server. The OPC UA server can be added using the "OPC UA Server" button. For further details, refer to "WinCC OPC UA client ".
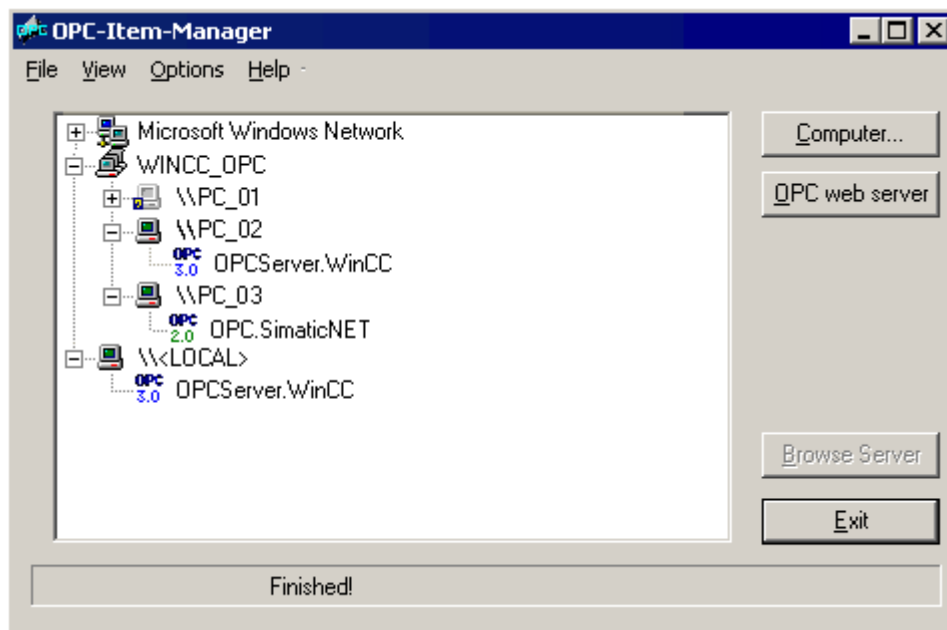
Icons of the OPC Item Manager	Description
	A networked computer has not yet been searched for installed OPC DA servers.
	The computer was not found in the network or the computer could not be accessed.
	A networked computer has been searched for installed OPC DA servers.
	A networked computer contains the OPC DA server designated with the OPC symbol. The number indicates which OPC DA specification of the WinCC OPC DA client is used.
\\<LOCAL>	Refers to the computer running the OPC Item Manager.

Icons of the OPC Item Manager	Description
OPC http://	Name of the OPC XML server. The OPC XML server can be added using the "OPC Web Server" button.
OPC opc.tcp://	Name of the OPC UA server. The OPC UA server can be added using the "OPC UA Server" button.

Creating a connection

The OPC Item Manager configures all required settings when creating a connection. If a connection to the OPC server has already been created, this function is not available.

Tag selection

You may use the tag selection dialog to select one or more tags on the OPC server which the WinCC OPC client is to access. Filter criteria can be used to limit the choices in the tag selection dialog.

Adding a tag
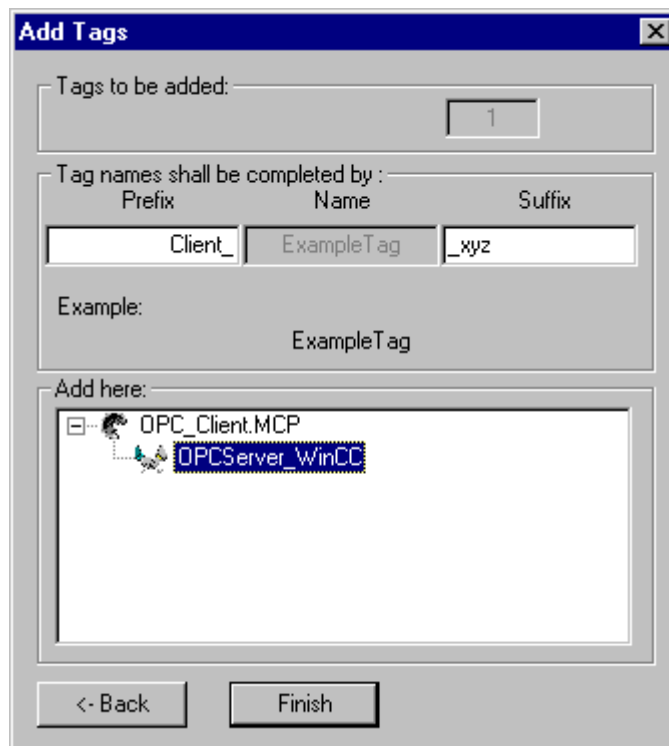
The names of the WinCC tags that access the tags of the OPC server can be set in the "Add Tags" dialog.

The WinCC tag name consists of the "prefix", "name" and "suffix". The "Name" field is preconfigured with the "ExampleTag" text. "ExampleTag" stands for the WinCC tag name of the WinCC OPC server.

You can assign a prefix or suffix to distinguish the WinCC tag name on the WinCC OPC client from the WinCC tag name on the WinCC OPC server. When configuring project monitoring, a prefix or suffix must be assigned.

The tag name may be assigned only once in a given WinCC project.

Example

The WinCC tag name on the WinCC OPC DA server is called "OPC_Server_Tag". The "Client_" value is entered in the prefix field and "_xyz" in the suffix field. In the WinCC project of the WinCC OPC DA client, the WinCC tag "Client_OPC_Server_Tag_xyz" is created.

If the tag name on the OPC server contains special characters, they are replaced by an underscore ("_"), because not all special characters occurring in tag names are supported by the OPC Item Manager.

Click "Finish" to add the WinCC tags to the WinCC project of the WinCC OPC DA client. The OPC Item Manager automatically sets the data type, the name and the address parameters for the WinCC tag.

See also

How to Access a WinCC Tag with the OPC Item Manager (Page 103)

How to Access a WinCC Tag with the OPC Item Manager (Page 87)

5.3   Overview of the Supported WinCC Data Types

The list below shows the data types that are supported by the WinCC OPC DA client and WinCC OPC DA server:

- Binary tags
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag, 8-bit character set
- Text tag, 16-bit character set
- Raw data type
- Structure types
- Text reference

Note

For structure types, only the structure elements are supported, not the structure itself. However, the structure can be configured later. For more information, refer to the topic "Using structures on the WinCC OPC DA client."

If a text tag is created with the OPC Item Manager, it is assigned a length of 160 characters. This length can be changed to any length.

See also

How to Use Structures on the WinCC OPC DA Client (Page 94)

5.4 WinCC OPC DA Client

5.4.1 Functionality of the WinCC OPC DA Client

Introduction

The OPC channel does not require a separate communication module. The OPC channel is an application which employs the OPC software interface to use an OPC DA server to access process data.

If WinCC is to be used as an OPC DA client, the OPC channel must be added to the WinCC project.

If a communication is established to a WinCC OPC DA server, the values of the WinCC tags are exchanged. To do this, a connection is set up in the WinCC project of the WinCC OPC DA client; it is used to handle access to the WinCC OPC DA server.

For the WinCC OPC DA client to access multiple OPC DA servers, a connection for each of the OPC DA servers must be set up in the WinCC project. For more information about troubleshooting channels and tags, refer to "Troubleshooting".

Note

The WinCC OPC channel establishes connections only to OPC servers which have the status "OPC_STATUS_RUNNING".

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Connection Monitoring

Three mechanisms are integrated for connection monitoring in the WinCC OPC-DA client. It is thus possible to take the best possible measures in the event of a network error or malfunction of an OPC DA server.

1. If the processing period for a DCOM activation exceeds warning value of 5 seconds, the tag is assigned the value "Addressing Error". If the processing period exceeds the cancellation value of 10 seconds, the connection to the OPC DA server is interrupted. This is displayed in the "Connection Status" dialog of the WinCC Explorers.

The OPC DA specification 3.00 is provided with the "Keep-Alive" feature. If the OPC DA server supports the OPC DA specifications 3.00, this feature is used. The feature causes the OPC DA server to automatically trigger cyclic updating (call OnDataChange) even if the tag values have not changed. If this regular updating is disabled, the WinCC OPC DA client terminates the connection.

The same behavior applies in the case of an OPC DA server which supports the OPC DA specifications 2.05a. In order to check the connection to the OPC DA server, the WinCC OPC DA client requests the status cyclically every 10 seconds. If this regular updating is disabled, the WinCC OPC DA client terminates the connection.

Generally, the WinCC OPC DA client terminates the connection to the OPC DA server when the connection is not capable of functioning. The WinCC OPC DA client attempts to re-establish the connection again, automatically, every 10 seconds.

See also

5.4.2 How to Access a WinCC Tag with the OPC Item Manager

5.4.2.1 How to Access a WinCC Tag with the OPC Item Manager

Introduction

When an OPC connection is made between WinCC and WinCC, data exchange occurs using WinCC tags. The WinCC OPC DA client uses an OPC connection to read the WinCC tag "OPC_Server_Tag" on the WinCC OPC DA server. To simplify the process, the OPC Item Manager is used.
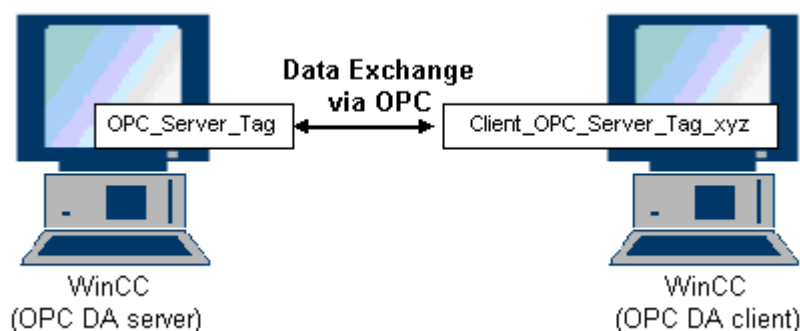
Requirements

- Two computers with WinCC projects.
- Both computers must be accessible via their IP addresses.

Configuration Steps

The following configurations are required in the WinCC project of the WinCC OPC DA client:

See also

Configuring Access with the OPC Item Manager (Page 88)

Configuring the OPC Channel on the WinCC OPC DA Client (Page 88)

Configuring Access with the OPC Item Manager (Page 103)

5.4.2.2   Configuring the OPC Channel on the WinCC OPC DA Client

Introduction

To use OPC for data exchange, the OPC channel must be set up in the WinCC project.

Procedure

1. Click the "Tag Management" icon in the navigation window of the WinCC Explorer on the WinCC OPC DA client.

2. Select "Add New Driver" from the "Tag Management" shortcut menu. The "Add New Driver" dialog is opened.

3. Select the "OPC.chn" driver and click the "Open" button. The channel is created and the communication driver is displayed in the tag management.

See also

Configuring Access with the OPC Item Manager (Page 88)

5.4.2.3   Configuring Access with the OPC Item Manager

Introduction

This section explains how to use the OPC Item Manager to configure access to the WinCC tag "OPC_Server_Tag" of the WinCC OPC DA server.

Requirements

- Configure an internal tag named "OPC_Server_Tag" of the data type "signed 16-bit value" in the WinCC project of the WinCC OPC DA server.

- Enable the WinCC project of the WinCC OPC DA server.

- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The "OPC Item Manager" opens.
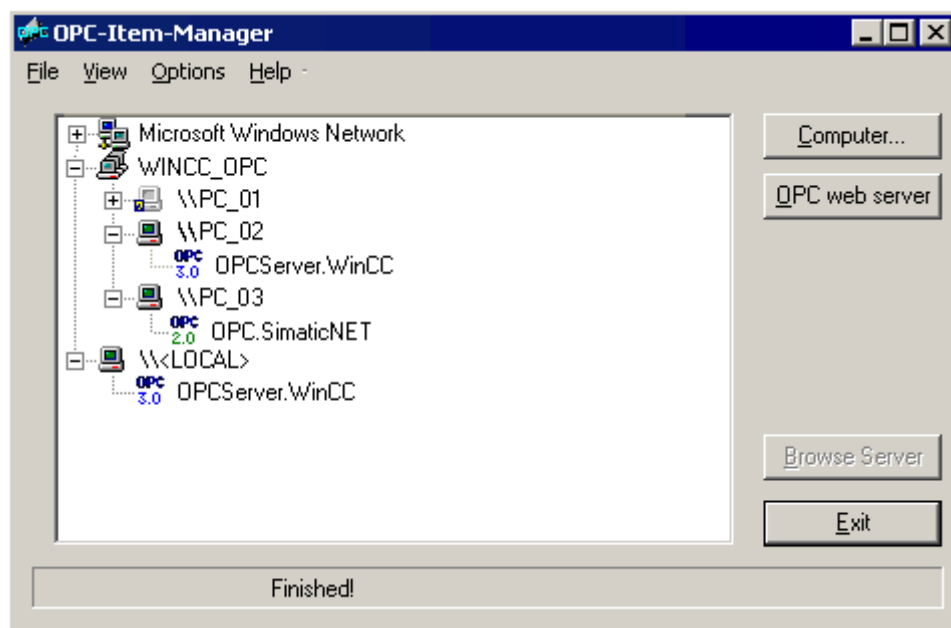
2. Choose the name of the computer to be used as the WinCC OPC DA server from the selection dialog.
   Select "OPCServer.WinCC" from the list displayed.

3. Click the "Browse Server" button.
   The "Filter criteria" dialog is opened.
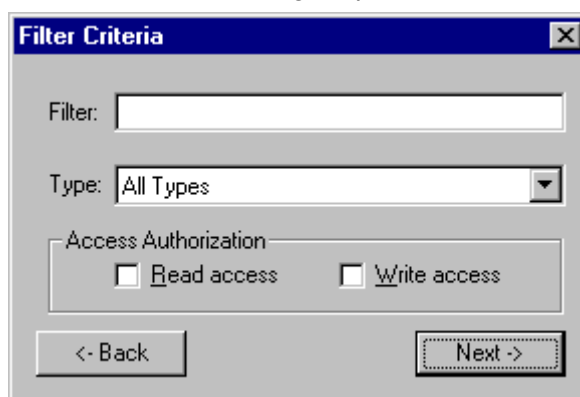
4. Click "Next".
   The "OPCServer.WinCC ..." dialog is opened.
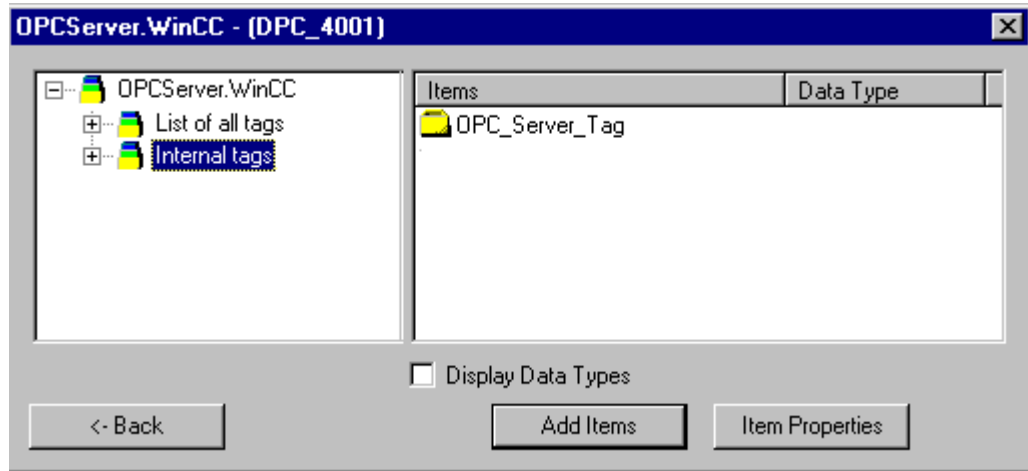
5. Select the WinCC tag "OPC_Server_Tag".
   Click the "Add Items" button.

6. If a connection to the WinCC OPC DA server already exists, continue with step 6.
   If a connection has not been created, a message will be displayed.
   Click on the "Yes" button. The "New Connection" dialog is opened.
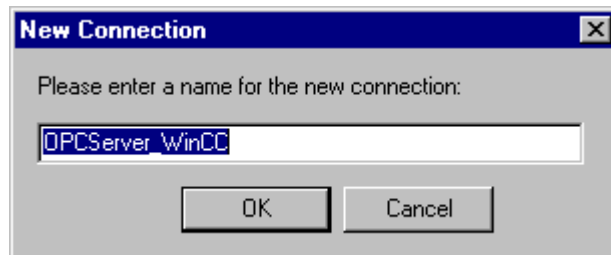
7. Enter "OPCServer_WinCC" as the name of the connection. Click "OK".
   The "Add Tags" dialog opens.
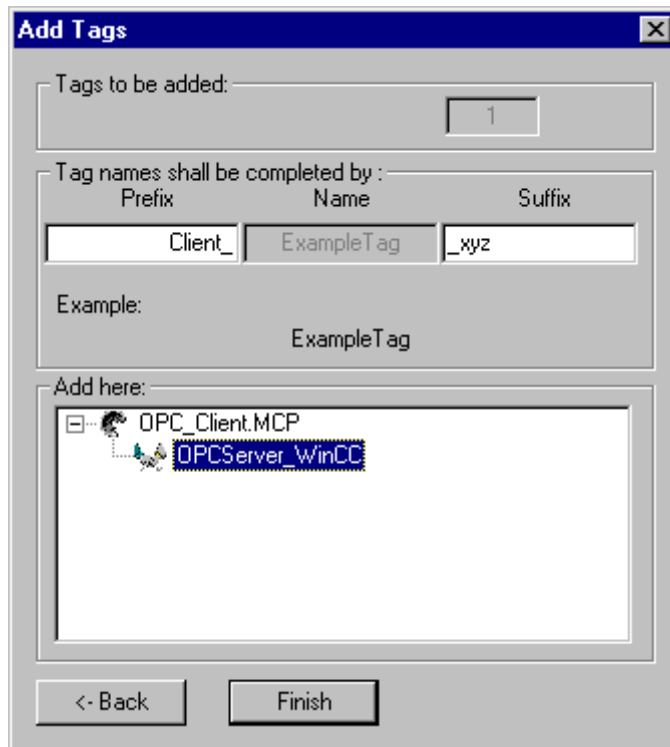
8. Enter the text "Client_" in the prefix field and the text "_xyz" in the suffix field.

9. Select connection "OPCServer_WinCC".
   Click "Finish".

10. Click the "Back" button in the "OPCServer.WinCC ..." dialog.
    Click "Exit" to close the OPC Item Manager.

See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 88)

5.4.3 Accessing a WinCC Tag without the OPC Item Manager

Introduction

OPC servers that do not support browser functionality require access to be configured manually. Configuration of WinCC tags on the WinCC OPC DA client is shown using an example of a WinCC-WinCC OPC connection.
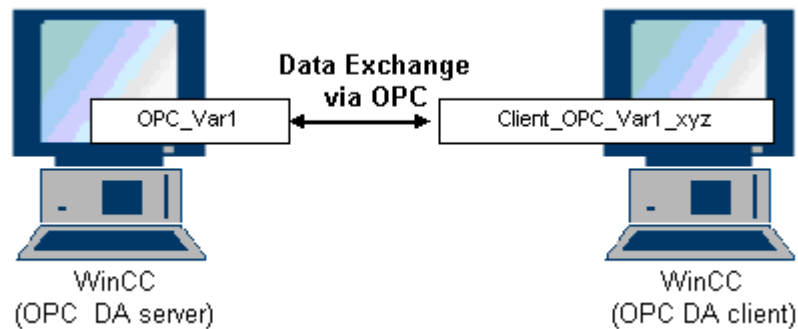
Note

To access a WinCC tag without the OPC Item Manager, the ItemID must be set manually. When addressing WinCC tags, the symbolic computer name (server prefix) can also be specified. The ItemID has the following syntax: Server prefix::WinCC tag. If the WinCC tag of the local WinCC project is addressed, the server prefix is omitted.

The following configurations are required in the WinCC project of the WinCC OPC DA client:

1. Selection of the "OPC_Var1" WinCC tag to be accessed.

2. Creation of a connection.

3. Configuration of the "Client_OPC_Var1_xyz" WinCC tag that accesses the WinCC tag of the WinCC OPC DA server.

Requirements

- Two computers with WinCC projects.

- Both computers must be accessible via their IP addresses.

- Configure an internal tag named "OPC_Var1" with data type "signed 16-bit value" in the WinCC project of the WinCC OPC DA server.

- Enable the WinCC project of the WinCC OPC DA server.

- Add the OPC channel to the WinCC project of the WinCC OPC DA client.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. Select "New Connection" from the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client. The "Connection Properties" dialog is opened. Enter a name for the connection in the corresponding field.

2. Click the "Properties" button. A dialog with the connection name in its title is displayed.
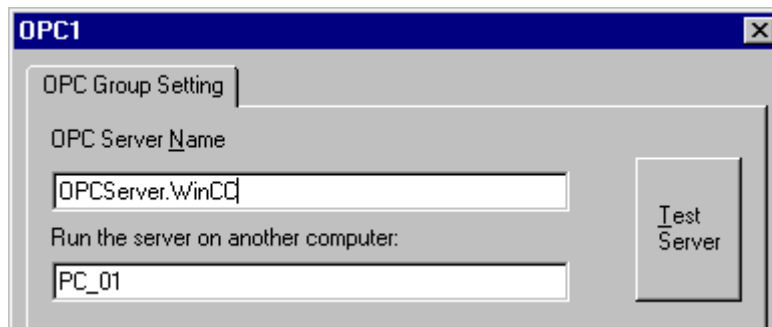
For connections to WinCC V 6, the entry in the "OPC Server Name" field must be "OPCServer.WinCC".

3. Enter the name of the computer to be used as the OPC DA server in the "Start Server on this Computer" field. Click "Test Server", to check the connection to the WinCC OPC DA server.

4. Select "New Tag" from the shortcut menu of the connection. The "Tag Properties" dialog opens.

5. Enter the name "Client_OPC_Var1_xyz" in the "Tag" field. Set the data type to "signed 16-bit".

6. In the "Tag Properties" dialog, click the "Select" button. The "Address Properties" dialog opens.

```
Address properties                              [X]
 ┌─ OPC Address ─┐
 │                │
   Item Name:      │OPC_Var1                    │

   Access Path:    │                            │

   Data Type:      │Signed 16-bit value      │▼│
```
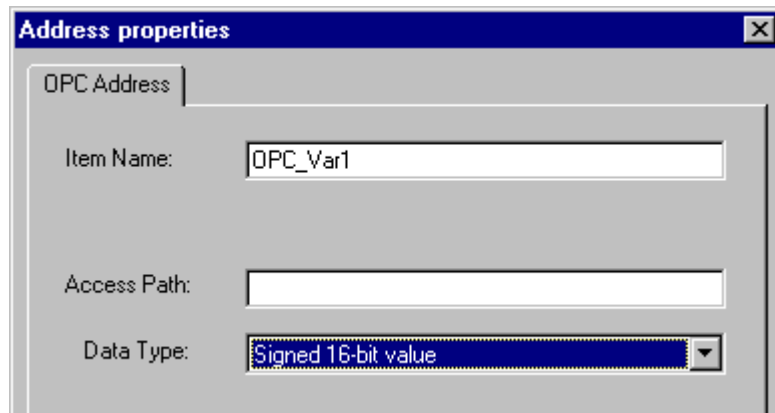
Enter the name of the WinCC tag of the WinCC OPC DA server in the "Item Name" field. Leave the entry in the "Access Path" field unchanged. Set the data type to "signed 16-bit".

7. Click "OK" to close all open dialogs.

5.4.4 Using Structures on a WinCC OPC DA Client

5.4.4.1 How to Use Structures on the WinCC OPC DA Client

Introduction

Structures are used to organize tags and tag types that form a logical unit. This allows them to be referenced using a single logical name.

Structures are not supported by the OPC DA specification. As a result, structures cannot be set up using the OPC Item Manager, only the individual tags in a structure. If you wish to use structures on the WinCC OPC DA client nonetheless, the data structure must be configured subsequently in the WinCC project of the WinCC OPC DA client in order to supply it with the relevant item names of the server tags.

Requirements

- Two computers with WinCC projects.
- Both computers must be accessible via their IP addresses.

Configuration steps

The following configuration steps are necessary to use structures on the WinCC OPC DA client:

- Configuring structures and structure tags on the WinCC OPC DA server
- Using structures on the WinCC OPC DA client in the WinCC project

See also

5.4.4.2 Configuring Structures and Structure Tags on the WinCC OPC DA Server

Introduction

In this section a structure and a structure tag is created in the WinCC project of the OPC DA server. This configuration is required for the OPC DA client to access the structure tag.

Procedure

1. Select "New Structure Type" from the structure types shortcut menu on the WinCC OPC DA server. The "Structure Properties" dialog is displayed.

2. Click "New Element" and create the internal tag "OPCServer_Struct" of data type SHORT.
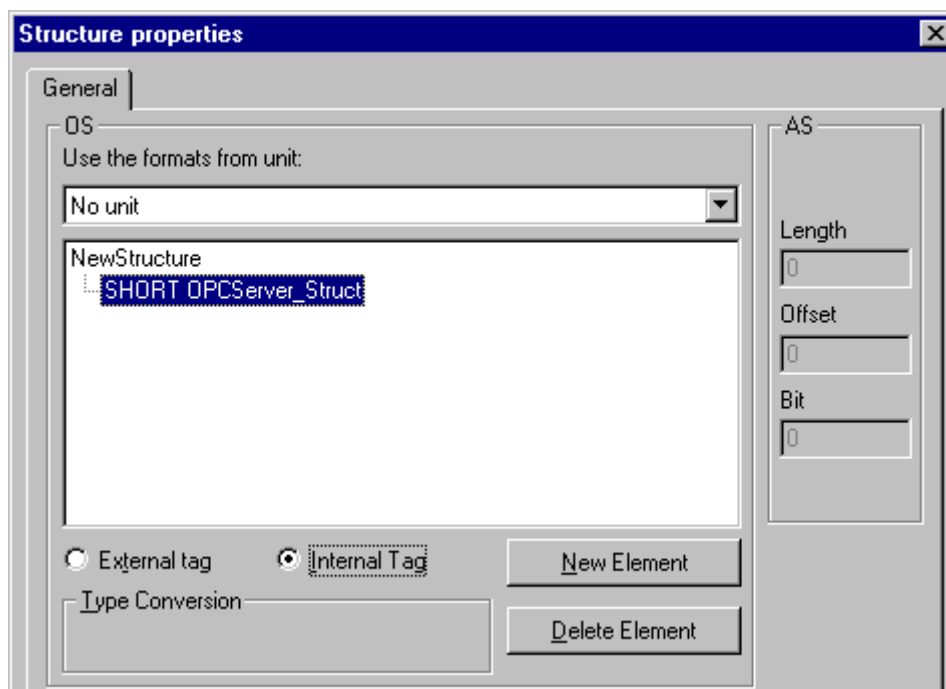
Click "OK" to close the dialog.

3. In the navigation window, click the plus sign in front of the icon for tag management. Select "New Tag" from the internal tag shortcut menu. Create a WinCC tag named "Var" with this structure type.

4. The data frame of the WinCC Explorer shows the single tag "Var" and the structure tag "Var.OPCServer_Struct".

5. Activate the WinCC project.

See also

How to Configure Structures on the WinCC OPC DA Client (Page 96)

5.4.4.3   How to Configure Structures on the WinCC OPC DA Client

Introduction

Structures are not supported by the OPC DA specification. As a result, structures cannot be set up using the OPC Item Manager. In this section, the structure already present in the WinCC project of the WinCC OPC DA server is configured for the WinCC project of the WinCC OPC DA client. A WinCC tag that accesses the existing structure tag on the WinCC OPC DA server is configured on the WinCC OPC DA client.

Requirements

- Create a structure and a structure tag named "Var.OPCServer_Struct" in the WinCC project of the WinCC OPC DA server.

- Enable the WinCC project of the WinCC OPC DA server.

- Add the OPC channel to the WinCC project of the WinCC OPC DA client.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. Select "New Structure Type" from the structure types shortcut menu on the WinCC OPC DA client. The "Structure Properties" dialog is displayed.

2. Click the "New Element" button and set up an external tag. Name the element exactly as it is in the WinCC project of the OPC-DA server. Click "OK" to close the "Structure Properties" dialog.

3. If a connection to the OPC DA server already exists, continue with step 6.
   If no connection has been created, select "New Connection" from the shortcut menu of the channel unit "OPC". The "Connection Properties" dialog is opened. Enter a name for the connection in the corresponding field.

4. Click the "Properties" button. A dialog with the connection name in its title is displayed. For connections to WinCC V 6, the entry in the field "OPC Server Name" must be "OPCServer.WinCC".

5. Enter the name of the computer to be used as the WinCC OPC DA server in the field "Start Server on this Computer". Click "Test Server", to check the connection to the WinCC OPC DA server. Click "OK" to close the dialog.

6. Select "New Tag" from the shortcut menu of the connection. The "Tag Properties" dialog opens. Select the newly created structure type as the data type.

7. In the "Tag Properties" dialog, click the "Select" button. The "Address properties" dialog opens.
   In the "Item Name" field, enter the name "Var.OPCServer_Struct" for the structure tag of the WinCC OPC DA server. Leave the entry in the "Access Path" field unchanged.

8. Click "OK" to close all open dialogs.

See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 88)

Configuring Structures and Structure Tags on the WinCC OPC DA Server (Page 95)

5.4.5 Error Handling in the Event of Disturbed OPC DA Communication

5.4.5.1 Error Handling in the Event of Disturbed OPC Communication

Introduction

The procedure for communication testing is independent of how WinCC is used.

WinCC Used as the OPC DA Server

Use the channel diagnostics on the WinCC OPC DA client to determine whether a connection to the OPC DA server can be established. For more information regarding channel problem analysis, refer to "Troubleshooting".

WinCC Used as the OPC DA Client

Use the channel diagnostics on the WinCC OPC DA client to determine whether a connection to the OPC DA server can be established. For more information regarding channel problem analysis, refer to "Troubleshooting".

See also

WinCC is used as the OPC DA client, and the connection is not established. (Page 101)

WinCC is used as the OPC DA client, and the connection is established. (Page 100)

WinCC is used as the OPC DA server, and the connection is not established. (Page 99)

WinCC is used as the OPC DA server, and the connection is established successfully. (Page 98)

Possibilities for Diagnosing the "OPC" Channel (Page 404)

5.4.5.2 WinCC as OPC DA Server

WinCC is used as the OPC DA server, and the connection is established successfully.

```
┌─────────────────────────────────┐
│ WinCC is used as the OPC DA      │
│ server.                          │
│ A connection is established but  │
│ the value of the tag is incorrect.│
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│ Check the configuration of the  │
│ item name and the data type of   │
│ the OPC DA client.               │
└─────────────────────────────────┘
                 │
                 ▼
          Are the item             No    ┌──────────────────────┐
          name and data    ────────────▶ │ Correct the entries. │
          type correct?                  └──────────────────────┘
                 │ Yes
                 ▼
          Is the                   Yes   ┌──────────────────────┐
        "Access path" field ─────────────▶│ Delete the entry.    │
          empty?                          └──────────────────────┘
                 │ No
                 ▼
          Are the                  No    ┌───────────────────────────────────┐
          DCOM settings    ─────────────▶│ Change the configuration on the    │
          correct?                       │ WinCC DA OPC server.               │
                 │ Yes                   │ You can find further information in │
                 ▼                       │ the documentation of the operating │
┌───────────────────────┐               │ system.                            │
│ Contact the WinCC      │              └───────────────────────────────────┘
│ Customer Support.      │
└───────────────────────┘
```

Check if the correct value is now displayed.

Yes → You can now use the OPC connection for data communication.

No → Check the OPC DA client.

WinCC is used as the OPC DA server, and the connection is not established.

WinCC is used as the OPC DA server.
Unable to establish a connection.

Open the registration editor. In the "Registration" menu, click "Connect with network registration ". Enter the name of the computer on which the OPC DA server is running in the "Computer name" field. Click the "OK" button.

Can you connect to the network registration of the OPC DA server?

No — Is the computer available on the network?

No → Make sure the computer is available on the network.

Yes

Is Runtime active on the WinCC OPC DA server?

No → Activate the WinCC project.

Yes

Open Task Manager on the WinCC DA OPC server. In the "Processes" tab, is the "SOPCSERVWinCC" process active?

No → Are the configured ProgID and the server name of the client application correct?

Yes

No → Correct the entries.

Yes

Change the configuration on the WinCC DA OPC server.
You can find further information in the documentation of the operating system.

No — Are the DCOM settings correct?

Yes

Contact the WinCC Customer Support.

Check if you can establish a communication connection now.

Yes → You can now use the OPC connection for data communication.

No

Check the OPC DA client.

5.4.5.3 WinCC as OPC DA Client

WinCC is used as the OPC DA client, and the connection is established.

```
┌──────────────────────────────────────────┐
│  WinCC is used as the OPC DA client.      │
│  A connection is being established, but    │
│  the tag value is incorrect.              │
└──────────────────────────────────────────┘
                    │
                    ▼
        If the OPC DA server
  Yes   has a browser interface, are   No
◄────── the tags of the OPC DA server ──────►
        displayed in the OPC Item Manager?
```

In the WinCC project of the OPC DA client, open the address properties of the WinCC tag that accesses the tag of the OPC DA server.

Enable the WinCC project of the OPC DA client. Start the WinCC "Channel Diagnosis" from the Start menu. Enable the trace function.

In the "Address Properties" dialog, are the entries in the fields "Item Name" and "Data Type" correct? → Yes → Are the DCOM settings correct? → Yes → Contact the WinCC Customer Support.

No

No

Correct the entries.

Change the configuration on the WinCC DA OPC client. For additional information, refer to the documentation of the operating system.

Check whether you can establish the communication connection now. → Yes → You can now use the OPC connection for data transmission.

No

Check the OPC DA server.

WinCC is used as the OPC DA client, and the connection is not established.

```
                    ┌──────────────────────────────┐
                    │ WinCC is used as the OPC DA   │
                    │ client.                       │
                    │ Unable to establish a         │
                    │ connection.                   │
                    └──────────────────────────────┘

┌──────────────────────┐              ╱ Is the   ╲
│ Go to the "Properties │             ╱ computer   ╲    No   ┌──────────────────────┐
│ Connection" dialog on │            ⟨ available on the ⟩────────│ Make sure the computer │
│ the OPC DA client and │             ╲ network?  ╱           │ is available on the    │
│ check the entries in   │             ╲         ╱            │ network.              │
│ the"OPC Connection" tab│               Yes                  └──────────────────────┘
└──────────────────────┘

      ╱ Are the ProgID ╲                ╱ Are the   ╲   Yes   ┌──────────────────────┐
     ⟨ and the computer  ⟩── Yes        ⟨ DCOM settings ⟩────────│ Contact the WinCC     │
      ╲ name correct? ╱                  ╲ correct?  ╱          │ Customer Support.     │
          No                                 No                └──────────────────────┘

┌──────────────────────┐              ┌────────────────────────────┐
│ Correct the entries.  │             │ Change the configuration on │
└──────────────────────┘             │ the WinCC DA OPC client.    │
                                      │ You can find further        │
                                      │ information in the          │
                                      │ documentation of the        │
                                      │ operating system.           │
                                      └────────────────────────────┘

                         ╱ Check if the ╲    Yes    ┌──────────────────────┐
                        ⟨ correct value is  ⟩────────│ You can now use the OPC│
                         ╲ now displayed. ╱          │ connection for data    │
                             No                      │ communication.         │
                                                     └──────────────────────┘
                    ┌──────────────────┐
                    │ Check the OPC DA  │
                    │ server.           │
                    └──────────────────┘
```

5.5 WinCC OPC XML Client

5.5.1 Functionality of the WinCC OPC XML Client

Introduction

The OPC channel does not require a separate communication module. The OPC communication driver can be implemented as the OPC XML client.

In order to use WinCC as the WinCC OPC XML client, the OPC channel must be added to the WinCC project.

The WinCC OPC XML client provides the OPC XML server with the OPC process data as a web page. Access can be made to the web page via the Internet / Intranet using HTTP. When a WinCC OPC XML client requests data, the web service is automatically started by the web server.

In order that the WinCC OPC XML client can access several OPC XML servers, a connection must be made to each OPC XML server in the WinCC project.

If a communication is established to a WinCC OPC XML server, the values of the WinCC tags are exchanged. A connection is set up in the WinCC project of the WinCC OPC XML client via which access to the WinCC OPC XML server can be processed. The connection monitoring is not activated in the case of a WinCC OPC XML client.

Note

The WinCC OPC channel establishes connections only to OPC servers which have the status "OPC_STATUS_RUNNING".

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

See also

Overview of the Supported WinCC Data Types (Page 85)

OPC Item Manager (Page 81)

5.5.2 How to Access a WinCC Tag with the OPC Item Manager

5.5.2.1 How to Access a WinCC Tag with the OPC Item Manager

Introduction

When an OPC connection is made between WinCC and WinCC, the data is exchanged by means of WinCC tags. The WinCC OPC DA client uses an OPC connection to read the WinCC tag "OPC_Server_Tag" on the WinCC OPC XML server. To simplify the process, the OPC Item Manager is used.
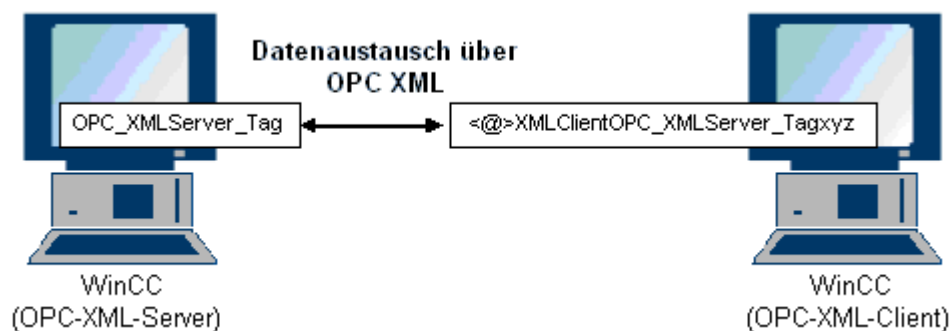
Requirements

- Two computers with WinCC projects.
- It must be possible to access both computers using HTTP.

Configuration Steps

The following configurations are required in the WinCC project of the WinCC OPC XML client:

See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 88)

Configuring Access with the OPC Item Manager (Page 103)

5.5.2.2 Configuring Access with the OPC Item Manager

Introduction

This section explains how to use the OPC Item Manager to configure access to the "OPC_XMLServer_Tag" WinCC tag of the WinCC OPC XML server.

Requirements

- Configure an internal tag named "OPC_XMLServer_Tag" of data type "signed 16-bit value" in the WinCC project of the WinCC OPC XML server.

- Activate the WinCC project of the WinCC OPC XML server.

- The OPC channel must be added to the WinCC project of the WinCC OPC XML client.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. Select "System Parameters" from the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC XML client. The "OPC Item Manager" opens.
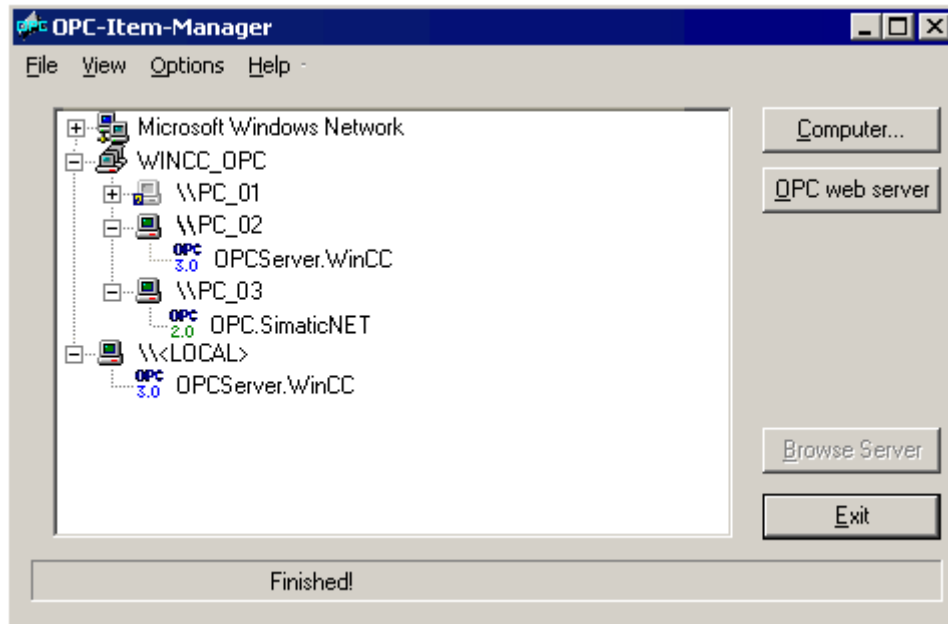
2. Click the "OPC Web Server" button. The "Add OPC Web Server" dialog is opened. Enter the URL of the WinCC OPC XML server in the "URL" field in following format: <http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx>. Replace xxx with either the IP address or the computer name on which the OPC XML web service is running.
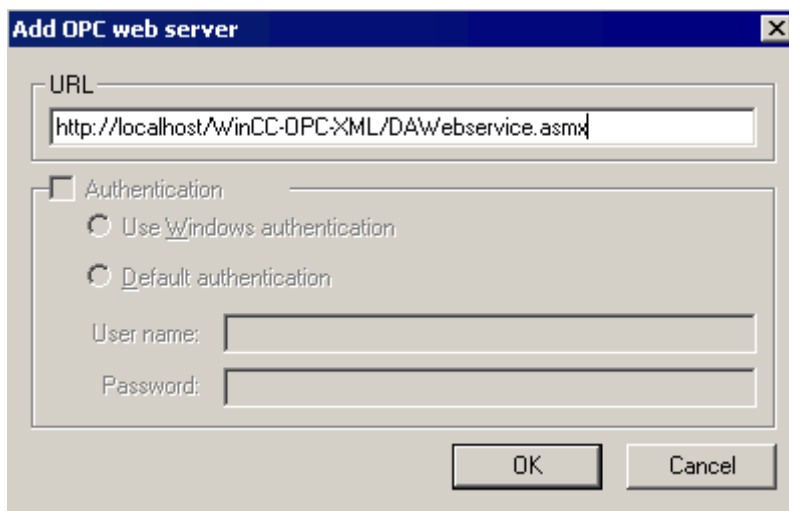
Click "OK" to close the dialog.

3. A list appears from which to select "/WinCC-OPC-XML/DAWebservice.asmx>". Click the "Browse Server" button. The "Filter criteria" dialog is opened.
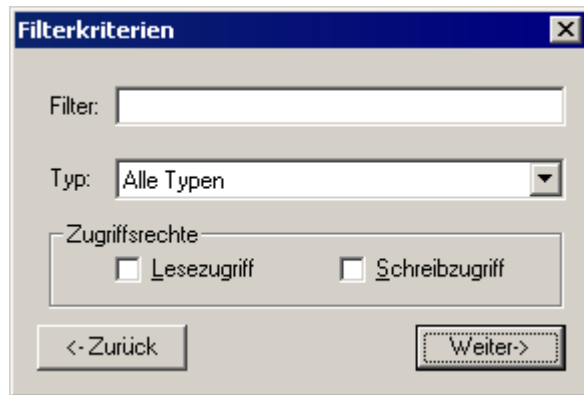
4. Click the "Next->" button in the "Filter Criteria" dialog. The "http:// ..." dialog is opened.
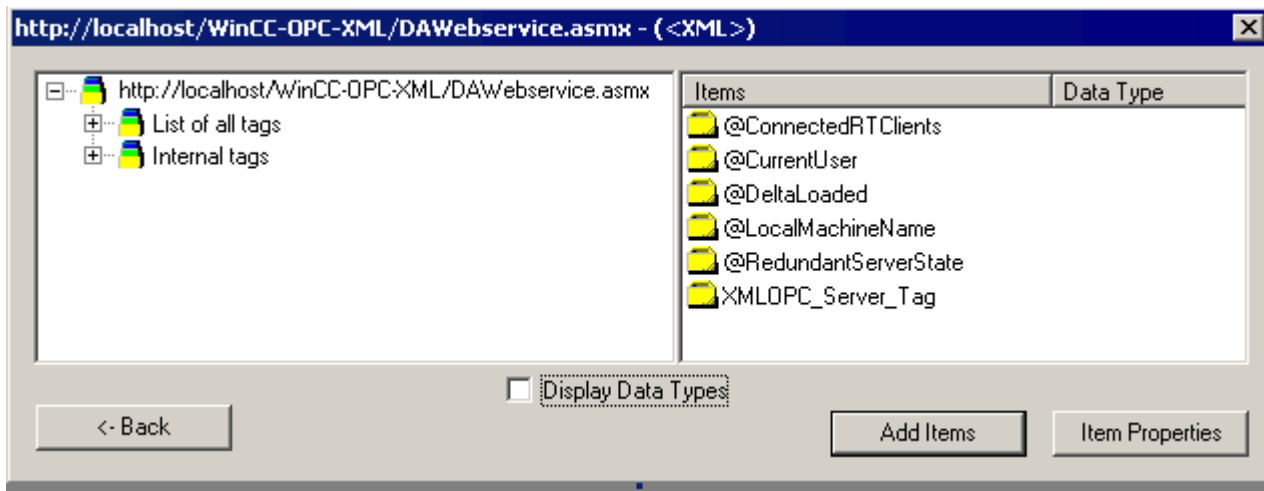
5. In the "http:// ..." dialog, select the WinCC tag "XMLOPC_Server_Tag". Click the "Add Items" button.

6. If a connection to the WinCC OPC XML server already exists, continue with step 7. If no connection has been configured, a corresponding message is output.
Click "Yes". The "New Connection" dialog is opened.
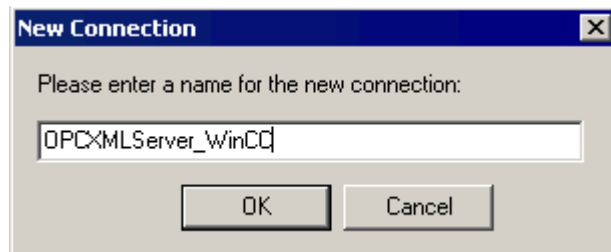
Enter the name "OPCXMLServer_WinCC" for the connection. Click "OK".

7. The "Add Tags" dialog opens.
   Enter the string "XMLClient_" in the prefix field, and "_xyz" in the suffix field. Select the
   "OPCXMLServer_WinCC" connection. Click "Finish".
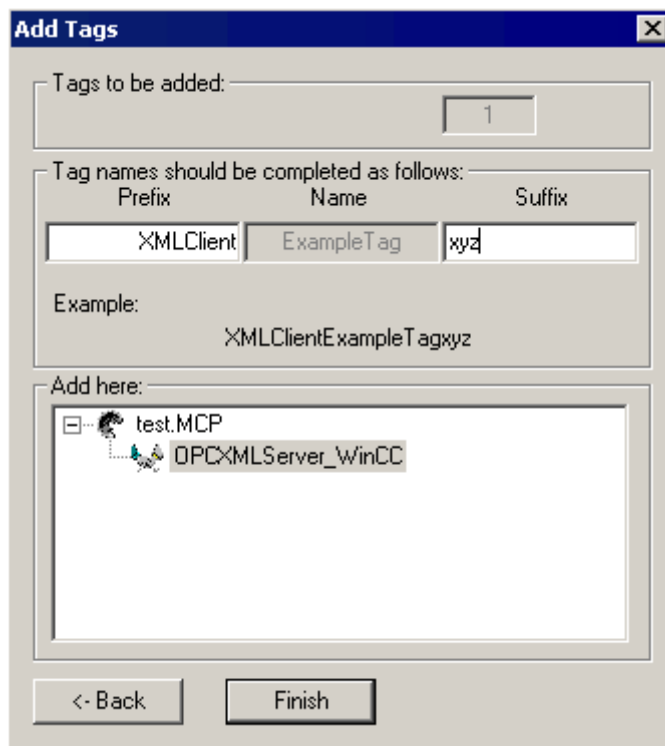
8. In the "http:// ..." dialog, click "<- Back". Click "Exit" to close the OPC Item Manager.

See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 88)

5.5.3 Accessing a WinCC Tag without the OPC Item Manager

Introduction

OPC servers that do not support browser functionality require access to be configured
manually. Configuration of WinCC tags on the WinCC OPC XML client is shown using an
example of a WinCC - WinCC OPC connection.

Note

To access a WinCC tag without the OPC Item Manager, the ItemID must be set manually.
When addressing WinCC tags, the symbolic computer name (server prefix) can also be
specified. The ItemID has the following syntax: server_prefix::<@>WinCC tag. If the WinCC
tag of the local WinCC project is addressed, the ItemID has the following syntax: <@>WinCC
tag.

Configuration Steps

The following configurations are required in the WinCC project of the WinCC OPC XML client:

1. Creation of a connection.

2. Configuring the "XMLClient_OPC_Var1_xyz" WinCC tag on the WinCC OPC XML client which accesses the WinCC tag of the WinCC OPC DA server.

Requirements

- Two computers with WinCC projects.

- It must be possible to access the computers using HTTP.

- Configure an internal tag named "XMLOPC_Server_Tag" with data type "signed 16-bit value" in the WinCC project of the WinCC OPC XML server.

- Enable the WinCC project of the WinCC OPC XML server.

- The OPC channel must be added to the WinCC project of the WinCC OPC XML client.

  ### Note

  When configuring external tags in the OPC channel, the preset values from WinCC in the type conversion field of the "Tag Properties" dialog must not be altered. The data type of the tag in the process is set in the data type field of the "Address Properties" dialog.

Procedure

1. Select "New Connection" from the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC XML client. The "Connection Properties" dialog is opened. Enter a name for the connection in the corresponding field.

2. Click the "Properties" button. A dialog with the connection name in its title is displayed.
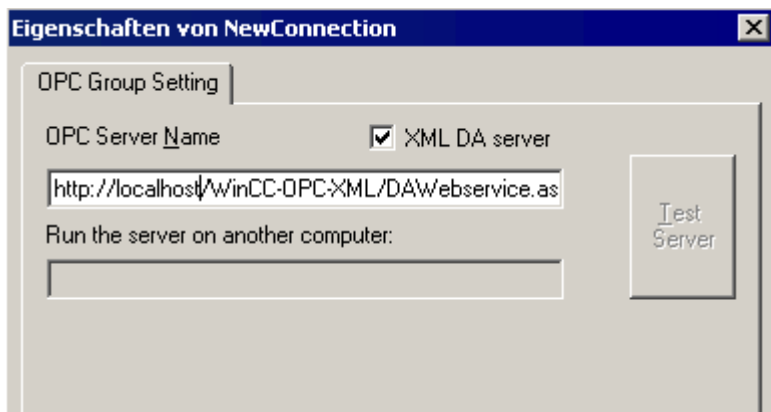
   Select the "XML DA Server" check box. In the case of a connection to the WinCC OPC XML server, the "OPC Server Name" field must contain the URL of the WinCC OPC XML server. The URL has the following syntax: "http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx". Replace xxx with either the IP address or the computer name on which the OPC-XML web service is running.

3. Select "New Tag" from the shortcut menu of the connection. The "Tag Properties" dialog is opened.

4. Enter the name "XMLClient_OPC_Var1_xyz" in the "Tag" field. Set the data type to "signed 16-bit".

5. In the "Tag Properties" dialog, click the "Select" button. A dialog with the tag name in its title is displayed.
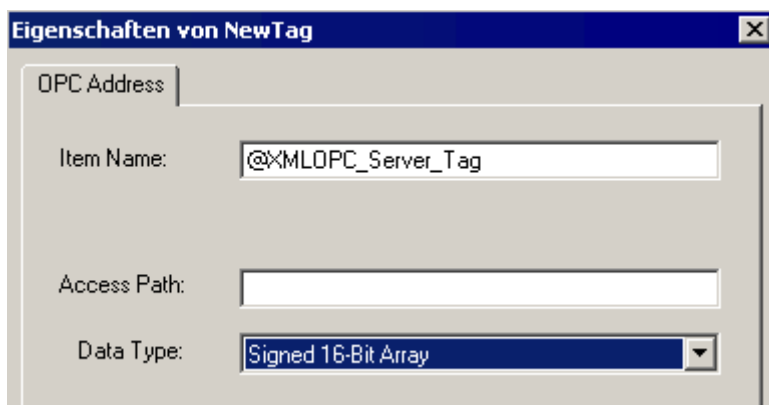
Enter the symbol "<@>" and the name of the WinCC tag of the WinCC OPC XML server in the "Item Name" field. Leave the entry in the "Access Path" field unchanged. Set the data type to "signed 16-bit".

6. Click "OK" to close all open dialogs.

5.6 WinCC OPC UA client

5.6.1 Functionality of the WinCC OPC UA client

Introduction

The WinCC OPC UA client enables data access to any OPC UA server in accordance with the OPC Unified Architecture specification. OPC Unified Architecture (OPC UA) also provides additional mechanisms such as authentication and encryption for secure communication between participating partners.

To use WinCC as WinCC OPC UA client, add the OPC communication driver (OPC channel) to the WinCC project. A separate communication module is not required.

To enable access to the OPC process data of different OPC UA servers, configure a connection to each server in the WinCC project. Use the WinCC OPC Item Manager to configure the connections.

Note

Connection monitoring is not activated if the OPC channel is used as OPC UA client.

Connections are only established to servers which signal the status "RUNNING".

5.6.2 Accessing a tag using the OPC Item Manager

5.6.2.1 Accessing an OPC tag using the OPC Item Manager

Introduction

If there is an OPC link from WinCC to an OPC UA server, OPC UA server data is mapped to a WinCC tag. In this process, the WinCC OPC UA client uses a secure connection to access the "OPC_UAServer_Tag" of the WinCC OPC UA server. Use OPC Item Manager for your configuration.

Note

You can only access servers that offer browse support during the configuration.

Distinguish between client and server certificates when configuring. Secure communication is only possible when client and server recognize each other's certificates.

Certificates are linked to the respective computers. After having moved, copied, or duplicated the WinCC project to a different computer, repeat this procedure so that each computer can verify the other's certificates.

Requirements

- Client computers with a WinCC project.
- Both computers must be accessible via TCP connection.
- The Firewall allows two-way communication with the UA server.

Configuration Steps

Complete the following configuration tasks in the WinCC project of the WinCC OPC UA client:

- Configuration of the OPC channel on the WinCC OPC UA client.
- Setup of a valid server certificate.
- Configuration of access with the OPC Item Manager.

5.6.2.2   How to set up a server certificate

Introduction

This section explains how to set up an OPC UA server certificate for the WinCC OPC UA client.

Requirements

- The OPC UA server is activated.
- The Firewall allows two-way communication with the UA server.
- The "OPC" channel has been added to the WinCC project of the WinCC OPC UA client.

Procedure

1. On the WinCC OPC UA client, select the "System parameters" command from the shortcut menu of the "OPC Groups(OPCHN Unit#1)" channel unit. The "OPC Item Manager" opens.
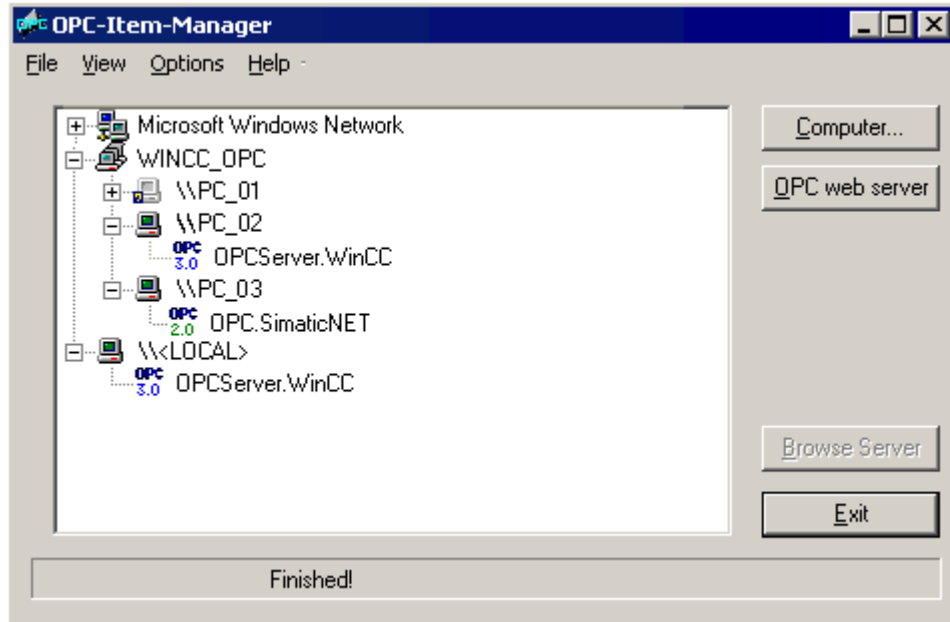

2. Click "OPC UA server". The "Add OPC UA server" dialog opens. Enter the URL of the WinCC OPC UA server in the "Server URL" field in the following format:
<opc.tcp://OPC UA server address>
Make the required security settings. If necessary, enter authentication data.
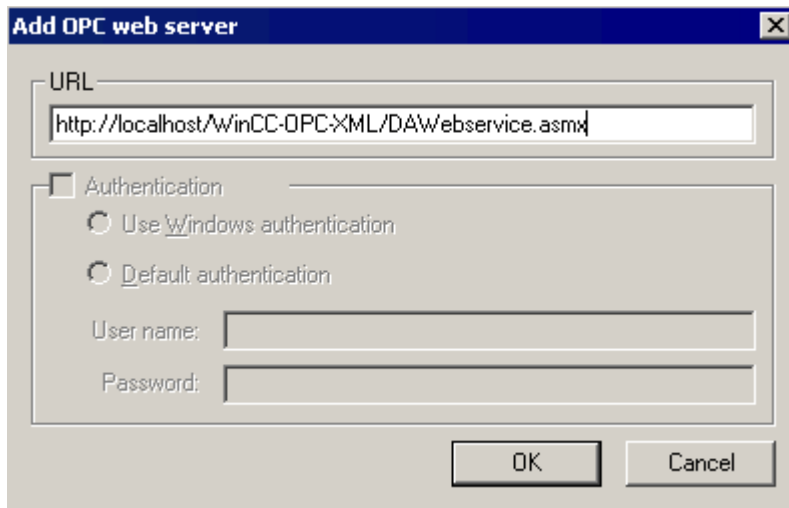
Close the dialog with "OK".

3. Select the "<opc.tcp://OPC UA server address>" entry from the list displayed.

4. Click "Browse Server".
An error dialog opens.

5. Close the error dialog.

6. Open Windows Explorer and change to the "OPC" folder in the WinCC installation path.

7. Open the "PKI/CA/" folder.
The "rejected" folder containing the rejected server certificate has been added.

8. Move the server certificate from the "rejected" folder to the "certs" folder.
The server certificate has been validated. It is now possible to activate a secure connection to the OPC UA server.

5.6.2.3   How to configure access to a tag using the OPC Item Manager

Introduction

This section explains how to use OPC Item Manager to configure access to the "OPC_UAServer_Tag" WinCC tag of the WinCC OPC UA server.

Note

You can only access servers that offer browse support during the configuration.

Requirements

- The OPC UA server is activated.

- The Firewall allows two-way communication with the UA server.

- The server certificate has been set up.

- The "OPC" channel has been added to the WinCC project of the WinCC OPC UA client.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. On the WinCC OPC UA client, select "System parameters" from the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)". The "OPC Item Manager" opens.
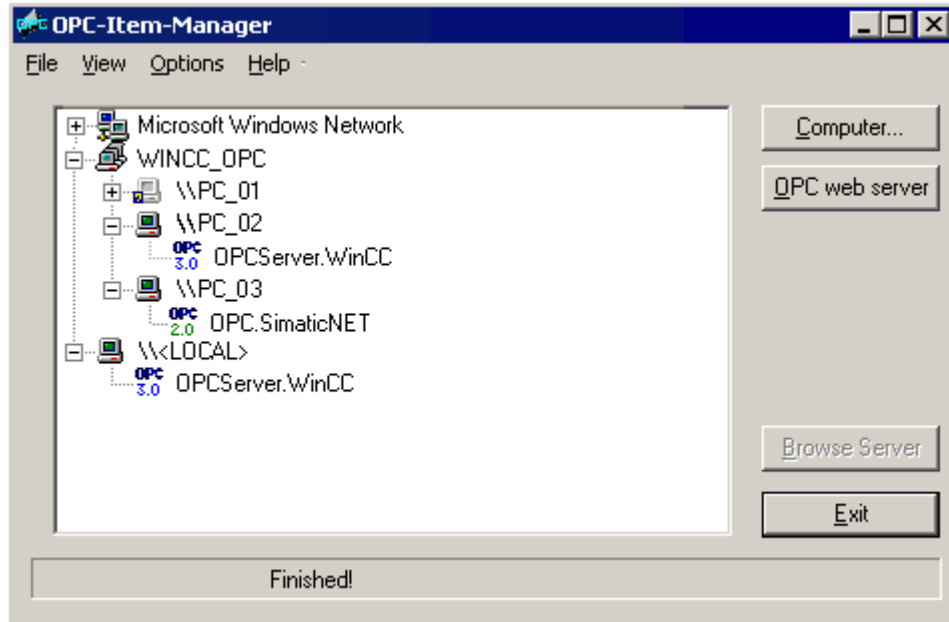
2. Click "OPC UA server". The "Add OPC UA server" dialog opens. Enter the URL of the WinCC OPC UA server in the "Server URL" field in the following format:
   <opc.tcp://OPC UA server address>
   Make the required security settings. If necessary, enter authentication data.
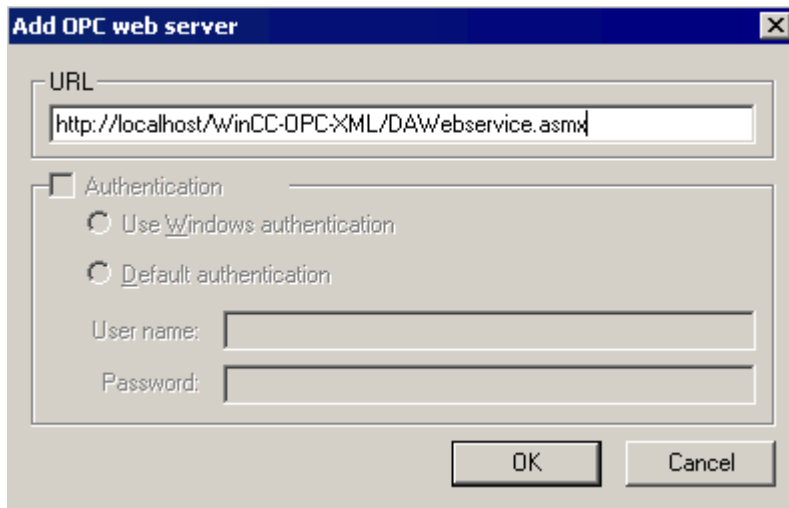
Click "OK" to close the dialog.

3. Select the "<opc.tcp://OPC UA server address>" entry from the list displayed. Click "Browse Server". The "Filter criteria" dialog is opened.

4. Click "Next". The "opc.tcp:// ..." dialog opens.
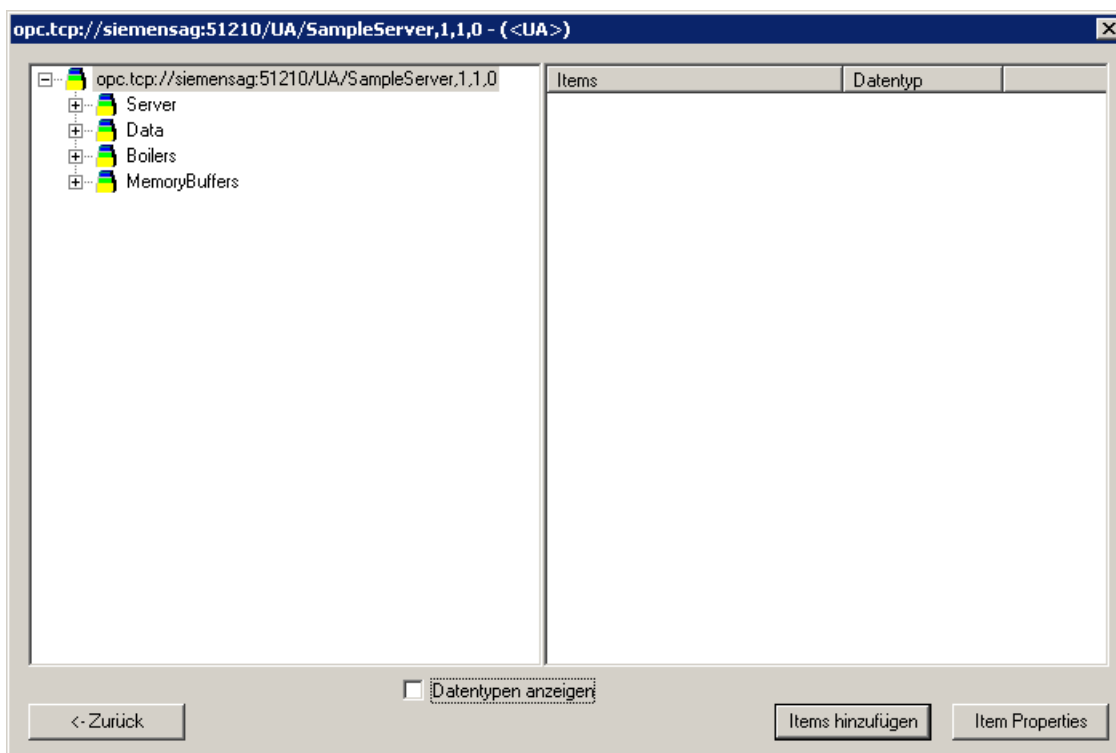
5. Select the tag to be mapped in the "opc.tcp:// ..." dialog, e.g. "OPC_UA_Server_Tag". Click "Add items".

6. Continue with step 7 if you have already set up a connection to the WinCC OPC UA server.
   A message will be output if this connection is not available.
   Click "Yes". The "New Connection" dialog is opened.
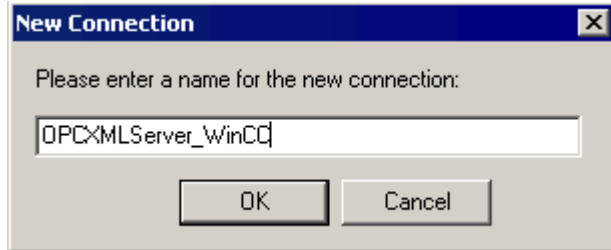
   Enter the connection name "OPCUAServer". Click "OK".

7. The "Add Tags" dialog opens.
   Enter the "UAClient_" string in the "Prefix" field and the "xyz" string in the "Suffix" field.
   Select the "OPCUAServer" connection. Click "Finish".
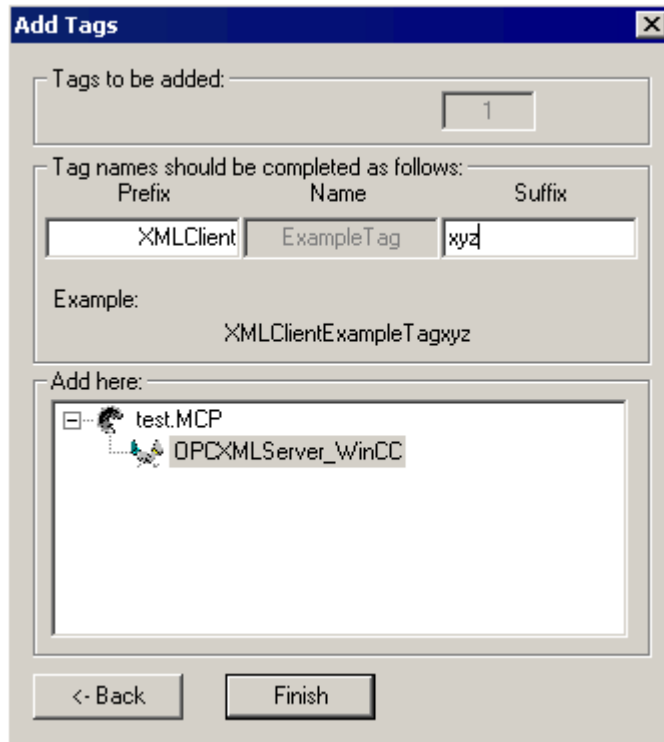
8. Click "Back" in the "opc.tcp:// ..." dialog.
   Click "Exit" in the "OPC Item Manager" dialog to close the OPC Item Manager.

5.6.3 Error handling in event of disturbed communication

Introduction

The procedure for communication testing is independent of how WinCC is used.

WinCC used as OPC UA server

First, run the channel diagnostics on the WinCC OPC UA client to check whether it is possible to set up a connection to the OPC UA server. For more information regarding channel problem analysis, refer to "Troubleshooting".

WinCC used as OPC UA client

First, run the channel diagnostics on the WinCC OPC UA client to check whether it is possible to set up a connection to the OPC UA server. For more information regarding channel problem analysis, refer to "Troubleshooting".

5.6.3.1 WinCC as OPC UA server

WinCC is used as OPC UA server, and the connection is set up.

```
┌─────────────────────────────┐
│ WinCC is used as OPC UA      │
│ server. A connection is      │
│ being established, but the   │
│ tag value is incorrect or    │
│ the subscription does not    │
│ provide any data             │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ Check the configuration of   │
│ the node ID and the data     │
│ types of the OPC UA client.  │
└─────────────────────────────┘
              │
              ▼
      ╱ Are the ╲          No      ┌──────────────────────┐
     ╱ node ID and the ╲─────────▶│ Correct the entries  │
     ╲ data type correct? ╱        │ in the OPC UA client.│
      ╲         ╱                  └──────────────────────┘
        │ Yes
        ▼
      ╱ Are the ╲          No   ┌──────────────────────────────────┐
     ╱ DCOM settings ╲────────▶│ Change the configuration on the  │
     ╲ correct?      ╱          │ WinCC UA OPC server.             │
      ╲           ╱             │ For additional information, refer│
        │ Yes                  │ to the documentation of the      │
        ▼                      │ operating system.                │
┌─────────────────────┐        │ The SIMATIC HMI group must have  │
│ Contact the WinCC   │        │ start and access permission to   │
│ Customer Support.   │        │ the OPC UA server, among other   │
└─────────────────────┘        │ things.                          │
                               └──────────────────────────────────┘
```

```
      ╱ Check        ╲     Yes   ┌──────────────────────┐
     ╱ whether the    ╲────────▶│ You can now use the  │
     ╲ correct value   ╱         │ OPC connection for   │
      ╲ is now displayed.╱       │ data transmission.   │
        │ No                     └──────────────────────┘
        ▼
┌──────────────────────────┐
│ Check the OPC UA client. │
└──────────────────────────┘
```

WinCC is used as the OPC UA server, and the connection is not set up.

```
┌─────────────────────────────────────┐
│ WinCC is used as the OPC UA server.  │
│ Unable to establish a connection.    │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────────────────┐
│ Open the registration editor. In the            │
│ "Registration" menu, click                       │
│ "Connect with network registration ". Enter the  │
│ name of the computer on which the OPC UA server  │
│ is running in the "Computer name" field. Click   │
│ the "OK" button.                                 │
└─────────────────────────────────────────────────┘
```

Can you connect to the network registration of the OPC UA server?

Is the computer available on the network?

No → Make sure the computer is available on the network.

Is WinCC in Runtime on the OPC UA server?

No → Start WinCC Runtime on the WinCC OPC UA server.

Open "Services" in Computer Management. Check if the "OpcUAServer-WinCC" has started or is running.

No → If the service of the WinCC OPC UA server has ended, start the service manually. Otherwise contact WinCC Customer Support

Are the server/client certificates set up correctly ?

No → Correct the setting.

Contact the WinCC Customer Support.

Check if you can establish a communication connection now.

Yes → You can now use the OPC connection for data communi-cation.

No → Check the OPC UA client.

5.6.3.2   WinCC as OPC UA client

WinCC is used as the OPC UA client, and the connection is set up.

```
┌──────────────────────────────────────┐
│  WinCC is used as the OPC UA client.  │
│  A connection is established but the  │
│  value of the tag is incorrect.       │
└──────────────────────────────────────┘
```

WinCC is used as the OPC UA client.
A connection is established but the value of the tag is
incorrect.

If the OPC UA server has a browser interface, are the tags of the OPC UA server displayed in the OPC Item Manager?

Yes — No

In the WinCC project of the OPC UA client, open the address properties of the WinCC tags that access the tag of the OPC UA server.

Activate the WinCC project of the OPC UA client. Start WinCC "Channel Diagnosis" from the Start menu. Activate the trace function.

In the "Properties Address" dialog, are the entries in the "Item name" and "Data type" fields correct?

Yes

Are the DCOM settings correct?

Yes

Contact the WinCC Customer Support.

No

No

Correct the entries.

Change the configuration on the WinCC UA OPC client.
You can find further information in the documentation of the operating system.

Check if the correct value is now displayed.

Yes

You can now use the OPC connection for data communication.

No

Check the OPC UA server.

WinCC is used as OPC UA client, and the connection is not set up.

```
                        ┌─────────────────────────────┐
                        │ WinCC is used as the OPC UA  │
                        │ client.                      │
                        │ A connection is not          │
                        │ established.                 │
                        └─────────────────────────────┘
```

WinCC is used as the OPC UA client. A connection is not established.

Check the entries in the "Connection Properties" dialog of the "OPC Connection" tab on the OPC UA client.

Are the OPC UA server address and the port number correct?

Correct the entries.

Is the computer available in the network?

No — Make sure that the computer is available in the network.

Check whether the OPC UA server supports the security setting.

No — Correct the settings.

Are the server and/or client certificates set up correctly?

Are the DCOM settings correct?

Yes — Contact the WinCC Customer Support.

Change the configuration on the WinCC UA OPC client.
For additional information, refer to the documentation of the operating system.

Check whether you can establish the communication connection now.

Yes — You can now use the OPC connection for data transmission.

No — Check the OPC UA server.

PROFIBUS FMS

<div style="text-align: right; font-size: large;">6</div>

6.1 WinCC channel "PROFIBUS FMS"

Introduction

The "PROFIBUS FMS" communication driver makes the connection between a WinCC station and automation systems such as S5 or S7.

This chapter describes

● how to configure data transfer with the "PROFIBUS FMS" channel

● how to configure a connection and a tag.

Channel Unit

The communication driver has a channel unit. This provides the following capability:

● Channel unit PROFIBUS FMS for SIMATIC NET PROFIBUS (communication module DP 5613).

> **Note**
>
> The configuration of the logical connection can take place in configuration mode as well as in runtime. Both methods are explained.
>
> Only WinCC can request data from the connected AS with the "PROFIBUS FMS" channel. Sending from the AS is not possible.

6.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag
- Unsigned 8-bit value
- Signed 8-bit value
- Unsigned 16-bit value
- Signed 16-bit value
- Unsigned 32-bit value
- Signed 32-bit value
- Floating-point number 32-bit IEEE 754
- Text tag, 8-bit character set.
- Raw data type

6.3 Configuring the Channel

6.3.1 Configuring a connection

6.3.1.1 Configuring a connection

Introduction

The automation system must be connected to PROFIBUS with a suitable communication processor. The PROFIBUS FMS protocol must be supported by the hardware / software.

In automation systems S5-115U, S5-135U and S5-155U, communication processor CP 5431 FMS is used for instance.

in WinCC system, the communication module CP 5613 and software SIMATIC NET must be installed.

When configuring the connection, the following situations are handled differently:

● Configuring in configuration mode

● Configuring in Runtime

6.3.1.2   Configuring the connection in configuration mode

Procedure

1. Select the connection and open the "Connection properties" dialog window with menu point "Properties" in the context menu.

2. Click the "Properties" button. A dialog with the connection name in its title is displayed.
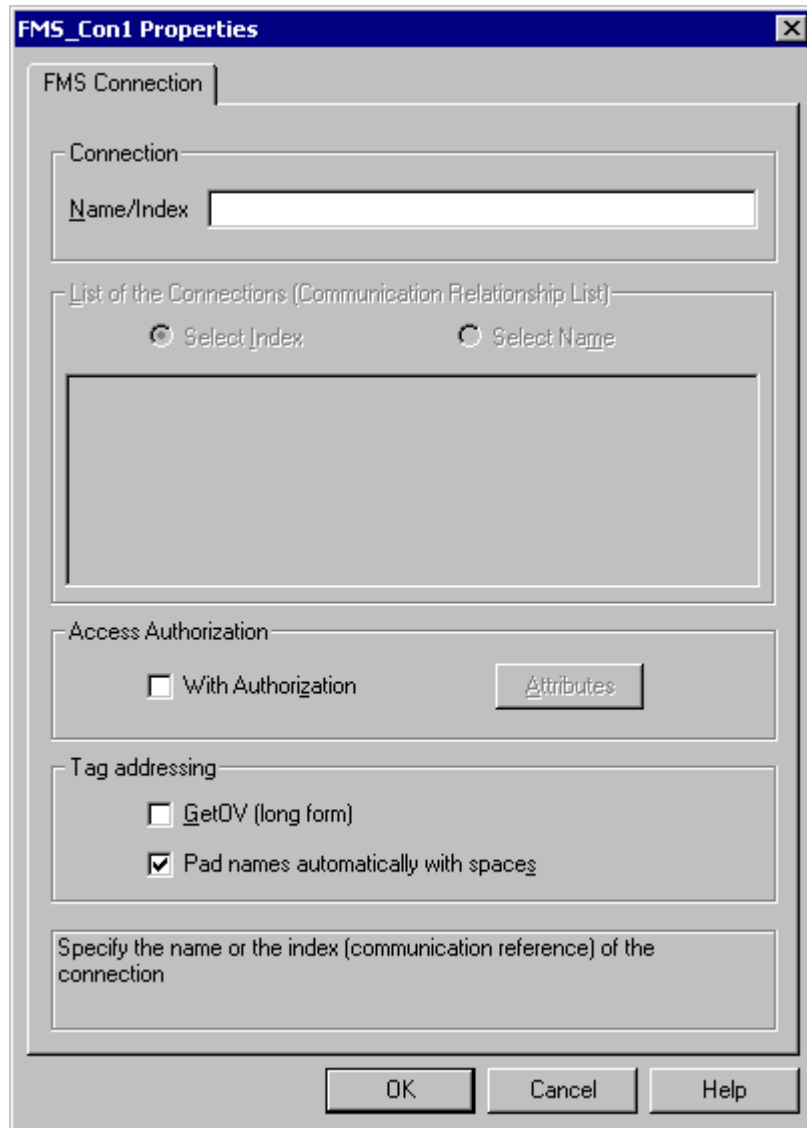
3. Enter the name or the index of the logical connection in the "Name/Index" field.
   The Name / Index of the logical connection must be configured in the local communication relationship list (CRL).

4. If access authorization has been assigned for the logical connection, please select the "With Authorization" field.

5. The "Attributes" button opens the "Access Authorization" dialog:

6. Now you can enter the password for the logical connection. Close the dialog by clicking on the "OK" button.

7. Define the required parameter in the "Tag addressing" area. Close the dialog by clicking on the "OK" button.

Note

To configure the Name or Index of the connection, you must access the communication relationship list, which can be created with SIMATIC NCM PC or STEP 7.

6.3.1.3 How to configure the connection in Runtime

Procedure

1. Select the connection and open the "Connection properties" dialog window with menu point "Properties" in the context menu.

2. Click the "Properties" button. A dialog with the connection name in its title is displayed.
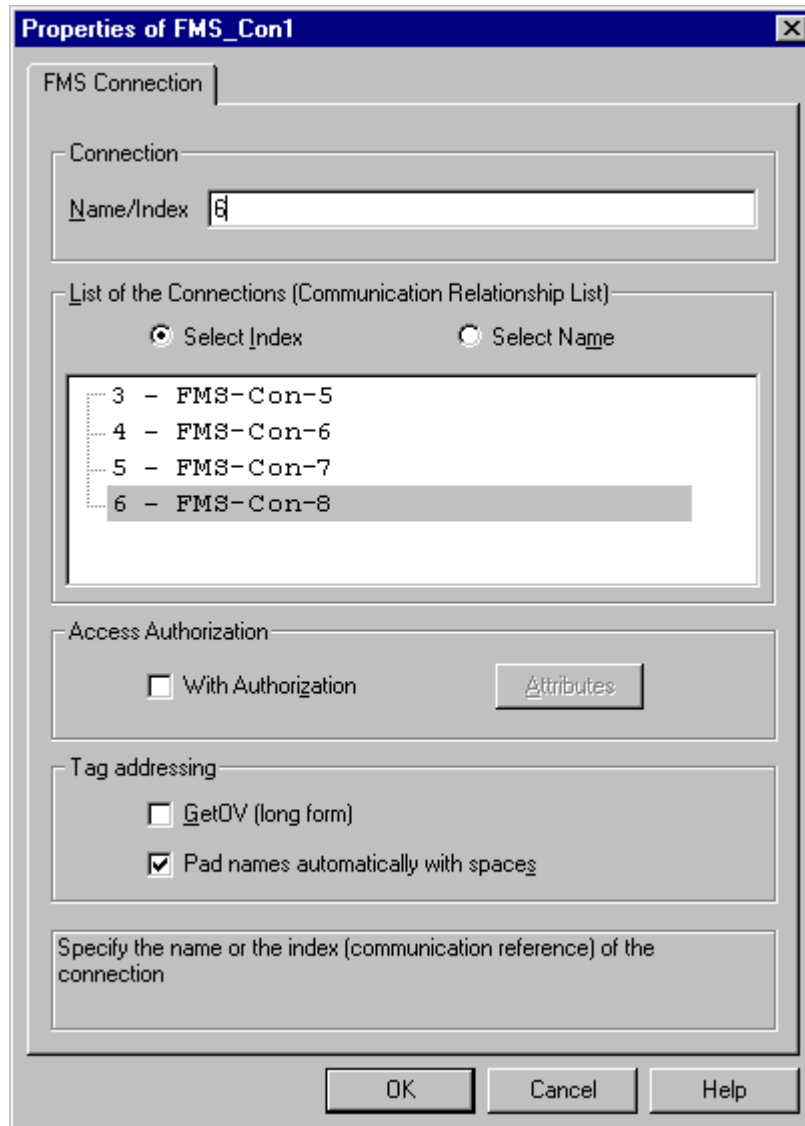
3. The configured connections from the communication relationship list (CRL) are shown in the "List of connections (communication relationship list)" area. Select whether the connection should be used with the respective index or name in the "Name/Index" field in the "Connection" area.

4. If access authorization has been assigned for the logical connection, please select the "With Authorization" field.

5. The "Attributes" button opens the "Access Authorization" dialog.

6. Now you can enter the password for the logical connection. Close the dialog by clicking on the "OK" button.

7. Define the required parameter in the "Tag addressing" area. Close the dialog by clicking on the "OK" button.

6.3.2   Configuring the tags

6.3.2.1   Configuring the tags

Introduction

The tag address is entered according to the address structure of the PROFIBUS FMS protocol.

When configuring addresses, a distinction is made between the following situations:

● in configuration mode

● in runtime

6.3.2.2   Configuring the address of the tags in configuration mode

Procedure

1. Select the tag and open the "Edit tag" dialog window with menu point "Properties" in the context menu.

2. Select the desired data type for the tags (e.g. Signed 16 bit value) from the "General" tab.

3. Click the "Select" button. A dialog with the tag name as its title is displayed.
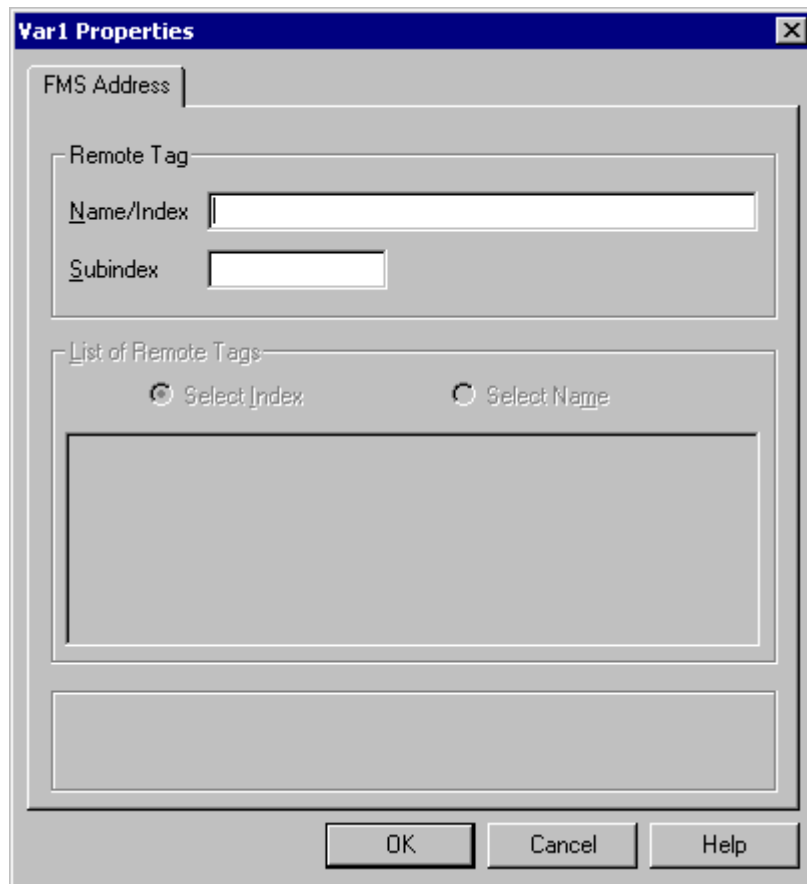
4. Enter the name or index of the tag from the remote automation system in the "Name/Index" field.

5. If the tag is a structured tag (structure or array), you must also enter - in the field "Subindex" - which part of the tag (structure component or array element) should be read or written.

Note

The name or index and the Subindex must match the name or index in the Object List for the logical connection of the external automation system. Otherwise it will not be possible to access the tag in runtime.

6.3.2.3 How to configure the address of the tags in runtime

Procedure

1. Select the tag and open the "Edit tag" dialog window with menu point "Properties" in the context menu.

2. Select the desired data type for the tags (e.g. Signed 16 bit value) from the "General" tab.

3. Click the "Select" button. A dialog with the tag name as its title is displayed.
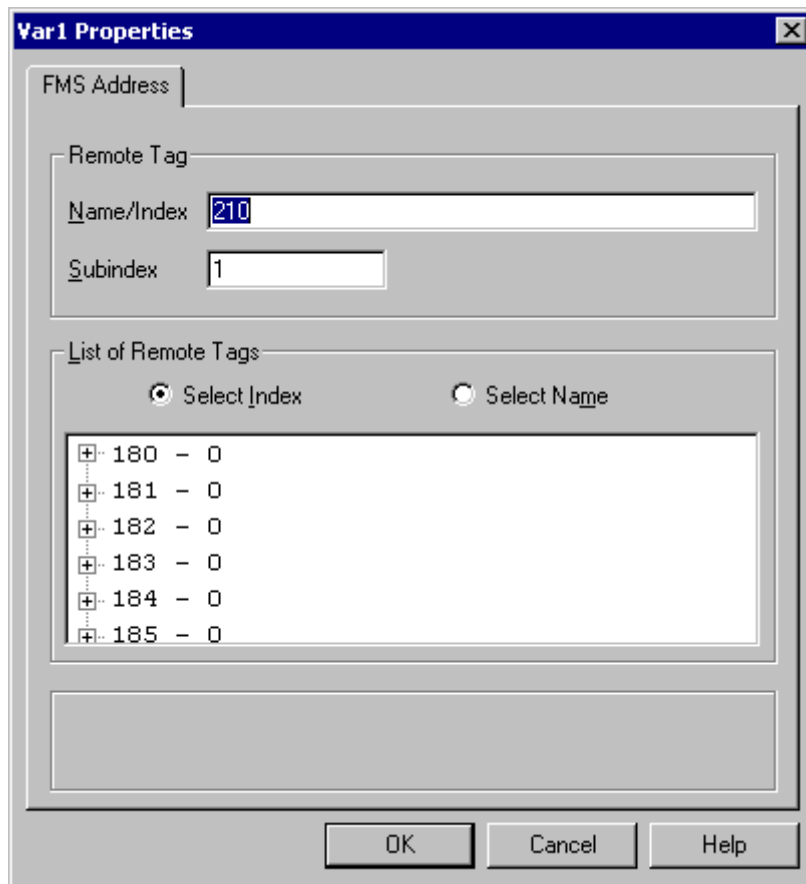
4. The tags that can be accessed through the selected connection are listed in the "List of Remote Tags" area. Select whether the tag should be chosen using its index or its name in the "Remote tag" area.

5. In the "List of the Remote Tags", double click on the desired tag to select it.

6. The selected tag will be displayed in the "Name/Index" field in the "Remote Tag" area.

7. If the tag is a structured tag (structure or array), you must also enter - in the field "Subindex" - which part of the tag (structure component or array element) should be read or written.

S5 Ethernet Layer 4

<div align="right">7</div>

7.1   WinCC Channel "SIMATIC S5 Ethernet Layer 4"

Introduction

The communication driver is used e.g. to connect automation systems SIMATIC S5-115U/H, SIMATIC S5-135U and SIMATIC S5-155U/H with the ISO transport protocol or the TCP/IP protocol.

Depending on the communication protocol that is used, the following communication partners will be used.

Communication protocol	WinCC side	SIMATIC S5 side
ISO transport protocol	CP1612 (3Com-compatible) CP1613 CP1623	CP1430 TF
TCP/IP (conforming with RFC1006)	CP1612 (3Com-compatible) CP1613 CP1623	CP1430 TCP

When using this channel, no local database is required.

Channel units

The communication driver has two channel units "CP1413-x" which can operate a maximum of two CP 1612, CP1613 or CP1623. The functionality of the channel unit is identical. They differ only in the logical device names of the two CPs. CP1623 is identical to CP1613, but is operated via PCI Express.

Communication can be established via the TCP/IP protocol with a CP1612, CP1613 or CP1623 using the third channel unit "TCP/IP".

The logical device name can be changed in the system parameters of a channel unit. Here, it is also possible to set the parameters for the protocol used.

The following application capabilities exist:

- Channel unit "S5-Transport (CP 1413-1)" for the communication modules for SIMATIC Industrial Ethernet (CP 1612/1613/1623).

- Channel unit "S5-Transport (CP 1413-2)" for the communication modules for SIMATIC Industrial Ethernet (CP 1612/1613/1623).

- Channel unit "S5-Transport (TCP/IP)" for the communication modules for SIMATIC Industrial Ethernet (CP 1612/1613/1623).

7.2   Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag

- Unsigned 8-bit value

- Signed 8-bit value

- Unsigned 16-bit value

- Signed 16-bit value

- Unsigned 32-bit value

- Signed 32-bit value

- Floating-point number 32-bit IEEE 754

- Text tag, 8-bit character set

- Raw data type

7.3 Configuring the Channel

7.3.1 Configuring the channel "SIMATIC S5 Ethernet Layer 4"

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Ethernet Layer 4".

1. Configuring the connection
2. Configuring the tags
3. System parameter configuration

7.3.2 How to configure the connection

Introduction

The connection parameters are almost identical for all protocols used. In the following example, communication is described using the ISO transport protocol with a channel unit "CP1413-x".

When implementing the TCP/IP protocol, the IP address of the AS is entered instead of the Ethernet address. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range of 0-255.

For a logical connection, WinCC establishes one connection in the transport layer for reading ("READ function" area) and one for writing ("WRITE function" area). The address parameters for both functions are defined in the dialog. Only if both connections are established is the logical connection also indicated as being "established".

Allocations for the READ function

WinCC side	SIMATIC S5 side
FETCH Active (Request "READ Active")	READ Passive (Request "READ Passive")
FETCH Passive (Request "WRITE Passive")	WRITE Active (Request "WRITE Active")

Note

It is not possible to write binary or byte variables in the data area of the AS, if the data from the AS is sent active, i.e. the READ function is set to "FETCH Passive" in the connections parameters.

A FETCH Passive connections is only assigned the "OK" status if at least one telegram has been sent from AS to WinCC.

Allocations for the WRITE function

WinCC side	SIMATIC S5 side
Request "WRITE Active"	Request "WRITE Passive"

Procedure

1. Select the connection and open dialog window "Properties" with the context menu.

2. Actuate the "Properties" button on the "General" tab and open the "Connection Parameters" dialog.
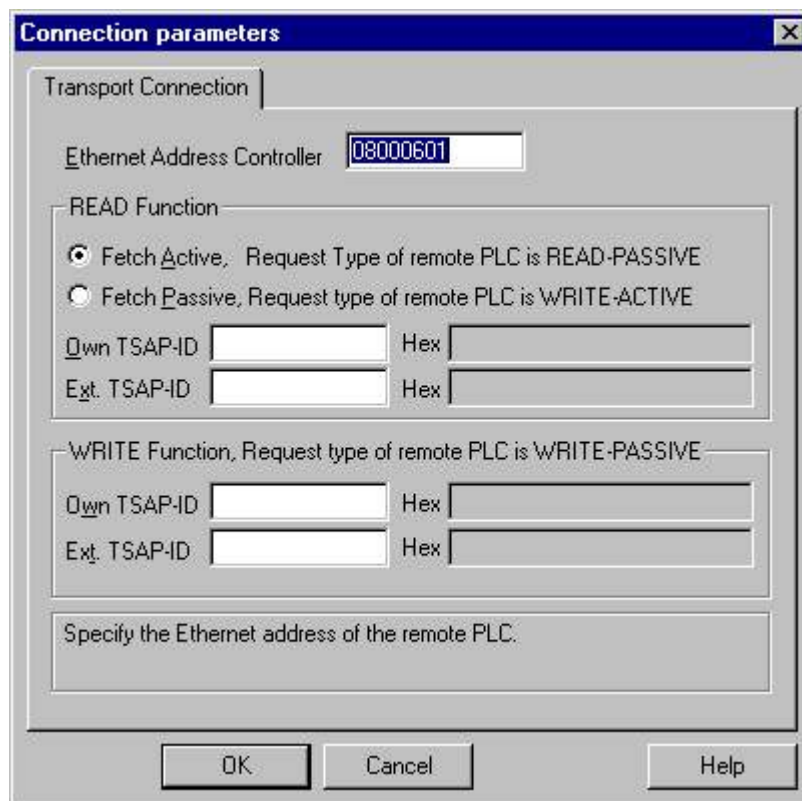
3. Enter the station address of the SIMATIC S5 on the industrial Ethernet bus in the field "Ethernet Address AG". When the TCP/IP protocol is being implemented, the IP address is entered here in the IP address AG" field.

4. Define the parameters for the READ function in the WinCC system. These are independent of the request used in the SIMATIC S5.

5. Then, enter the value in the allocated field "Own TSAP" that was configured in the "Remote parameter" as "TSAP" while configuring the CP1430 TF.

6. Now, enter the value in the allocated field "Remote TSAP" that was configured in the "Local parameter" as "TSAP" while configuring the CP1430 TF.

7. Define the parameters "Own TSAP" and "Remote TSAP" for the WRITE function accordingly.

Note

In the entries for "TSAP", you must not use any spaces.

7.3.3   Configuring the tags

7.3.3.1   Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 Ethernet Layer 4", tags of different data types can be created in WinCC. The following describes how to configure a tag of these data types.

- Addresses of tags

- Configuring a tag with bit by bit access

- Configuring a tag with byte by byte access

- Configuring a tag with word by word access

- Configuring a raw data tag

7.3.3.2   Addresses of tags

Introduction

The tag address is entered according to the address structure of the SIMATIC S5.

Depending on the tag type, the access to memory areas in the AS is bit by bit, byte by byte or word by word. For this purpose, the addressed memory area is read from the AS for every

single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

It is not possible to write binary or byte variables in the data area of the AS, if the data from the AS is sent active, i.e. the READ function is set to "FETCH Passive" in the connections parameters.


Configuring the address of a tag is done independent of the tag type:

- With tags of type "binary" or 8 bit value", first the "Bits-/Bytes-tag" dialog is opened, in which the bitwise or byte-wise access to the memory area of the AS is defined.
  Afterwards, the address of the tag in the AS memory is defined in the "Address properties" dialog.

- For word-oriented tags, the address of the tag in the AS memory is defined in the "Address properties" dialog.
  The dialog "Bits-/Bytes-tag" is not opened, since the access to the AS memory is word by word.

7.3.3.3   How to Configure a Tag with Bit by Bit Access

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "binary tag" on the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.
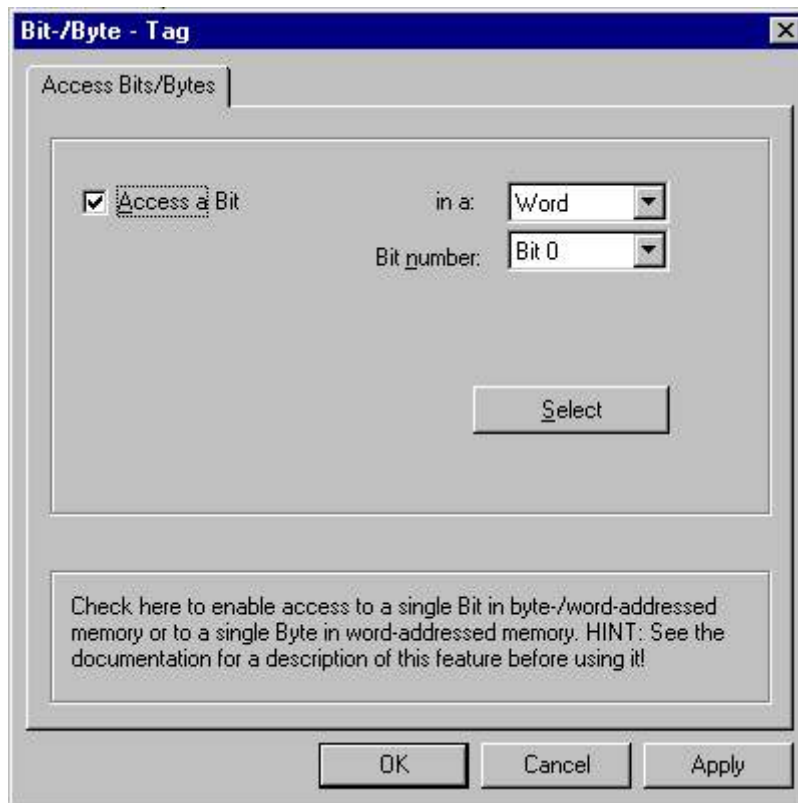
4. Use the checkbox to define whether access should be enabled for reading and writing certain bits in the memory area.

5. Select the addressing methods for the AS memory in the selection field e.g. "Word" or "Byte".

6. Select the number of bits to be changed in the selection field.

7. Use the "Selection" button to open the "Address properties" dialog for defining the tag address in AS.

Note

With the S5, flags, inputs and outputs can be addressed by byte, data blocks (DB, DX) are addressed by word.

Activating the checkbox "Access a bit" affects the display of the fields of the "Address properties" dialog.

For word-oriented tags, the described "Bit-/Byte-tag" is not opened because the address of the tags and therefore the access to PLC memory is by word.

7.3.3.4   How to Configure a Tag with Byte by Byte Access

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "Unsigned 8 bit value" or "Signed 8 bit value" in the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Use the checkbox to define whether access should be enabled for reading and writing certain bytes in the memory area.

5. Only "Word" is shown as the AS memory addressing type in the selection field.

6. Select the number of bytes to be changed in the selection field.

7. Use the "Selection" button to open the "Address properties" dialog for defining the tag address in AS.

Note

With the S5, flags, inputs and outputs can be addressed by byte, data blocks (DB, DX) are addressed by word.

Activating the checkbox "Access a byte" affects the display of the fields of the "Address properties" dialog.

For word-oriented tags, the described "Bit-/Byte-tag" is not opened because the address of the tags and therefore the access to PLC memory is by word.

7.3.3.5 How to Configure a Tag with Word by Word Access

Introduction

The addresses of tags in AS are defined with the dialog that is described here.

- With tags of type "binary" or 8 bit value", first the "Bits-/Bytes-tag" dialog is opened, in which the bitwise or byte-wise access to the memory area of the AS is defined.

- For word-oriented tags, the "Bit-/Byte-tag" dialog is not opened because the address of the tags and therefore the access to AS memory is by word.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select the desired data type for the tags (e.g. Signed 16 bit value) from the "General" tab.

3. Click the "Select" button. The "Address properties" dialog is opened.
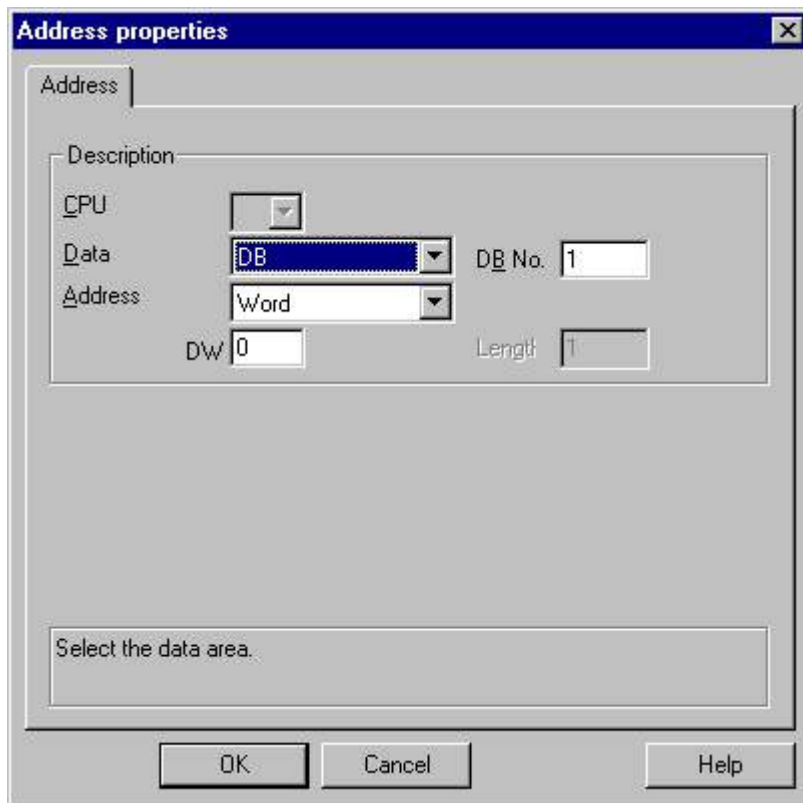
4. Choose whether the tag is located in a data block, the flag area, an input area or an output area in the "Data Area" field of the "Address" tab.

5. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.

6. The type of addressing is entered in the "Addressing" field. Normally, you can use the default definition.

7. Enter the address in the respective field (e.g. "DW ").

Note

For tags of type "binary" or "8 bit value", displaying the fields of this dialog depends on the selection made for "Access to bits/bytes" in the "Bits-/Bytes-tag" dialog.

If the tag of a word-oriented data area is to be written, the start address must be in the left byte and the length of the tags must be an even number.

7.3.3.6   How to configure a raw data tag

Introduction

The following is a description of how the address of a raw data tag is defined.

Note

If the tag of a word-oriented data area is to be written, the start address must be in the left byte and the length of the tags must be an even number.

Procedure

1. Select the tag and open dialog window "`Tag properties" with the "Properties" context menu.

2. Select the entry "Raw data type" on the "General" tab.

3. Click the "Select" button to open the "Address properties" dialog.
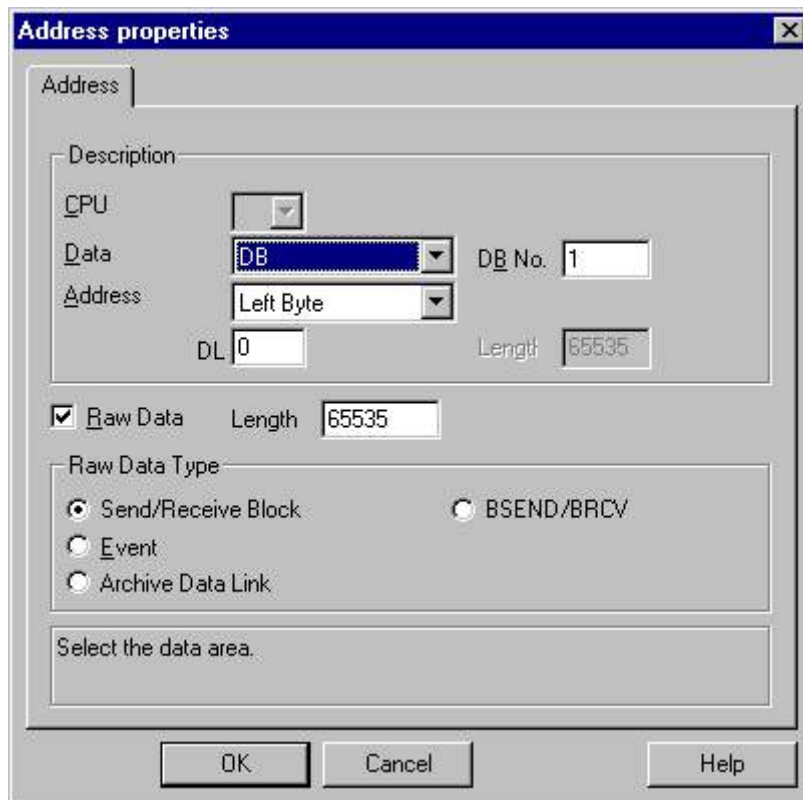
4. Choose whether the tag is located in a data block, a flag, an input area or an output area in the "Data Area" field.

5. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.

6. The type of addressing is entered in the "Addressing" field. Normally, the default definition can be used.

7. Enter the data address in the field underneath. The label on the field depends on the entry in the "Addressing" field, e.g. "DBW" for Word addressing.

8. Enter the length of the raw data block in bytes in the Length field.

9. Define which type of raw data tag is concerned in the "Raw data type" area.

7.3.4   System parameters

7.3.4.1   System parameters of the channel unit

Introduction

If you require a configuration that deviates from the standard WinCC settings, you can make all the required changes using the "System Parameter" dialog of the channel unit.

The system parameters are almost identical for all protocols used. When implementing the TCP/IP protocol, only the device name given during the installation is different.

The following individual points can be changed:

- the device name
- the transport parameter

> **Note**
>
> The system parameters apply for all CPs in the AS.

Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communication module and are unit-specific. The device name represents the logical device name. The logical device name is given the name "/CP_H1_1:/ SCP" with the ISO transport protocol and "/TCP_IP:/SCP" with the TCP/IP protocol as a default definition.

> **Note**
>
> When using the TCP/IP protocol, you must check whether the device name in WinCC matches the "Access point of the application" in the "Set PG/PC interface" dialog. The device name must also be changed in "Set PG/PC interface".

Transport Parameter

Specific settings for the channel unit are made in the transport parameters, e.g. PDU size, setup attempts, etc.

7.3.4.2 How to Change the Device Name

Introduction

Parameters of the channel unit are set with the system parameters, e.g. the logical device name or the transport parameters.

The system parameters are almost identical for all protocols used.

In the following example, communication is described using the ISO transport protocol with a channel unit "CP1413-x".

When implementing the TCP/IP protocol, only the device name given during the installation is different.

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.

2. Select the "Device Name" Tab.
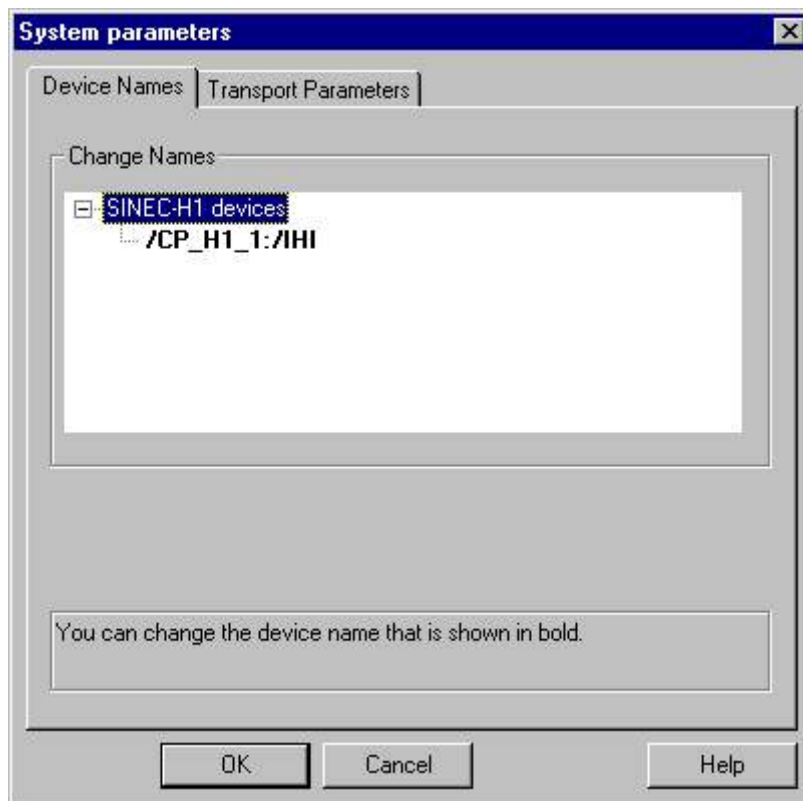
3. Now, you can select the device name shown in bold print with the mouse and change it with a mouse click in the name field for the device name.

Note

The device name is defined during the installation of the hardware driver. Only if you have defined another name there, which is not recommended, will you have to change the device name here as well.

7.3.4.3   How to change the transport parameter

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.

2. Select the "Transport parameter" tab.
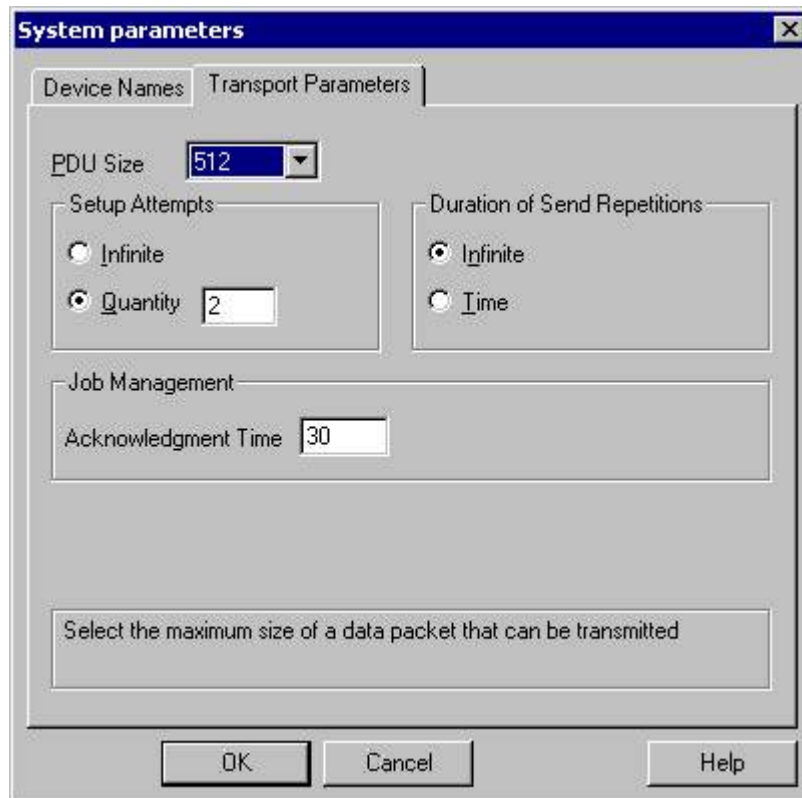
3. Set the value for "PDU size" to the value that was configured on the communication module CP 1430.

4. Define how often a connection establishment should be attempted in the "Setup attempt" filed.

5. Select "Infinite" in the "Duration of Send Repetitions" area.

6. Enter value e.g. 30 in the "Acknowledgement time" field so that you are informed of the tag status after 30 seconds at the most, if the communication partner has not responded within this time (e.g. AS in "Stop" status).

7.4 Appendix

7.4.1 Appendix

Introduction

Added information on the channel "SIMATIC S5 Ethernet Layer 4" is provided in the appendix.

7.4.2 Internal error codes and constants

7.4.2.1 Internal error codes and constants

Introduction

The following tables contain the most important error codes and constants. The information is intended for "insiders". Therefore, we have not gone into more detail on the meanings of the codes and constants.

- Error codes during connection disturbances
- iNA960 messages
- SCI messages

7.4.2.2 Error codes during connection disturbances

Introduction

The most important error codes are listed in this section. If an error with an error code that is not in the table occurs, please call the WinCC hotline.

Fehler_0002-INVALID_RQ

Faulty request block.

Fehler_0004-NO_RESOURCES

No resources free in CP.

Fehler_0006-UNKNOWN_REFERENCE

Incorrect OPEN reference defined.

Fehler_0008-BUFFER_TOO_SHORT

User buffer too short.

Fehler_0010-BUFFER_TOO_LONG

User buffer too long.

Fehler_0012-ILLEGAL_REQ

Incorrect "negot_options" defined.

Fehler_0014-REM_ABORT

Connection aborted by remote station.

Fehler_0016-LOC_TIMEOUT

Timeout.

Fehler_0018-UNKNOWN_CONN_CLASS

Unknown connection class.

Fehler_0020-DUP_REQ

Connection already established.

Fehler_0022-CONN_REJECT

Connection request rejected by remote.

Fehler_0024-NEGOT_FAILED

Connection abort faulty "negot-option".

Fehler_0026-ILLEGAL_ADDRESS

Faulty transport address.

Fehler_0028-NETWORK_ERROR

Bus or CP disrupted.

Fehler_0030-PROTOCOL_ERR

Protocol error.

Fehler_0032-ILLEGAL_RB_LENGTH

Incorrect request block length.

Fehler_0784-E_NO_HW

No communication hardware found.

- Communication module defective.
- Communication module not installed correctly.
- Wrong port address defined.

Fehler_0786-E_CNF

Driver configured incorrectly or invalid parameter in the registry.

Fehler_0787-E_BAUDRATE

Incorrect baudrate or incorrect interrupt vector defined.

Fehler_0788-E_HSA

Incorrect HSA (Highest Station Address) defined.

Fehler_0789-E_TS

The defined local participant number (TS_ADR) is already assigned.

Fehler_0791-E_INT_NOT_PROV

The defined interrupt vector (IRQ) is not available on the communication module.

Fehler_0792-E_INT_BUSY

The defined interrupt vector (IRQ) is already occupied on the communication module.

Fehler_0800-E_NO_FILE

The selected communication driver cannot be loaded; the file was not found.

- Communication driver not installed correctly.

Fehler_0897-E_LOGDEV

The logical device is not defined in the registry.

- Communication driver not installed correctly.
- Entry damaged or deleted in the registry.

- Check the setting of the logical device name with the "Set PG/PC interface" program.
- Check the setting for the logical device name in the "System parameter - Device" mask.

Fehler_0898-E_L2DRIVER

The entry "L2DRIVER" is missing in the registry.

- Module error or module installed incorrectly.

Fehler_0900-E_L4DRIVER

The entry "L4DRIVER" is missing in the registry.

- Module error or module installed incorrectly.

Fehler_30000-EC_WATCHDOG

Watchdog error.

Fehler_30001-EC_PDUERROR

PDU not expected.

Fehler_30005-EC_ONLERROR

Fault loading the S7-Online-DLL.

7.4.2.3   iNA960 messages

General iNA960 messages

OK_RESP	1	0x01	Request executed with no errors
OK_EOM_RESP	3	0x03	Data block received with no errors
OK_DECIDE_REQ_RESP	5	0x05	Request executed with no errors
OK_CLOSED_RESP	7	0x07	Connection aborted by local user

iNA960 error messages

INVALID_REQ	2	0x02	Faulty request block
NO_RESOURCES	4	0x04	No resources free in CP
UNKNOWN_REFERENCE	6	0x06	Incorrect OPEN reference defined
BUFFER_TOO_SHORT	8	0x08	User buffer too short
BUFFER_TOO_LONG	10	0x0A	User buffer too long

ILLEGAL_REQ	12	0x0C	Incorrect "negot_options" defined
REM_ABORT	14	0x0E	Connection aborted by remote station
LOC_TIMEOUT	16	0x10	Timeout
UNKNOWN_CONN_CLASS	18	0x12	Unknown connection class
DUP_REQ	20	0x14	Connection already established
CONN_REJECT	22	0x16	Connection request rejected by remote
NEGOT_FAILED	24	0x18	Connection abort faulty negot-option
ILLEGAL_ADDRESS	26	0x1A	Faulty transport address
NETWORK_ERROR	28	0x1C	Bus or CP disrupted
PROTOCOL_ERR	30	0x1E	Protocol error
ILLEGAL_RB_LENGTH	32	0x20	Incorrect request block length

7.4.2.4   SCI messages

See description in the "SINEC Communication Interface SCI" manual (A/5-15).

SCI messages

SCP_OK	0	0x00	No error
SCP_INCONS	201	0xC9	Minor device number is not 00
SCP_RESOURCE	202	0xCA	DPRAM request invalid
SCP_CONFIG	203	0xCB	Configuration error (NUM_PROCS)
SCP_NOCONFIG	204	0xCC	SCP driver not configured
SCP_PARAM	206	0xCE	Incorrect mode
SCP_DEVOPEN	207	0xCF	Open already performed
SCP_BOARD	208	0xD0	Board not inserted/recognized
SCP_SOFTWARE	209	0xD1	IRQ error or software not found
SCP_MEM	210	0xD2	Low memory in DPRAM
SCP_MODE	211	0xD3	Download process not yet ended
SCP_LOADER	212	0xD4	No response from loader
SCP_SIGNAL	213	0xD5	Process started asynchronously
SCP_NOMESS	215	0xD7	No message arrived for the process
SCP_USERMEM	216	0xD8	length_of_buffer too small
SCP_WINDOW	217	0xD9	Too many SEND calls
SCP_TIMEOUT	219	0xDB	Timeout on SCP
SCP_ATTACH	220	0xDC	Reset not executed/Channel still active
SCP_ILLEGAL_REQUEST	221	0xDD	Illegal request
SCP_ERECOVERF	223	0xDF	Buffer not retrieved with scp_receive
SCP_ECLOSED	224	0xE0	All buffers assigned for the connection
EUSERMAX	225	0xE1	
SCP_EINTR	226	0xE2	
SCP_BOARD_OPEN	231	0xE7	

SCP_NO_WIN_SERV	233	0xE9	
EPROTECT	234	0xEA	License not found

SCI messages

SCP_DB_FILE_DOES_NOT_EXIST	240	0xF0
SCP_DB_FILE_CLOSE_NOT_OK	241	0xF1
SCP_SEND_NOT_SUCCESSFUL	242	0xF2
SCP_RECEIVE_NOT_SUCCESSFUL	243	0xF3
SCP_NO_DEVICE_AVAILABLE	244	0xF4
SCP_ILLEGAL_SUBSYSTEM	245	0xF5
SCP_ILLEGAL_OPCODE	246	0xF6
SCP_BUFFER_TOO_SHORT	247	0xF7
SCP_BUFFER_1_TOO_SHORT	248	0xF8
SCP_ILLEGAL_PROTOCOL_SEQUENCE	249	0xF9
SCP_ILLEGAL_PDU_ARRIVED	250	0xFA
SCP_REQUEST_ERROR	251	0xFB
SCP_NO_LICENSE	252	0xFC

Additional online DLL messages on the SCP interface

E_TIMER_INIT	768	0x0300	WIN Set-timer request unsuccessful
E_INIT_COM	769	0x0301	
E_NO_HW	784	0x0310	MPI module not found
E_HW_DEFEKT	785	0x0311	Problem with the hardware
E_CNF	786	0x0312	Incorrect configuration parameter
E_BAUDRATE	787	0x0313	Incorrect baudrate/incorrect IntVector
E_HSA	788	0x0314	Incorrect HSA configured
E_TS	789	0x0315	Configured address already assigned
E_OCC	790	0x0316	HW_Device already assigned
E_INT_NOT_PROV	791	0x0317	Interrupt not available
E_INT_BUSY	792	0x0318	Interrupt occupied
E_SAP	793	0x0319	SAP deactivate: SAP not occupied
E_UNPLUGGED	794	0x031a	No remote station found
E_SYNI	795	0x031b	Syni Error occurred
E_AMPRO	796	0x031c	AMPRO 2 reported a system error
E_BUFFSIZE	797	0x031d	No buffer of this size created
E_NO_FILE	800	0x0320	DLL/VxD File not found or entries in registry destroyed
E_NO_ENTRY	801	0x0321	Address does not exist in DLL
E_VERSION	816	0x0330	Version conflict between SMC driver and SMC firmware
E_COMCNF	817	0x0331	Problem with the COM port configuration
E_NO_SMC	818	0x0332	SMC no longer responds

E_COMMBADID	819	0x0333	COM port is not configured
E_COMMOPEN	820	0x0334	COM port is not available
E_SMCBUSY	821	0x0335	Serial driver is currently in use with another configuration
E_SMCMODEM	822	0x0336	No connection exists to a PC/MPI cable
E_SMCNOLEG	823	0x0337	PC/MPI cable rejects request, necessary authorization is missing
E_ONLINE	896	0x0380	Internal error at the IOCTL interface
E_LOGDEV	897	0x0381	Logical device not in registry
E_L2DRIVER	898	0x0382	L2DRIVER entry is missing in the registry
E_L4DRIVER	900	0x0384	L4DRIVER entry is missing in the registry
E_SYSERROR	1023	0x03FF	System error

Channel-specific error codes

EC_WATCHDOG	30000	0x7530	Watchdog error
EC_PDUERROR	30001	0x7531	PDU not expected
EC_ONLERROR	30005	0x7535	Fault loading the S7-Online-DLL

S5 PROFIBUS FDL

<div align="right">

8

</div>

8.1 WinCC channel "SIMATIC S5 Profibus FDL"

Introduction

The channel "SIMATIC S5 Profibus FDL" is used for communication between WinCC Station and a SIMATIC S5 automation system. The PROFIBUS (Process Field Bus) network type and protocol FDL (Field Data Link) are used in this case.

PROFIBUS is the network for small to medium sized data quantities. A broad spectrum of automation tasks can be completed with a maximum of 127 possible participants.
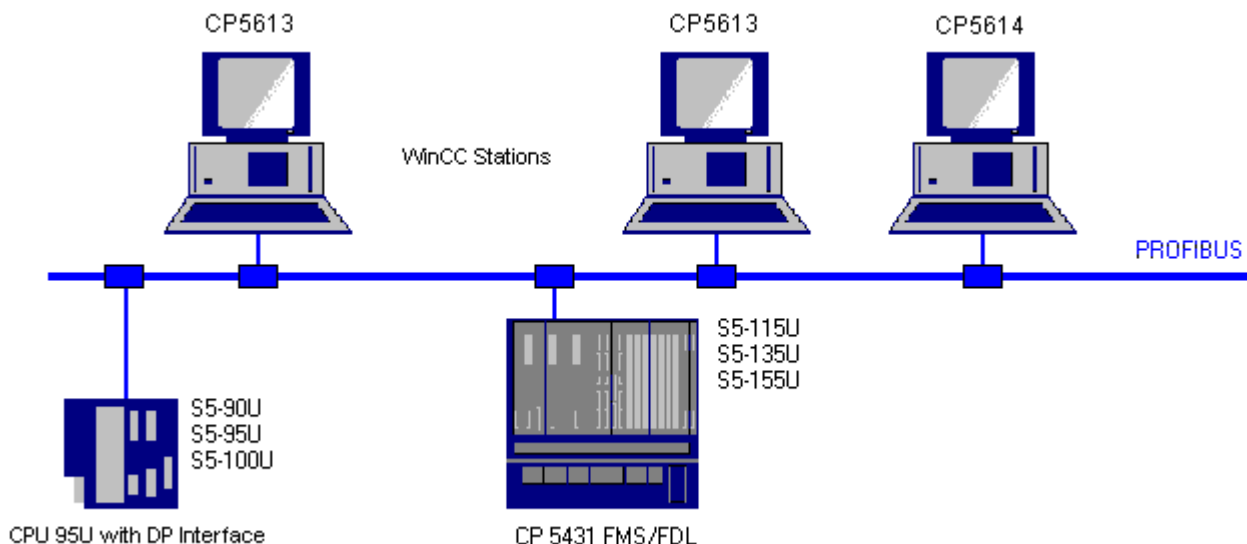
Reading/writing tags via PROFIBUS, using the FDL protocol, is done using request and response telegrams. The request telegram is sent to the automation device from WinCC. The AS answers with the response telegram.

An FDL connection is specified with the local and remote connection end point (Service Access Point).

This chapter describes

● how to configure the data transfer with the "SIMATIC S5 Profibus FDL" channel.

● how to create a sample project

Channel unit FDL (CP5412/A2-1)

Independent of the communication processor used, CP 5613 or CP 5614, connection can be made to the SIMATIC S5 via the channel unit "FDL (CP5412/A2-1)". This channel unit supports

up to a maximum of 24 connections. In order for the channel to function, a channel unit and a connection must be set up.

Service Access Point

SAPs are local data interfaces within a PROFIBUS node. The SAPs must be configured in WinCC and on the AS. A unique identification is defined with the Service Access Point. This unique identification is required for communication between the WinCC and the AS.

Active connection

An active connection is also called a Fetch connection. This is a connection, with which an active partner retrieves data with a communication partner. The communication partner that the data is retrieved from is designated as a passive partner.

Passive connection

A passive connection exists if the active AS sends data to the passive WinCC partner asynchronously without a request telegram.

8.2   Supported data types and data ranges

Introduction

Only certain data types and data ranges are supported for communication from SIMATIC S5 via PROFIBUS FDL.

Supported data types

WinCC Data type	SIMATIC S5 data type
Binary tag	BIT
Signed 8-bit value	non-existent in the SIMATIC S5
Unsigned 8-bit value	BYTE
Signed 16-bit value	WORD
Unsigned 16-bit value	WORD
Signed 32-bit value	DWORD
Unsigned 32-bit value	DWORD
Floating-point number 32-bit IEEE 754	DWORD
Floating-point number 64-bit IEEE 754	non-existent in the SIMATIC S5
Text tag, 8-bit character set	ARRAY OF BYTE
Text tag, 16-bit character set	non-existent in the SIMATIC S5
Raw data type	ARRAY OF BYTE

Access to SIMATIC S5 tags

The access to SIMATIC S5 tags is done word by word to data block DB or extended data blocks DX. This allows read and write access.

Access to a SIMATIC S5 tag of data type BIT

SIMATIC S5 tags of data type BIT only allow read access. This restriction applies for active or passive connections.

Access to a SIMATIC S5 tag of data type BYTE

SIMATIC S5 tags of data type BYTE only allow read access.

To configure a byte tag, the "left byte" or "right byte" of a 16 bit data word must be selected for addressing.

Access to a SIMATIC S5 tag of data type ARRAY OF BYTE

SIMATIC S5 tags of data type ARRAY OF BYTE only allow read access.

8.3 Features of the WinCC channel "SIMATIC S5 Profibus FDL"

Introduction

The capabilities of communication, from WinCC via the communication driver for PROFIBUS, are listed in the following section. All supported data types and the respective capabilities for format adaptation are also listed.

Note

Features for writing tags:
When configuring in WinCC, make sure that every tag is transferred individually when writing more than one tag into data areas of the automation system.

This behavior is especially important when writing more than one tag to a script with function "SetTagMultiWait". Since this function is only executed at the end of a transfer of all the tags sent to it, noticeable waiting times can occur with a greater number of tags.

Check for whether the utilization of the "Wait" function is required for a larger amount of tags. In this case, the use of a raw data tag may also be a good idea, especially if the data is sequential in the AS data area.

Type conversion

A format adaptation is then required if a certain value range or a conversion e.g. of decimal to BCD ("Unsigned 8 bit value" converted to "ByteToBCDWord") if necessary. No format adaptation is performed as standard procedure.

The following table lists the supported WinCC data types and the respective format adaptation capabilities.

WinCC Data type	Type conversion
Binary tag	No
Signed 8-bit value	non-existent in the S5
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Floating-point number 64-bit IEEE 754	non-existent in the S5
Text tag, 8-bit character set	No
Text tag, 16-bit character set	non-existent in the S5
Raw data type	No

WinCC side

The communication driver SIMATIC S5 Profibus FDL supports the communication with the following communication processors:

Communications processor	Bus type
CP5613	PCI
CP5614	PCI

AS side

Automation devices can generally be connected to the PROFIBUS network in two different ways. The connection can be done via the integrated interface on the central module or with special communication modules.

System	Modules
S5-90U, S5-95U, S5-100U	CPU95U
S5-115U, S5-135U, S5-155U	CP5431 FMS/DP

Note

With connection to S5-95U with L2-SS, no Fetch connections are possible since WinCC can only be a passive partner.

8.4   Configuring the Channel

## 8.4.1   How to configure the channel "SIMATIC S5 Profibus FDL"

Introduction

The steps in configuring the channel "SIMATIC S5 Profibus FDL" are described in this and the following chapters.

This section shows how the channel "SIMATIC S5 Profibus FDL" is configured.

Procedure

1. Select the entry "Add New Driver" in the shortcut menu of Tag Management. The "Add New Driver" dialog is opened.
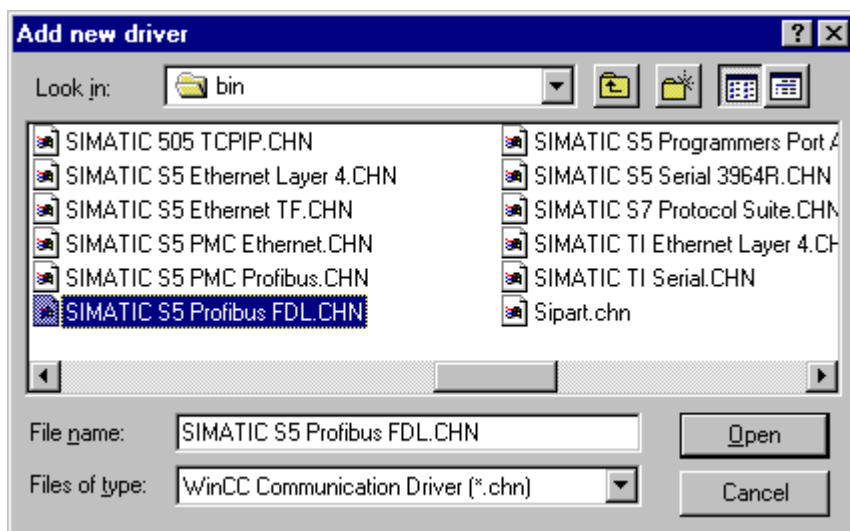
2. Select the "SIMATIC S5 Profibus FDL.CHN" driver and click the "Open" button. The channel is created and the communication driver is displayed in the tag management.

8.4.2   Channel unit "FDL (CP5412/A2-1)"

Introduction

The communication driver "SIMATIC S5 Profibus FDL" only contains the channel unit "FDL (CP5412/A2-1)".

The communication between the WinCC and the SIMATIC S5 automation device occurs via the channel unit "FDL (CP5412/A2-1)". A maximum of 24 connections can be created within

a channel unit. Special connection parameters are to be defined for every configured connection. Every configured tag must be defined with tag parameters.

Note

The name of the channel unit "FDL (CP5412/A2-1)" is bound to the communication driver, "SIMATIC S5 Profibus FDL.CHN", and is independent of the communication processor used. As communication processors, e.g. CP5613 and CP5614 can be used.

Tag parameter

The following tag parameters must be defined for every configured tag:

- Data area (e.g DB)
- Data block number
- Addressing (e.g. "left byte")
- Start address (e.g. DL 0, if "left byte" has been selected for addressing)

Connection parameters

The following connection parameters must be defined for every configured connection:

- The station address of the AS
- The priority
- Own and foreign SAPs (Service Access Point) must be defined for the read and write function

For the read function, configuring whether it should be an active or a passive connection is also necessary. For an active read connection, the values are requested from the WinCC station. For a passive connection, the transfer of the values to the WinCC station is initiated by the AS.

8.4.3   How to configure a connection

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. Open the context menu of channel unit "FDL (CP5412/A2-1)". In this shortcut menu, select "New Connection".

2. In the "Connection Properties" dialog, enter the connection name.
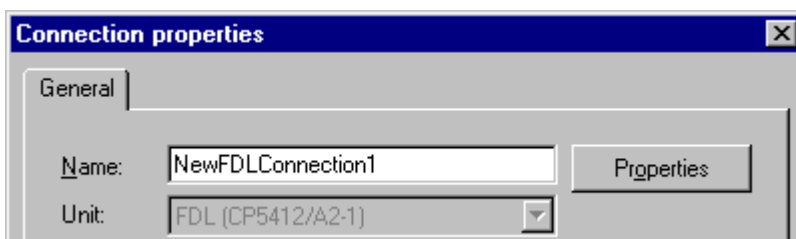
3. In the "Connection Properties" dialog, click the "Properties" button. The "Connection Parameters" dialog opens. You can configure the connection in this dialog.
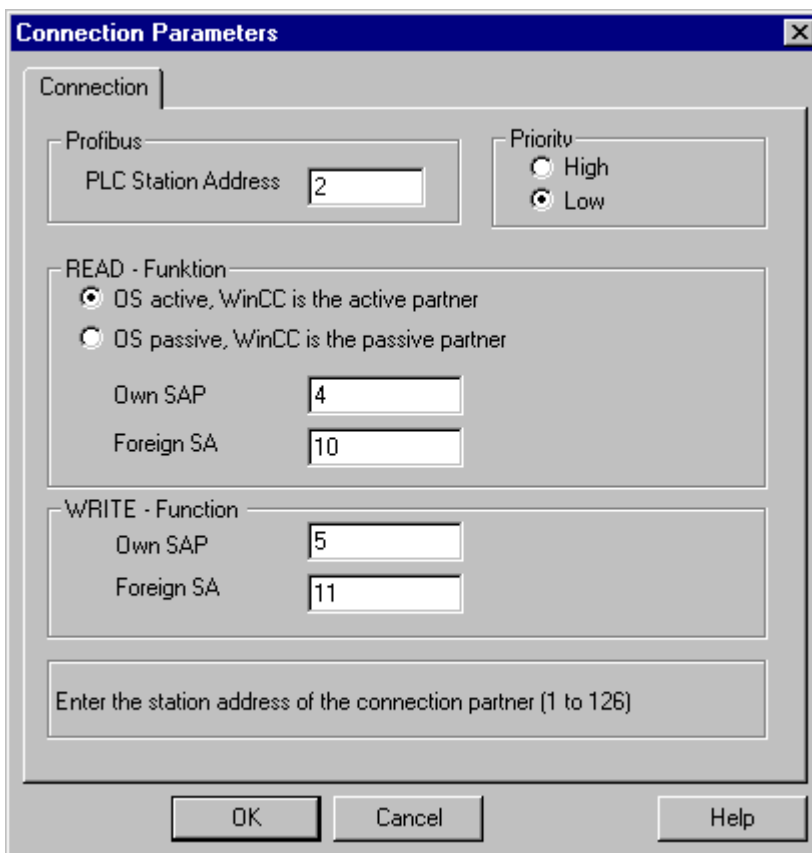
4. Enter the unique address of the AS in the "PLC Station Address" field.

5. The "Priority" option button must always be set to "Low" for an FDL connection.

6. The function of the WinCC station is defined with fields "OS active, WinCC is the active partner" or "OS passive, WinCC is the passive partner". Activate the required option button.

7. Enter the SAP addresses configured for the reading and writing access in the fields "Own SAP" and "Foreign SAP". The SAPs value range is between 2 and 54.

8. Click "OK" to close all open dialogs.

9. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens. Configure the tag.

10. Click "OK" to close all open dialogs.

8.4.4   Configuring the tags

8.4.4.1   Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 Profibus FDL", data types binary, byte and word can be defined within WinCC. The following describes how to configure a tag of these data types.

8.4.4.2   How to Configure a Tag with Bit by Bit Access

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in the AS.

> **Note**
>
> The bit by bit access to a tag is only read access.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.
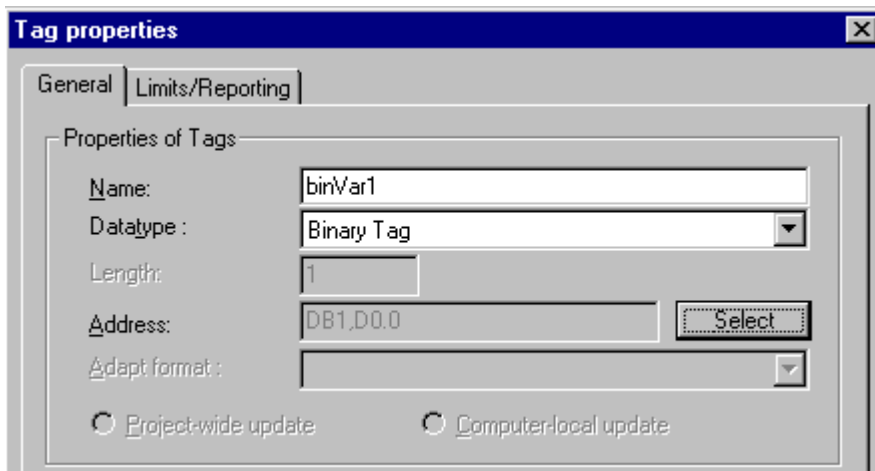
2. Enter the name of the tag in the "Name" field of the "General" tab, e.g. "binVar1". Define data type "Binary tag" in the "Data type" field.

3. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.
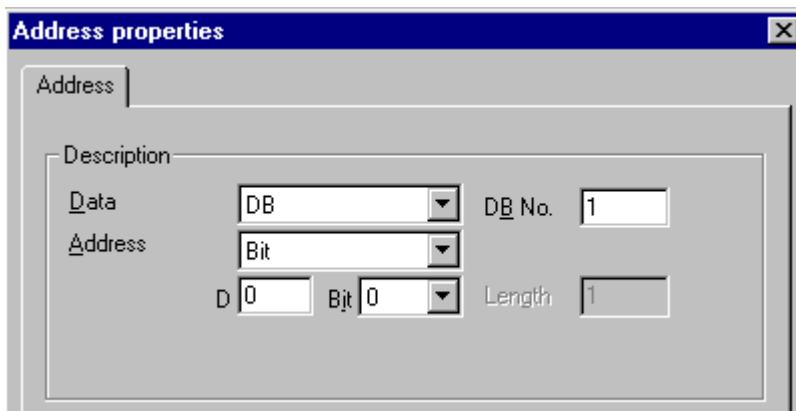
4. Enter the byte address in field "D" and the bit address in field "Bit". The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and binary tag as the data type of the tag.

5. Click "OK" to close all open dialogs.

Note

You cannot change the "Bit" entry in the Addressing field because it is defined by the Binary tag data type of the WinCC tag.

8.4.4.3 How to Configure a Tag with Byte by Byte Access

Introduction

This section shows you how to configure a tag for byte by byte access for the address area in the AS.

Note

The byte-wise access to a tag is only read access.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.
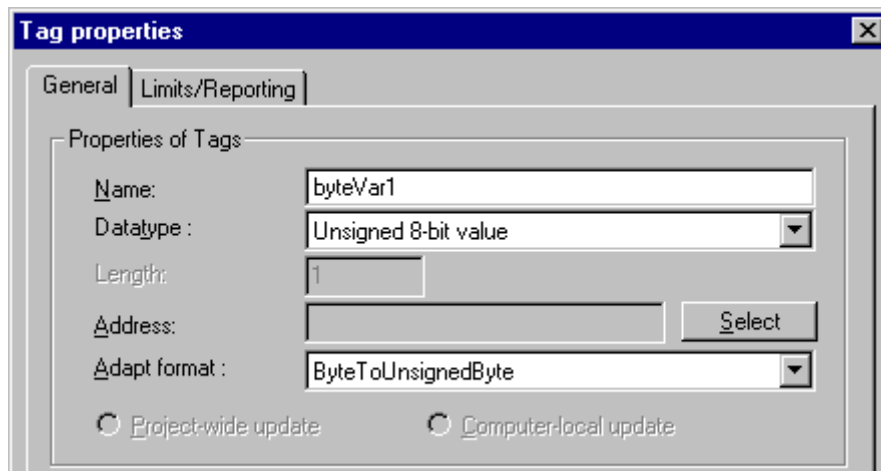
2. Enter the name of the tag in the "Name" field of the "General" tab, e.g. "byteVar1". Define data type "Unsigned 8 bit value" in the "Data type" field.

3. Click the "Select" button to open the "Address properties" dialog. Define the data area of the AS in the "Data" field. The communication driver "SIMATIC S5 Profibus FDL" only supports data areas DB and DX. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.
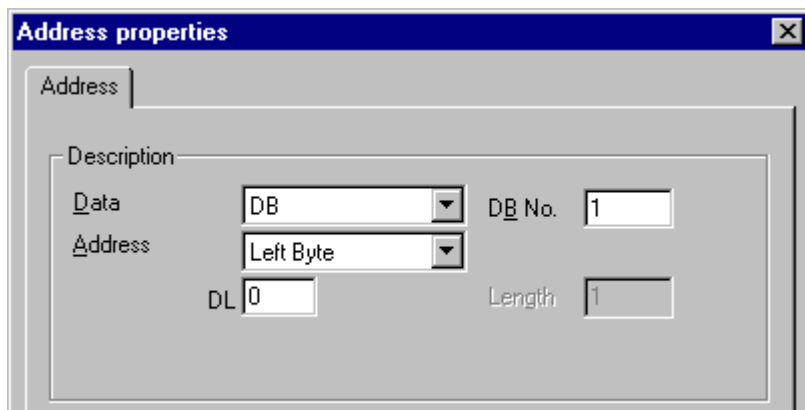
4. You can choose between "Left byte" and "Right byte" in the "Addressing" field.

5. Enter the byte address in the "DL" field. The label on the field depends on the entry in the "Addressing" field, e.g. "DL" for "Left byte" addressing.

6. Click "OK" to close all open dialogs.

8.4.4.4 How to configure a tag with word by word access

Introduction

This section shows you how to configure a tag for word-wise access for the address area in the AS.

Note

The word by word access to a tag is read and/or write access.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.
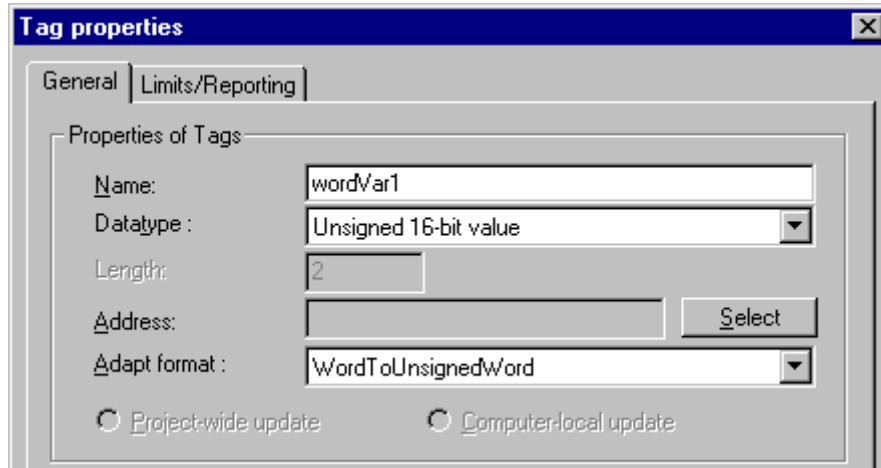
2. Enter the name of the tag in the "Name" field of the "General" tab, e.g. "wordVar1". Set the data type to "unsigned 16-bit".

3. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.
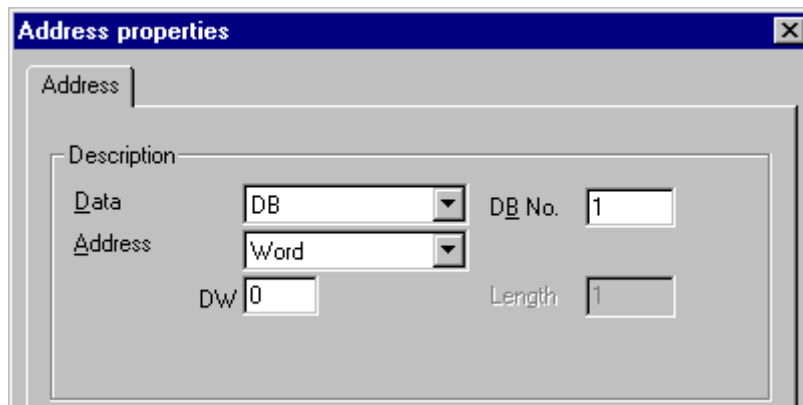
4. Enter the word address in the field below. The label on the field depends on the entry in the "Addressing" field, e.g. "DW" for "Word" addressing.

5. Click "OK" to close all open dialogs.

___

Note

You cannot change the "Word" entry in the "Addressing" field because it is defined by the "Unsigned 16-bit" data type of the WinCC tag.

___

8.4.5   System parameters

8.4.5.1   System parameters of the channel unit

Introduction

If you require a configuration that deviates from the standard WinCC settings, you can make all the required changes using the "System Parameter" dialog of the channel unit.

The following individual points can be changed:

● the device name

● the Write/Read monitoring time

Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communication module and are unit-specific. The device name represents the logical device name. This field is defined with the entry "/CP_L2_1:/SCP" as default.

Write/Read monitoring time

The write/read monitoring time is the maximum waiting time in seconds for write/read responses of the AS. If no response is made by the AS within the defined time, the connection is broken. This field is assigned a waiting time value of 30 seconds as default

---

Note

The system parameters apply for all CPs in the AS.

8.4.5.2 How to Change the Device Name

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. In the channel unit shortcut menu, select "System parameters". The "System Parameters" dialog opens.
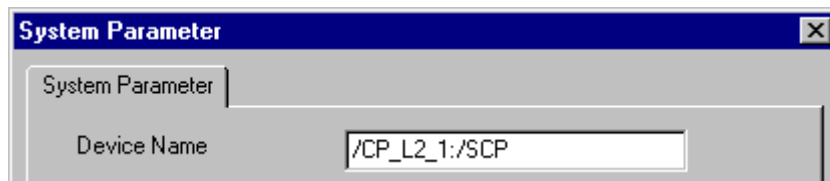
2. Enter the name of the access point in the "Device name" field. This name must match the setting that you have made under Windows via "Start" ➜ "Settings" ➜ "Control panel" ➜ "Set PG/PC interface".

3. Close the dialog by clicking the "OK" button.

Note

The changes only take effect after WinCC is restarted.

8.4.5.3 How to change the write/read monitoring time of process values

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. In the channel unit shortcut menu, select "System parameters". The "System Parameters" dialog opens.
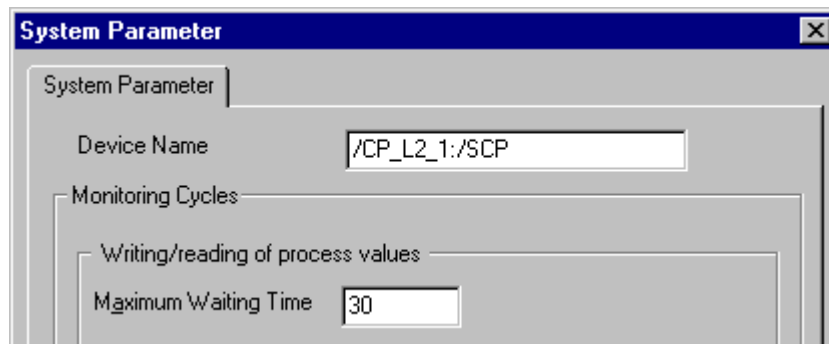
2. Enter the required value in seconds in the "Maximum waiting time" field. You can define a value between 1 and 3600 seconds. This field is assigned with a default value of 30 seconds.

3. Close the dialog by clicking the "OK" button.

Note

The changes only take effect after WinCC is restarted.

8.5 Special Functions

8.5.1 Special functions of the "SIMATIC S5 Profibus FDL" Channel

Introduction

The "SIMATIC S5 Profibus FDL" channel has some special functions, the functionality of which is described in this chapter.

8.5.2 Raw data tags of the "SIMATIC S5 Profibus FDL" channel

8.5.2.1 Raw data tags of the "SIMATIC S5 Profibus FDL" channel

Introduction

A tag of the type "raw data type" is a data telegram.

Raw data tags are required for transferring user data blocks from/to the AS

A raw data tag used by SIMATIC S5 Profibus FDL can be a maximum of 220 bytes in length.

Raw data tag as byte array

A raw data tag as byte array is handled like a normal process tag that is addressed via the address and length of the data block (for e.g. DB 100, DW 20, length 40 Byte).

Writing raw data tags using scripts

If a raw data tag, which is longer than the tag length configured in WinCC, is written to S5 using a VB script, the write process is aborted.

Instead write the raw data tag via a C script using the "SetTagRaw" function. For this function, specify the length of the tags that are to be written.

8.5.2.2 How to configure raw data tags

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter the name of the tag in the "Name" field of the "General" tab, e.g. "Var1_raw_byte". Select "Raw Data Type" in the "Data type" field.
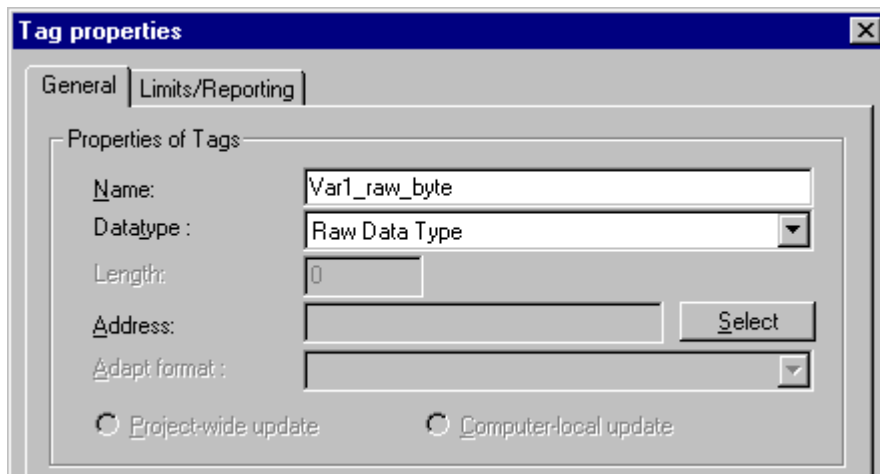
3. Click the "Select" button to open the "Address properties" dialog.
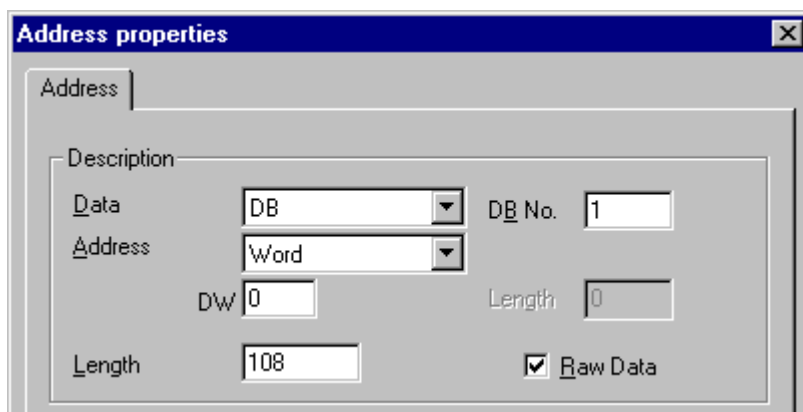
4. Mark the "Raw Data" check box.

5. Enter the length of the raw data block (in bytes) in the "Length" field.

6. In the "Data area" set the data area of the PLC where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

7. Set up the addressing type in the "Addressing" field. The entries "Left byte", "Right byte", "Word" and "Double word" are possible for data type "Raw data type" of the WinCC tag.

8. Enter the value of the start address in the underlying field. The label on the left field depends on the entry in the Data Area and Addressing field, for e.g. "DW" for data area "DB" "Word" for addressing type.

9. Click "OK" to close all open dialogs.

8.5.3 Configuring the communication types

8.5.3.1 Configuring the communication types

Introduction

An FDL connection can be configured so that WinCC runs as an active or passive partner.

If WinCC is configured as an active partner, the values are requested by the WinCC station.

If WinCC is configured as a passive partner, the transfer of values to the WinCC station is initiated by the AS.

8.5.3.2 How to configure an active data transfer

Introduction

This section shows you how to configure an active data transfer to the address area in the AS.

Note

If more than one connection is configured, note that an SAP can only be assigned one time.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

3. You must know the SAP address defined by the AS.

Procedure

1. Choose the "Properties" option from the context menu of the connection. The "Connection Properties" dialog is opened.

2. Click the "Properties" button to open the "Connection parameters" dialog.

3. Enter the station address of the AS in the "PLC Station Address" field on the "Connection" tab.

4. The following settings must be made in the "READ - Function" area:

5. Activate the option "OS active, WinCC is the active partner".

6. Enter the SAP-ID of the WinCC station in the "Own SAP" field.

7. Enter the SAP-ID of the AS in the "Foreign SAP" field.

8. The following settings must be made in the "WRITE - Function" area:

9. Enter the SAP-ID of the WinCC station in the "Own SAP" field.

10. Enter the SAP-ID of the AS in the "Foreign SAP" field.

11. Click "OK" to close all open dialogs.

8.5.3.3   How to configure a passive data transfer

Introduction

This section shows you how to configure a passive data transfer to the address area in the AS.

Note

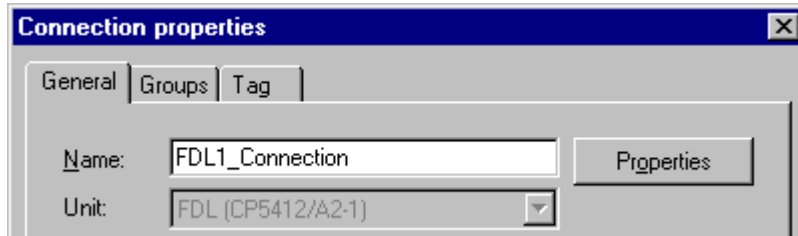If more than one connection is configured, note that an SAP can only be assigned one time.
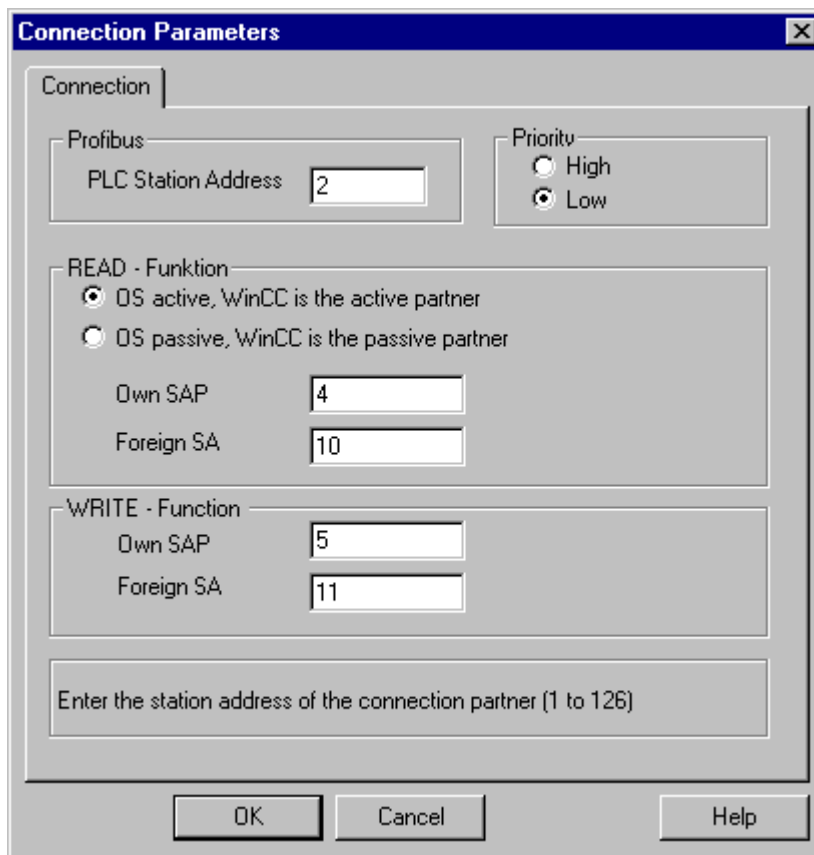

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

3. You must know the SAP address defined by the AS.

Procedure

1. Choose the "Properties" option from the context menu of the connection. The "Connection Properties" dialog is opened.

2. Click the "Properties" button to open the "Connection parameters" dialog.

3. Enter the station address of the AS in the "PLC Station Address" field on the "Connection" tab.

4. The following settings must be made in the "READ - Function" area:

5. Activate the option "OS passive, WinCC is the passive partner".

6. Enter the SAP-ID of the WinCC station in the "Own SAP" field.

7. Enter the SAP-ID of the AS in the "Foreign SAP" field.

8. The following settings must be made in the "WRITE - Function" area:

9. Enter the SAP-ID of the WinCC station in the "Own SAP" field.

10. Enter the SAP-ID of the AS in the "Foreign SAP" field.

11. Click "OK" to close all open dialogs.

8.6 Example of configuring the "SIMATIC S5 Profibus FDL" channel

8.6.1 Example of configuring the "SIMATIC S5 Profibus FDL" channel

Introduction

In this example, you will configure an I/O Field in the Graphics Designer and assign the necessary values to the data handling blocks in the AS.

8.6.2 How to configure the data handling blocks in the AS

Introduction

In this section, you will configure the standard function blocks OB 21 (L2ANLAUF) and OB 1 (L2SNDRCV) in the AS.

By default, the data traffic for the SIMATIC S5 connection by means of PROFIBUS FDL is handled by means of the following blocks.

Within the example, the following SAP numbers are used:

	WinCC	Automation device
SAP number for the READ function	12	6
SAP number for the WRITE function	11	4

Blocks

Function	Block
For the startup OB 20, 21, 22	FB-L2ANLAUF (FB 9)
For cyclic operation OB 1	FB-L2SNDRCV (FB 10)
As internal work DBs for both FBs	DB-L2DBVC3 (DB 10) DB-L2DBVC4 (DB 11) DB-L2DBVC5 (DB 12)

Startup blocks

The communication parameters are defined in the startup blocks, the work DBs are logged in and the communication processor is synchronized. These work steps are executed by calling function block FB9 L2ANLAUF, for example.

Cyclic block

The telegram traffic is handled in the cyclic FB. Received telegrams are entered in the destination data blocks. Should an error occur while this is being done, the telegram is rejected and an error message is issued. Telegrams that have to be sent will be specified by the user in a similar manner to the standard data handling blocks. An acknowledgement is sent when the transfer has been completed.

Requirements

- The data handling blocks SYNCHRON, CONTROL, SEND and RECEIVE must be available on the automation system.

Procedure

1. The startup block (OB 20, 21, 22) is created in STEP5 by choosing the menu option "Editor" ➔ "STEP5 Block" ➔ "in the program file". The program block is called "L2ANLAUF" in the example.

2. The following parameters must be preassigned:

   – Interface number (SSNR) of the CP (e.g. the CP5431)

   – The PROFIBUS address (RADR) of communication processor CP 5613 on the WinCC computer. This number must be unique in the network.

   – The connection parameters of the request types being used, for example, are the parameters RVC4 (for write) and RVC5 (for read), which specify the SAPs of the WinCC station. These SAPs are created when the connection is being set up in WinCC.

   – The request numbers (ANR4 and ANR5), which are set when the FDL connections are being configured for the communication processor are

   – numbers of the work data blocks, DBX4 (for write) and DBX5 (for read)

```
OB   21
NETZWERK 1 von 1              Synchronisieren CP 5431
       :
       :SPA FB 9
NAME  :L2ANLAUF
SSNR  :     KF  +0                SSNR of CP 5431
TIM3  :     KT  000.0             nr
TIM7  :     KT  000.0             nr
RADR  :     KF  +8                PROFIBUS-Address WinCC Station
RVC3  :     KF  +0                nr
RVC4  :     KF  +4                SAP WRITE
RVC5  :     KF  +6                SAP READ
RVC6  :     KF  +0                nr
RVC7  :     KF  +0                nr
ANR3  :     KF  +0                nr
ANR4  :     KF  +134             ANR WRITE
ANR5  :     KF  +135             ANR READ
ANR6  :     KF  +0                nr
ANR7  :     KF  +0                nr
DBX3  :     KY  000,000          nr
DBX4  :     KY  000,011          Work-DB WRITE
DBX5  :     KY  000,012          Work-DB READ
DBX6  :     KY  000,000          nr
DBX7  :     KY  000,000          nr
S/R3  :     KF  +0                nr
       :
       :BE
```

3. An OB 1(cyclic operation) is created in the STEP5 software by choosing the menu option "Editor" ➔ "STEP5 Block" ➔ "in the program file". The program block is called "L2SNDRCV" in the example.

4. Communication with WinCC is performed by means of communication processor CP5431 and function block FB10 L2SNDRCV. When WinCC sends data and has to request data, only two applicable transfer parameters have to be specified for this purpose. These are the parameters DBX4 (for write) and DBX5 (for read), which specify the numbers of the two work data blocks for the request types being used. These SAPs are created when the connection is being set up in WinCC.

```
OB   1
NETZWERK 1 von 1           Communication Manual
     :
     :SPA FB 10                              Communication
NAME :L2SNDRCV
STR3 :    M 0.0                              nr
STR7 :    M 0.0                              nr
RDY  :    MB 0                               nr
FAIL :    MB 0                               nr
TUC3 :    T 0                                nr
TUC7 :    T 0                                nr
DBX3 :    KY 000,000                         nr
DBX4 :    KY 000,011                         Work-DB WRITE
DBX5 :    KY 000,012                         Work-DB READ
DBX6 :    KY 000,000                         nr
DBX7 :    KY 000,000                         nr
     :
     :SPA FB 6                               Add and Increment
NAME :ADD_INC
     :
     :BE
```

5. Load the STEP5 program into the automation device.
   This is done in STEP5 by choosing the menu option "Object" → "Blocks" → "Transfer" → "PLC file". Select the option "All blocks" in the "Selection" field to load all previously created blocks onto the automation system.

8.6.3 How to configure an I/O Field

Introduction

You will configure an I/O Field in this section.

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. Choose the "New Connection" option from the pop-up menu of the channel unit "FDL (CP5412/A2-1)" and set up a connection called "TestFDL".

2. By clicking the "Properties" button you can configure the connection parameters.
   Enter the station address of the automation system in the " PLC station address" field.
   You can configure a FDL connection so that WinCC is either the active or the passive peer.
   If WinCC is configured as an active partner, the values are requested by the WinCC station.
   If WinCC is configured as a passive partner, the transfer of values to the WinCC station is initiated by the AS.
   Click "OK" to close all open dialogs

3. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.
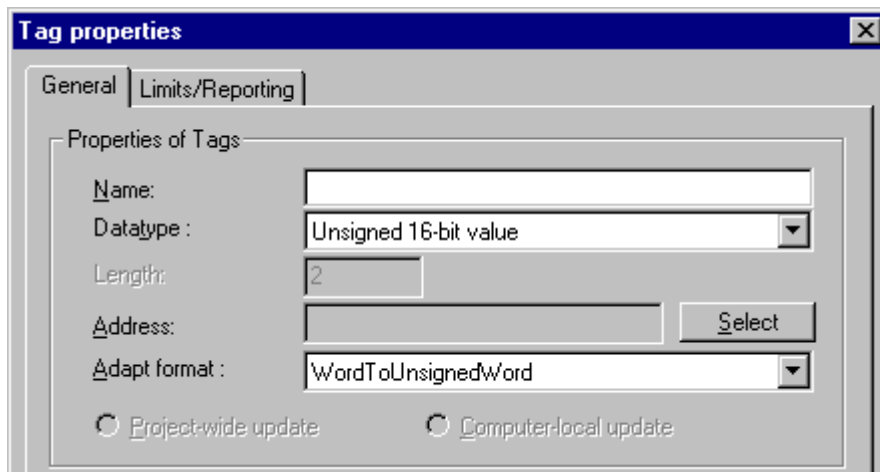
4. Enter tag name "FDLWord1_Test" in the "Name" field. Set the data type to "unsigned 16-bit".

5. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PLC where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field. Enter the byte address in the "DW" field.
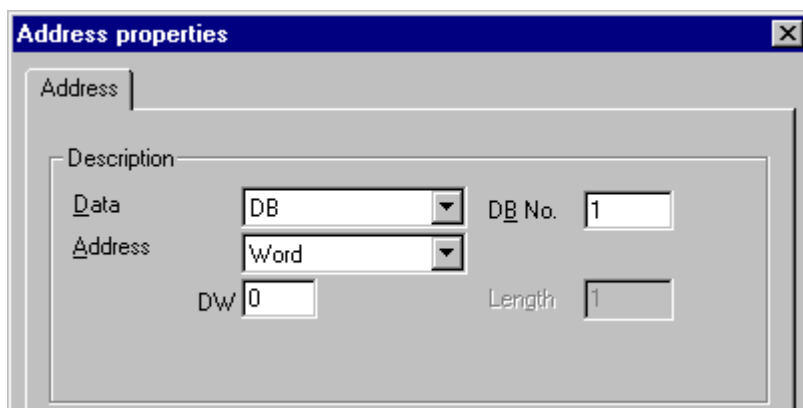
6. Click "OK" to close all open dialogs.

7. You must integrate the smart object "I/O field" into a picture within Graphics Designer.

8. Connect the "I/O field" to a configured tag.

9. Activate the project by clicking the "Activate Runtime" button in the toolbar or by selecting "Activate Runtime" in the "File" menu. All changes to the configured tag are displayed in the "I/O field" in Runtime.

S5 Programmers Port AS511   9

9.1   WinCC channel "SIMATIC S5 Programmers Port AS511"

Introduction

The communication driver "SIMATIC S5 Programmers Port AS511" is utilized for the serial connection through a TTY interface to the SIMATIC S5 automation system.

This chapter describes

- how to configure the data transfer with the "SIMATIC S5 Programmers Port AS511" channel.

- how to configure a connection and a tag.

Channel Unit

The communication driver has one channel unit for controlling a COM port for the serial connection.

The following capability is available:

- Channel unit S5-AS511 for serial communication via a "Siemens-specific" protocol.

9.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag
- Unsigned 8-bit value
- Signed 8-bit value
- Unsigned 16-bit value
- Signed 16-bit value
- Unsigned 32-bit value
- Signed 32-bit value
- Floating-point number 32-bit IEEE 754
- Text tag, 8-bit character set
- Raw data type

9.3 Configuring the Channel

9.3.1 Configuring the "SIMATIC S5 Programmers Port AS511" channel

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Programmers Port AS511".

● Configuring the connection

● Configuring the tags

9.3.2 How to configure the connection

Introduction

The process connection using a serial connection is possible with the SIMATIC S5 automation system. The AS 511 communication processor is used in the automation system.

No additional communication module is required in WinCC. Communication is set up via either the TTY port on a PG 760 or a COM Port that is part of the system's standard equipment. In the later case, an additional port converter is required V.24/V.28 <---> TTY.

This serial link supports transmission rates of up to 19200 baud.

The following procedure can be used to assign one of the PC's serial ports to the AS511-NT drivers.


Note

During communication between WinCC and a S5 automation system via the "AS511" channel, data blocks may not be transferred, created or deleted in the AS. The memory in the S5 may also not be compressed. This last restriction is the result of the absolute addressing of memory in the S5. If changes are necessary, the link to WinCC must be disconnected.

Procedure

1. Select the connection under channel unit "S5-AS511" and open the "Connection properties" dialog window with menu point "Properties" in the context menu.

2. Click on the "Properties" button. The "Connection parameters" dialog is opened.
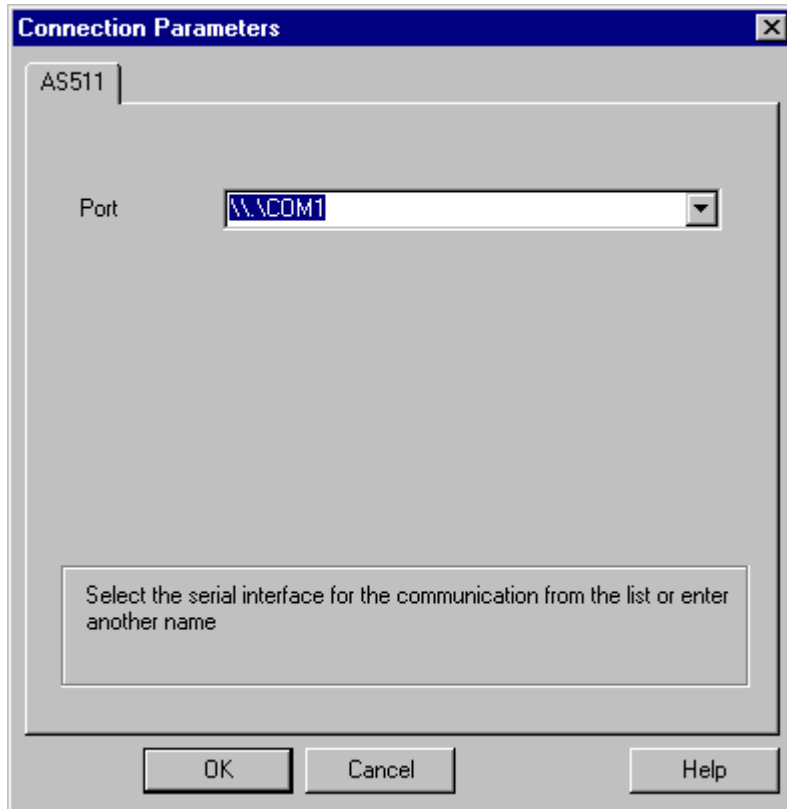
3. In the "Port" field, select the port to be used for the serial link.

9.3.3 Configuring the tags

9.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 Programmers Port AS511", tags of different data types can be created within WinCC. This is described in the following section.

Note

Addresses of the tags are not checked for plausibility in WinCC. If an address is used, which is not available in the AS, the status "Addressing error" will be set.

In DB and DX data blocks, reads and writes can only be made up to address 255.

Times cannot be written.

### 9.3.3.2 How to configure the address of a tag

Introduction

The tag address is entered according to the address structure of the SIMATIC S5.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.
2. Select the desired data type (e.g. Signed 8 bit value) from the "General" tab.
3. Click the "Select" button. The "Address properties" dialog is opened.

4. Select the "SYSVAR" tab.
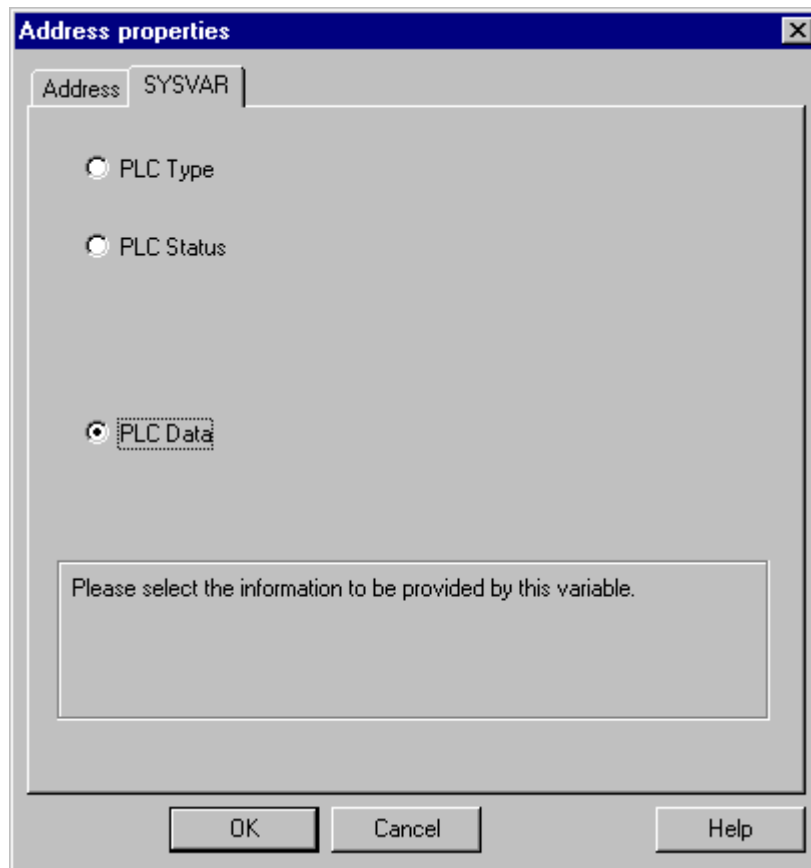
5. Click on the selection field to choose whether the tag should transfer the "PLC Type", the current status ("PLC Status") or other data ("PLC Data").

6. Only if you have selected "PLC data" will you have to click on the "Address" tab to define the S5 address of the tag.
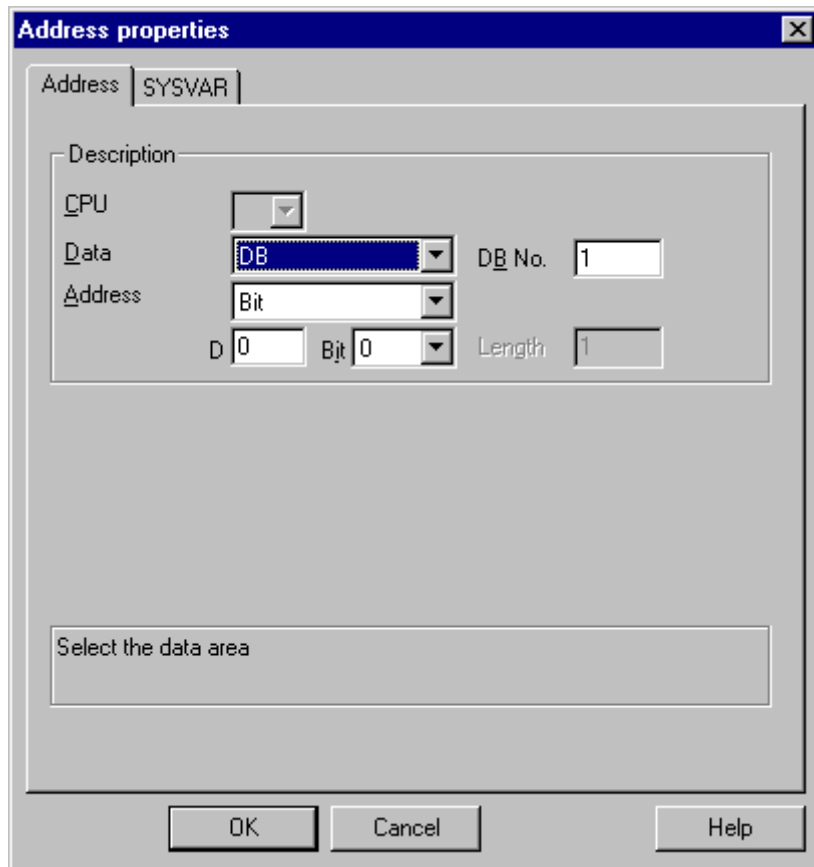
7. Choose whether the tag is located in a data block, in an extended data block, in a flag area, an input range or an output range in the "Data Area" field.

8. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.

9. The type of addressing is entered in the "Addressing" field. Normally, the default definition can be used.

10. Enter the address in the respective field (e.g. "DW ").

Frequently, the memory in the PLC can only be accessed by byte or word. When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to change individual bits in the memory of the PLC as well. For this purpose, the addressed memory area is read from the PLC for every single write request and the

corresponding bits and/or bytes are modified. Afterwards, the data is written back to the PLC's memory.

Note

Changes that have been made by the PLC in a read data area are overwritten when writing back into the data area.

Depending on the type of tag, you can access the automation system's memory bit-wise or byte-wise.

Addresses of the tags are not checked for plausibility in WinCC. If an address is used, which is not available in the AS, the status "Addressing error" will be set.

In DB and DX data blocks, reads and writes can only be made up to address 255.
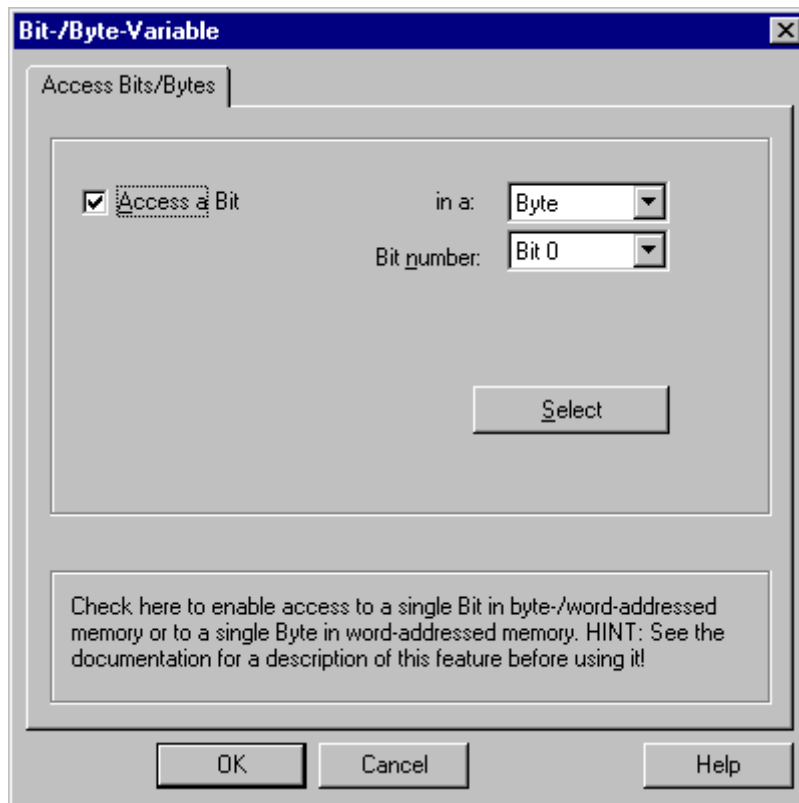
Times cannot be written.

9.3.3.3   How to configure a tag with bit-wise access

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.
2. Select data type "binary tag" on the "General" tab.
3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a bit" checkbox and define the addressing for the bit.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the selection field.

7. Select the number of bit to be changed in the selection field.

9.3.3.4 How to Configure a Tag with Byte by Byte Access

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "Unsigned 8 bit value" or "Signed 8 bit value" in the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a byte" checkbox and define the addressing for the byte.
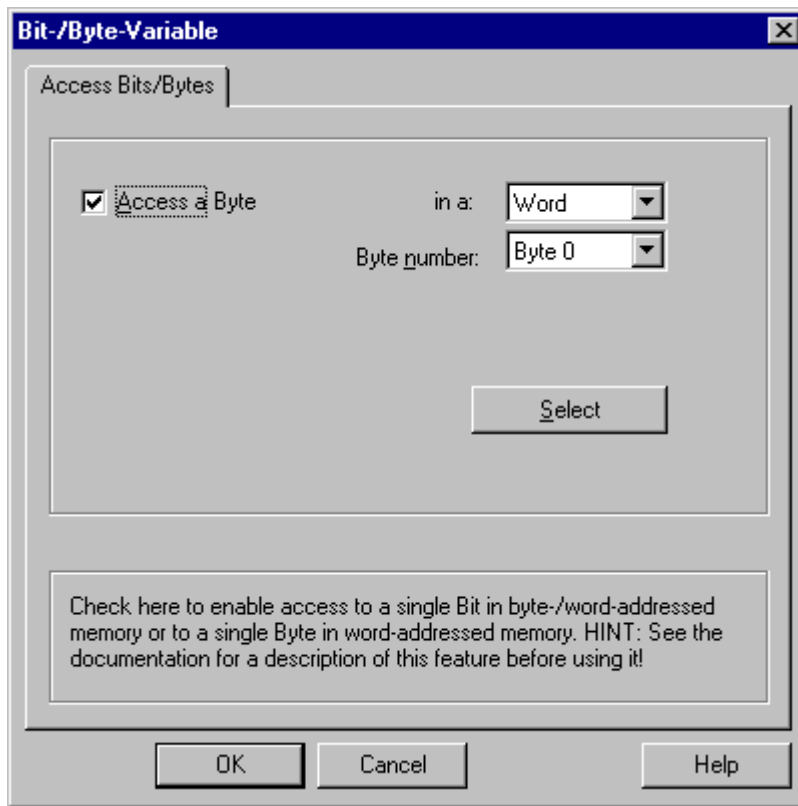

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the selection field.

7. Select the number of byte to be changed in the selection field.

S5 Serial 3964R

10

10.1   WinCC channel "SIMATIC S5 Serial 3964R"

Introduction

The communication driver "SIMATIC S5 Serial 3964R" is implemented for the serial link between a WinCC station and a SIMATIC S5 automation system.

This chapter describes

- how to configure the data transfer with the "SIMATIC S5 Serial 3964R" channel.

- how to configure a connection and a tag.

Channel Unit

The communication driver has one channel unit for controlling a COM port for the serial link.

The following capability is available:

- Channel unit S5-RK512 (3964R) for serial communication via the 3964R or 3964 protocol.

10.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag
- Unsigned 8-bit value
- Signed 8-bit value
- Unsigned 16-bit value
- Signed 16-bit value
- Unsigned 32-bit value
- Signed 32-bit value
- Floating-point number 32-bit IEEE 754
- Text tag, 8-bit character set
- Raw data type

10.3 Configuring the Channel

10.3.1 Configuring the "SIMATIC S5 Serial 3964R" channel

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Serial 3964R".

10.3.2 How to configure the connection

Introduction

The process connection using a serial connection is possible with the SIMATIC S5 automation system. On the automation system, the communication processor CP 544 or a second, plug-in serial port is used on the CPU module (module receptacle SI2).

No additional communication module is required in WinCC. Communication takes place by means of the default COM ports available on the system.

This serial link supports transmission rates of up to 19200 baud.

Note

When the SIMATIC S5 is actively sending with job type "Pseudowrite", the message length must not exceed 64 words.


Procedure

1. Select the connection and open the "Connection properties" dialog window with menu point "Properties" in the context menu.

2. Click on the "Properties" button. The "Connection parameters" dialog is opened.

3. Select the Serial 3964R tab.
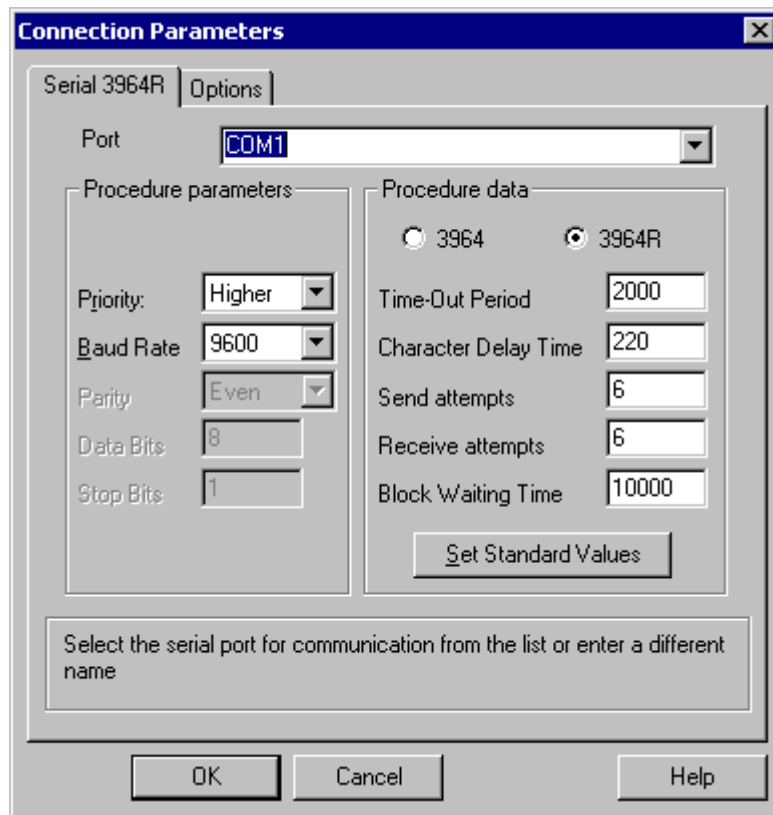
4. Select the communications port (COM1 or COM2) for the connection in the "Port" field.

5. Set the data transfer speed to the value used in the "Baud rate" field of the "Procedure parameters" area. The priority in the case of an initiation conflict (simultaneous line bid by WinCC and the automation system) is set in the "Priority" field.
   The priority has to be set differently from the SIMATIC S5.

6. In the "Procedure data" area, select either the "3964" or "3964R" line protocol. You should only change the default values for the procedure data (such as acknowledgement time, character delay time, etc.) in exceptional cases. Make sure that they match the parameters on the automation system.

7. Now select the "Options" tab.
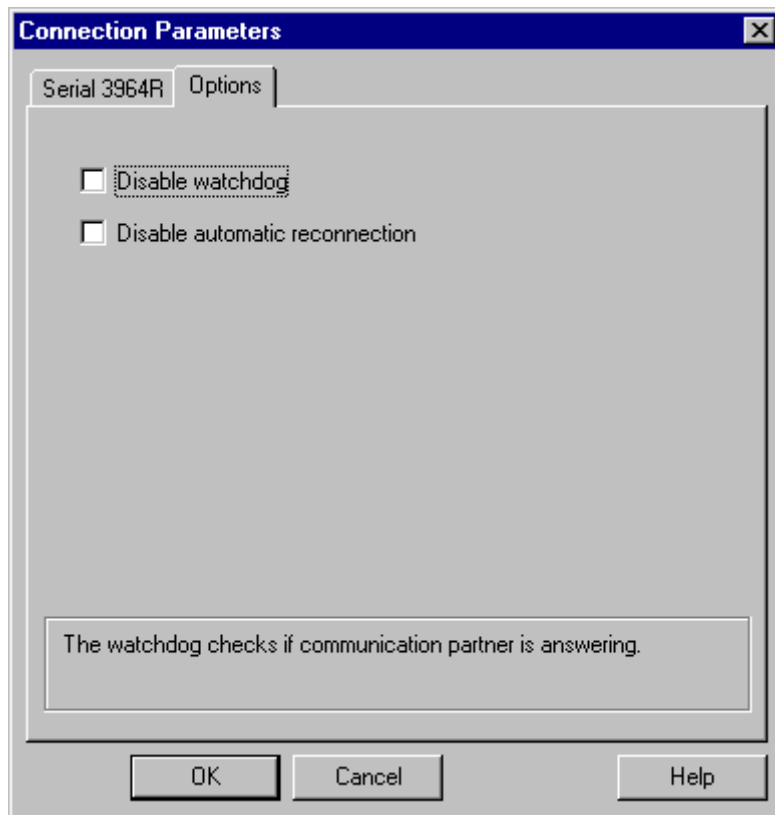
8. You can disable cyclic life beat monitoring and disable the automatic reconnection on the "Options" tab.

10.3.3 Configuring the tags

10.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 3964R", data types binary and byte can be defined within WinCC. The following describes how to configure a tag of these data types.
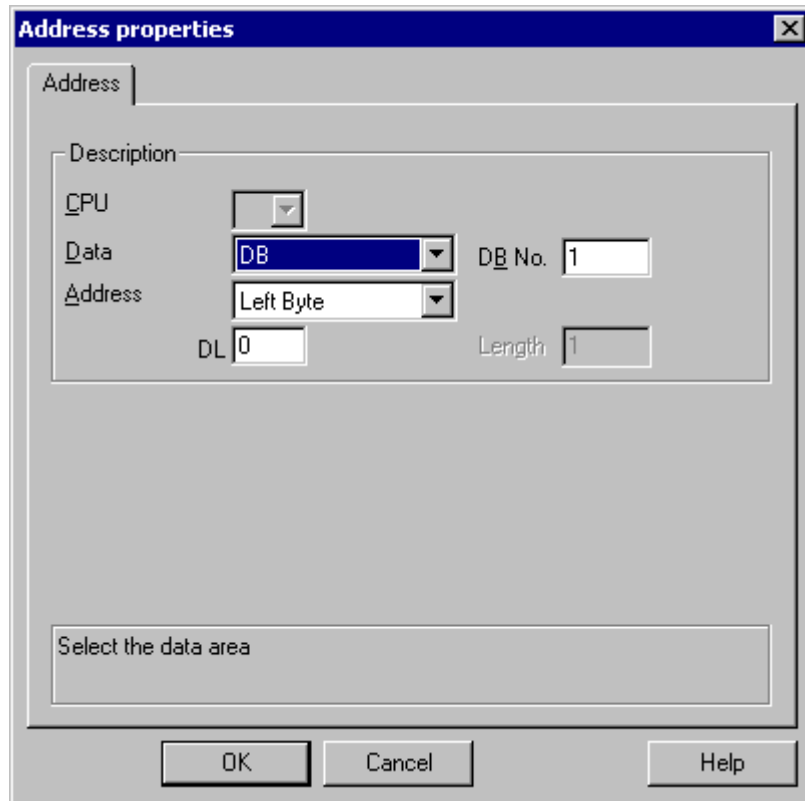
10.3.3.2 How to configure the address of the tag

Introduction

The tag address is entered according to the address structure of the SIMATIC S5.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select the desired data type (e.g. Signed 8 bit value) from the "General" tab.

3. Click the "Select" button. The "Address properties" dialog is opened.

4. Choose whether the tag is located in a data block, in an extended data block, in a flag area, an input range or an output range in the "Data Area" field.

5. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.

6. The type of addressing is entered in the "Addressing" field. Normally, the default definition can be used.

7. Enter the address in the respective field (e.g. "DL ").

Note

Only read access is possible to the Inputs, Outputs, Timers and Counters address areas. Read and write access are possible to data blocks (DB, DX).

Do not use data word addresses which are greater than 255. Due a system characteristic of the RK 512, only the data word addresses 0 to 255 are permissible.
Higher addresses can be configured but lead to data violation on all tags configured in the connection.

Frequently, the memory in the PLC can only be accessed by byte or word. When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to change individual bits in the memory of the PLC as well. For this purpose, the addressed memory area is read from the PLC for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the PLC's memory.

Note

Changes that have been made by the PLC in a read data area are overwritten when writing back into the data area.
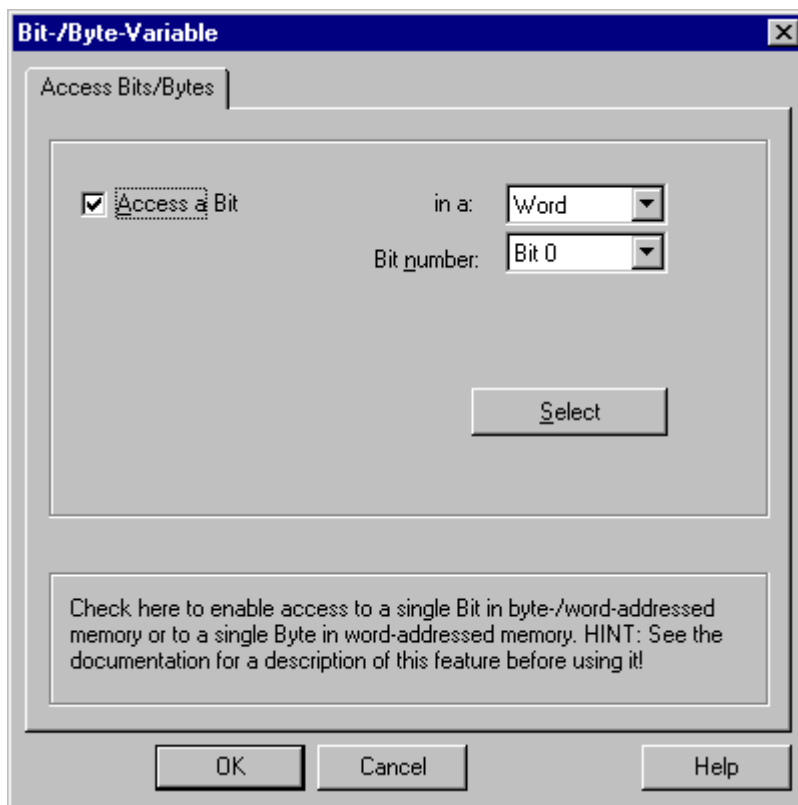

Depending on the type of tag, you can access the automation system's memory bit-wise or byte-wise.

10.3.3.3   How to configure a tag with bit-wise access

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "binary tag" on the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a bit" checkbox and define the addressing for the bit.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the selection field.

7. Select the number of bit to be changed in the selection field.

Note

With the S5, flags, inputs and outputs can be addressed by byte, data blocks (DB, DX) are addressed by word.
Only read access is possible to the Inputs, Outputs, Timers and Counters address areas.
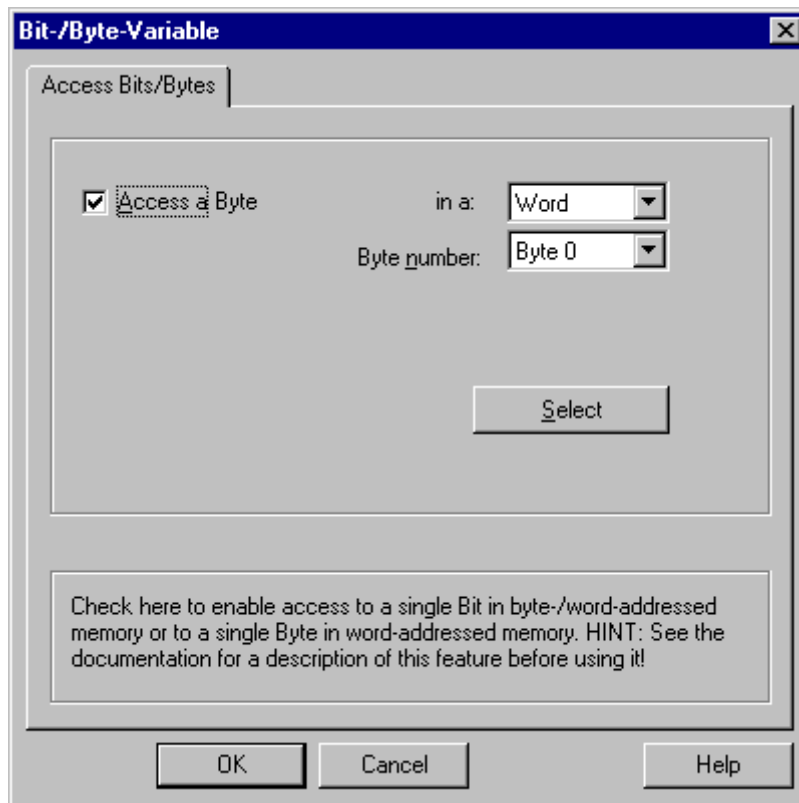Read and write access are possible to data blocks (DB, DX).

10.3.3.4 How to Configure a Tag with Byte by Byte Access

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "Unsigned 8 bit value" or "Signed 8 bit value" in the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a byte" checkbox and define the addressing for the byte.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the selection field.

7. Select the number of byte to be changed in the selection field.

Note

With the S5, flags, inputs and outputs can be addressed by byte, data blocks (DB, DX) are addressed by word.
Only read access is possible to the Inputs, Outputs, Timers and Counters address areas.
Read and write access are possible to data blocks (DB, DX).

SIMATIC S7 Protocol Suite 11

11.1 "SIMATIC S7 Protocol Suite" Channel

Content

The "SIMATIC S7 Protocol Suite" channel supports communication between a WinCC station and the SIMATIC S7 automation systems. The suite supports a variety of protocols and types of networks.

This section shows you

- how to configure various connections and tags for the channel

- how to create a sample project

- how to use the channel's special functions such as the AR_SEND function, raw data tags and software redundancy

11.2 WinCC Channel "SIMATIC S7 Protocol Suite"

Function principle

The Channel "SIMATIC S7 Protocol Suite" is used to link SIMATIC S7-300 and SIMATIC S7-400 automation systems.

Depending on the communication hardware used, the system supports connections via the following channel units:

- Industrial Ethernet and Industrial Ethernet (II): For communication via a communication processor (such as CP 1612; CP1613) with SIMATIC NET Industrial Ethernet.

- MPI: For communication via the external MPI port of a programming device (e.g. PG 760/ PC RI45), via an MPI communications processor or a communications module (e.g. CP 5511, CP 5613).

- Named Connections: to communicate with STEP 7 via a symbolic connection. These symbolic connections are configured using STEP 7 and are needed e.g. to provide high-availability communication with the PLC S7-400 in conjunction with the redundancy in H/F systems.

- PROFIBUS and PROFIBUS (II): For communication via a communication processor (such as CP 5613) with SIMATIC NET PROFIBUS.

- Slot-PLC: to communicate with a Slot PLC (e.g. WinAC Pro) that is installed as a PC card in the WinCC computer.

- Soft-PLC: to communicate with a Software PLC (e.g. WinAC Basis), that is installed as an application on the WinCC computer.

- TCP/IP: to communicate with networks using the TCP/IP protocol.

For more information regarding the diagnosis of channels and tags, refer to "Communication Diagnostics".

Detailed procedures

Communication Manual: The communication manual contains additional information and extensive examples for the channel configuration. You can download this manual at "www.siemens.com/automation/csi_de_WW/product".

In the field in the top right corner, enter the order number "6AV6392-1CA06-0AA0" for the Communication Manual. Click on "Search" to start searching for the manual.

See also

11.3   Channel unit selection

Introduction

To setup a communication link to an existing or planned network, a selection must be made for:

● one of the channel's channel units

● a suitable communication processor for the WinCC station

● a suitable communication module for a specific automation system

This section provides an overview of the various options.

There are two different types of communication processors for WinCC:

● Communication processors for the so-called Hardnet. They have their own microprocessors and reduce the load on the system's CPU. It is possible to use two different protocols at the same time (multi-protocol operation).

● Communication processors for the so-called Softnet. They do not have their own microprocessors. Only one protocol can be used at a time (mono-protocol operation).

Channel Unit Assignment

The table below presents the assignment of the channel units of the Channel "SIMATIC S7 Protocol Suite" to networks and automation systems.

Channel unit of the channel	Communication Network	Automation system
MPI	MPI	S7-300 and S7-400
PROFIBUS and PROFIBUS (II):	PROFIBUS	S7-300 and S7-400
Industrial Ethernet + Industrial Ethernet (II)	Industrial Ethernet	S7-300 and S7-400
TCP/IP	Industrial Ethernet via TCP/IP	S7-300 and S7-400
Named Connections	Industrial Ethernet or PROFIBUS	S7-400 H/F Systems
Slot PLC	"Soft K-Bus" (internal)	PC (internal)
Soft PLC	"Soft K-Bus" (internal)	PC (internal)

MPI

For communication with the S7-300 and S7-400 PLCs via MPI, Channel "SIMATIC S7 Protocol Suite" includes the "MPI" channel unit.

The MPI network corresponds in largest part to the PROFIBUS network with preset parameters and limitations with regard to the number of subscribers and the transfer rate. The same communication processors and modules are used for communication via MPI as for the PROFIBUS network. The same communication protocols are also used.

Automation System Communication Links

The S7-300 and S7-400 programmable controllers can communicate via a MPI network using their internal MPI port or a suitable communication module. The table shows the recommended components.

System	CPU or Communication Module (recommended)
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

Communication Processors for WinCC

The following table shows the communication processors recommended for connecting a WinCC station to a MPI network. Only one communication processor per WinCC computer can be used for the MPI communication. Each card must also have a driver suitable for the respective communication protocol.

Communication Processor (WinCC)	Assembly/Type
CP 5613	PCI Card/ Hardnet
CP 5511	PCMCIA Card/ Softnet
CP 5611	PCI Card/ Softnet

PROFIBUS

For communication with the S7-300 and S7-400 PLCs via PROFIBUS, Channel "SIMATIC S7 Protocol Suite" includes the "PROFIBUS" and "PROFIBUS II" channel units.

The channel units support communication via the Hardnet and Softnet modules.

Automation System Communication Links

The S7-300 and S7-400 programmable controllers can communicate via a PROFIBUS network using their internal port or a suitable communication module. The table shows the recommended components.

System	CPU or Communication Module
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

Communication Processors for WinCC

The following table shows the communication processors recommended for connecting a WinCC station to the PROFIBUS. The "PROFIBUS" channel unit supports communication via Hardnet and Softnet cards. A WinCC station supports the use of up to two of these modules. Each communication processor must also have a driver suitable for the respective communication protocol.

Communication Processor (WinCC)	Assembly/Type
CP 5613	PCI Card/ Hardnet
CP 5511	PCMCIA Card/ Softnet
CP 5611	PCI Card/ Softnet

Industrial Ethernet and TCP/IP

In WinCC, the Channel "SIMATIC S7 Protocol Suite" supports multiple channel units for communication via Industrial Ethernet:

- "Industrial Ethernet" and "Industrial Ethernet (II)" Channel Units for "ISO" Protocol with S7 Functions

- "TCP/IP" Channel Unit for "ISO-on-TCP" Protocol with S7 Functions

The channel units support communication via the Hardnet and Softnet modules.

Automation System Communication Modules

If the S7-300 or S7-400 PLC should communicate via an Industrial Ethernet using the "ISO" or "ISO-on-TCP" protocol, it must be equipped with a suitable communication module. The table shows the recommended components.

System	Communication module for Industrial Ethernet	Communication module for TCP/IP-Protocol
S7-300	CP 343-1	CP 343-1 TCP
S7-400	CP 443-1	CP 443-1 TCP CP 443-1 IT

Communication Processors for WinCC

A WinCC station uses Industrial Ethernet to communicate via the "ISO" or "ISO-on-TCP" protocol with one of the communication processors recommended in the table below.

Each communication processor also has a driver suitable for each of the respective communication protocols.

Communication Processor (WinCC)	Assembly/Type
CP 1612	PCI Card/ Softnet
CP 1613	PCI Card/ Hardnet
CP 1512	PCMCIA Card/ Softnet

11.4   Overview of the supported data types

Introduction

For configuring a tag, you need to define data type and type conversion according to the data format in AS.

The table shows the data types supported by the channel and the use of type conversions.

Supported data types

Data Types	Type conversion
Binary tag	No
Signed 8-bit value	Yes
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Text tag, 8-bit font	No
Raw data type	No

You will find additional information about type conversion in the "Communication" section.

11.5 Configuring the Channel

11.5.1 "SIMATIC S7 Protocol Suite" Channel - Configuration

Introduction

This section will show you how to configure the "SIMATIC S7 Protocol Suite" channel.

1. Installing the Channel
2. Channel unit selection
3. Configuring a connection
4. Tag configuration

System parameter configuration

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

See also

System Parameters of the Channel Unit (Page 236)

Configuring the tags (Page 230)

Channel units of the "SIMATIC S7 Protocol Suite" channel (Page 213)

Diagnosis of Channels and Tags (Page 369)

11.5.2 How to configure the "SIMATIC S7 Protocol Suite" channel

Introduction

This section will show you how to install the "SIMATIC S7 Protocol Suite" channel.

1. Installing the Channel
2. Channel unit selection
3. Creating a connection
4. Inserting a tag
5. Configuring the system parameters in a customized WinCC installation

Prerequisites:

- The communication module is built in.
- The hardware driver has been installed.
- Cable connection to AS exists.

Procedure

1. Select the entry "Add New Driver" in the shortcut menu of Tag Management. A selection dialog is opened.

2. Select "SIMATIC S7 Protocol Suite.chn" and close the dialog by clicking OK. The channel is set up and the channel driver is displayed in Tag Management along with the channel units.

3. Select the desired channel unit and select the "New Connection" entry in the pop-up menu.

4. In the "Connection Properties" dialog, enter the connection name. Close the dialog with the "OK" button.

5. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

6. Define the name and data type of the tags in the General tab.
   Alternately, you can define a start value and a substitute value for the tags in the Limits/Reporting tab.
   For a detailed description about configuring the tags for connecting a specific channel unit, close the dialog and continue with the topic "Configuring tags" within the respective channel unit.

7. Click the "Select" button to open the "Address properties" dialog. Set up the address area of the desired data.

8. Close both of the dialogs by clicking the "OK" button.

9. If your WinCC system and the communication hardware is non-standard, you also need to set the system parameters to the non-standard values. Additional information on this topic may be found under "System parameters".

11.5.3 Channel units

11.5.3.1 Channel units of the "SIMATIC S7 Protocol Suite" channel

Introduction

The following chapters describe how to configure the channel units and a corresponding connection. There can be multiple connections in the same channel unit.

See also

11.5.3.2   "Industrial Ethernet (I+II)" channel units"

Channel Units "Industrial Ethernet" + "Industrial Ethernet (II)"

Principle of Operation

The channel unit "Industrial Ethernet" is used to connect WinCC to the S7 automation systems via the Industrial Ethernet. Communication is possible via the communications modules (CP), e.g. in the case of automation system S7-300 via CP 343-1 and in the case of S7-400 via CP 443-1.

In WinCC different communications processors can be used, e.g. CP 1613. Via the channel unit "Industrial Ethernet (II)" a second communications processor can be addressed. Because communication takes place via the "ISO" transport protocol, it is not necessary to configure the logical connection in the local database.

The function and configurations regarding these channel units are identical.

Typical unit terminology

Communications processor

A communications processor (CP) is a module that supports communication between the WinCC computer and a specific network.

"ISO" transport protocol

ISO transport is a layer of the ISO-OSI reference model and offers services related to the transfer of data via connections. The transport layer handles data flow control, blocking and acknowledgment tasks.

The protocol defines the structure of the data traffic with regards to content on the physical line. It defines, among other things, the mode of operation, the procedure when establishing a connection, data backup or the transmission speed.

Industrial Ethernet

The Industrial Ethernet is the most efficient subnet in the industrial environment. It is suitable for the factory and cell levels and facilitates the exchange of large data volumes over large distances between a large number of participants.

The Industrial Ethernet is an open communications network and is standardized in accordance with IEEE 802.3. Its prime advantages are its speed, simple extendibility and openness as well as high availability and worldwide utilization. The configuration process requires a minimum of effort.

See also

Configuring the tags (Page 230)

How to configure a "Industrial Ethernet" channel unit connection (Page 215)

How to configure a "Industrial Ethernet" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

For S7 automation systems, a communications module, e.g. a CP 343-1 in the S7-300 or a CP 443-1 in the S7-400, is used for the communication.

A communications processor, for example CP 1613, is used in WinCC. A second communications processor can be addressed/increased via the "Industrial Ethernet II" channel unit.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".
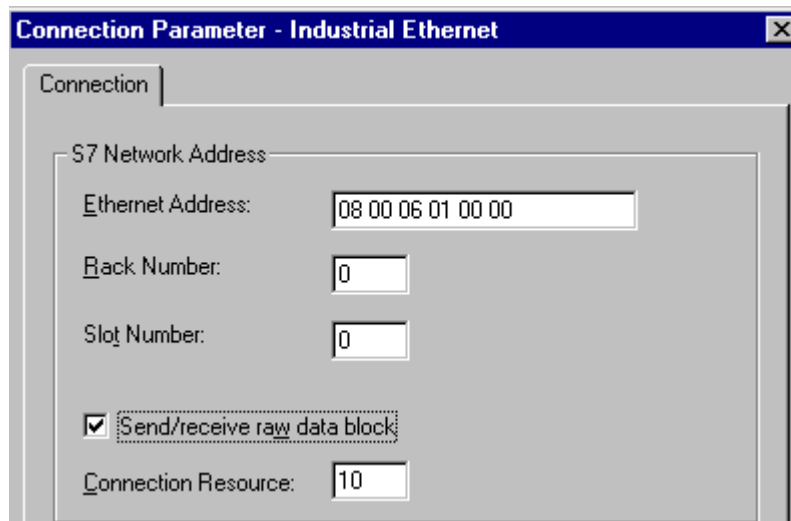
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the channel unit "Industrial Ethernet". The "Connection Properties" dialog is opened.

2. Enter a connection name, for example "Test_Ind_Eth", in the General tab.

3. Click the "Properties" button to open the "Connection Parameters - Industrial Ethernet" dialog.

Connection Parameter - Industrial Ethernet

Connection

S7 Network Address

Ethernet Address: `08 00 06 01 00 00`

Rack Number: `0`

Slot Number: `0`

☑ Send/receive raw data block

Connection Resource: `10`

4. Enter the station address of the automation system on the bus in the field "Ethernet Address".

5. Enter the number of the rack in which the CPU that is to be addressed is located in the "Rack Number" field.

6. The CPU's slot number in the specified rack must be entered in the corresponding field "Slot Number".

7. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/ BRCV data blocks via the connection. If the check box is active, the field "Connection Resource" can be edited. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP7 when the connection is configured in the PLC.

8. Close both of the dialogs by clicking the "OK" button.

Note

When using an S7-300 or S7-400 with an external communications module, you must enter the Rack/Slot number of the CPU.

If the wrong Rack or Slot Number is entered, the communications link will not be established!

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

Diagnosis of Channels and Tags (Page 369)

11.5.3.3 "MPI" channel unit

"MPI" channel unit

Principle of Operation

The MPI channel unit is used to connect WinCC to the SIMATIC S7-300 and S7-400 automation systems via MPI.

This can be done in WinCC via

- the internal MPI interface of the programming devices such as PG 760/PC RI45

- a communication processor such as CP 5613 (PCI card)

The so-called MPI module (ISA card) is also suitable - it exists but is hard to get. It was replaced by the communication processors.

In PLC, linking is done via the MPI interface of the CPU or a corresponding communication module.

Typical unit terminology

MPI

MPI means Multi Point Interface and is a communication link where there can be multiple participants. The connection to the communication network takes place as follows:

- In PLC via the MPI interface of the CPU or a communication module,

- In WinCC via the built-in MPI interface, for e.g., of a programming device or a communication processor (network card).

Communications processor

A communications processor (CP) is a module that supports communication between the WinCC computer and a specific network.

See also

Configuring the tags (Page 230)

How to configure a "MPI" channel unit connection (Page 217)

How to configure a "MPI" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

S7-300 and S7-400 PLCs either use the internal MPI interface or a communication module such as CP 342-5 (SIMATIC S7-300) or CP 443-5 (SIMATIC S7-400).

If WinCC is installed on a PG 760/PC RI45, the internal MPI interface can be used; otherwise, you need to have a built-in MPI module. Alternately, you can also use a communication module such as CP 5511 (PCMCIA card).

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Requirements

- The "SIMATIC S7 Protocol Suite" (communication) driver must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the channel unit "MPI". The "Connection Properties" dialog is opened.
2. Enter a connection name, for example "Test_MPI", in the General tab.
3. Click the "Properties" button to open the "Connection Parameters - MPI" dialog.
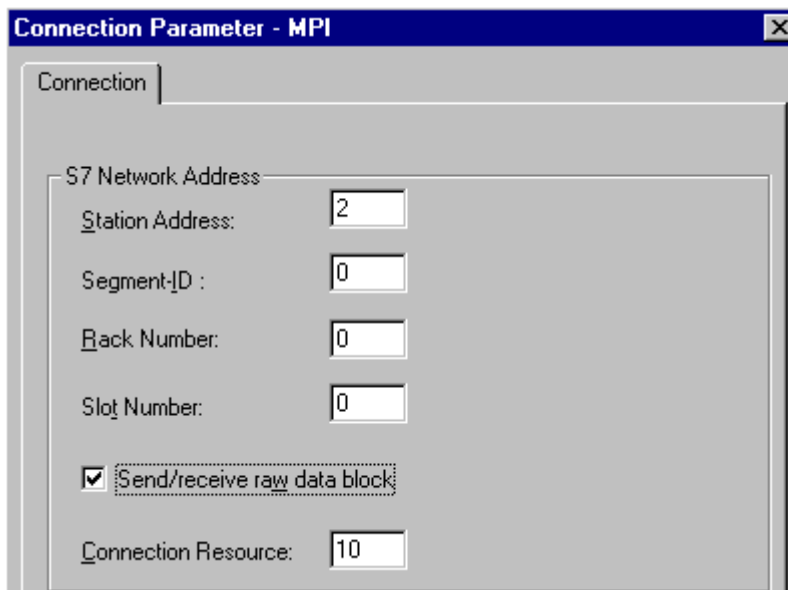
4. Enter the station address in the Station Address field of the automation system on the bus in the appropriate field.
5. The field "Segment ID" is currently not supported. The value must remain at "0".
6. Enter the number of the rack in which the CPU that is to be addressed is located in the "Rack Number" field.
7. Enter the "Slot Number" of the CPU in the specified rack.

8. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/ BRCV data blocks via the connection. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP7 when the connection is configured in the PLC.

9. Close both of the dialogs by clicking the "OK" button.

Note

When using an S7-300 or S7-400 with an external communications processor, you must enter the Rack/Slot number of the CPU.

If the wrong Rack or Slot Number is entered, the communications link will not be established!

While using a S7-300, for the link via the internal MPI interface of the CPU, the rack/ slot number = 0 must be given.

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.3.4 "Named Connections" channel unit

"Named Connections" channel unit

Principle of Operation

This channel unit is used for setting up a symbolic connection configured using STEP 7. WinCC can thus access redundant and non-redundant connection via a symbolic connection name. The symbolic connections are required, for e.g., for high availability communication using S7-400 PLC in connection with redundancy in H/F systems.

The symbolic connection names are configured in STEP 7 using the NETPRO application. The connection names, connection parameters and the application names are stored in a database (*.XDB). This databased is automatically stored by the PLC/OS Engineering Tool "Mapper" in the corresponding WinCC project directory; however, it can also be copied outside this directory, for e.g. if you are not using the "Mapper".

Note

There should only be one XDB file per communication participant in the WinCC system. Hence, a XDB file should not be copied and used on multiple WinCC computers.

You have the following options to activate this database in WinCC:

- If the XDB file is located outside the project directory (for e.g. because the Mapper tool is not used), you need to enter the path and name of the XDB file in the "Set PG/PC interface" (Control Panel) in the STEP 7 Configuration tab before starting WinCC.

On starting WinCC, this XDB file is read from this external directory provided no file exists within the project directory. This procedure is helpful when multiple projects have to use the same centrally stored database.

- If the Mapper tool is used, it automatically copies the XDB file to the WinCC project directory. On starting WinCC and opening the project, the data is read from the S7 channel and entered in the registration database of Windows.

Thereafter, a connection can be configured in WinCC by assigning one of the symbolic connection names to the selected application name.

Note

The application and connection name can also be entered manually, for e.g. if an XDB file does not exist for the symbolic connection name or if the project is to be transferred to another computer. It is necessary to check the correct writing of the name in STEP 7 because there is no name validation in the CS mode.

Typical unit terminology

Communications processor

A communications processor (CP) is a module that supports communication between the PLC and a specific network.

See also

Configuring the tags (Page 230)

How to configure a "Named Connections" channel unit connection (Page 220)

How to configure a "Named Connections" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the S7-400 PLC via a symbolic connection.

For setting up a logical connection, one of the symbolic connection names listed in the "Connection name" field is assigned to a selected application name.

The symbolic connection names and application names are configured in STEP 7.
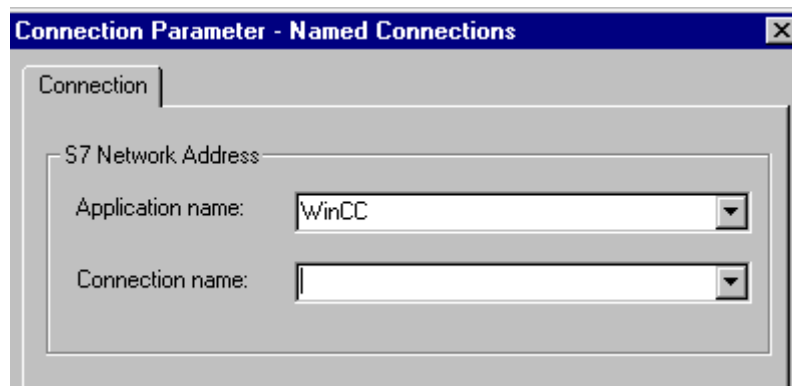
Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the channel unit "Named Connections". The "Connection Properties" dialog is opened.

2. Enter a connection name, for example "Test_NC", in the General tab.

3. Click the "Properties" button to open the "Connection Parameters - Named Connections" dialog.

```
Connection Parameter - Named Connections               [X]
┌─ Connection ─────────────────────────────────────────────┐
│                                                           │
│  ┌─ S7 Network Address ────────────────────────────────┐  │
│  │                                                      │  │
│  │   Application name:    [WinCC              ▼]        │  │
│  │                                                      │  │
│  │   Connection name:     [|                  ▼]        │  │
│  │                                                      │  │
│  └──────────────────────────────────────────────────────┘ │
│                                                           │
└───────────────────────────────────────────────────────────┘
```

4. In the Application name field, enter the application name that has been configured in STEP 7. Default value is WinCC.

5. In the Connection name field, enter the symbolic connection name that has been configured in STEP 7.

6. Close both of the dialogs by clicking the "OK" button.

Note

The application and connection name can also be entered manually, for e.g. if an XDB file does not exist for the symbolic connection name or if the project is to be transferred to another computer. It is necessary to check the correct writing of the name in STEP 7 because there is no name validation in the CS mode.

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.3.5   "PROFIBUS (I+II)" channel units

Channel Units "PROFIBUS (I + II)"

Principle of Operation

The channel unit is used to connect WinCC to the SIMATIC S7-300 and S7-400 automation systems via a PROFIBUS network.

For the S7 automation systems, a communications module is used, e.g. a CP 342-5 in a S7-300 or a CP 443-5 in a S7-400.

A communications processor, for example CP 5613, is used in WinCC.

A second communications processor can be addressed via the "PROFIBUS II" channel unit. In this manner, the maximum number of connections is increased.

Typical unit terminology

PROFIBUS

The PROFIBUS is an open, nonproprietary communications system for the cell and field levels and has been designed for a maximum of 127 participants. The PROFIBUS is based on the European Standard EN 50170, Volume 2, PROFIBUS. PROFIBUS uses token passing with an underlying master/slave access procedure.

Communications processor

A communications processor (CP) is a module that supports communication between the WinCC computer and a specific network.

See also

Configuring the tags (Page 230)

How to configure a "PROFIBUS" channel unit connection (Page 222)

How to configure a "PROFIBUS" channel unit connection

Introduction

In addition to the channel unit, WinCC must also have a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

For S7 automation systems, a communications module is used, e.g. a CP 342-5 in a S7-300 or a CP 443-5 in a S7-400.

A communications processor, for example CP 5613, is used in WinCC. A second communications processor can be addressed via the "PROFIBUS II" channel unit.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the channel unit "PROFIBUS". The "Connection Properties" dialog is opened.

2. Enter a connection name, for example "Test_PROFIBUS", in the General Tab.

3. Click the "Properties" button to open the "Connection Parameters - PROFIBUS" dialog.
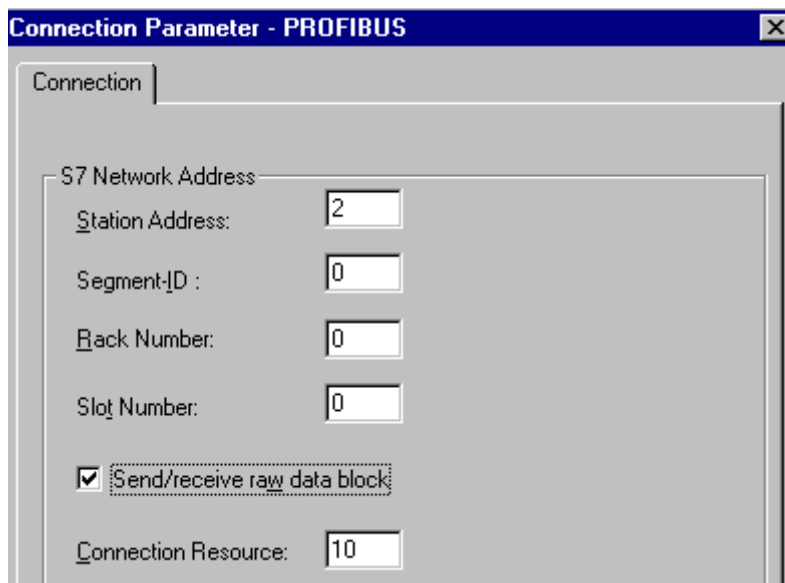
4. Enter the "Station Address" of the automation system on the bus in the appropriate field.

5. The field "Segment ID" is currently not supported. The value must remain at "0".

6. Enter the "rack number" in which the CPU that is to be addressed is located.

7. Enter the "Slot Number" of the CPU in the specified rack.

8. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/ BRCV data blocks via the connection. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP7 when the connection is configured in the PLC.

9. Close both of the dialogs by clicking the "OK" button.

Note

When starting up the PROFIBUS communication, PROFIBUS errors can occur if the communication processor was connected to the PROFIBUS while the WinCC computer was ON. Therefore, it is recommended that the computer be switched OFF before connecting it to the PROFIBUS. Otherwise, (in accordance with the PROFIBUS standard) multiple tokens may be generated on the bus, which will cause a bus error.

When using an S7-300 or S7-400 with an external communications module, you must enter the Rack/Slot number of the CPU.

If the wrong Rack or Slot Number is entered, the communications link will not be established!

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.3.6   "Slot PLC" channel unit

"Slot PLC" channel unit

Principle of Operation

Channel unit "Slot PLC" serves the communication between WinCC and up to four Slot PLC (WinAC Pro) installed in the WinCC computer. Since the Slot PLC has an integrated interface, no additional communication hardware is required for the connection between WinCC and Slot PLC.

See also

Configuring the tags (Page 230)

How to Configure a "Slot PLC" Channel Unit Connection (Page 224)

How to Configure a "Slot PLC" Channel Unit Connection

Introduction

In order to communicate with the installed SPS cards, WinCC requires a logical connection in addition to the channel unit. All the specific parameters are defined while establishing a logical connection.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- If several Slot PLC are to be configured, Slot PLC Version 3.4 is required.

Procedure

1. Select the entry "New connection" in the pop-up menu of the channel unit "Slot PLC". The "Connection Properties" dialog is opened.

2. Enter a connection name, for example "Test_SPLC", in the General tab.

3. Click on the "Properties" button to open the "Connection parameters - Slot PLC" dialog.
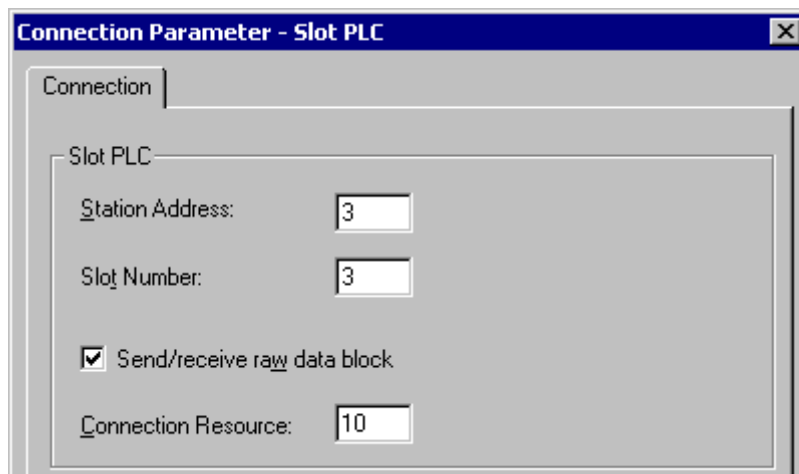
4. In the field "Station address", enter the station address of the Slot PLC on the Soft K-Bus.

5. In the field "Slot No.", enter the number of the slot in which the Slot PLC is installed.

6. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/ BRCV data blocks via the connection.

7. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP 7 when the connection is configured within the PLC.

8. Close both of the dialogs by clicking the "OK" button.

Note

Connection parameters "Station Address" and "Slot No." must be identical for several installed Slot PLCs and must start with "Slot No." "3".

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.3.7   "Soft PLC" channel unit

"Soft PLC" channel unit

Principle of Operation

Channel unit "Slot PLC" serves the communication between WinCC and a Soft PLC (WinAC Basic) installed in the WinCC computer. No other communication hardware is required for connecting WinCC to the Soft PLC.

See also

Configuring the tags (Page 230)

How to configure a connection on the "Soft PLC" channel unit (Page 226)

How to configure a connection on the "Soft PLC" channel unit

Introduction

In addition to the channel unit, WinCC must also have a logical connection to communicate with the Soft PLC. All the specific parameters are defined while establishing a logical connection.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New connection" in the pop-up menu of the channel unit "Soft PLC". The "Connection Properties" dialog is opened.

2. Enter a connection name, for example "Test_SOFTPLC", in the General tab.

3. Click on the "Properties" button to open the "Connection parameters - Soft PLC" dialog.
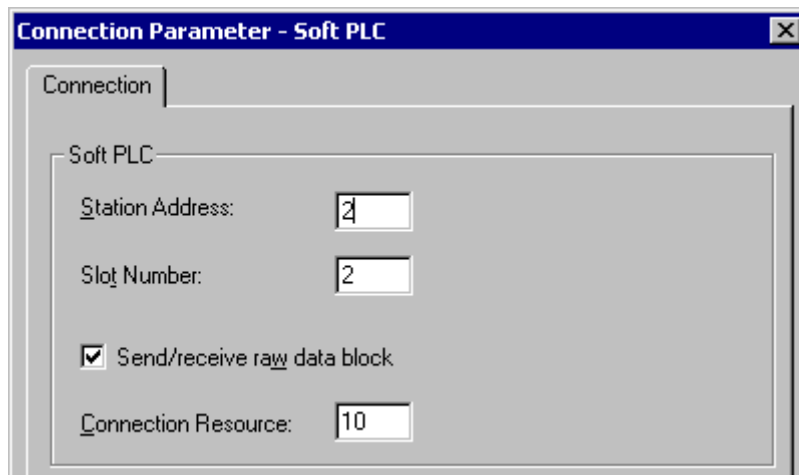
4. In the field "Station address", enter the station address of the Soft PLC on the Soft K-Bus.

5. In the field "Slot No.", enter the number of the slot. The slot number is configured in the hardware configuration of Soft PLC and is required when you want to use multiple Soft PLC in the same WinCC computer.

6. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/ BRCV data blocks via the connection.

7. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP 7 when the connection is configured within the PLC.

8. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.3.8 "TCP/IP" channel unit

"TCP/IP" channel unit

Principle of Operation

The channel unit "TCP/IP" is used to connect WinCC to the automation systems SIMATIC S7-300 and S7-400 via an Industrial Ethernet with protocol "ISO-on-TCP Transport".

The protocol corresponds to the standard TCP/IP with the extension RFC 1006. This extension is necessary, since TCP/IP uses communication where there is no blocking of data.

In the case of the automation system S7-300, communication takes place via a communications module, e.g. CP 343-1 TCP, and via CP 443-1 TCP or CP 443-1 IT in the case of S7-400.

In WinCC a communications processor such as CP 1613 is used.

Because communication takes place via the ISO-on-TCP transport protocol, it is not necessary to configure the logical connection in the local database.

Typical unit terminology

Communications processor

A communications processor (CP) is a module that supports communication between the WinCC computer and a specific network.

ISO transport protocol

ISO transport is a layer of the ISO-OSI reference model and offers services related to the transfer of data via connections. The transport layer handles data flow control, blocking and acknowledgment tasks.

The protocol defines the structure of the data traffic with regards to content on the physical line. It defines, among other things, the mode of operation, the procedure when establishing a connection, data backup or the transmission speed.

Industrial Ethernet

The Industrial Ethernet is the most efficient subnet in the industrial environment. It is suitable for the factory and cell levels and facilitates the exchange of large data volumes over large distances between a large number of participants.

The Industrial Ethernet is an open communications network and is standardized in accordance with IEEE 802.3. Its prime advantages are its speed, simple extendibility and openness as well as high availability and worldwide utilization. The configuration process requires a minimum of effort.

See also

How to configure a "TCP/IP" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

In the case of the S7-300 automation system, communication takes place via a communications module, e.g. CP 343-1 TCP, and via CP 443-1 TCP or CP 443-1 IT in the case of S7-400.

In WinCC a communications processor such as CP 1613 is used.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the pop-up menu of the "TCP/IP" channel unit. The "Connection Properties" dialog is opened.

2. Enter a connection name, for example "Test_TCP", in the General Tab.

3. Click on the "Properties" button to open the "Connection Parameters - TCP/IP" dialog.
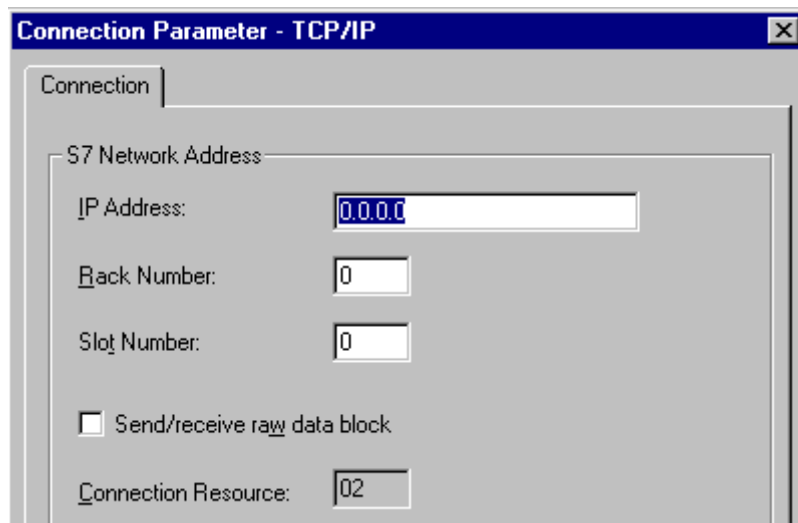
4. Enter the Internet protocol address of the automation system on the bus in the field "IP Address".

5. Enter the number of the rack in which the CPU that is to be addressed is located in the "Rack Number" field.

6. The CPU's slot number in the specified rack must be entered in the corresponding field "Slot Number".

7. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/ BRCV data blocks via the connection. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP7 when the connection is configured in the PLC.

8. Close both of the dialogs by clicking the "OK" button.

Note

When using an S7-300 or S7-400 with an external communications module, you must enter the Rack/Slot number of the CPU.

If the wrong Rack or Slot Number is entered, the communications link will not be established!

See also

Configuring the tags (Page 230)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.4 Configuring the tags

11.5.4.1 Configuring the tags

Introduction

The following sections describe how to configure the tags. It is different in the way the data area in the PLC is accessed and the data type of the WinCC tags.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

See also

How to Configure a Text Tag (Page 234)

How to Configure a Tag with Word by Word Access (Page 233)

How to Configure a Tag with Byte by Byte Access (Page 232)

How to Configure a Tag with Bit by Bit Access (Page 230)

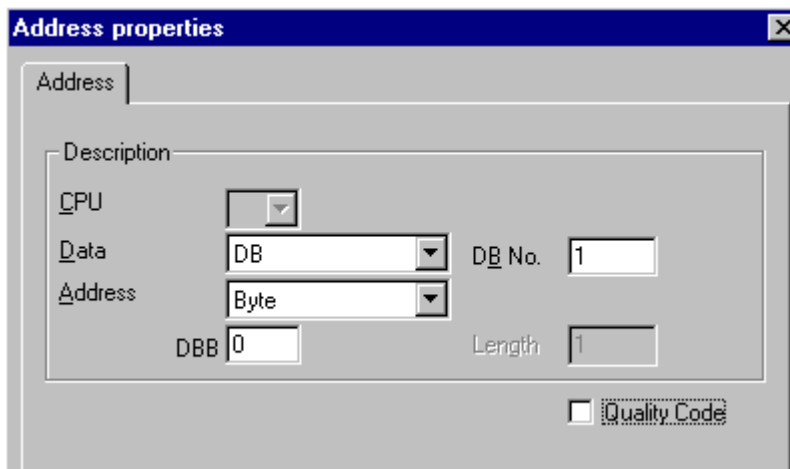### 11.5.4.2 How to Configure a Tag with Bit by Bit Access

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in PLC.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the "Test_Ind_Eth" connection. The "Tag Properties" dialog opens.

2. Enter "ETH_Var1_bit" as the name of the tag in the "Name" field. Select "Binary tag" in the "Data Type" field.

3. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. You cannot change the "Bit" entry in the Addressing field because it is defined by the Binary Variable data type of the WinCC tag.

5. Enter the byte and bit address in the two fields below it. The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and Binary Variable as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".

7. Close both of the dialogs by clicking the "OK" button.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.4.3   How to Configure a Tag with Byte by Byte Access

Introduction

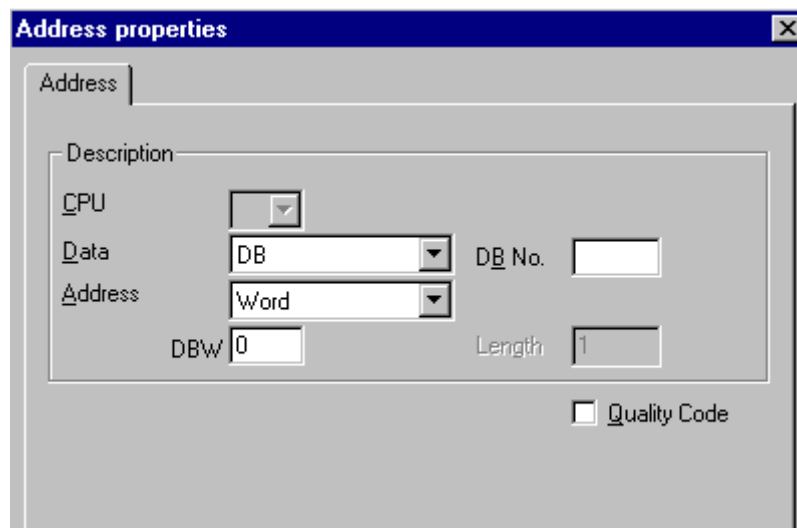This section shows you how to configure a tag for byte by byte access for the address area in PLC.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the "Test_Ind_Eth" connection. The "Tag Properties" dialog opens.

2. Enter "ETH_Var2_byte" as the name of the tag in the "Name" field. Set the data type to "unsigned 8-bit".

3. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. You cannot change the "Bit" entry in the Addressing field because it is defined by the "Unsigned 8-bit" data type of the WinCC tag.

5. Enter the byte addresss in the field below. The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and Unsigned 8-bit as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".

7. Close both of the dialogs by clicking the "OK" button.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.4.4   How to Configure a Tag with Word by Word Access

Introduction

This section shows you how to configure a tag for word by word access for the address area in PLC.

This procedure is also applicable for tags with length of 4 byte ("double word") and more.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the "Test_Ind_Eth" connection. The "Tag Properties" dialog opens.

2. Enter "ETH_Var3_word" as the name of the tag in the "Name" field. Set the data type to "unsigned 16-bit".

3. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. You cannot change the "Word" entry in the Addressing field because it is defined by the "Unsigned 16-bit" data type of the WinCC tag.

5. Enter the numeric value of the address in the Addressing field. The label on the left field depends on the entry in the Data Area field, for e.g. "DBW" for Unsigned 16-bit as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".

7. Click "OK" to close all open dialogs.

See also

11.5.4.5   How to Configure a Text Tag

Introduction

This section show you how to configure a text tag.

For a text tag in the SIMATIC S7 Protocol Suite channel, WinCC only supports S7 string type consisting of a control word and the actual user data of the string:

- To configure a text tag in WinCC, enter the address of the control word that exists in the PLC memory before the user data. The first byte of the control word contains the customized maximum length of the string, the second byte the actual length.

- To insert the data structure in the PLC memory, you must note that the length of the text tag configured in WinCC is extended by 2 bytes of the control word. If the data structures of the text tag are inserted in the memory directly one after the other, then the subsequent data will get overwritten.

- New mapping is required for switching the PCS-7 version from V4.01 to V5.0 SP1 because in the versions before V5.0 the address of the user data was also mentioned while configuring the text tags; from version V5.0 onwards the address of the control word is to be given.

- While reading, the control word is read along with the user data and the current length is evaluated in the second byte. Only the user data according to the current length included in the second control byte is trasferred at the text tags of WinCC.

- While writing, the actual length of the string is ascertained ("0" character) and the control byte with the current length is sent to the PLC along with the user data.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the "Test_Ind_Eth" connection. The "Tag Properties" dialog opens.

2. Enter "ETH_Var3_Text" as the name of the tag in the "Name" field. In the Data Type field, set "Text tag, 8-bit font" as the data type. Enter the tag length in bytes in the Length field.

3. Click the "Select" button to open the "Address properties" dialog. In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. The entry in the Addressing field can only be changed to Byte or Word because it is defined by the "Text tag, 8-bit font" data type of the WinCC tag.

5. Enter the numeric value of the address in the Addressing field. Mention the address of the control word. The label on the left field depends on the entry in the Data Area field, for e.g. "DBW" for Word as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".

7. Click "OK" to close all open dialogs.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 212)

11.5.5 System parameters

11.5.5.1 System Parameters of the Channel Unit

Introduction

If you require a configuration that deviates from the WinCC standard settings, you will therefore be able to make all the required changes using the "System Parameter" dialog of the channel unit.

The following can be modified:

* the logical device name

* the use of the cyclic read service in the AS

Logical Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communications card and are unit-specific. This field will be filled with a default unit-specific entry, e.g. "MPI" in the case of channel unit "MPI".

Using cyclic reading services in the PLC

It is possible to specify whether or not the cyclic read services of S7-PLC (also referred to as cyclic tag services) should be used. These cyclic read services group the tags that are to be read cyclically into individual request and transfer these to the PLC. The PLC will transfer the required data immediately on receipt of the request and will also transfer the data each time the cycle time elapses.

When the cyclic read services are activated, modification transfers can also be used. The data will then only be transferred when the values have changed. The function must be supported by the PLC.

Note

The system parameters on the SIMATIC S7 and Unit tabs are unit-specific and can thus be set separately for each channel unit of the channel.

See also

11.5.5.2   Cyclic read services in PLC

Introduction

In the system parameters of the "SIMATIC S7 Protocol Suite" channel, it is also possible to specify whether or not the cyclic read services of the S7-AS(also referred to as cyclic tag services) should be used. These cyclic read services group the tags that are to be read cyclically into individual request and transfer these to the PLC. The PLC will transfer the required data immediately on receipt of the request and will also transfer the data each time the cycle time elapses. When the requested data is no longer required, e.g. in the case of a screen change, WinCC will delete the cyclic read service in the PLC.

In normal cases, use should be made of the cyclic read services in the PLC. For this reason, the corresponding check box is already activated (default setting) in the system parameters of the channel unit. This setting should only be changed if you do not wish to use the cyclic services.

Modification transfers can only be used when the cyclic read services are activated. The data will then only transferred from the AS when a value has changed and only once per AS cycle. The function must be supported by the PLC.

The use of the cyclic read services and modification transfers relieves both the AS and AS-OS communication, since read requests need not be continually sent to the AS and processed there.

In the case of acyclic read services, the tags that are to be read are combined in an individual request and transferred to the PLC. The PLC only sends the required data once. The formation of the cycle for the request is carried out by WinCC.

The number of cyclic read services in a CPU

The number of cyclic read services will depend on the resources that are available in the S7-PLC. A maximum of four cyclic services are available for an S7-300 max. and a maximum of 32 for an S7-416 or 417. This number applies for all participants communicating with the PLC, i.e. if several WinCC systems are communicating with an S7-PLC, they will have to share the resources that are available. If the maximum number of resources is exceeded, access to a further cyclic read service will be refused. WinCC will then have to request this data using acyclic read requests and will also have to execute cycle formation.

Requesting external tags in scripts

The utilization of the cyclic read service has no influence on the initial update once a picture has been opened if the picture that has been selected does not contain any scripts that request external tags using the function "GetTagWord()". If scripts are executed with "GetTagWord()" when a picture is opened, the incorrect configuration of this script could result in new tag requests being sent to this channel repeatedly following a picture change. If external tags are required in a script, "Tag" should be entered as a trigger event.

11.5.5.3 How to Configure the System Parameters

Introduction

In this section, we will show you how to configure the system parameters of the Channel "SIMATIC S7 Protocol Suite".

The "System Parameters" dialog comprises two tabs:

- SIMATIC S7 tab

- Unit tab

The system parameters on the SIMATIC S7 and Unit tabs are unit-specific and can thus be set separately for each channel unit of the channel.

These tabs are identical for all channel units of the S7 channel. Consequently, the dialog for the channel unit "MPI" is used in all examples.

Any changes that are made to the parameter values will only take effect after WinCC has been restarted.

Note

When copying the project to another computer, the settings on the Unit tab will be retained, the settings on the SIMATIC S7 tab, however, will not.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the Channel "SIMATIC S7 Protocol Suite" in Tag Management. Open the dialog "System Parameters" using the pop-up menu of the desired channel unit.

2. Select the SIMATIC S7 tab. Place a tick in the check boxes "by AS" and "with modification transfer", if you want to activate cyclic reading of tags by the channel and the use of modification transfers. If available, the cyclic services in the PLC will be used here. Further information can be found under "PLC Cyclic Read Services".
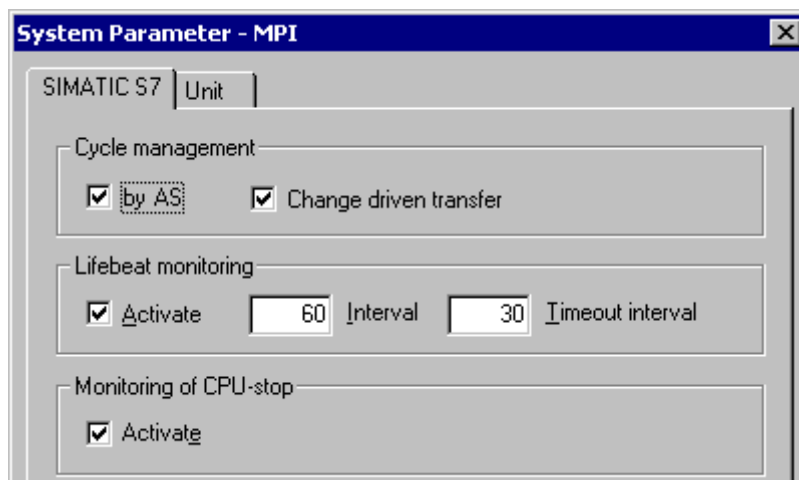
3. Activate the check box "Enable" in the "Lifebeat Monitoring" area if you wish to use this function. In the Interval field enter the time interval in seconds for transferring the lifebeat telegrams.
   In the Monitoring Time field enter the seconds value for monitoring the response to a lifebeat telegram.

4. If WinCC should indicate that communication is faulty when the S7-CPU is in the Stop status, activate the check box "Enable" in the "CPU Stop Monitoring" area.

5. Select the Unit tab. A name, which will depend on the communications processor installed, will be displayed in the field "Logical Device Name". You should only change this name if you have selected a different name when installing the communications processor. Further information can be found under "Changing Logical Device Names".
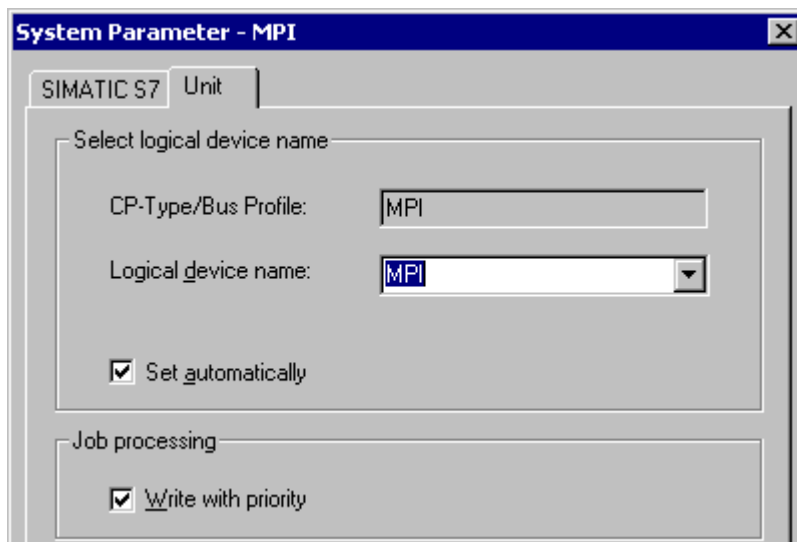
6. If only one communications processor has been installed for this communication type, activate the check box "Set Automatically", if the device name should be set automatically when Runtime is started.

7. Activate the check box "Write with Priority", if the processing of write requests should take priority over the processing of read requests.

8. Close the dialog with the "OK" button.

See also

How to Change the Logical Device Name (Page 240)

Cyclic read services in PLC (Page 237)

11.5.5.4   How to Change the Logical Device Name

Introduction

Communication with the S7 takes place via logical device names. These names are assigned during the installation of the communications processor and are unit-specific.

Certain presettings have now been established for the device names depending on the communications processor that has been installed. These are listed in the table "Default Device Names" below.

The tabs for all units of the S7 channel are identical and, for this reason, the dialog for the channel unit "MPI" is shown in the description.

Default Device Names

Channel Unit	Default Device Name
Industrial Ethernet	CP_H1_1:
Industrial Ethernet (II)	CP_H1_2:
MPI	MPI
Named Connections	VM/
PROFIBUS	CP_L2_1:
PROFIBUS (II)	CP_L2_2:
Slot PLC	SLOT_PLC
Soft PLC	SOFT_PLC
TCP/IP	CP-TCPIP

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection must be created to a channel unit e.g. "MPI".

Procedure

1. Select the desired channel unit in Tag Management.

2. Open the dialog window "System Parameters" using the pop-up menu.

3. Select the Unit tab.

4. Specify a device name in the field "Logical Device Name". You can either select an entry from the selection list or enter a new name manually.
   All possible names will be determined by the "Configure PG/PC Interface" (Control Panel) tool. If this has not been installed, only the device name that is currently set will be displayed.
   If you specify a different logical device name, a message will be displayed.
   Manual entries should only be made if the target station uses a communications card which is not installed on the configuring station.

5. Close the dialog by clicking the "OK" button.

Note

Logical devices names must be exactly the same - to the letter - as in the device settings. This being the case, the default logical device names for the "Industrial Ethernet"and "PROFIBUS" have, for example, a colon at the end of the name.

Any changes that are made to the parameter values will only take effect after WinCC has been restarted.

11.6 Special functions

11.6.1 Special functions of the "SIMATIC S7 Protocol Suite" Channel

Introduction

The SIMATIC S7 Protocol Suite contains some special functions; their functionality is described in this chapter.

See also

Software Redundancy (Page 283)

Raw data tags of the Channel "SIMATIC S7 Protocol Suite" (Page 275)

Data exchange with the S7 function block AR_SEND (Page 242)

11.6.2 Data exchange with the S7 function block AR_SEND

11.6.2.1 Data exchange with the S7 function block AR_SEND

Introduction

The S7 function block AR_SEND in the S7-400 AS is used to transfer process values to the process value archives.

Principle of Operation

To transfer PLC process values to a process value archive in WinCC, the S7-400 PLC has an integrated function component called SFB 37 "AR_SEND".

The basic function of AR_SEND component can supply data to archive tag. Data can be supplied to multiple tags if the AR_ID-Subnumber is used. If AR_SEND component is used, the process values are not sent individually to the archive; they are first collect in PLC and transferred as a package. This reduces the load on the used network.

In a PLC, you can use a CPI-dependent number of AR_SEND components (for e.g. CPU 416 max. 32 AR_SEND). A AR_ID can in turn be assigned to aach AR_SEND component. The sub-number is used to increase the amount of transferable process data because up to 4095 sub-numbers are possible for each AR_ID.
In reality, the number of archive tags per AR_SEND component is limited by the maximum length of the data area to be transferred. For more information about "The Structure and Parameters of Data Block Structures", please see the description of the "Number of Process Values" parameter.

AR_ID and AR_ID-Subnumber establish the assignment between data in the PLC and the archive tags and are defined using other parameters while configuring the data structure in

the databases in the PLC.
This assignment is configured in WinCC where as the other parameters are evaluated automatically.

SFB 37 "AR_SEND" must have first been configured in the PLC and the data block structure must have been done because configuration in WinCC is based on these values in the PLC. For information about configuring the AR_SEND function component can be found in the S7-400 PLC documentation.

Overview of AR_SEND Variants

Variants: AR_SEND for ...	Number of process-controlled archive tags for each AR_SEND	Intended use
an archive tag	a	For transferring process values for an archive tag where the process values can also be read in very small time intervals.
multiple archive tags	corresponds to the number AR_ID-Subnumbers	For transferring process values for multiple archive tags where the process values can also be read in very small time intervals.
multiple archive tags (optimized)	corresponds to the number AR_ID-Subnumbers	For cyclic data supply to maximum number of archive tags each one value each at one time

See also

11.6.2.2   Data Block - Structure and Parameters

Introduction

Before data is transferred from a PLC with the "AR_SEND" function block, it is first organized as one or more data blocks. The structure of a data block depends on various parameters, e.g. whether a time stamp or the process value's data type is appended to the AR_SEND variant used.

The parameters used in data blocks are described below. The individual parameter values are set in the data block in the AS and in the "AR_SEND" function block parameters.

The parameters are checked when the data block is evaluated in WinCC. If WinCC detects an error in the structure of the data block or if the archive tag configuration does not match the received data, an entry with the following structure will be recorded in the WinCC Diagnosis logbook:

"Date, Time, 1003080 ,4 ,user name, computer name, NRMS7PMC, PdeReceive: Unknown parameter AR_SEND from connection connectionname ...+ additional information for error description"

If the message system has been configured with the WinCC System Messages, this diagnosis entry will also trigger the OS process control message numbered 1003080. The text of the logbook entry will be found in this message's comment.

Structure of a Data Block

Each data block consists of a header and a user data area:

● The header contains information about the process values and their cycle, and possibly a time stamp.

● The user data area contains the actual process values.

One or more data blocks form the data area to be transferred.

Header Type			
Year		Month	
Day		Hours	
Minutes		Seconds	
1/10 s	1/100 s	1/1000 s	Weekday
Cycle			
Unit (Type)		Unit (Range)	
AR_ID-Subnumber			
Process data - data type			
Number of process values			
Process value 1			
Process value 2			
Process value i			

Header Type

Time stamp (no time stamp in header types 0 or 8) ⎫
Cycle information ⎬ — Header
Archive ID ⎭
Information about process values
Process values — Useful data block
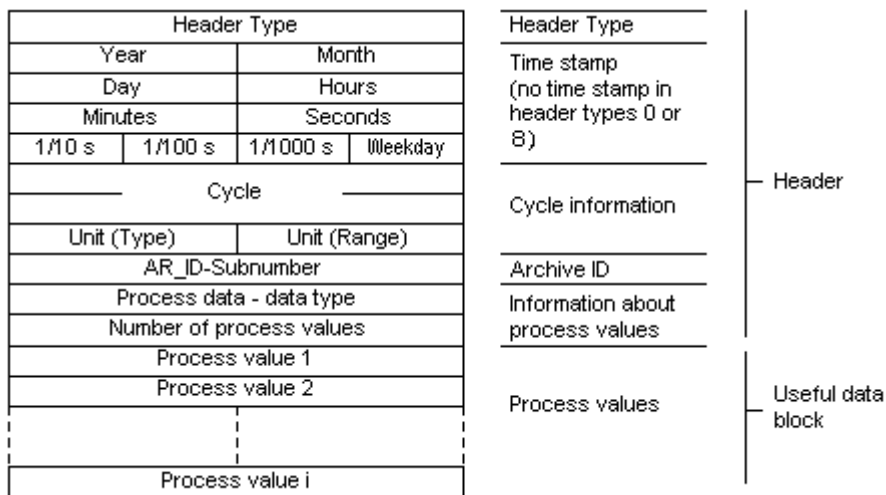
Note

In the data blocks, each line represents two bytes. Process values can be one or more bytes long, depending on their data type. Further information can be found in the description of the "Number of Process Values" parameter.

Description of parameters

Header Type

The Header Type defines the type of information that is contained in the header.

Header Type	Time stamp	AR_ID Subnumber
0	Header without Time Stamp	Header without AR_ID Subnumber
1	Header with Time Stamp	Header without AR_ID Subnumber
8	Header without Time Stamp	Header with AR_ID Subnumber
9	Header with Time Stamp	Header with AR_ID Subnumber

Note

In the case of the header types 0 and 8, the bytes for the time stamp will not be included in the header. Since these bytes are also not reserved in the data block, the header will be shortened accordingly by 8 bytes.

AR_ID Subnumber

Establishes the assignment between the AS user data and the WinCC archive tags and is configured at two positions:

- In WinCC, when configuring the process controlled archive tags
- In the PLC, when setting up the user data area to be transferred

The Subnumber is only relevant for the header types 8 or 9. Valid values for the Subnumber are in the range from 1 to 4095. The parameter is entered in WinCC as a hexadecimal value (1 - 0FFF).

Time stamp

The time stamp contains the date and the time in SIMATIC S7 BCD format. The weekday entry is not utilized by WinCC.

Note

The automation system S7 does not recognize the Summer/Wintertime switching. In the AS the local Winter time must be set as the system time. The time stamp on Summer or Winter time is corrected in WinCC by the standardization DLL. The corrected time and Summer/ Winter time ID are then available in the WinCC applications. The corrected time and ID are then added to the archive e.g. in Tag Logging.

Cycle

The cycle in which the process values are to be read. This parameter is a factor for the units of time specified under Unit (Range). Data length: Double word.

E.g.:

"Cycle" = 10 ; "Unit(area)" = 4 means: Reading cycle for process values = 10 seconds

Unit (Type)

Specifies the type of time information and modifies the parameter "Number of Process Values".

No.	Meaning
1	The process values are read at equal intervals. Start time is mentioned in the timestamp of the header and is mandatory. The time interval between the process values is defined by the time units in the "Units (Range)" and the factor "Cycle".
2	Each process value has a timestamp A time stamp given perhaps in the header will not be evaluated. The format corresponds to the time stamp in the header with a length of 8 bytes.
3	Each process value has a relative time difference in units of time with a data length of 2 words. The absolute time is the sum of the time stamp in the header (= start time) and the relative time difference in the time unit set in "Unit(Range)". A time stamp entry in the header is mandatory.
4	Each process value contains the AR_ID-Subnumber. The time stamp given in the header applies for the process value. A time stamp entry in the header is mandatory.

Units (Range)

Specifies the units of time used for Unit (Type) = 1 or 3.

No.	Meaning
1	Reserved
2	Reserved
3	Milliseconds
4	Seconds
5	Minutes
6	Hours
7	Days

Process Data - Data Type

The process values are stored directly in the S7 format.

No.	S7 data type	WinCC Data Type
0	BYTE	BYTE
1	WORD	WORD
2	INT	SWORD
3	DWORD	DWORD
4	DINT	SDWORD
5	REAL	FLOAT

Number of process values

Depending on the entry in "Unit (Type)", the transferred data area can contain a specific number of process values. The number is limited by the maximum length of the transferred data area - 16 Kbytes.

Please note the resource restrictions when using the S7 functions "AR_SEND" and "BSEND/ BRCV" to communicate with a S7-400, i.e. no more than 16 Kbytes of data can be simultaneously sent by the AS to WinCC using AR_SEND and/or BSEND/BRCV.

Note

In the case of the AR_SEND variant "Multiple Archive Tags", the following limitation applies for this parameter:

The data blocks for the various archive tags must always begin on a word boundary. Therefore, in the case of the combination "Data Type Process Value" = 0 (BYTE) and "Unit (Type)" = 1 (Process value with equally spaced time intervals) an even number of process values (=Bytes) must be entered for the parameter "Number of Process Values". This restriction only applies for this AR_SEND variant and this combination of data type and "Unit (Type)".

Example:

1x BSEND with a max. of 16 Kbytes

or 1x AR_SEND with 8 Kbytes + 1x BSEND with 8 Kbytes

or 1x AR_SEND with 10 Kbytes + 1x AR_SEND with 2 Kbytes + 1x BSEND with 4 Kbytes

Unit (Type)	The meaning of the number of process values
1	Reading process values at equal time intervals: 8000 process values of the data type WORD or INT or 4000 values of data type DWORD, DINT or REAL can be transferred.
2	Process values with timestamp: Each element of the user data area consists of a timestamp (8 bytes) and a value. Therefore, 1600 process values of the data type WORD or INT or 1333 values of data type DWORD, DINT or REAL can be transferred.
3	Process values with time difference: Each element of the user data area consists of a timestamp (4 bytes) and a value. Therefore, 2666 process values of the data type WORD or INT or 2000 values of data type DWORD, DINT or REAL can be transferred.
4	Process value contains AR_ID-Subnumber (AR-SEND with multiple variables - optimized) In Type 4 , the process value consists of one word with the AR_ID-Subnumber (Value range: 1 - 0x0FFF) and one value. Thus, the user data area consists of an array of process values preceded by AR_ID Subnumbers. Therefore, 3992 process values of the data type WORD or INT or 2660 values of data type DWORD, DINT or REAL can be transferred.

Note

The AR_ID Subnumbers given in the data blocks must all be configured in WinCC. WinCC will stop interpreting the user data, if a not-configured subnumber is found.

The data blocks for the various archive tags must always begin on a word boundary. Therefore, with the data type BYTE and "Unit (Type)" = 1 (Process value with equally spaced time intervals), an even number of process values (=Bytes) must be entered for the parameter "Number of Process Values". This restriction only applies for this AR_SEND variant and this combination of data type and "Unit (Type)".

See also

How to configure the AR _SEND variant for multiple archive tags (Page 272)

How to configure the AR _SEND variant for an archive tag (Page 269)

AR_SEND variant for multiple archive tags (optimized) (Page 267)

AR_SEND variant for multiple archive tags (Page 255)

AR_SEND variant for an archive tag (Page 250)

11.6.2.3 Overview of the properties of the AR_SEND variants

Introduction

On the basis of examples, the tables show the properties and possible parameter values for different AR_SEND variants.

The tables do not display all of the possible combinations.

The columns "Header Type" to "Process Value Data Type" are presented in the order that they appear in the header.

Note

The values for AR_ID and AR_ID Subnumber are set together with those of the other parameters while configuring the function block "AR_SEND" and the data structure in the data block in the AS.

Variants for an Archive Tag

Example / Property	E.g.-No.	Header type	Date / Time (Timestamp t in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process value structure in the e.g.
Each process value (byte) with its own time stamp	1	0	does not exist	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 Byte process value
Process value with equally spaced time stamp	2	1	Relevant	>=1	1	3 to 7	0	0 1; 2 3; 4; 5	16000 8000 4000	1 word process value
Each process value (word) with its own time stamp	3	1	not relevant	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 word process value
Each process value with time difference	4	1	Relevant	>=1	3	3 to 7	0	0 1; 2 3; 4; 5	5332 2666 2000	8 byte time stamp + 1 Byte process value

Variants for Multiple Archive Tags

Example / Property	E.g.-No.	Header type	Date / Time (Timestamp t in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process Value structure in the e.g.
Each process value (byte) with its own time stamp	5	8	does not exist	0	2	0	1 to 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 Byte process value
Process value with equally spaced time stamp	6	9	Relevant	>=1	1	3 to 7	1 to 4095	0 1; 2 3; 4; 5	16000 8000 4000	1 word process value

Example / Property	E.g.-No.	Header type	Date / Time (Timestampt in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process Value structure in the e.g.
Each process value (word) with its own time stamp	7	9	not relevant	0	2	0	1 to 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 word process value
Each process value with time difference	8	9	Relevant	0	3	3 to 7	1 to 4095	0 1; 2 3; 4; 5	5332 2666 2000	8 byte time stamp + 1 Byte process value

Variants for Multiple Archive Tags - Optimized

Example / Property	E.g.-No.	Header type	Date / Time (Timestampt in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process value structure in the e.g.
Each process value with an AR_ID Subnumber	9	1	Relevant	0	4	0	0	1; 2 3; 4; 5	3992 2660	1 Word Subnumber + 1 Word process value

See also

11.6.2.4   AR_SEND variant for an archive tag

Introduction

This variant can be used to supply an archive tag with process values. It can also be used with older versions of WinCC (prior to V5.0).

The properties of this variant:

- The Header Type must be 0 or 1, i.e. without AR_ID Subnumber and with/without time stamp.

- The AR_ID Subnumber in the header will not be evaluated.

- In WinCC, the archive tag name does not contain an AR_ID Subnumber, since only the process values for an archive tag will be transferred.

An example of the data area's structure

The data area to be transferred consists of one data block.

*1) = time stamp omitted with header type 0

See also

11.6.2.5   Example 1 for data block structure: An archive tag; each process value has a time stamp

Introduction

In this example, the process values are transferred for one archive tag only. There is no time stamp in the header and the corresponding number of bytes are also not reserved. Hence, each process value (1byte) is preceded by a time stamp (8byte).

Data type of the process values is BYTE.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 0			
2.0	Cycle =0			
4.0				
6.0	Unit (Type) = 2		Unit (Range) = 0	
8.0	AR_ID-Subnumber = 0			
10.0	Process data - data type = 0			
12.0	Number of process values = 3 (max. 3200)			
14.0	Year=2001		Month=10	
16.0	Day=05		Hours=13	
18.0	Minutes=40		Seconds=00	
20.0	1/10 s	1/100 s	1/1000 s	Weekday
22.0	Process value 1		-	
24.0	Year=2001		Month=10	
26.0	Day=05		Hours=14	
28.0	Minutes=40		Seconds=00	
30.0	1/10 s	1/100 s	1/1000 s	Weekday
32.0	Process value 2		-	
34.0	Year=2001		Month=10	
36.0	Day=05		Hours=15	
38.0	Minutes=40		Seconds=00	
40.0	1/10 s	1/100 s	1/1000 s	Weekday
42.0	Process value 3		-	

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.6   Example 2 for data block structure: One archive tag; equally spaced time stamp

Introduction

In this example, the process values are transferred for one archive tag.

The equally spaced time stamp of one second is formed using the parameters "Cycle" = 1 and "Unit (Range)" = 4 (= seconds).

Data type of the process values is WORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =1			
12.0				
14.0	Unit (Type) = 1		Unit (Range) = 4	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 1			
20.0	Number of process values = 8 (max. 8000)			
22.0	Process value 1			
24.0	Process value 2			
26.0	Process value 3			
28.0	Process value 4			
30.0	Process value 5			
32.0	Process value 6			
34.0	Process value 7			
36.0	Process value 8			

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.7 Example 3 for data block structure: An archive tag; each process value has its own time stamp

Introduction

In this example, the process values are transferred for one archive tag only. The time stamp in the header is not important. Hence, each process value (1Word) is preceded by a time stamp (8byte).

Data type of the process values is SWORD.

Data block structure in the data component

Adress in the DB	Data block sent

Address	Content
0.0	Header Type = 1
2.0	Year=0 / Month=0
4.0	Day=0 / Hours=0
6.0	Minutes=0 / Seconds=0
8.0	1/10 s / 1/100 s / 1/1000 s / Weekday
10.0	Cycle =0
12.0	
14.0	Unit (Type) = 2 / Unit (Range) = 0
16.0	AR_ID-Subnumber = 0
18.0	Process data - data type = 2
20.0	Number of process values = 2 (max. 1600)
22.0	Year=2001 / Month=10
24.0	Day=05 / Hours=13
26.0	Minutes=40 / Seconds=00
28.0	1/10 s / 1/100 s / 1/1000 s / Weekday
30.0	Process value 1
32.0	Year=2001 / Month=10
34.0	Day=05 / Hours=14
36.0	Minutes=40 / Seconds=00
38.0	1/10 s / 1/100 s / 1/1000 s / Weekday
40.0	Process value 2

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.8 Example 4 for data block structure: An archive tag; each process value with relative time stamp (time difference)

Introduction

In this example, the process values are transferred for one archive tag with time stamp.

The parameter "Unit(Type)" = 3 gives each process value a time difference (4Byte) for the time stamp in the header. The unit of time difference is set by the parameter "Unit(range)" = 4 in seconds.

Data type of the process values is DWORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 3		Unit (Range) = 4	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 3			
20.0	Number of process values = 3 (max. 2000)			
22.0	Relative time difference in seconds			
24.0				
26.0	Process value 1			
28.0				
30.0	Relative time difference in seconds			
32.0				
34.0	Process value 2			
36.0				
38.0	Relative time difference in seconds			
40.0				
42.0	Process value 3			
44.0				

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.9   AR_SEND variant for multiple archive tags

Introduction

With this variant, you can supply process values to one or more archive tags. For each archive tag, an AR_ID Subnumber will be assigned and a data block will be created in the data area to be transferred.

"x" process values can be transferred for each AR_ID Subnumber. For more information about "The Structure and Parameters of Data Block Structures", please see the description of the "Number of Process Values" parameter.

The time stamp for the value of an archive tag is taken or derived from the data area to be transferred in accordance with the given "Unit (Type)" and "Unit (Range)". It is then sent on to the WinCC process value archive.

The properties of this variant:

- The Header Type must be 8 or 9 (with/without time stamp and with AR_ID Subnumber).
- For every AR_ID Subnumber, a data block must be created in the data area to be transferred.
- The AR_ID Subnumber in each data block must be greater than zero.
- In WinCC, the archive tag name has an AR_ID Subnumber.

Note

The AR_ID Subnumbers given in the data blocks must all be configured in WinCC. WinCC will stop interpreting the user data, if a not-configured subnumber is found.

The data blocks for the various archive tags must always begin on a word boundary. Therefore, in the case of the combination "Data Type Process Value" = 0 (BYTE) and "Unit (Type)" = 1 (Process values with equally spaced time intervals) an even number of process values (=Bytes) must be entered for the "Number of Process Values" parameter. This restriction only applies for this AR_SEND variant and this combination of data type and "Unit (Type)".

An example of the data area's structure

The data area to be transferred consists of one or more data blocks corresponding to the number of archive tags to be supplied.

Data area transferred
AR_ID = 0x40

Process value archive, WinCC

Data block 1

Header type = 9	
Year	Month
Day of month	Hours
Minutes	Seconds
1/10 s | 1/100 s	1/1000 s | Weekday
Cycle	
Unit (Type) = 3	Unit (Range) = 3
AR_ID subnumber = 1	
Data type of process data	
Number of process values = i	
Process value 1	
Process value 2	
Process value i	

*1)

Archive tag name:
#Var_raw1#A#00000040#0001

AR_ID
AR_Subnumber

Process values 1 to i

Data block 2

Header type = 9	
Year	Month
Day of month	Hours
Minutes	Seconds
1/10 s | 1/100 s	1/1000 s | Weekday
Cycle	
Unit (Type) = 3	Unit (Range) = 3
AR_ID subnumber = 2	
Data type of process data	
Number of process values = j	
Process value 1	
Process value 2	
Process value j	

*1)

#Var_raw1#A#00000040#0002

Process values 1 to j

Data block m

Header type = 9	
Year	Month
Day of month	Hours
Minutes	Seconds
1/10 s | 1/100 s	1/1000 s | Weekday
Cycle	
Unit (Type) = 3	Unit (Range) = 3
AR_ID subnumber = m	
Data type of process data	
Number of process values = k	
Process value 1	
Process value 2	
Process value k	

*1)

#Var_raw1#A#00000040#000m

Process values 1 to k

*1) = omitted with header type 0 or 8

See also

Example 8 for data block structure: Multiple archive tags; process values with relative time stamp (time difference) (Page 265)

Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp (Page 263)

Example 6 for data block structure: Multiple archive tags; equally spaced time stamp (Page 261)

Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp (Page 259)

Overview of the properties of the AR_SEND variants (Page 248)

11.6.2.10 Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp

Introduction

In this example, process values are transferred for multiple archive tags.

The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in each data block.

There is no time stamp in the header and the corresponding number of bytes for it are also not reserved. Hence, each process value (1byte) is preceded by a time stamp (8byte).

Data type of the process values is BYTE.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 8			
2.0	Cycle =0			
4.0				
6.0	Unit (Type) = 2		Unit (Range) = 0	
8.0	AR_ID-Subnumber = 1			
10.0	Process data - data type = 0			
12.0	Number of process values = 3			
14.0	Year=2001		Month=10	
16.0	Day=05		Hours=13	
18.0	Minutes=40		Seconds=00	
20.0	1/10 s	1/100 s	1/1000 s	Weekday
22.0	Process value 1		-	
24.0	Year=2001		Monat=10	
26.0	Day=05		Stunden=14	
28.0	Minutes=40		Sekunden=00	
30.0	1/10 s	1/100 s	1/1000 s	Weekday
32.0	Process value 2		-	
34.0	Year=2001		Month=10	
36.0	Day=05		Hours=15	
38.0	Minutes=40		Seconds=00	
40.0	1/10 s	1/100 s	1/1000 s	Weekday
42.0	Process value 3		-	
44.0	Header Type = 8			
46.0	Cycle =0			
48.0				
50.0	Unit (Type) = 2		Unit (Range) = 0	
52.0	AR_ID-Subnumber = 2			
54.0	Process data - data type = 0			
56.0	Number of process values = 2			
58.0	Year=2001		Month=10	
60.0	Day=05		Hours=12	
62.0	Minutes=40		Seconds=00	
64.0	1/10 s	1/100 s	1/1000 s	Weekday
66.0	Process value 1		-	
68.0	Year=2001		Month=10	
70.0	Day=05		Hours=13	
72.0	Minutes=40		Seconds=00	
74.0	1/10 s	1/100 s	1/1000 s	Weekday
76.0	Process value 2		-	

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.11   Example 6 for data block structure: Multiple archive tags; equally spaced time stamp

Introduction

In this example, process values are transferred for multiple archive tags. The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in the header in each data block.

The equally spaced time stamp of one second is formed using the parameters "Cycle" = 1 and "Unit (Range)" = 4 (= seconds).

Data type of the process values is WORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 9			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =1			
12.0				
14.0	Unit (Type) = 1		Unit (Range) = 4	
16.0	AR_ID-Subnumber = 1			
18.0	Process data - data type = 1			
20.0	Number of process values = 8			
22.0	Process value 1			
24.0	Process value 2			
26.0	Process value 3			
28.0	Process value 4			
30.0	Process value 5			
32.0	Process value 6			
34.0	Process value 7			
36.0	Process value 8			
38.0	Header Type = 9			
40.0	Year=2001		Month=10	
42.0	Day=05		Hours=12	
44.0	Minutes=40		Seconds=00	
46.0	1/10 s	1/100 s	1/1000 s	Weekday
48.0	Cycle =1			
50.0				
52.0	Unit (Type) = 1		Unit (Range) = 4	
54.0	AR_ID-Subnumber = 2			
56.0	Process data - data type = 1			
58.0	Number of process values = 5			
60.0	Process value 1			
62.0	Process value 2			
64.0	Process value 3			
66.0	Process value 4			
68.0	Process value 5			

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.12 Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp

Introduction

In this example, process values are transferred for multiple archive tags. The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in each data block.

The time stamp in the header is not important. Hence, each process value (1Word) is preceded by a time stamp (8byte).

Data type of the process values is SWORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 9			
2.0	Year=0		Month=0	
4.0	Day=0		Hours=0	
6.0	Minutes=0		Seconds=0	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 2		Unit (Range) = 0	
16.0	AR_ID-Subnumber = 1			
18.0	Process data - data type = 2			
20.0	Number of process values = 3			
22.0	Year=2001		Month=10	
24.0	Day=05		Hours=13	
26.0	Minutes=40		Seconds=00	
28.0	1/10 s	1/100 s	1/1000 s	Weekday
30.0	Process value 1			
32.0	Year=2001		Month=10	
34.0	Day=05		Hours=14	
36.0	Minutes=40		Seconds=00	
38.0	1/10 s	1/100 s	1/1000 s	Weekday
40.0	Process value 2			
42.0	Year=2001		Month=10	
44.0	Day=05		Hours=15	
46.0	Minutes=40		Seconds=00	
48.0	1/10 s	1/100 s	1/1000 s	Weekday
50.0	Process value 3			
52.0	Header Type = 9			
54.0	Year=0		Month=0	
56.0	Day=0		Hours=0	
58.0	Minutes=0		Seconds=0	
60.0	1/10 s	1/100 s	1/1000 s	Weekday
62.0	Cycle =0			
64.0				
66.0	Unit (Type) = 2		Unit (Range) = 0	
68.0	AR_ID-Subnumber = 2			
70.0	Process data - data type = 2			
72.0	Number of process values = 2			
74.0	Year=2001		Month=10	
76.0	Day=05		Hours=12	
78.0	Minutes=40		Seconds=00	
80.0	1/10 s	1/100 s	1/1000 s	Weekday
82.0	Process value 1			
84.0	Year=2001		Month=10	
86.0	Day=05		Hours=13	
88.0	Minutes=40		Seconds=00	
90.0	1/10 s	1/100 s	1/1000 s	Weekday
92.0	Process value 2			

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.13 Example 8 for data block structure: Multiple archive tags; process values with relative time stamp (time difference)

Introduction

In this example, the process values are transferred for one archive tag with time stamp.

The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in each data block.

The parameter "Unit(Type)" = 3 gives each process value a time difference (4Byte) for the time stamp in the header. The unit of time difference is individually defined by the "Unit(Range)" parameter for each archive tag and hence for each data block.

Data type of the process values is DWORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 9			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 3		Unit (Range) = 5	
16.0	AR_ID-Subnumber = 1			
18.0	Process data - data type = 3			
20.0	Number of process values = 3			
22.0	Relative time difference in minutes			
24.0				
26.0	Process value 1			
28.0				
30.0	Relative time difference in minutes			
32.0				
34.0	Process value 2			
36.0				
38.0	Relative time difference in minutes			
40.0				
42.0	Process value 3			
44.0				
46.0	Header Type = 9			
48.0	Year=2001		Month=10	
50.0	Day=05		Hours=12	
52.0	Minutes=40		Seconds=00	
54.0	1/10 s	1/100 s	1/1000 s	Weekday
56.0	Cycle =0			
58.0				
60.0	Unit (Type) = 3		Unit (Range) = 6	
62.0	AR_ID-Subnumber = 2			
64.0	Process data - data type = 3			
66.0	Number of process values = 2			
68.0	Relative time difference in hours			
70.0				
72.0	Process value 1			
74.0				
76.0	Relative time difference in hours			
78.0				
80.0	Process value 2			
82.0				

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.14   AR_SEND variant for multiple archive tags (optimized)

Introduction

This variant is to be used when the maximum number of archive tags should each be supplied a process value at one time. In this case, the data area to be transferred consists of just one data block and each process value has just its AR_ID Subnumber and its associated value.

The data type is the same for the process values of all of the archive tags in this data block.

The properties of this variant:

- The Header Type must be 1 (with time stamp and without AR_ID Subnumber).

- The AR_ID Subnumbers for the associated process values in the data block must be greater than zero. The AR_ID Subnumber in the header will not be evaluated.

- The "Unit (Type)" parameter must be 4, i.e. the process value has an AR_ID Subnumber.

- The "Units (Range)" parameter must be 0, i.e. the time stamp in the Header is valid for all process values and there are no relative times.

- In WinCC, the archive tag name has an AR_ID Subnumber.

Note

If a process value has an AR_ID Subnumber for which no WinCC archive tag is found, this will result in an entry in the WinCC Diagnosis Log. The remaining process values will then continue to be processed.

An example of the data area's structure

The data area to be transferred consists of just one data block.
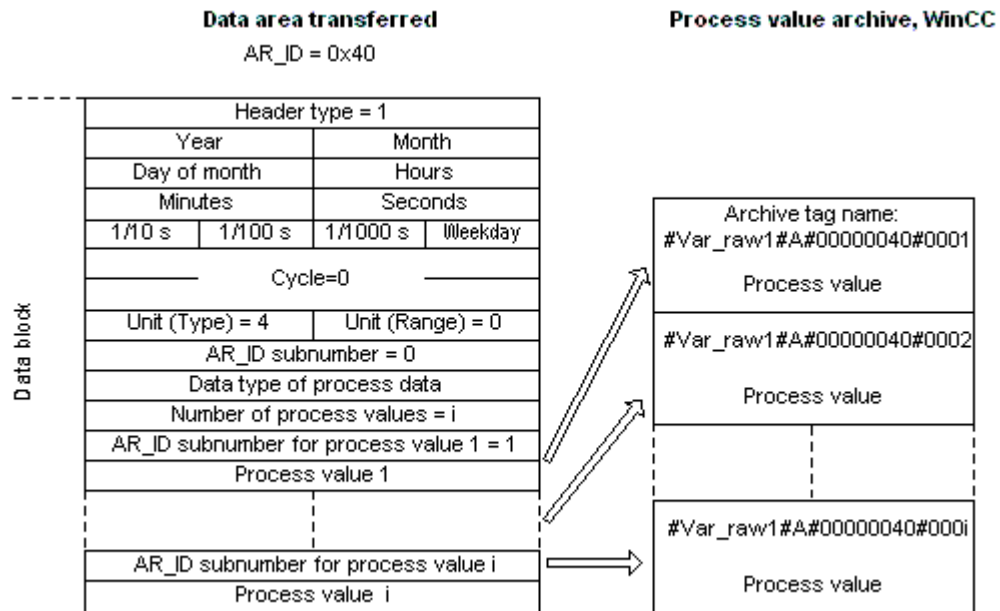
See also

Overview of the properties of the AR_SEND variants (Page 248)

Example 9 for data block structure: multiple archive tags;optimized (Page 268)

11.6.2.15 Example 9 for data block structure: multiple archive tags;optimized

Introduction

In this example, the process values are transferred for one archive tag with time stamp. The time stamp is applicable to all archive tags.

The corresponding AR_ID-Subnumber is placed before each process value.

Data type of the process values is WORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 4		Unit (Range) = 0	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 1			
20.0	Number of process values = 5 (max. 3992)			
22.0	AR_ID-Subnumber			
24.0	Process value 1			
26.0	AR_ID-Subnumber			
28.0	Process value 2			
30.0	AR_ID-Subnumber			
32.0	Process value 3			
34.0	AR_ID-Subnumber			
36.0	Process value 4			
38.0	AR_ID-Subnumber			
40.0	Process value 5			

See also

Data Block - Structure and Parameters (Page 243)

11.6.2.16   How to configure the AR _SEND variant for an archive tag

Introduction

There are a number of variants for using the AR_SEND function to exchange data. For the "One Archive Tag" variant, only the AR_ID is used. The AR_ID Subnumber is not used.

The AR_ID is used to establish the assignment of the data in the AS to the archive tags and is configured together with other parameters while configuring the data blocks and the SFB 37 "AR_SEND" function block in the AS.

In WinCC, this allocation is performed in the properties for the process controlled tag. This assignment is the only configuration that is necessary in WinCC and will be described in this section.

The other parameters in WinCC need not be configured, since they are evaluated automatically.

Note

To use this variant with just the AR_ID, the Header Type must be configure as 0 or 1. The AR_ID Subnumber must be set to zero.

Since they do not have an AR_ID Subnumber, all of the archive tags, which were configured before WinCC Version V5.0, can be used with this variant.

Since WinCC Version 5.1 Hotfix 4 it has been possible to specify an alias for the archive tag name with process-controlled tags or to use the internal names generated by the system. The name generated by the system contains the name of the assigned raw data tag instead of the raw data ID from Version V5.1 HF4. In projects migrated to a version from V5.1 HF4, the archive tag names can be used in their original format or can be converted. The names are converted by once opening and closing the properties dialog box of the process-controlled archive tags. An alias does not have to be assigned.

If in a project all external tags are remapped using the "AS-OS-Transfer" function, the archive tag name must therefore be converted once to the new structure! The new structure is then maintained.

Requirements

- The "AR_SEND" function block and the data block structure must first be configured in the AS and this configuration information must be available during the following procedure.
- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".
- A process value archive must be configured in the "Tag Logging" editor.

Procedure

1. In Channel "SIMATIC S7 Protocol Suite", select the connection which should be used for the data transfer. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter "Var_raw_arsend" as the name of the tag in the "Name" field. The name can be no longer than 24 characters. While configuring the process-controlled tags, the raw data tag name becomes part of the archive tag name and is therefore limited to the length of the archive tag name.
   In the Data Type field, set the type to Raw Data Type. Click the "Select" button to open the "Address properties" dialog.

3. Mark the "Raw Data" check box. In the "Raw Data Type" area select the type "Archive Data Link". Click "OK" to close all open dialogs.

4. Open the "Tag Logging" editor and in the pop-up menu of a process value archive select "New Process Controlled Tag". The "Properties of process controlled tag" dialog will open.

5. Click on "Select" to open a selection dialog and then select the raw data tag that should be used to supply the archive tags. In this example that is "VAR_raw_arsend". Close the dialog with the "OK" button.

6. In the "Conversion DLL" field set the entry "nrms7pmc.nll". Click on "Options". The input dialog will open.

7. Enter the AR_ID as a hexadecimal value. The value is defined through the configuration in AS.
Uncheck the checkbox "Subnumber" because there is no subnumber used in this AR_SEND variant.
Close the dialog by clicking "OK".

8. The "Internal Archive Tag Name" field shows the internal archive tag name generated by the system. It contains the name of the assigned raw data tag and the AR_ID. In the "Archive Tag Name" field, if required, you can define an alias for this archive tag. If no alias is entered, the internal archive tag name is used for management in the process value archive and for addressing the archive tag in WinCC.

9. Close the dialog with the "OK" button.

10. Save your changes and close Tag Logging.

11.6.2.17 How to configure the AR _SEND variant for multiple archive tags

Introduction

There are a number of variants for using the AR_SEND function to transfer data for multiple archive tags.

● Use the "Multiple Archive Tags" variant to supply multiple values to multiple archive tags at various times.

● Use the "Multiple Archive Tags - optimized" variant to supply one value each to the maximum number of archive tags at one time.

The AR_ID and AR_ID Subnumber are used in both of these variants.

AR_ID and AR_ID-Subnumber establish the assignment between the data in the AS and the archive tag. They are defined in the AS with other parameters when configuring the data blocks and the function modue SFB 37 "AR_SEND".

In WinCC, this allocation is performed in the properties for the process controlled tag. This assignment is the only configuration that is necessary in WinCC and will be described in this

section.
The other parameters in WinCC need not be configured, since they are evaluated automatically.

Note

To use the AR_ID Subnumber, the Header Type must be configured as 8 or 9.

Archive tags configured in WinCC Version V5.0 have no AR_ID Subnumber and can therefore only be used in the "One Tag" variant.

Since WinCC Version 5.1 Hotfix 4 it has been possible to specify an alias for the archive tag name with process-controlled tags or to use the internal names generated by the system. The name generated by the system contains the name of the assigned raw data tag instead of the raw data ID from Version V5.1 HF4. In projects migrated to a version from V5.1 HF4, the archive tag names can be used in their original format or can be converted. The names are converted by once opening and closing the properties dialog box of the process-controlled archive tags. An alias does not have to be assigned.

If in a project all external tags are remapped using the "AS-OS-Transfer" function, the archive tag name must therefore be converted once to the new structure! The new structure is then maintained.

Requirements

- The "AR_SEND" function block and the data block structure must first be configured in the AS and this configuration information must be available during the following procedure.

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

- A process value archive must be configured in the "Tag Logging" editor.

Procedure

1. In Channel "SIMATIC S7 Protocol Suite", select the connection which should be used for the data transfer. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter "Var_raw_arsend" as the name of the tag in the "Name" field. The name can be no longer than 24 characters. While configuring the process-controlled tags, the raw data tag name becomes part of the archive tag name and is therefore limited to the length of the archive tag name.
   In the Data Type field, set the type to Raw Data Type. Click the "Select" button to open the "Address properties" dialog.

3. Mark the "Raw Data" check box. In the "Raw Data Type" area select the type "Archive Data Link". Click "OK" to close all open dialogs.

4. Open the "Tag Logging" editor and in the pop-up menu of a process value archive select "New Process Controlled Tag". The "Properties of process controlled tag" dialog will open.

5. Click on "Select" to open a selection dialog and then select the raw data tag that should be used to supply the archive tags. In this example that is "VAR_raw_arsend". Close the dialog with the "OK" button.

6. In the "Conversion DLL" field set the entry "nrms7pmc.nll". Click on "Options". The input dialog will open.

7. Enter the AR_ID as a hexadecimal value. The value is defined through the configuration in AS.
   Activate the "Subnumber" checkbox. Enter the AR_ID Subnumber as a hexadecimal value. The value is also specified by the configuration in AS.
   Close the dialog by clicking "OK".

8. The "Internal Archive Tag Name" field shows the internal archive tag name generated by the system. It contains the name of the assigned raw data tag, the AR_ID and the AR_ID Subnumber. In the "Archive Tag Name" field, if required, you can define an alias for this archive tag. If no alias is entered, the internal archive tag name is used for management in the process value archive and for addressing the archive tag in WinCC.

9. Close the dialog with the "OK" button.

10. Save your changes and close Tag Logging.

11.6.3 Raw data tags of the Channel "SIMATIC S7 Protocol Suite"

11.6.3.1 Raw data tags of the Channel "SIMATIC S7 Protocol Suite"

Introduction

- A tag of the type raw data type is a data telegram on a transport level. The contents of the raw data tag are not fixed and therefore only the sender and the receiver can interpret the transmitted data. There are no format changes in WinCC for this data type. Maximum length is 65535 Byte.

- WinCC distinguishes between two types of raw data tags: Raw data tag for free application use and raw data tag for handling S7 functions.

Raw data tag for free application use

Raw data tags for free application use are used for transferring user data blocks between WinCC and PLC and handle only user data. It distinguishes between:

Raw data tag as byte array

Raw data tag for BSEND/BRCV functions

Raw data tag for handling S7 functions

These raw data tags do not have any channel-specific header and are normally used by the message system and for process data entry in WinCC.

No further description is needed here as these are tags and functions internal to the channel.

See also

Raw data tag for BSEND/BRCV functions of S7 communication (Page 279)

Raw data tag as byte array (Page 276)

11.6.3.2   Raw data tag as byte array

Introduction

Raw data tags as byte array are used for transferring user data blocks between WinCC and PLC and handle only user data.

A raw data tag as byte array is handled in the channel like a normal process tag that is addressed via the address and length of the data area (for e.g. DB 100, DW 20, length 40 Byte).

The raw data length is limited to one transferable data block and must be fully transferable uisng a PDU (Protocol Data Unit). The maximum length of the data blocks that can be transferred using the communication driver depends on the PDU length negotiated while establishing the connection minus the header and additional information. The PDU lengths normally used in SIMATIC S7 thus result in the following maximum lengths:

- S7-300: PDU length 240 Byte, max. data block length 208 Byte

- S7-400: PDU length 480 Byte, max. data block length 448 Byte

Data must be blocked if larger data blocks are to be transferred. In PLC, the S7 software forms the blocks; in WinCC through scripts.

How to Configure a Raw Data Tag as Byte Array

The raw data tags for transferring data blocks are configured as raw data of "Send/receive block" type with one address and one length detail.

The following illustration shows a configuration example for a data area with length of 40 bytes in the data component 100 from data word 20:
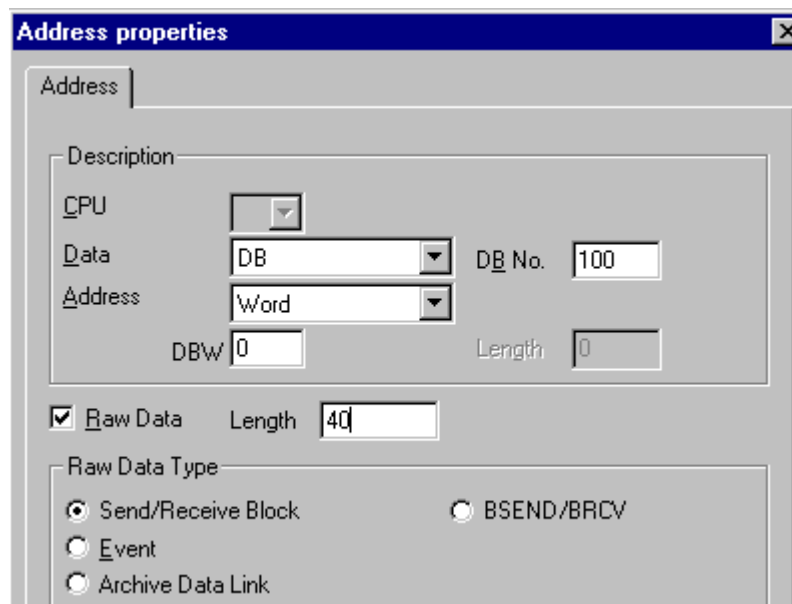
Read a Raw data tag as Byte Array

Raw data tag is read in the same way as a "normal" process tag. The corresponding data block is requested in AS and transferred to the user when the data is received.

Data is transferred always at the initiative of WinCC. Sporadic or event-controlled data reception at the initiative of AS cannot be done using this raw data tag.

Write a Raw data tag as Byte Array

Raw data tag is written in the same way as a "normal" process tag. After sending the data block and receiving a positive acknowledgement from AS, the data block is transferred to the image of the Data Manager.

See also

How to Configure a Raw Data Tag as Byte Array (Page 277)

11.6.3.3   How to Configure a Raw Data Tag as Byte Array

Introduction

This section will show you how to configure as byte array a raw data tag of the "SIMATIC S7 Protocol Suite" channel.

The configuration is identical for all channel units of the channel. The "MPI channel unit and its connection is used in the example.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection must be created to a channel unit e.g. "MPI".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.
2. Enter "Var1_raw_byte" as the name of the tag in the "Name" field. Select "Raw Data Type" in the "Data Type" field.
3. Click the "Select" button to open the "Address properties" dialog.
4. Mark the "Raw Data" check box. In the "Raw Data Type" area select the type "Send/receive block". The display of the fields in the "Address description" area and the field next to the checkbox "Raw data" depend on this setting.
5. Enter the length of the raw data block (in bytes) in the Length field.
6. In the "Data area" set the data area of the PCL where the data is located. If data area is selected as "DB" enter the number of the data component in the enabled "DB-No." field. The "Cpu" field is disabled for a connection of the "MPI" channel unit.
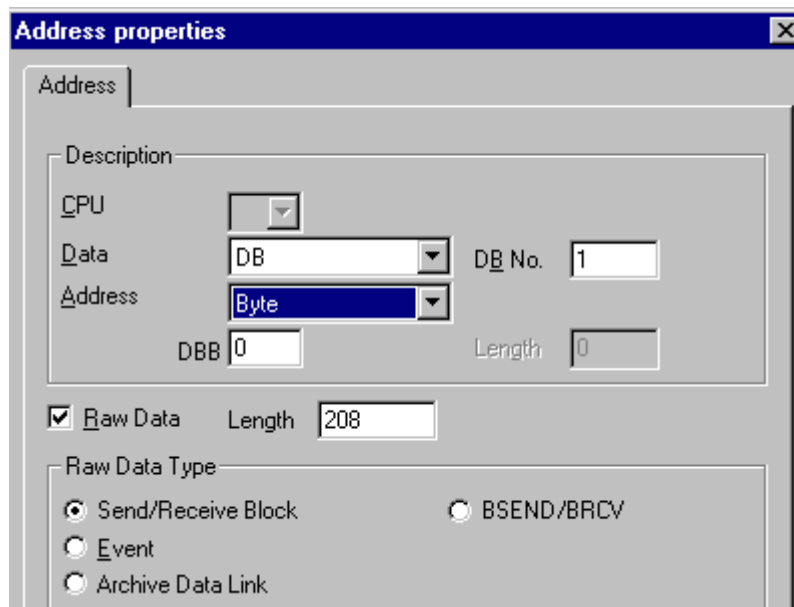
7. Set up the addressing type in the "Addressing" field. The entries "Byte", "Word" or "Doubleword" are possible for data type "Raw data type" of the WinCC tag.
8. Enter the value of the start address in the underlying field. The label on the left field depends on the entry in the Data Area and Addressing field, for e.g. "DBB" for data area "DB" "Byte" for addressing type.
9. Click "OK" to close all open dialogs.

Note

The raw data length is limited to one transferable data block and must be fully transferable uisng a PDU (Protocol Data Unit). The maximum length of the data block that can be transferred using the communication driver depends on the PDU length negotiated while establishing the connection minus the header and additional information. The PDU lengths normally used in SIMATIC S7 thus result in the following maximum lengths:

- S7-300: PDU length 240 Byte, max. data block length 208 Byte
- S7-400: PDU length 480 Byte, max. data block length 448 Byte

Wrong length will cause the read/write job to be rejected with a display.

11.6.3.4 Raw data tag for BSEND/BRCV functions of S7 communication

Introduction

Raw data tags for "BSEND/BRCV" functions are used for transferring user data blocks between WinCC and AS and handle only user data.

This raw data type can be used to access the "BSEND/BRCV" functions of S7 communication. The functions are available for the following automation systems:

- S7-400
- S7-300
  - CPU319-3 PN/DP as of V2.5
  - CPU317-2 PN/DP as of V2.6
  - CPU315-2 PN/DP as of V3.1

  Firmware version V3.x or higher is recommended for S7-300 controllers.
- WinAC RTX 2010

The initiative of data transfer always lies with the sending partner; hence "BSEND/BRCV" functions can also be used to implement event-controlled or sporadic data block transfers.

For resource reasons, it is advisable to keep the number of BSEND/BRCV raw data tags low.

Resource limitations for using S7 functions "AR_SEND" and "BSEND/BRCV" for communication with S7-400

The maximum data volume that can be sent simultaneously using AR_SEND and/or BSEND/BRCV functions from AS to WinCC is limited to 16 kByte.

Example:

- 1x BSEND with a max. of 16 Kbytes

- 1x AR_SEND with 8 kBytes + 1x BSEND with 8 kBytes

- 1x AR_SEND with 10 Kbytes + 1x AR_SEND with 2 Kbytes + 1x BSEND with 4 Kbytes

Note

If the data block of a write job is transferred to AS and has not yet been deleted or fully deleted from the receiving buffer, then the next write job will be rejected with an error message. During such an error display, write jobs with R_ID > 0x8000 0000 are written to a connection-specific queue and the system tries to repeat the write job for 6 seconds.

The responsibility for time co-ordination for transfer rests with the user and needs to be noted as shorter time intervals for write jobs.

Configuring a PBK Connection for Using "BSEND/BRCV" functions

"BSEND/BRCV" functions can only be used via a "hard-configured connection", a so-called PBK connection (programmed component communication). To configure a hard-configured connection, you must mention a connection resource (hex: 10 ... DF) in the connection parameters. This connection resource will be assigned by STEP 7 when the connection is configured within the PLC. The connection must be configured in AS as passive connection end-point.

A hard-configured connection can also be used to handle "normal" read and write jobs. If very large data areas are to be transferred via the connection, then the data blocks are transferred in multiple PDUs. For performance reasons, it would therefore be better to create a separate connection for "BSEND/BRCV" functions.

Configuring Raw data tag for BSEND/BRCV functions

Raw data tags for transferring "BSEND/BRCV" data blocks are configured as raw data of type "BSEND/BRCV" with a "R_ID". The data length is derived implicitly from the sent or received data volume.

"R_ID" Parameter

For the "BSEND/BRCV" functionality, you must specify a 32-bit long R_ID as hexadecimal number. The R_ID is assigned at the time of configuration in AS and is used for distinguishing multiple data block transfers over one connection. The send and receive calls are always notified with reference to this R_ID in the underlying communication sub-system (SIMATIC Device Drivers). A raw data tag is thus assigned to one unique R_ID.

Sending a "BSEND/BRCV" raw data tag

Sending a "BSEND/BRCV" raw data tag takes place in the same way as writing a "normal" process tag. After sending the data block and receiving a positive acknowledgement from AS, the data block is transferred to the image of the Data Manager.

Receiving a "BSEND/BRCV" raw data tag

"BSEND/BRCV" raw data is sproradically sent to the channel on the initiative of the AS. Hence it is not possible to explicitly read S7 raw data tags.

The BSEND/BRCV mechanisms do not include any synchronization functions. If no user has logged in to receive the data during the start-up phase, the data blocks sent by AS will bounce

on the receiver side. Hence, the user has to take care of the synchronization and, for e.g., release the sending direction on the AS by setting a flag with a data word.

See also

How to Configure a Raw Data Tag for ""BSEND/BRCV" functions (Page 282)

11.6.3.5 How to Configure a Raw Data Tag for ""BSEND/BRCV" functions

Introduction

This section will show you how to configure a raw data tag of the "SIMATIC S7 Protocol Suite" channel for "BSEND/BRCV" functions.

The configuration is identical for all channel units of the channel. The "MPI channel unit and its connection is used in the example.

### Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

- A connection must be created to a channel unit e.g. "MPI".

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter "Var2_raw_bsend" as the name of the tag in the "Name" field. Select "Raw Data Type" in the "Data Type" field.

3. Click the "Select" button to open the "Address properties" dialog.

4. Mark the "Raw Data" check box. In the "Raw Data Type" area select the type "BSEND/ BRCV". The display of the fields of the "Address description" area is now deactivated.

5. Enter the hexadecimal value of the ID in the "R_ID" field. The R_ID is assigned in AS at the time of configuration
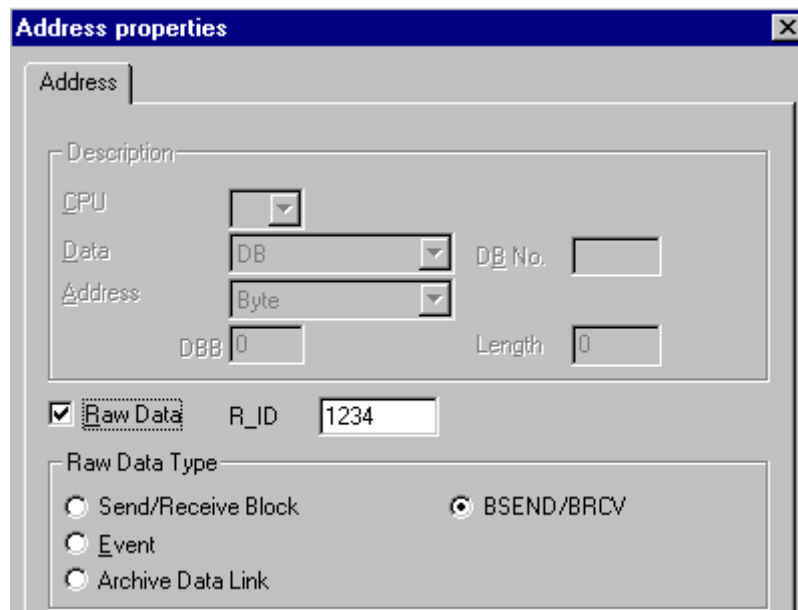
6. Close both of the dialogs by clicking the "OK" buttons.

11.6.4 Software Redundancy

11.6.4.1 Software Redundancy

Introduction

The software redundancy offers a cost-effective option for monitoring the safety-related parts of the system that do not have time criticality through a redundant connection of two automation systems S7-300 or S7-400.

Note

The software redundancy of this channel does not have the same functions as the H Layer Redundancy of SIMATIC S7-400 H.

Configurations in AS and in WinCC are required for functioning.

WinCC (Tag management)
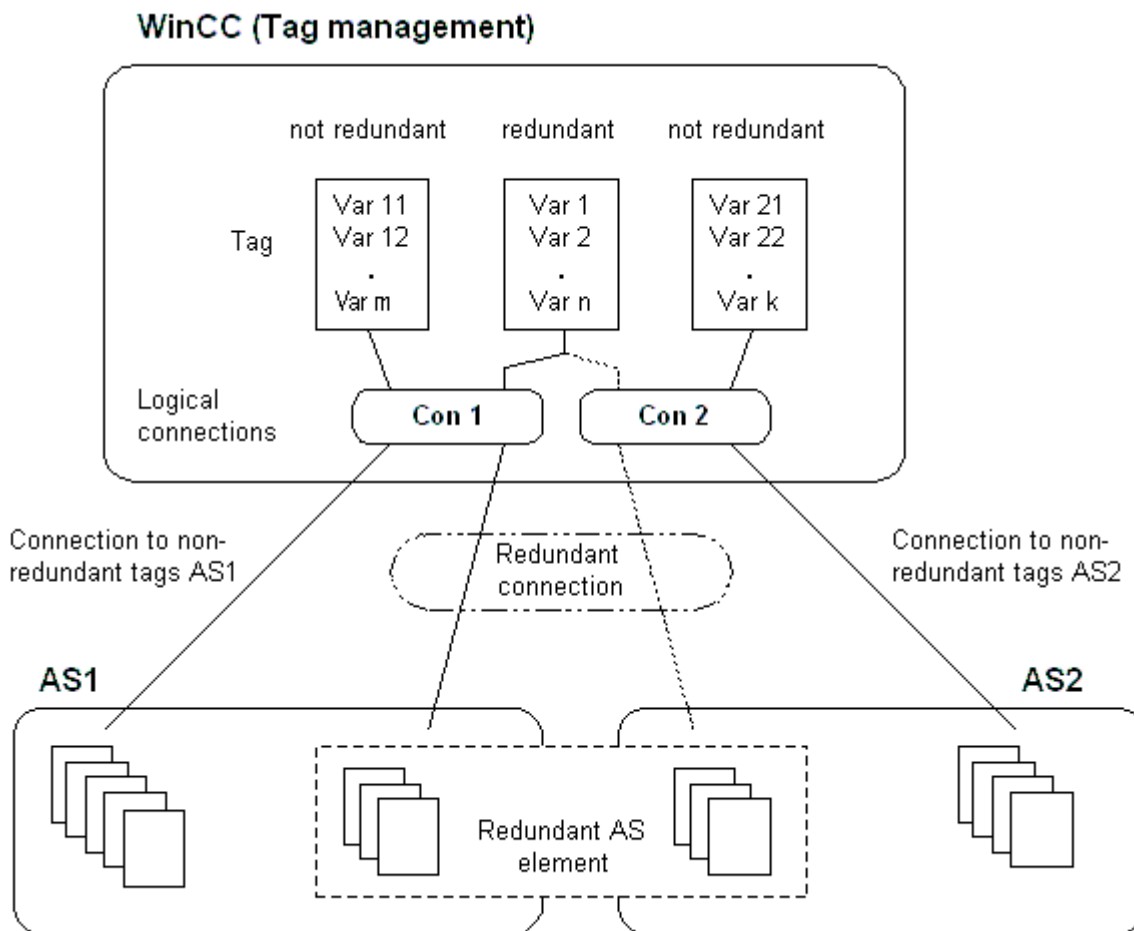
AS

If a redundant connection exists between two automation systems, then in the event of failure of one AS the other one can take over the monitoring. Monitoring can cover the entire process or just parts thereof.

Apart from the application program, the software redundancy package is installed on both the automation systems. This program package is not included in the scope of WinCC delivery. For matching data, you need a redundancy connection via MPI, PROFIBUS-DP or Industrial Ethernet between the two automation systems whereby even the existing communication links can be used.

WinCC

The redundancy connection can also be configured amongst the connection of the same channel unit. At the time of configuration, only one connection, the so-called main connection is configured. The reserve connection is only inserted via the Dynamic Wizard "Set up redundant connection".

The Wizard also inserts the connection-specific internal tags and a script. This controls the switching between connections during runtime and also the corresponding messages.

During runtime, it is possible to use the script to automatically switch between the connections when there is an error. However, it is possible to also switch manually without the script by

describing the connection-specific internal tags
"@<connectionname>@ForceConnectionAddress".

For AS2, you need a second fixed configuration connection if data is to be used from the non-redundant part of AS2 in WinCC.

The use of software redundancy does not means that the hard configured connections can only be used for the redundancy. Each single connection can also be used without redundance.

See also

11.6.4.2 Software Redundancy - Connection-specific internal tags

Introduction

Connection-specific internal tags are used to control the redundant connections. These tags are setup using the Dynamic Wizard "Redundant Connection Configuration" and are gathered in a Tag Group called "@<connectionname>" in the associated connection.

These tags can be used to determine the connection status and to control the establishment of a connection. Thus, they can also be used to implement other applications, e.g. the addressing of multiple PLCs via a single connection. However, if these tags are to be used without the Software Redundancy package, they must be created manually.

Name Format

The name of a connection-specific internal tag is composed of the name of the associated connection and an identifier.

"@<connectionname>@<identifier>"

The connection name is prefixed by a "@" to identify it as a system tag. The identifier is placed as a separator before the connection name.

Example: "@CPU_3@ConnectionState"

<connectionname> = CPU_3

<identifier> = ConnectionState

> **Note**
>
> Connection-specific internal tags are counted as external tags (eight external tags per connection).
>
> WinCC's data manager only permits access to external tags when the associated connection is ready. The connection-specific internal tags can, however, be written and read regardless of the connection status.
>
> In Runtime, the current values of some of the connection-specific internal tags may be called using the "WinCC Channel Diagnosis" tool. When the main connection is selected, the tags will be displayed in the "Counters" column. In addition, in WinCC Explorers' Tag Management, it is possible to display the tag's current value as a tooltip.

An Overview of the Tags

The following identifiers are available for the internal tags of Software Redundancy:

ConnectionState

Meaning	Connection Status
	This tag can be used to find out the current connection status.
Type	DWORD
Access	Read
Default value	0 = "faulty"
Values	0 = Connection faulty
	1 = Connection ready-to-use
	2 = Connection redundant (only in case of redundancy in H systems)

ConnectionError

Meaning	Cause of fault
	The tag contains a description of the cause for the fault. Default = 0, i.e. connection not yet established or without error. When establishing a connection, the tag is loaded with 0 (no error) again. The error code is interpreted in a channel-specific manner. The S7 channel passes the SIMATIC Device Driver's error code in this tag.
Type	DWORD
Access	Read
Default value	0 = "No error"
Values	0 = No error
	<> 0 = S7 Error codes

ConnectionErrorString

Meaning	Cause of fault as string
	The tag contains the reason for the connection error as string. The string is put out in the language currently selected. Default = '', i.e. connection not yet established or without error. In S7 channel, the following text is output in "English" regardless of the selected language.
Type	TEXT8 [128]
Access	Read
Default value	"" = "No error"
Values	"No Error"
	"Error hhhh" = Error hhhh has occurred (whereby hhhh = S7 error code hexadezimal)

ConnectionErrorCount

Meaning	Counter for connection error
	The value of this tag is incremented by 1 every time there is a connection error. When there is an overflow, the counting starts back at 0.
Type	DWORD
Access	Read
Default value	0

ConnectionEstablishMode

Meaning	Connection Establish Mode
	This tag can be used to set the automatic mode to establish a connection. If enabled, the S7 channel attempts to reestablish a failed connection at intervals of approx. 4 seconds. If the value in this tag = 0, there will be no attempt after every 4 seconds to reestablish the connection after a fault; rather it will remain disconnected.
Type	DWORD
Access	Write
Default value	1
Values	A Write to the tag brings about the following actions:
	0 = Manual connection establish mode
	Action: Deactivate automatic connection establishment
	<>0 = Automatic connection establishment mode
	Action: Activate automatic connection establishment mode

ForceConnectionState

Meaning	Preferred connection status
This tag can be used to inform the channel about the preferred connection status. Usually this tag has the value 1, i.e. the channel attempts to establish the connection (at regular intervals of approx. 4 seconds, if applicable). If the value 0 is written to this tag, the channel interrupts the connection.
Type	DWORD
Access	Write
Default value	1
Values	Any write to the tag has the following effects:
0 = preferred connection status: Connection broken
Action: if connection established, initiate disconnection
1 = Preferred Connection Status: Connection broken
Action: if connection disconnected, initiate establishment of connection

ForceConnectionAddress

Meaning	Select the connection address
This tag defines which of the connection addresses is to be used to establish the connection.
Type	DWORD
Access	Write
Default value	0
Values	If ConnectionEstablishMode is set to "Automatic", the connection will be setup automatically to the corresponding address.
A Write to the tag brings about the following actions:
0 = Connection via configured connection parameters
Action: If @ForceConnectionAddress has been earlier set to 1, then initiate disconnection.
1 = Connection via alternative connection parameters
Action: If @ForceConnectionAddress previously on 0, then initiate disconnect.

AlternateConnectionAddress

Meaning	Alternative Connection Address
	In this tag, you can enter the alternative connection address string. The string is the same as the one that will be displayed as the connection parameter in WinCC Explorer. The string is channel-specific. Upon system startup (Runtime), the configured address is entered here as default for the S7 channel. If an address has not been configured yet, the text "Illegal Address" is entered for the S7 channel.
	Example of address detail for a S7-AS with station address 3 via MPI: "MPI, 3 0,,0,0,02"
Type	TEXT8 [255]
Access	Write
Default value	"..." = "configured address
Values	Writing to this tag gives rise to the following actions:
	- If the address changes due to the write process, then the connection is disconnected fro the setting "Connection via alternative connection parameter".
	- If connection mode is set to "automatic", then the connection is automatically established with the address that has just been written.

11.6.4.3 How To Configure a Software Redundancy

Introduction

This section describes how to configure the software redundancy for connections of the Channel "SIMATIC S7 Protocol Suite" in WinCC. The PLC must also be configured to use this function, but this will not be described in this document.

Requirements

1. The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

2. A connection must be created in one of this channel's channel-units to which a redundant connection should be configured.

Procedure

1. In the computer's startup parameters, activate the "Global Script Runtime", "Alarm Logging Runtime" and "Graphics Runtime" modules.
   For further information, please see "Checking startup parameters".

2. Load WinCC's system messages into Alarm Logging. These system messages include messages about software redundancy.
   You can find additional information about the topic under "Read WinCC system messages in Alarm Logging".

3. Open a picture in Graphics Designer. In the "Dynamic Wizard" window select the "System Functions" tab. Double-click to start the Dynamic Wizard "Setup Redundant Connection".
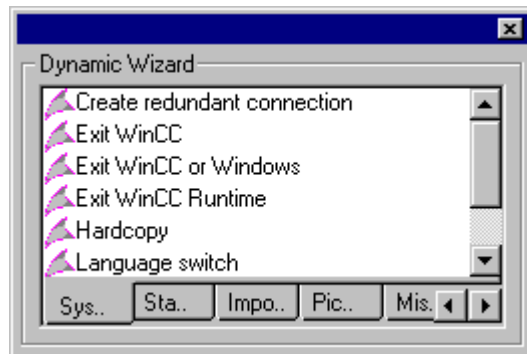
4. The procedure for using the wizard is described briefly in the "Welcome". Click "Next" to open the "Set Options" dialog.

5. Select the connection that should be used as the main connection and then click on "Next". The wizard will now create the connection-specific internal tags and will save them in a tag group "@" under the main connection.

6. Enter the address of the PLC to which the reserve connection should be established in the "Parameter" field.
Mark the "Automatic Switching" check box to have the wizard generate a script for automatically switching connections.
Click "Next"
A graphic of the redundancy - showing a MPI connection to two PLCs - will appear in the picture:
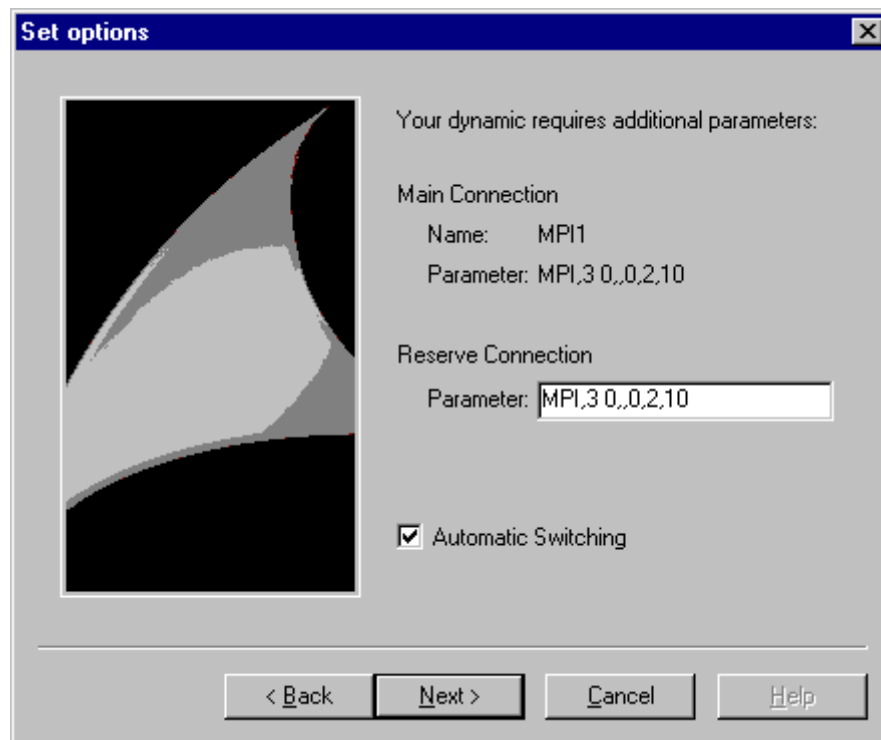
7. All of the settings made will be displayed once more in the "Finished!" dialog. If you want to make any corrections, simply click on "Back". Click "Finish".
The Wizard will now generate a script and save it under "@<connectionname>.pas" in the directory "C-Editor \ Actions \ Actions : <computername> of the Global Script Editor.

Note

In the following procedure, the "Setup redundant connection" wizard will be used. This wizard generates - when Step 6 is completed - the connection-specific internal variables. If the wizard is canceled at this point or the procedure is not completed by clicking on "Finish", these tags will remain unchanged.

See also

11.6.4.4   How to Clear a Software Redundancy in WinCC

Introduction

This section describes how to delete the software redundancy for connections of the Channel "SIMATIC S7 Protocol Suite" in WinCC. The PLC must also be configured to use this function to return to non-redundant connections, but this is not described in this document.

Requirements

- The WinCC project must be deactivated.

Procedure

A software redundancy is deleted in two steps:

- Delete the tag group "@<computername" including its tags in "Tag Management".
- Delete the script "@<connectionname>.pas" in "Global Script".

Procedure

1. In the Tag Management, select the connection that should be configured as the main connection for the software redundancy. It contains a tag group "@<connectionname" with the software redundancy's connection-specific internal tags. Delete this tag group.

2. Delete the script for the Action "@<connectionname>.pas". To do this, open the C-Editor in "Global Script". Several subdirectories will be displayed.

3. Select the "Actions \ Action : " directory. <computername>". In the data window, delete the script "@<connectionsname>.pas" for the "Actions" type.

4. Close the "Global Script" editor.

11.6.4.5   How to Check the WinCC Startup Parameters

Procedure

1. In WinCC Explorer, select "Computer".
2. Open the shortcut menu and select "Properties".
3. The "Computer List Properties" dialog is opened. Click the "Properties" button.
4. The "Computer Properties" dialog is opened.
5. Click the "Startup" tab and check the entries. If necessary, activate or deactivate Runtime modules or add other applications to the startup list.
6. Close the open dialogs.

11.6.4.6   How To Load WinCC's system messages into Alarm Logging

Introduction

In this section you will see how you can load the WinCC system messages into the project.

Procedure

1. Open Alarm Logging.

2. In "Tools" mneu, use "WinCC System Messages" item to open the "WinCC System Messages" dialog.

3. In the Create System Messages area, click the "Create only New System Messages" option button.

4. Click "Create" and close the dialog after completing the transaction.

5. Save your changes and close Alarm Logging.

11.6.4.7   Error codes during connection disturbances

Documentation of Error Codes

SIMATIC S7-1200, S7-1500 Channel   12

12.1   "SIMATIC S7-1200, S7-1500 Channel" channel

Introduction

The "SIMATIC S7-1200, S7-1500 Channel" channel is used for communication between a WinCC station and an automation system S7 1200 or S7 1500. Communication takes place using the TCP/IP protocol.

Note

Data blocks in the TIA Portal must not be configured with "Optimized block access". This could lead to errors during reading from the database.

Modification of the configuration of the automation system can only be done in the TIA Portal.

Channel unit

The "SIMATIC S7-1200, S7-1500 Channel" channel comes with the "TCP/IP" channel unit.

Recommended communications processors

The following communications processors are recommended for the communication of a WinCC station with the automation systems S7 1200 or S7 1500:

● CP 1612

● CP 1613

● CP 1512

12.2 Overview of the supported data types

Introduction

The data type and the format adaptation to the data format in the automation system (AS) are specified when the tag is configured.

The table shows the data types supported by the channel and the use of type conversions.

Supported data types

Data Types	Type conversion
Binary tag	No
Signed 8-bit value	Yes
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Floating-point number 64-bit IEEE 754	Yes
Text tag, 8-bit font	No
Raw data type	No

12.3 Configuring the channel

12.3.1 Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS). This section describes how to configure the "SIMATIC S7-1200, S7-1500 Channel" channel.

When implementing the TCP/IP protocol, you must define the IP address of the AS for the logic connection. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range of 0-255.

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol. The check-back message can take up to a minute.

Linkable automation systems

Links can be implemented for the following SIMATIC automation systems:

- S7 1200
- S7 1500

Online Configuration

The Online configuration is not supported.

12.3.2 How to configure a connection

Introduction

The following steps are required for configuring the "SIMATIC S7-1200, S7-1500 Channel" channel:

1. Configuring a connection
2. Configuring Tags

Requirements

- The communication driver for "SIMATIC S7-1200, S7-1500 Channel" channel is installed and integrated into the project.
- The SIMATIC project is configured and is available in the automation system (AS).

Procedure

1. Open the menu structure for the "SIMATIC S7-1200, S7-1500 Channel" communication driver in the "Tag Management" editor of the WinCC Explorer.

2. Select the entry "New Connection" in the pop-up menu of the "TCP/IP" channel unit. The "Connection Properties" dialog is opened.

3. Enter the name of the connection.

4. Click the "Properties" button to open the "New connection" dialog.
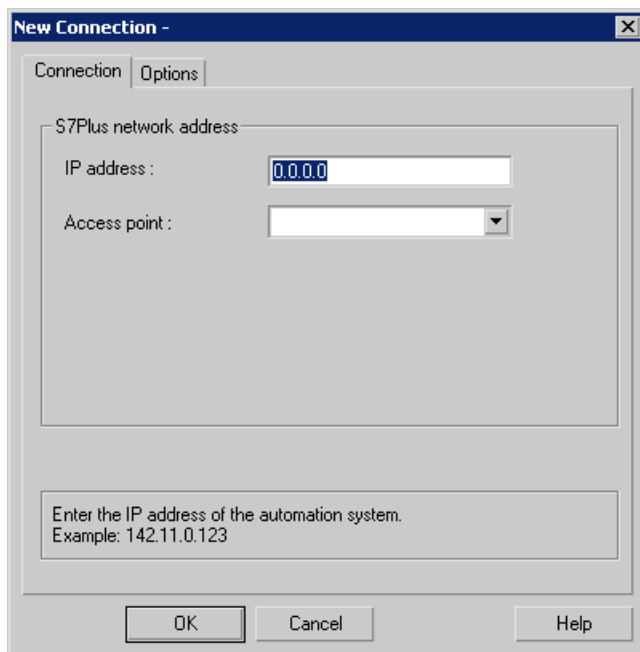
5. Enter the IP address of the AS.

6. Select or enter an "access point".

7. If you want to use password protection, click "Options".

Note

Access protection is only possible on the S7 1500 automation system.

---
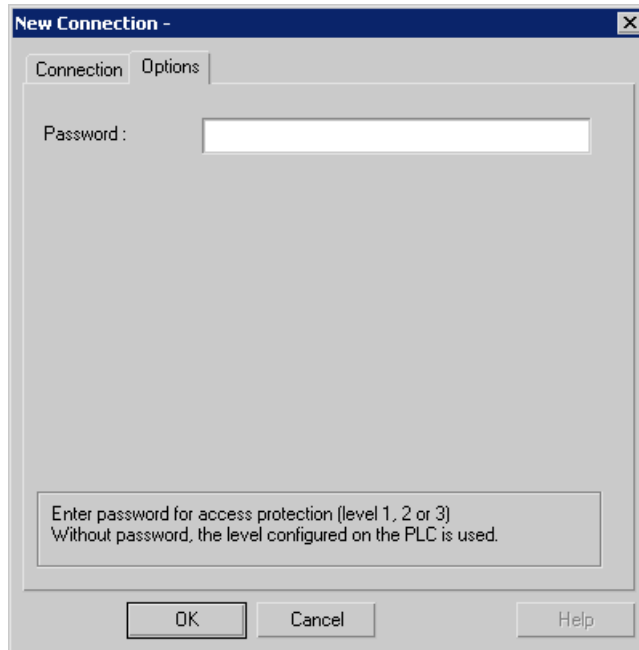
8. Enter the password for access protection.

9. Close both of the dialogs by clicking the "OK" button.

12.3.3 Configuring the tags

12.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the automation system (AS) via "SIMATIC S7-1200, S7-1500 Channel" channel, tags of different data types can be created in WinCC.

The following sections describe how to configure the tags. The addressing of the data range in the AS and the data type of the WinCC tags are different.

12.3.3.2 How to Configure a Tag with Bit by Bit Access

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

Requirements

- The "SIMATIC S7-1200, S7-1500 Channel" channel must be integrated into the project.

- A connection must be created in the "TCP/IP" channel unit.

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter the name of the tag. Select "Binary tag" in the "Data Type" field.

3. Click the "Select" button to open the "Address properties" dialog. Define the data area in which the AS data are stored. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. You cannot change the "Bit" entry in the Addressing field because it is defined by the Binary Variable data type of the WinCC tag.

5. Enter the byte and bit address in the two fields below it. The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and Binary Variable as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the AS. The check-box is enables only if the data area is selected as "DB".

7. Close both of the dialogs by clicking the "OK" button.

12.3.3.3 How to Configure a Tag with Byte by Byte Access

Introduction

This section shows you how to configure a tag for byte by byte access for the address area in the automation system (AS).

Requirements

- The "SIMATIC S7-1200, S7-1500 Channel" channel must be integrated into the project.

- A connection must be created in the "TCP/IP" channel unit.

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter name of the tag. Set the data type to "unsigned 8-bit".

3. Click the "Select" button to open the "Address properties" dialog. Define the data area in which the AS data are stored. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. You cannot change the "Bit" entry in the Addressing field because it is defined by the "Unsigned 8-bit" data type of the WinCC tag.

5. Enter the byte addresss in the field below. The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and Unsigned 8-bit as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the AS. The check-box is enables only if the data area is selected as "DB".

7. Close both of the dialogs by clicking the "OK" button.

12.3.3.4 How to Configure a Tag with Word by Word Access

Introduction

This section shows you how to configure a tag for word by word access for the address area in the automation system (AS).

This procedure is also applicable for tags with length of 4 byte ("double word") and more.

Requirements

- The "SIMATIC S7-1200, S7-1500 Channel" channel must be integrated into the project.
- A connection must be created in the "TCP/IP" channel unit.

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter the name of the tag. Set the data type to "unsigned 16-bit".

3. Click the "Select" button to open the "Address properties" dialog. Define the data area in which the AS data are stored. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. You cannot change the "Word" entry in the Addressing field because it is defined by the "Unsigned 16-bit" data type of the WinCC tag.

5. Enter the numeric value of the address in the Addressing field. The label on the left field depends on the entry in the Data Area field, for e.g. "DBW" for Unsigned 16-bit as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the AS. The check-box is enables only if the data area is selected as "DB".

7. Close both of the dialogs by clicking the "OK" button.

12.3.3.5 How to Configure a Text Tag

Introduction

This section show you how to configure a text tag.

For a text tag in the "SIMATIC S7-1200, S7-1500 Channel" channel, WinCC only supports S7 string type consisting of a control word and the actual user data of the string:

- To configure a text tag in WinCC, enter the address of the control word that exists in the automation system memory before the user data. The first byte of the control word contains the customized maximum length of the string, the second byte the actual length.

- To insert the data structure in the PLC memory, you must note that the length of the text tag configured in WinCC is extended by 2 bytes of the control word. If the data structures of the text tag are inserted in the memory directly one after the other, then the subsequent data will get overwritten.

- While reading, the control word is read along with the user data and the current length is evaluated in the second byte. Only the user data according to the current length included in the second control byte is trasferred at the text tags of WinCC.

- While writing, the actual length of the string is ascertained ("0" character) and the control byte with the current length is sent to the PLC along with the user data.

Requirements

- The "SIMATIC S7-1200, S7-1500 Channel" channel must be integrated into the project.

- A connection must be created in the "TCP/IP" channel unit.

Procedure

1. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.

2. Enter the name of the tag. In the Data Type field, set "Text tag, 8-bit font" as the data type. Enter the tag length in bytes in the Length field.

3. Click the "Select" button to open the "Address properties" dialog. Define the data area in which the AS data are stored. If you select "DB" as data area, enter the data component in the enabled "DB-No." field.

4. The entry in the Addressing field can only be changed to Byte or Word because it is defined by the "Text tag, 8-bit font" data type of the WinCC tag.

5. Enter the numeric value of the address in the Addressing field. Mention the address of the control word. The label on the left field depends on the entry in the Data Area field, for e.g. "DBW" for Word as type.

6. Check the Quality Code check-box if the tag is with Quality Code that is to be used in WinCC. For this, the code must also exist in the AS. The check-box is enables only if the data area is selected as "DB".

7. Close both of the dialogs by clicking the "OK" button.

SIMATIC TI Ethernet Layer 4

<div style="text-align: right; font-size: 3em;">13</div>

13.1   WinCC channel "SIMATIC TI Ethernet Layer 4"

Introduction

The communication driver "SIMATIC TI Ethernet Layer 4" handles the link between a WinCC station and a SIMATIC TI505 automation system via Industrial Ethernet. The communication is handled with the ISO transport protocol.

This chapter describes

- how to configure the data transfer with the "SIMATIC TI Ethernet Layer 4" channel.
- how to configure a connection and a tag.

Channel units

IT has two channel units to run a maximum of two CP 1613. The functionality of the channel unit is identical. They differ only in the logical device names of the two CP 1613.

The logical device name can be changed via the system parameters of the channel unit. Here, it is also possible to set the parameters for the ISO transport protocol.

The following application capabilities exist:

- Channel unit 505 Ethernet (CP 1413-1) for the communication modules for SIMATIC Industrial Ethernet (e.g. CP 1613).
- Channel unit 505 Ethernet (CP 1413-2) for the communication modules for SIMATIC Industrial Ethernet (e.g. CP 1613).

13.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag

- Unsigned 8 bit value (is only supported by VMS addressing)

- Signed 8 bit value (is only supported by VMS addressing)

- Unsigned 16-bit value

- Signed 16-bit value

- Unsigned 32-bit value

- Signed 32-bit value

- Floating-point number 32-bit IEEE 754

- Raw data type

13.3   Configuring the Channel

13.3.1   Configuring the channel "SIMATIC TI Ethernet Layer 4"

Introduction

The following steps are required for configuring the channel "SIMATIC TI Ethernet Layer 4".

13.3.2   How to configure the connection

Introduction

The process connection via Industrial Ethernet is possible with the SIMATIC TI505 automation system. Communication module CP 1434 TF is used in the automation system. The communication is handled with the ISO transport protocol.

The communication module CP 1613 is used in the WinCC system. Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

For a logical connection, WinCC establishes one connection in the transport layer for reading (READ) and one for writing (WRITE). Only if both connections are established is the logical connection also indicated as being established.

Parameters for the READ function

When configuring the connection, parameters are defined for the READ function in WinCC. These are independent of the request used in the SIMATIC TI.

WinCC side	SIMATIC TI side
FETCH Active (Request "READ Active")	READ Passive (Request "READ Passive")
FETCH Passive (Request "WRITE Passive")	WRITE Active (Request "WRITE Active")

Parameters "Own TSAP-ID" and "External TSAP-ID" for the WRITE function

WinCC side	SIMATIC TI side
Request "WRITE Active"	Request "WRITE Passive"

Procedure

1. Select the connection and open the "Connection properties" dialog window with menu point "Properties" in the context menu.

2. Click on the "Properties" button. The "Connection properties" dialog is opened.
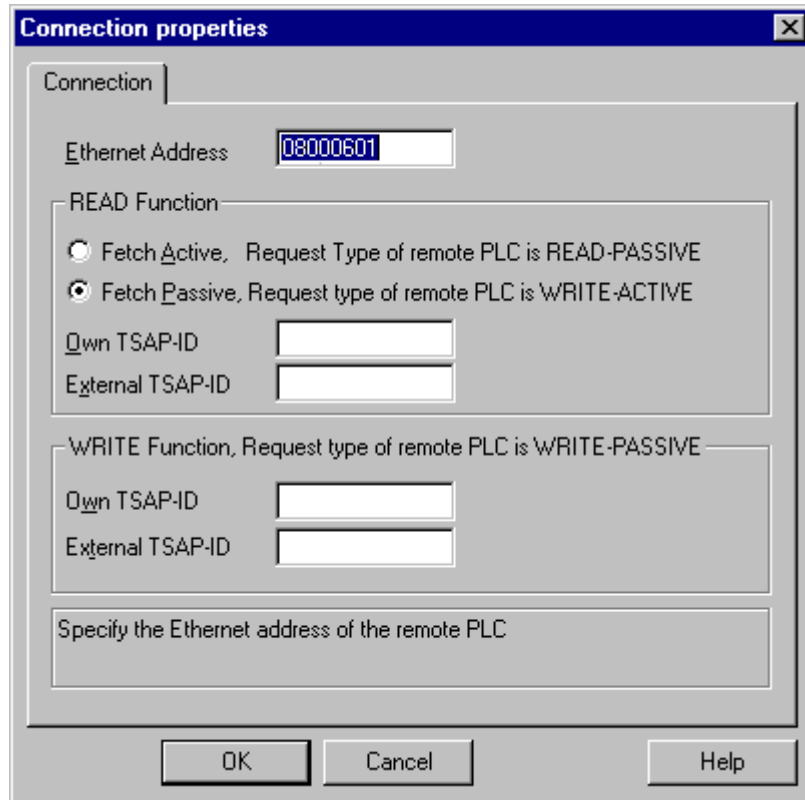
3. Enter the station address of the SIMATIC TI on the industrial Ethernet bus in the field "Ethernet Address".

4. Now, define the parameters for the READ function in the WinCC system. These are independent of the request used in the SIMATIC TI.

5. Then, enter the value in the allocated field "Own TSAP-ID" that was configured in the "Remote parameter" area as "TSAP" while configuring the CP 1434 TF.

6. Now, enter the value in the allocated field "External TSAP-ID" that was configured in the "Local parameter" as "TSAP" while configuring the CP1434 TF.

7. Define the parameters "Own TSAP-ID" and "External TSAP-ID" for the WRITE function accordingly.

13.3.3 Configuring the tags

13.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC TI Ethernet Layer 4", tags of different data types can be created in WinCC. The following describes how to configure a tag of these data types.

13.3.3.2 How to configure the address of a tag

Introduction

The tag address is entered according to the address structure of the SIMATIC TI505.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select the desired data type for the tags (e.g. Signed 16 bit value) from the "General" tab.

3. Click the "Select" button. If it is a "Binary" or "8-Bit" tag, the "Bit/Byte tag" dialog will now open. Tick the corresponding check box "Access a Bit/Byte", if data should be written to the AS memory.

4. Click the "Select" button. The "Address properties" dialog is opened.

5. Select the location of the tag in the automation system in the "Address type" field. Depending the selected address type, more definitions have to be made (e.g. "V-tag memory" for address type in the "Data element" field).

6. In the field "Read-Only Tag", you can specify that the tag cannot be written by WinCC.

Note

Structure tags are not supported.

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Depending on the type of tag, you can access the memory in the AS bit-wise or byte-wise.

13.3.3.3 How to configure a tag with bit-wise access

Introduction

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.
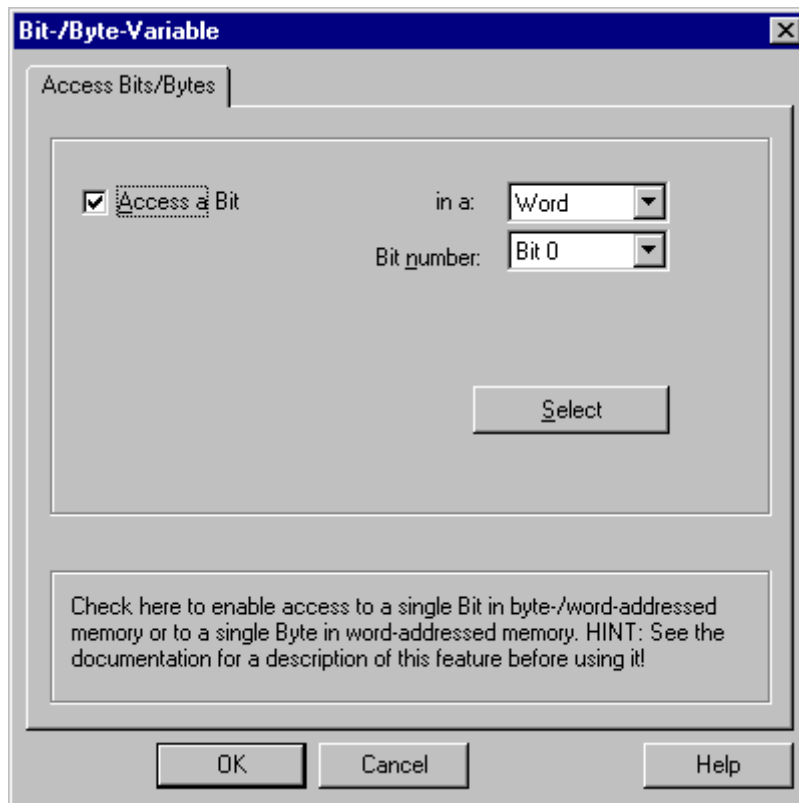

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "binary tag" on the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a bit" checkbox and define the addressing for the bit.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the "Address type" selection field.

7. From the list below, select the number of the element to be changed.

Note

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

13.3.3.4 How to Configure a Tag with Byte by Byte Access

Introduction

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single
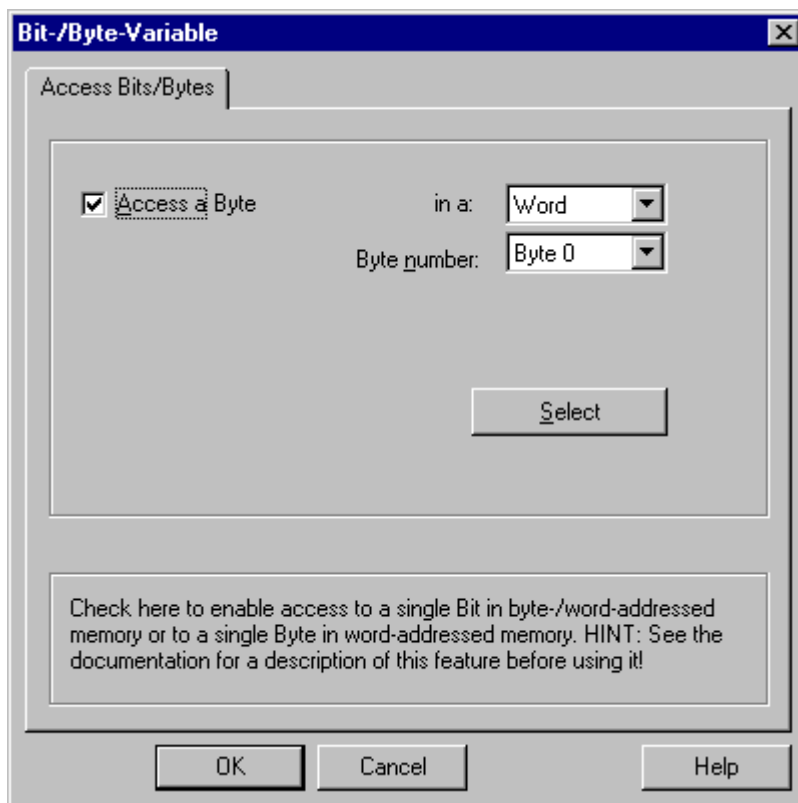
write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "Unsigned 8 bit value" or "Signed 8 bit value" in the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a byte" checkbox and define the addressing for the byte.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the "Address type" selection field.

7. From the list below, select the number of the element to be changed.

> **Note**
>
> A description of address types may be found in the SIMATIC TI505 Technical Documentation.

13.3.4 System parameters

13.3.4.1 System parameters of the channel unit

Introduction

If you require a configuration that deviates from the standard WinCC settings, you can make all the required changes using the "System parameters" dialog of the channel unit.

The following individual points can be changed:

- the device name
- the transport parameter

Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communication module and are unit-specific. The device name represents the logical device name. The logical device name is initially defined as "/CP_H1_1:/SCP" as default.

Transport Parameter

Specific settings for the channel unit are made in the transport parameters, e.g. PDU size, setup attempts, etc.

> **Note**
>
> The system parameters apply for all CPs in the AS.

13.3.4.2 How to Change the Device Name

Introduction

The process connection via Industrial Ethernet is possible with the SIMATIC TI505 automation system. Communication module CP 1434 TF is used in the automation system. The communication is handled with the ISO transport protocol.

The communication module CP 1613 is used in the WinCC system. Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

Requirements

- The channel "SIMATIC TI Ethernet Layer 4" must be integrated in the project.

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.

2. Select the "Device Name" Tab.

3. Now, you can select the logical device name shown in bold print with the mouse and change it with a mouse click in the name field.
   The logical device name is defined as "/CP_H1_1:/SCP" as default during the hardware driver installation. Only if you have defined another name there, which is not recommended, will you have to change the device name here as well.

13.3.4.3 How to change the transport parameter

Introduction

The process connection via Industrial Ethernet is possible with the SIMATIC TI505 automation system. Communication module CP 1434 TF is used in the automation system. The communication is handled with the ISO transport protocol.

The communication module CP 1613 is used in the WinCC system. Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

Requirements

- The channel "SIMATIC TI Ethernet Layer 4" must be integrated in the project.

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.

2. Select the "Transport parameter" tab.

3. Set the value for "PDU size" to the value that was configured on the CP 1434 TF.

4. Define how often a connection establishment should be attempted in the "Setup attempt" filed.

5. Select "Infinite" in the "Duration of Send Repetitions" area.

6. Enter value 30 in the "Acknowledgement time" field so that you are informed of the tag status after 30 seconds at the most, if the communication partner has not responded within this time (e.g. AS in "Stop" status).

SIMATIC TI Serial 14

14.1 WinCC channel "SIMATIC TI Serial"

Introduction

The communication driver "SIMATIC TI Serial" is used for establishing a serial link between WinCC station and an SIMATIC TI505 automation device.

This chapter describes

- how to configure the data transfer with the "SIMATIC TI Serial" channel.
- how to configure a connection and a tag.

Channel units

The communication driver has one channel unit for controlling a COM port for the serial connection.

The following capability is available:

- Channel unit "505 Serial Unit #1" for serial communication, either via the TBP protocol or the NITP protocol.

Note

It is possible to run more than one logical connections (with different COM ports) through one channel unit.

14.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag

- Unsigned 8 bit value (is only supported by VMS addressing)

- Signed 8 bit value (is only supported by VMS addressing)

- Unsigned 16-bit value

- Signed 16-bit value

- Unsigned 32-bit value

- Signed 32-bit value

- Floating-point number 32-bit IEEE 754

- Raw data type

14.3   Configuring the Channel

14.3.1   Configuring the "SIMATIC TI Serial" channel

Introduction

The following steps are required for configuring the channel "SIMATIC TI Serial".

14.3.2   How to configure the connection

Introduction

The process connection using a serial connection is possible with the SIMATIC TI505 automation system. The serial interface on the CPU module is used in the automation system.

No additional communication module is required in WinCC. Communication can take place by means of the default COM ports available on the system.

Procedure

1. Select the connection and open the "Connection properties" dialog window with menu point "Properties" in the context menu.

2. Click on the "Properties" button. The "Connection properties" dialog is opened.
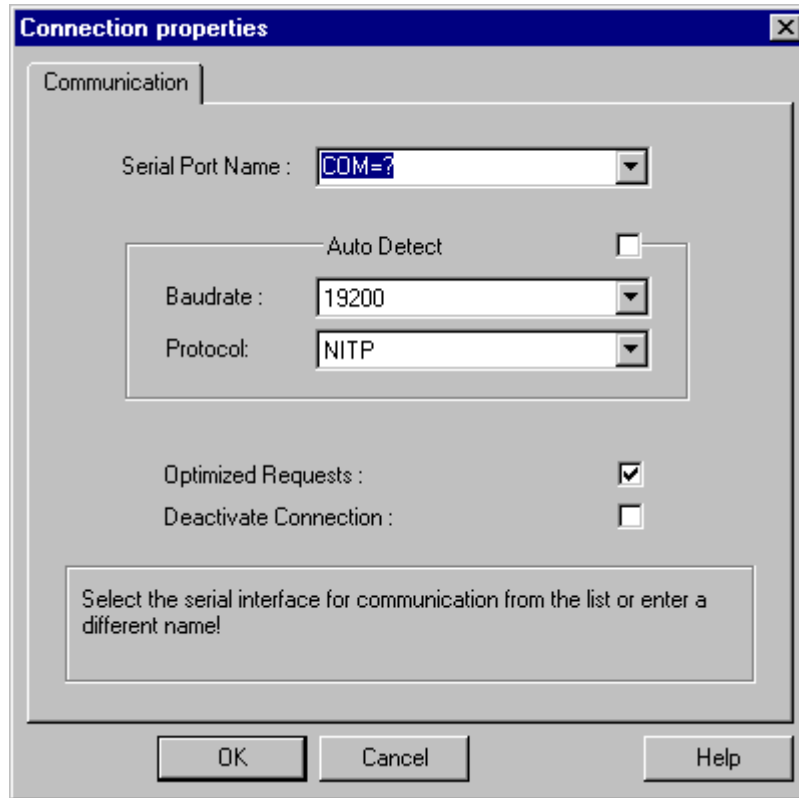
3. Select the communications interface (e.g. COM1, COM2 or a configured port) for the serial link in the "Serial port" field.

4. Select the field "Detect automatically" when the data transfer speed and the protocol used by the PLC are required to be detected automatically by the channel unit.

5. Set the data transfer rate and the protocol being used in the fields "Baudrate" and "Protocol".

6. By selecting the field "Optimized Requests," you can optimize data transfer to transfer several tags with one request.

7. If you select the "Deactivate connection" field, the logical connection is deactivated. This is often a good idea during commissioning to temporarily deactivate the connection.

14.3.3   Configuring the tags

14.3.3.1   Configuring the tags

Introduction

For a connection between WinCC and the PLC via channel "SIMATIC TI Serial", tags of different data types can be created within WinCC. This is described in the following section.

14.3.3.2   How to configure the address of a tag

Introduction

The tag address is entered according to the address structure of the SIMATIC TI505.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select the desired data type (e.g. Signed 16 bit value) from the "General" tab.

3. Click the "Select" button. If it is a "Binary" or "8-Bit" tag, the "Bit/Byte tag" dialog will now open. Tick the corresponding check box "Access a Bit/Byte", if data should be written to the AS memory.

4. Click the "Select" button. The "Address properties" dialog is opened.
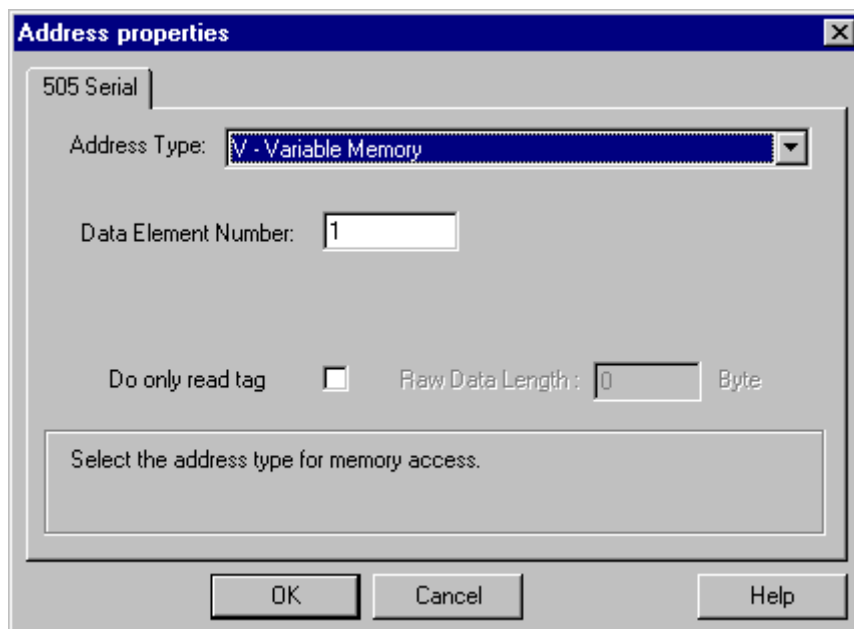
5. Select the location of the tag in the automation system in the "Address type" field. Depending the selected address type, more definitions have to be made (e.g. "V-tag memory" for address type in the "Data Element Number" field).

6. In the field "Read-Only Tag", you can specify that the tag cannot be written by WinCC.

Note

Structure tags are supported in address areas V, K, X, Y and C.

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

Write access to memory areas in the AS can only performed bit-wise or word-wise in channel "TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Depending on the type of tag, you can access the memory in the AS bit-wise or byte-wise.

14.3.3.3 How to configure a tag with bit-wise access

Introduction

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "SIMATIC TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "binary tag" on the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a bit" checkbox and define the addressing for the bit.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the selection field.

7. Select the number of bit to be changed in the selection field.

Note

With the S5, flags, inputs and outputs can be addressed by byte, data blocks (DB, DX) are addressed by word.

14.3.3.4   How to Configure a Tag with Byte by Byte Access

Introduction

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "SIMATIC TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single

write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag and open dialog window "Properties" with the context menu.

2. Select data type "Unsigned 8 bit value" or "Signed 8 bit value" in the "General" tab.

3. Click the "Select" button. The "Bit-/Byte-tag" dialog is opened.

4. Check the "Access to a byte" checkbox and define the addressing for the byte.

5. Click the "Select" button. The "Address properties" dialog is opened.

6. Select the addressing type of the PLC memory in the selection field

7. Select the number of byte to be changed in the selection field.

SIMOTION

<div style="text-align: right; font-size: 3em;">15</div>

15.1 WinCC channel "SIMOTION"

Introduction

The "SIMOTION" channel connects a WinCC station with a SIMOTION automation system. The connection is established via Industrial Ethernet using the TCP/IP protocol.

SIMOTION is a system platform for automation and drive solutions with an emphasis on motion control applications and technology tasks.

The SIMOTION modular system consists of the SIMOTION SCOUT Engineering System and a common runtime system for various hardware platforms.

Export the project from SIMOTION SCOUT to access the data of a SIMOTION SCOUT project.. Then create a WinCC project from the exported files with the Simotion Mapper.

You can configure certain changes to the configuration of the SIMOTION automation system, e.g. change an IP address, later in WinCC. Further changes must be configured in SIMOTION SCOUT, exported again and transferred with the Simotion Mapper.

Note

You must be familiar with the SIMOTION SCOUT Engineering System and the configuration of WinCC to configure the "SIMOTION" channel in WinCC.

15.2 Overview of the supported data types

Introduction

The data type and the format adaptation to the data format in the automation system are determined in the configuration of a tag.

The table shows the data types supported by the channel and the application of format adaptations.

Supported data types

Data Types	Type conversion
Binary tag	No
Signed 8-bit value	Yes
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Text tag, 8-bit font	No
Raw data type	No

15.3 Configuring the channel

15.3.1 Configuration of the "SIMOTION" channel

Introduction

This chapter describes how to configure the "SIMOTION" channel.

Note

You must be familiar with the SIMOTION SCOUT Engineering System and the configuration of WinCC to configure the "SIMOTION" channel.

Proceed as follows to configure the "SIMOTION" channel:

1. Export the SIMOTION SCOUT project from SIMOTION SCOUT.

2. Create WinCC project with the Simotion Mapper.

3. Open WinCC project.

4. Configure system parameters.

Further information about the diagnosis of the channel, the connection and the tags can be found in the "Diagnosis 'SIMOTION' channel" chapter.

15.3.2 How to export a SIMOTION SCOUT project

Introduction

This section describes how to export tags and message definitions from SIMOTION SCOUT.

Requirements:

- You are familiar with the SIMOTION SCOUT Engineering System.
- You have access to the SIMOTION SCOUT project to be exported.

Procedure

1. Open the SIMOTION SCOUT project to be exported in SIMOTION SCOUT.

2. Select "Export OPC Data" under "Tools."

3. Select version "SIMATIC NET V6.4", the desired scope and at least the "OPC-Alarm/Event" option for the export.

   Note

   Simotion Mapper does not process other export versions than "SIMATIC NET V6.4".

4. Select the destination directory.

5. Select the communication interface.
   The project is exported.

6. Enter the routing information if you are using routing.

The SIMOTION SCOUT project is exported. The "OPC_Data.sti" and "OPC_AE.xml" files are saved in the destination directory.

15.3.3   How to create a WinCC project with Simotion Mapper

Introduction

This section describes how to create a WinCC project from the exported SIMOTION SCOUT project with the Simotion Mapper.

Note

If a WinCC project was already created for an older version of the SIMOTION SCOUT project, only the SIMOTION parameters are changed in a transmission. All other configuration settings in the WinCC project (such as archiving) remain the same.

Requirements:

- You have access to the export files "OPC_Data.sti" and "OPC_AE.xml" of the SIMOTION SCOUT project.

- You have access rights to the WinCC installation directory.

  Note

  In the SIMOTION SCOUT programming environment, the tags to be exported can be filtered using the watch tables. Use the watch tables to keep the number of tags in the WinCC project low. More information on the watch tables can be found in the SIMOTION SCOUT online help.

Procedure

1. Launch the "SimotionMapper.exe" program in the WinCC installation directory.

2. Click "Open". Navigate to the directory with the files "OPC_Data.sti" and "OPC_AE.xml". The data is read and displayed in Simotion Mapper.

3. In the Simotion Mapper Explorer, select the groups and tags you need in your WinCC project.

4. Select "Create new WinCC project".

5. If you want to change the "WinCC connection name", click on the name displayed and enter the new name.

6. Specify the "First TA message number" for the technological alarm. The value must be selected in such a way that it does not lead to collisions with messages of other communication channels. The default value is 100.
   The Simotion Mapper creates one message and a total of six template messages for each Simotion connection starting from the number specified.

7. Click "Start mapping". Select the destination folder for the WinCC project.
   The WinCC project is created. The progress bar indicates the progress of the procedure.

8. Close Simotion Mapper.

The WinCC project is created and can now be opened and edited in WinCC.

Note

You may have to set the system parameters of the "SIMOTION" channel in WinCC to use the created WinCC project.

15.3.4 How to change a WinCC project with Simotion Mapper

Introduction

This section describes how to add an exported SIMOTION SCOUT project to an existing WinCC project with the Simotion Mapper. In this way, you can use the same Simotion project several times in one WinCC project, for example.

Note

If a WinCC project was already created for an older version of the SIMOTION SCOUT project, only the SIMOTION parameters are changed in a transmission. All other configuration settings in the WinCC project (such as archiving) remain the same.

Requirements:

- You have access to the export files "OPC_Data.sti" and "OPC_AE.xml" of the SIMOTION SCOUT project.
- You have access rights to the WinCC installation directory.

  ### Note

  In the SIMOTION SCOUT programming environment, the tags to be exported can be filtered using the watch tables. Use the watch tables to limit the number of tags in the WinCC project. More information on the watch tables can be found in the SIMOTION SCOUT online help.

Procedure

1. Open the WinCC project to be edited.
2. Launch the "SimotionMapper.exe" program in the WinCC installation directory.
3. Click "Open". Navigate to the directory with the files "OPC_Data.sti" and "OPC_AE.xml". The data is read and displayed in Simotion Mapper.
4. In the Simotion Mapper Explorer, select the groups and tags you need in your WinCC project.
5. Select "Add to the open project".
6. If you want to re-add a group or tag that has already been created, you must change the "WinCC connection name" by clicking on the name displayed.
7. If you do not want to transfer any messages, groups or tags for a connection, unselect "WinCC connection name".
8. Specify whether tags should be overwritten.
9. Specify the "First TA message number" for the technological alarm. The value must be selected in such a way that it does not lead to collisions with messages of other communication channels. The default value is 100.
   The Simotion Mapper creates one message and a total of six template messages for each Simotion connection starting from the number specified.

   ### Note

   Please do change any "First TA message number" that has already been mapped. If you do, you may experience unpredictable message behavior.

10. Click "Start mapping". Select the destination folder for the WinCC project.
    The SIMOTION SCOUT project is added to the open WinCC project. The progress bar indicates the progress of the procedure.
11. Close Simotion Mapper.

The WinCC project was expanded by the SIMOTION SCOUT project and saved with your settings.

15.3.5 How to change the connection parameters

Introduction

In this section, you will learn how to change the connection parameters of the SIMOTION network address.

Note

Change only the connection parameters listed here. Do not create new connection for the "SIMOTION" channel. Incorrectly set connections may result in control errors in the PLC. Configure new connections according to the description in the chapter "Configuration of the "SIMOTION" channel (Page 327)".

Requirements

- The SIMOTION communication driver is integrated in the WinCC project.

Procedure

1. Open the directory structure for the "SIMOTION" communication driver in the "Tag Management" editor.

2. Select "Properties" in the shortcut menu of the channel unit.

3. Click "Properties" in the open dialog box.
   The "Connection Properties - SIMOTION" dialog box opens.
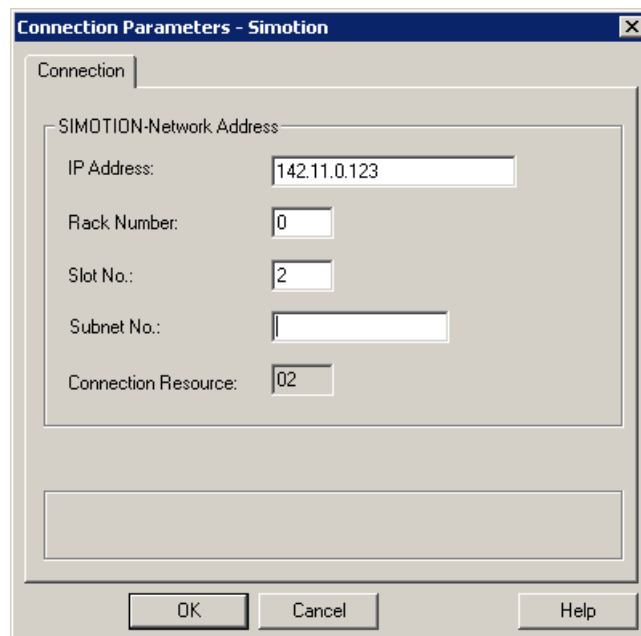
4. Change the connection parameters for the SIMOTION network address in the respective fields.

5. Close each open dialog box by clicking "OK."

15.3.6 How to change the tag address

Introduction

This chapter describes how to change a tag address in the "SIMOTION" channel.

Note

You must have very good knowledge of the use of ANY pointers to change the tag address in the "SIMOTION" channel. No communication connection may be established if the tag address is entered incorrectly.

Requirements

- The "SIMOTION" channel in integrated into the WinCC project.

- A connection with tags has been created in the "SIMOTION" channel unit.

Procedure

1. Click "Addressing" in the shortcut menu of the tag to be changed.
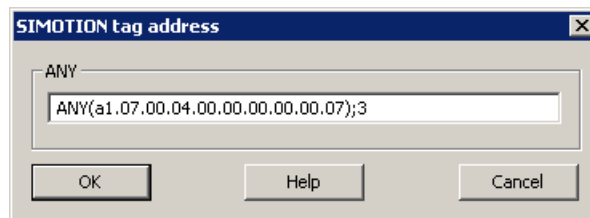
2. Change the tag address.

15.3.7 System parameter configuration

15.3.7.1 System Parameters of the Channel Unit

Introduction

If you require a different configuration than the WinCC default settings, make these settings in the "System Parameters" dialog box.

You can change the following system parameters:

● Logical device name

● The channel uses cyclic read services in the AS

Logical device name

WinCC and the PLC communicate by means of logical device names that are assigned when the communications processor is installed in the PLC.

The channel uses cyclic read services in the AS

The PLC cyclic read services group the tags that are to be read cyclically into individual requests and transfer these to the PLC. The PLC sends the requested data the first time on receipt of the request and then again each time the cycle time elapses.

When cyclic read services are enabled, you can use the change-driven transfer function. If the PLC supports change-driven transfer, the data are then transferred only when values are changed.

15.3.7.2 How to Configure the System Parameters

Introduction

This section shows how to configure the system parameters of the "SIMOTION" channel.

The "System Parameters" dialog comprises two tabs:

● "SIMOTION" tab

● "Unit" tab

Note

When the project is copied to another computer, the settings in the "Unit" tab are retained. The settings on the "SIMOTION" tab are deleted on the other hand.

Requirements

- The "SIMOTION" channel in integrated into the WinCC project.

Procedure

1. Select the "SIMOTION" channel in the variable management. Open the "System Parameters" dialog box in the shortcut menu of the "Simotion" channel unit.

2. Select the "SIMOTION" tab.
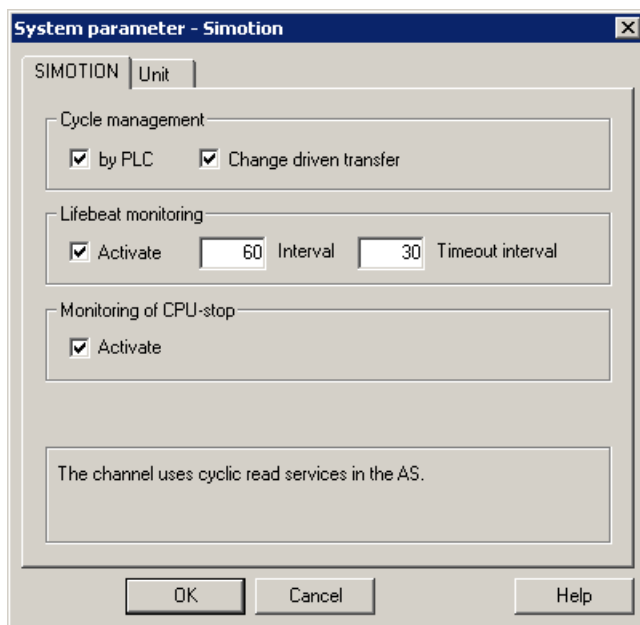
3. To enable cyclic reading of tags and change-driven transfer, select "by PLC" and "Change-driven transfer."

Note

The "cycle management", "lifebeat monitoring" and "stop monitoring" functions are not supported by the integrated SINAMICS servo control. The "SIMOTION" channel therefore ignores corresponding settings for connections to SINAMICS servo controls. The channel determines whether the AS supports the respective function when establishing the connection.

---

4. Select the "Lifebeat monitoring" function if required.
   Determine the interval in seconds for sending lifebeat monitoring messages.
   Determine the monitoring time in seconds for monitoring the response to a lifebeat monitoring message.

5. Enable "CPU Stop Monitoring" if you want WinCC to signal a fault in the communication when the SIMOTION CPU is in the stopped state.

6. Select the "Unit" tab.
   "S7ONLINE" is displayed as a default for "logical device name". You must change the device name if a different name was selected when installing the used communications processor.
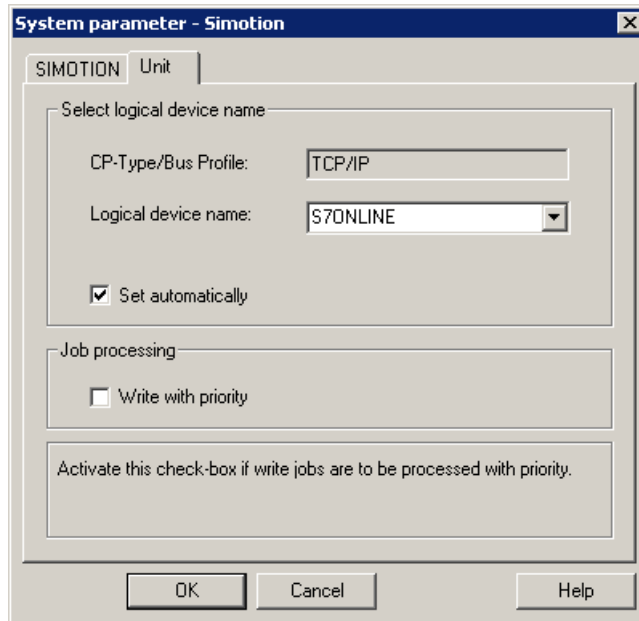
7. To set the device name automatically at the start of runtime, select "Set automatically."

8. To give write jobs higher priority than read jobs during processing, select "Write with priority."

9. Close the dialog by clicking "OK."

Note

Setting changes only take effect after WinCC is restarted.

15.3.7.3   How to Change the Logical Device Name

Introduction

WinCC and SIMOTION communicate through logical device names. These logical device names are assigned when the communications processor is installed.

Requirements

- The "SIMOTION" channel in integrated into the project.

- A connection has been created in the "SIMOTION" channel unit.

Procedure

1. Select the SIMOTION channel in Tag Management.

2. Open the "System Parameters" dialog box in the shortcut menu.

3. Select the "Unit" tab.

4. Enter the device name in the "Logical device name" field. You can select an entry from the list or enter a new name.
   The device names are determined by the "Set PG/PC interface" tool. You call the tool in the system control. Only the currently set device name is displayed if it is not installed.
   If you specify a different logical device name, a message is displayed.
   Only enter a name if the communications processor being used on the target station is not installed on the configuring system.

5. Close the dialog by clicking "OK."

Note

Setting changes only take effect after WinCC is restarted.

15.4   Diagnosis "SIMOTION" channel

15.4.1   Diagnosis possibilities of the "SIMOTION" channel

The following possibilities exist for the diagnosis and error detection of the "SIMOTION" channel and its tags.

Checking the Communication Processor Configuration

After checking the access point, the communication processor can be tested with the "Set PG/ PC interface" application. The communication processor can be checked under SIMATIC NET in the same way.

Checking the Configuration of the Connection and Tags

There may be errors in the configuration of the system and connection parameters. An incorrect tag addressing may also be responsible for wrong tag values.

Diagnosis of the Channel with "Channel Diagnosis"

You can query the status of the channel and the connection in runtime with "Channel Diagnosis". Errors are displayed by "Error Codes".

Diagnosis of the Channel Tags

You can query the current value, the current quality code and the last change time of the tag in runtime in the tag management.

15.4.2   Description of Log File Entries

Introduction

The channel enters important status changes and errors in the logfile. The entries support the analysis of communication faults.

Every entry in the file contains a date and time stamp with the following flag names and description.

Example of a logbook entry:

2009-10-28 12:10:11,467 INFO Log starting ...

2009-10-28 12:10:11,483 INFO | LogFileName : D:\SIEMENS\WINCC\Diagnosis \Simotion_01.LOG

2009-10-28 12:10:11,483 INFO | LogFileCount : 3

2009-10-28 12:10:11,483 INFO | LogFileSize : 1400000

2009-10-28 12:10:11,483 INFO | TraceFlags : fa000000

2009-10-28 12:10:11,498 INFO SIMOTION channel DLL started!

2009-10-28 12:10:11,498 INFO SIMOTION channel with own cycle creation!

2009-10-28 12:10:11,967 INFO Connection "D445": StartRegisterEvVariable for dwVariableCount = 89

2009-10-28 12:10:11,967 INFO Connection "D445": RegisterEvVariable for Variable "@D445@CheckSum"!

...

2009-10-28 12:10:11,983 INFO Connection "D445": EndRegisterEvVariable

2009-10-28 12:10:12,436 INFO S7DOS release: @(#)TIS-Block Library DLL Version R8.0.0.0-REL-BASIS

2009-10-28 12:10:12,436 INFO S7DOS version: V8.0 / 0

2009-10-28 12:10:12,436 INFO SIMOTION version: V6.0 / Sep 15 2009 / 08:06:43

2009-10-28 12:10:12,436 INFO SIMOTION channel unit "Simotion" activated!

2009-10-28 12:10:12,451 ERROR Cannot connect to "SINAMICS_Integrated": Errorcode 0xFFDF 42C2!

2009-10-28 12:10:12,451 ERROR Cannot connect to "D445": Errorcode 0xFFDF 42C2!

Description of the Most Important Entries for the "INFO" Flag

Message text	Meaning
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	Name of the log file with path
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : c4000000	Displays the flags used by the Trace function as a hexadecimal number
SIMOTION channel DLL started!	Start message
SIMOTION channel DLL terminated!	End message

Description of the Most Important Entries for the "ERROR" Flag

Message text	Meaning
Cannot connect to <connectionname>: Errorcode 0x0000 7<xxx>!	Communication error Communication to SIMOTION could not be established immediately after activating WinCC. <connectionname> = Name of connection <xxx> 1...fff The channel has received all other error codes as the result of a function call of S7DOS, a lower layer or from the AS.
Connectionerror <nnn> <connectionname>: Errorcode 0x0000 7xxx!	Communication error Communication to SIMOTION could not be established after activating WinCC. The connection was broken. <nnn> = Number of connection terminations for this connection <connectionname> = Name of connection <xxx> 1...fff The channel has received all other error codes as the result of a function call of S7DOS, a lower layer or from the AS.
Channel API error: errorstring	Channel API error The channel passed the error string 'errorstring' to WinCC Explorer. The error string is displayed in an information box depending on the error relevance. See API error texts for a description of the error strings.
Max. count of API errors reached - API logbook deactivated	Channel API error Depending on the error and function, errors can occur cyclically on the API. To avoid filling the logbook file with these error messages, a maximum of 32 messages are output for an API error.
Cannot write storage data! Cannot read storage data / use default data Storage data illegal or destroyed / use default data! No storage data / use default data!	General Channel Error Messages
Devicename in unit "unitname" changed from "old devicename" to "new devicename"	Initialization message

Note

The error codes of the "SIMOTION" channel correspond to those of the "SIMATIC S7 Protocol Suite" channel. You will find the description of the error code in the "Error codes for connection fault" chapter in the help for the "SIMATIC S7 Protocol Suite" channel.

In addition, SIMOTION reports the error code 0x000 7301 if the consistency check failed. The reason for the error message is that the data exported from SIMOTION SCOUT for the AS entered checksum do not match the checksum in the connected device.

System Info

<div style="text-align: right; font-size: 3em;">16</div>

16.1   "System Info" Channel

Contents

The "System Info" channel is used to evaluate system information such as the time, date, disk capacity and provides functions such as timers and counters.

This chapter will show you

- configure the channel, connection and tags
- display system information in a process picture
- use system information to trigger and display a message
- display system information graphically
- display the system information from several servers in a multi-user system

16.2 WinCC System Info Channel

Principle

The System Info channel is used to evaluate system information such as the time, date, disk capacity and provides functions such as timers and counters.

Possible applications are:

- Display of the time, date and day of the week in process pictures
- Triggering of events through evaluation of system information in scripts
- Display of the CPU load in a trend
- Display and monitoring of the available drive space on different servers of a client system
- Monitoring of the available disk capacity and triggering of a message

The channel requires no hardware, since it directly accesses the system information of the computer on which it has been installed. In order for the channel to function, a connection must be set up. Additional connections are possible, but not required for the proper operation.

For more information regarding the diagnosis of channels and tags, refer to "Communication Diagnostics".

Note

The process tags required for the System Info channel need no licenses. Thus, the tags are not entered in the license count.

If you have no administrator rights under Windows 2003 Server, you must be a member of the superusers of the "System Monitor User" group in order to use the System Info channel.

Communication Manual

The communication manual contains additional information and extensive examples for the channel configuration. This manual is available for download on the Internet:

- http://support.automation.siemens.com/

Search by order number:

● A5E00391327

See also

16.3 Overview of the Supported System Information

Introduction

In the "Function" field in the "System Info" dialog, you can specify the system information to be assigned to a WinCC tag. The display format is set in the "Format" field.
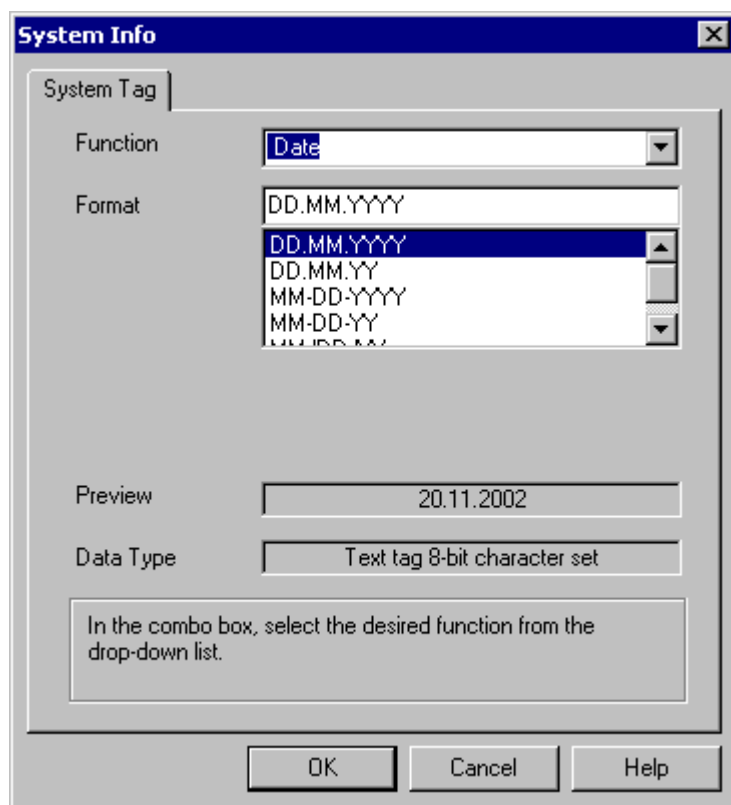
System Info Channel Supported System Information - Overview

Function	Data type	Format	Preview
Date	Text tag 8-bit character set	DD.MM.YYYY	21.10.1999
		DD.MM.YY	21.10.99
		MM-DD-YYYY	10-21-1999
		MM-DD-YY	10-21-99
		MM/DD/YY	10/21/99
Day	Unsigned 16-bit value	DD	1...31
Month	Unsigned 16-bit value	MM	1...12
Year	Unsigned 16-bit value	YYYY	2000

Function	Data type	Format	Preview
Weekday	Unsigned 16-bit value Text tag 8-bit character set	Text: 1 for Monday to 7 for Sunday Text: Mon,Tue,Wed, Thu,Fri,Sat,Sun	1...7 Mon ... Sun
Time	Text tag 8-bit character set; length = 10 bytes Length = 12 bytes	HH:MM:SS HH:MM HH:MM AM,PM HH:MM:SS AM,PM	23:45:37 23:45 23:45 PM 23:45:37 PM
Hour	Unsigned 16-bit value	HH	0...23
Minute	Unsigned 16-bit value	MM	0...59
Second	Unsigned 16-bit value	SS	0...59
Milliseconds	Unsigned 16-bit value	MSC	0...999
Counter	Signed 32-bit value	ZZZZ	0...9999
CPU load	Floating-point number 32-bit IEEE 754	Total load in % idle load in % process load in %	0...100%
Timer	Signed 32-bit value	TTTT	0...9999
Free main memory	Floating-point number 32-bit IEEE 754	Free capacity in kB Free in % Free in bytes	0...n kB 0...100% 0...n B
Free disk capacity (local disks)	Floating-point number 32-bit IEEE 754	Free in MB Free in %	0...n MB 0...100%
Printer monitoring	Unsigned 32-bit value	Filled capacity of spooler disk Printer status Job status Free spooler disk area in kB Free PRT_OUT-Disk areas in kB Size of spooler directory in kB Size of PRT_OUT directory in kB	0...n % 0...n 0...n 0...n kB 0...n kB 0...n kB 0...n kB
Status of swap file	Floating-point number 32-bit IEEE 754	Used in kB Used in % Available in kB	0...n kB 0...100% 0...n kB

Counter

This function is useable for test purposes in scripts.

Timer

When this function is selected, the "System Info" dialog is extended with fields "Limits from" and "to".

After every second, the timer is incremented or decremented. The direction in which the changes are made is determined by the starting and ending values in the fields "Limits from" and "to". If the start value is smaller than the end value, the timer is incremented. If the start value is greater than the end value, the timer is decremented.

If in Runtime, a value is entered in the I/O field linked to the timer, the start and current timer values is set to this value. Example: timer configured from 0 to 60. If "0" is written in Runtime, the timer is reset.

After the deactivation, the original start value is reapplied.

CPU load

For the formats "Total load in %" and "Idle load in %" in connection with multiprocessor PCs enter the CPU number beginning with "0".
For the format "Process load in %" enter the instance number, if there are several instances of a process.

Free disk capacity

The system can only determine the space available on a local hard disk or diskette.

Printer monitoring

With the "Printer status" and "Job status" formats the server name must be entered in the "Printer" field. The printers in use must support this status information to be able to use this system information.

In order to analyze the printer status, please observe the following:

- The port monitor is responsible for the transmission of the printer status to the spooler. Depending on the selected printer port, different port monitor DLLs are installed. From the port monitors supplied with Windows, only "TCPMON.DLL" is capable of transmitting the printer status using the TCP/IP port. "LOCALMON.DLL" using the LPT port does not communicate the printer status.

- The printer status is assessed only after a print job has been submitted, but not during polling of the status at the port.

In the case of the "Free PRT_OUT drive space" and "PRT_OUT Directory Size" formats, the channel automatically determines the path for the "Directory" field.

Error codes for the "Printer status" format

Status	Error code
PRINTER_STATUS_PAUSED	0x00000001
PRINTER_STATUS_ERROR	0x00000002
PRINTER_STATUS_PENDING_DELETION	0x00000004

Status	Error code
PRINTER_STATUS_PAPER_JAM	0x00000008
PRINTER_STATUS_PAPER_OUT	0x00000010
PRINTER_STATUS_MANUAL_FEED	0x00000020
PRINTER_STATUS_PAPER_PROBLEM	0x00000040
PRINTER_STATUS_OFFLINE	0x00000080
PRINTER_STATUS_IO_ACTIVE	0x00000100
PRINTER_STATUS_BUSY	0x00000200
PRINTER_STATUS_PRINTING	0x00000400
PRINTER_STATUS_OUTPUT_BIN_FULL	0x00000800
PRINTER_STATUS_NOT_AVAILABLE	0x00001000
PRINTER_STATUS_WAITING	0x00002000
PRINTER_STATUS_PROCESSING	0x00004000
PRINTER_STATUS_INITIALIZING	0x00008000
PRINTER_STATUS_WARMING_UP	0x00010000
PRINTER_STATUS_TONER_LOW	0x00020000
PRINTER_STATUS_NO_TONER	0x00040000
PRINTER_STATUS_PAGE_PUNT	0x00080000
PRINTER_STATUS_USER_INTERVENTION	0x00100000
PRINTER_STATUS_OUT_OF_MEMORY	0x00200000
PRINTER_STATUS_DOOR_OPEN	0x00400000
PRINTER_STATUS_SERVER_UNKNOWN	0x00800000
PRINTER_STATUS_POWER_SAVE	0x01000000

Error codes for the "Job status" format

Status	Error code
JOB_STATUS_PAUSED	0x00000001
JOB_STATUS_ERROR	0x00000002
JOB_STATUS_DELETING	0x00000004
JOB_STATUS_SPOOLING	0x00000008
JOB_STATUS_PRINTING	0x00000010
JOB_STATUS_OFFLINE	0x00000020
JOB_STATUS_PAPEROUT	0x00000040
JOB_STATUS_PRINTED	0x00000080
JOB_STATUS_DELETED	0x00000100
JOB_STATUS_BLOCKED_DEVQ	0x00000200
JOB_STATUS_USER_INTERVENTION	0x00000400
JOB_STATUS_RESTART	0x00000800

Note

The error codes of the "Printer status" and "Job status" formats corresponds to the values in the Visual C-referenced file "Winspool.h" .

16.4   Differences to Other Software Components

Introduction

Some of the System Info channel system information can also be evaluated or displayed by WinCC using ActiveX controls.

Once fundamental difference to ActiveX controls can be seen in the fact that the System Info channel system information is assigned to a WinCC tag. The continued evaluation (e.g. of messages, limit values) can be carried out repeatedly and is then configured individually. The ActiveX controls are intended for use in the specified applications and may also be used for multi-user or client systems.

For the following system information, differences between the ActiveX control and the channel exist:

Time

The ActiveX control "WinCC Digital/Analog Clock Control" is used to display the time in WinCC. This control also supports an analog display of the time. The control does not require the System Info channel even when if it is used in the process control system options. Using the control, it is possible to display a WinCC client's time in its process picture. This is not possible with the System Info channel, since this always displays the server's system time.

Free disk capacity

The ActiveX control "IX Diskspace" is supplied to display the disk space available in WinCC. This ActiveX control can also display the space available on network drives and supports other configuration options such as the setting of multiple limit values directly in the control.

16.5 Configuring the Channel

16.5.1 How to Configure the System Info Channel

Introduction

This section illustrates how to configure the System Info channel.

Procedure

1. Select the entry "Add New Driver" in the shortcut menu of Tag Management. A selection dialog is opened.

2. Select the "System Info.chn" channel and close the dialog. The channel is set up and the channel driver is displayed in Tag Management.

3. Select the associated System Info channel unit and call up the shortcut menu. In this shortcut menu, select "New Connection".

4. In the "Connection Properties" dialog, enter the connection name. Close the dialog.

5. Choose the "New Tag" option from the shortcut menu for the connection. A dialog is opened.

6. Enter the tag name and set its data type.
   If you want to use the example for this channel, close the dialog and continue with the topic "How to Configure a Tag".

7. Click the "Select" button to open the "System Info" dialog. Select the desired function and format. The associated type of the WinCC tag is displayed in the "Data Type" field.

8. Close the open dialogs.

16.6 Examples of Evaluating and Displaying System Information

16.6.1 How To Call Up and Evaluate System Information

This section uses examples to illustrate how system information can be displayed and evaluated in a variety of ways.

See also

16.6.2 How to Configure a Tag in the System Info Channel

Introduction

This section illustrates how to configure tags in the System Info channel. These tags are used in the examples.

Requirement

Install the "SystemInfo.chn" channel.

Table of the Data Types Used

The table below shows the tag types and formats used in the System Info channel.

Example	System Information	Tag name	Data type	Format
I/O field	Time	Sysinfo_Time	Text tag 8-bit character set	Hours: minutes: seconds (HH:MM:SS)
Bar, message	Free disk capacity (disk: C)	Sysinfo_Drive_C	Floating-point number 32-bit IEEE 754	0-100% (free in %)
Trends	CPU load	Sysinfo_CPU	Floating-point number 32-bit IEEE 754	0-100% (total load in %)
Printer status	Printer monitoring	Sysinfo_Printerstate	Unsigned 32-bit value	0-n (hex) (printer status)

Procedure

1. In the shortcut menu of the associated System Info channel unit, select the entry "New Connection" and create a connection named "Testinfo".

2. Choose the "New Tag" option from the shortcut menu for the connection.

3. In the "Tag Properties" dialog, enter the tag name. For the example, take the appropriate tag name from the table of data types used and enter it in the "Name" field. In the "Data Type" field, set the tag's data type in accordance with the table.

4. Click the "Select" button to open the "System Info" dialog. Set the system information and display format in accordance with the information in the table.

5. Close all open dialogs.

See also

How to Configure a Message Regarding Free Disk Capacity (Page 356)

How to Configure the System Info Channel (Page 350)

How to Display the Printer Status in a Status Display (Page 359)

How to Display the CPU Load in a Trend Window (Page 355)

How to Display the Free Disk Capacity in a Bar Graph (Page 354)

How to Display the Time in an I/O Field (Page 352)

16.6.3   How to Display the Time in an I/O Field

Requirements

Configure a "Sysinfo_Time" tag with data type "Text tag 8-bit character set". This tag must be assigned the "Time" system information with display format "HH:MM:SS".

Procedure

1. Start Graphics Designer and open a picture.

2. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
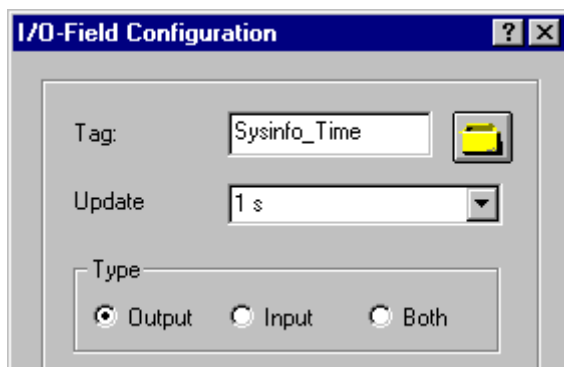
3. In the "Tag" field, enter the name "Sysinfo_Time".

4. Set the update to "1 s".

5. Set the field type to "Output". Close the dialog.

6. Click "Properties" in the I/O field's shortcut menu to open the "Object Properties" dialog.

7. On the "Properties" tab, select "Output/Input". Set the "Data Format" attribute to "String".

8. Close the dialog and save the picture.

9. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a time display every 250 ms is detrimental to the system performance.

See also

How to Start Runtime (Page 362)

How to Insert an I/O Field (Page 362)

How to Configure a Tag in the System Info Channel (Page 351)

16.6.4 How to Display the Free Disk Capacity in a Bar Graph

Requirements

Configure a tag "Sysinfo_Drive_C" with data type "Floating-point number 32-bit IEEE754". This tag must be assigned the "Free Disk Space" system information for drive "C" with display format "Free capacity in %".

Procedure

1. Start Graphics Designer and open a picture.

2. Insert a bar graph into the picture. For this purpose, select the object "Bar" from "Smart Objects" in the object palette. The "Bar Configuration" dialog is opened.
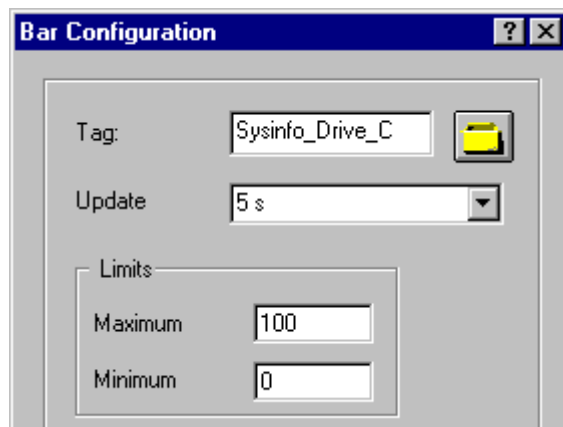
3. In the "Tag" field, enter the name "Sysinfo_Drive_C".

4. Set the update to "5 s".

5. Set the maximum value to "100" and the minimum value to "0". Close the dialog.

6. Click on "Properties" in the bar graph shortcut menu to open the "Object Properties" dialog.

7. On the "Properties" tab, select "Axis". Set the attribute "Decimal Places" to "0".

8. Close the dialog and save the picture.

9. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating an available drive space display every second is detrimental to the system performance.

See also

How to Start Runtime (Page 362)

How to Insert a Bar Graph (Page 361)

How to Configure a Tag in the System Info Channel (Page 351)

16.6.5 How to Display the CPU Load in a Trend Window

Requirements

Configure a tag named "Sysinfo_CPU" with data type "Floating-point number 32-bit IEEE754". This tag must be assigned the "CPU Load" system information with display format "Total load in %".

Procedure

1. Start Graphics Designer and open a picture.

2. Insert a trend display into the picture. You can accomplish this by selecting the "WinCC Online Trend Control" object from the "Controls" object palette. The "Properties of WinCC Online Trend Control" dialog is opened.
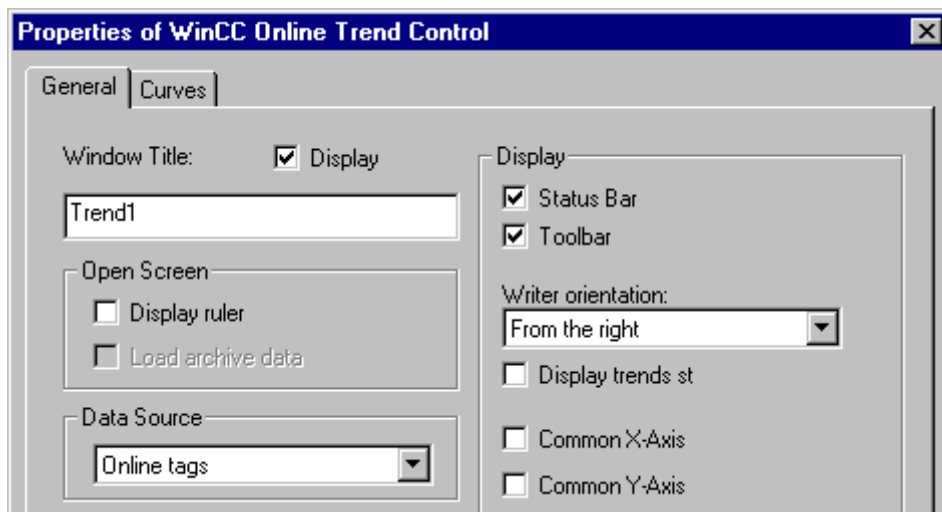
3. On the "General" tab, enter the name "Trend1" in the "Window Title" field.

4. In the "Data Source" field, select "Online Tags".

5. Click the "Trends" tab and then the "Selection" button to open the "Tag Configuration" dialog.

6. Enter "Sysinfo_CPU" as the name of the tag and then select a cycle time of "2 s". Close the dialog.
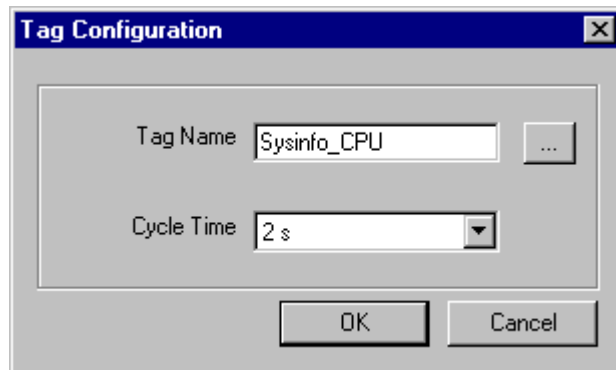
7. Close the "Properties of WinCC Online Trend Control" dialog and save the picture.

8. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a CPU load display every 500 ms is detrimental to the system performance.

See also

How to Start Runtime (Page 362)

How to Configure a Tag in the System Info Channel (Page 351)

16.6.6 How to Configure a Message Regarding Free Disk Capacity

Requirements

Configure a tag "Sysinfo_Drive_C" with data type "Floating-point number 32-bit IEEE754". This tag must be assigned the "Free Disk Space" system information for drive "C" with display format "Free capacity in %".

Procedure

1. Start Alarm Logging and create a new message by adding a new line in the table window below.

2. Open the "Single Message" dialog by clicking on "Properties" in the shortcut menu for the new message.

3. Click the "Text" tab and then enter the text "Memory Space" in the "Message Text" field, and the text "Hard Disk" in the "Point of Error" field. Close the dialog.
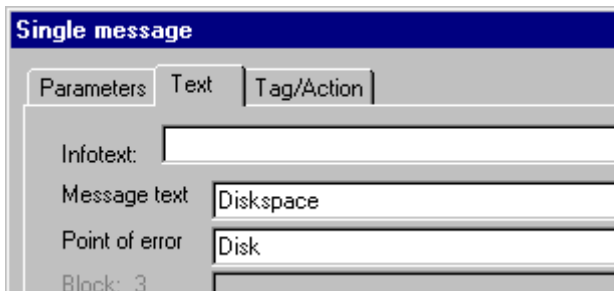
4. Select the "Extras / Add Ins" menu item to open the "Add Ins" dialog. If necessary, select the "Analog Alarm" check box. Close the dialog.

5. Select "Analog Alarm" in the navigation window for Alarm Logging and then click on "New" in the shortcut menu to open the "Properties" dialog. In the "Tag to be Monitored" field, enter the name "Sysinfo_Drive_C".
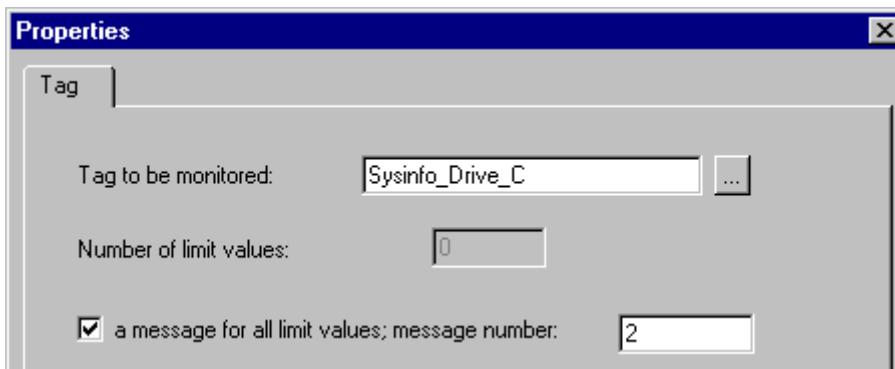
6. Select the "One message for all limit values" check box. In the "Message Number" field, enter the number of the newly created message. Close the dialog.

7. Select "Analog Alarm" in the navigation window. Select the entry "Sysinfo_Drive_C" in the data window. Click "New" in the shortcut menu to open the "Properties" dialog. Click the radio button "Lower Limit" and enter the value "30" in the "Limit value or tag" field.

8. Close the dialog. Save and exit Alarm Logging.

See also

How to Configure a Tag in the System Info Channel (Page 351)

How to Display a Message regarding the Available Disk Capacity (Page 358)

16.6.7 How to Display a Message regarding the Available Disk Capacity

Requirements

- A tag "Sysinfo_Drive_C" with data type "Floating-point number 32-bit IEEE754". The tag must be assigned the "Free Disk Space" system information for drive "C" with the display format "Free mem in %".

- A message text and the lower limit value for the limit value monitoring of this tag.

- "Alarm Logging Runtime" must be set in the computer's startup parameters.

Procedure

1. Start Graphics Designer and open a picture.

2. Insert a message window into the picture. Select the "WinCC Alarm Control" object from the "Controls" object palette and place it in the picture.
   Close the quick configuration dialog without making any further changes.

3. Double-click the inserted control to open the "WinCC Alarm Control Properties" dialog.

4. Select the "User Text Blocks" entry from the "Type" field on the "Message Blocks" tab.
   Under "Selection", select the check boxes "Message Text" and "Point of Error".
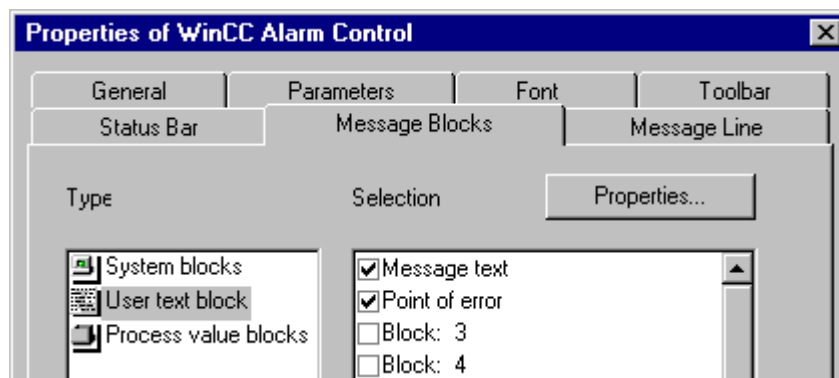
5. Click the "Message Line" tab. In the "Available Message Blocks" field, mark the entries "Message Text" and "Point of Error" and then click the "Right Arrow" button to transfer them to the "Elements of message line" field. Close the dialog.
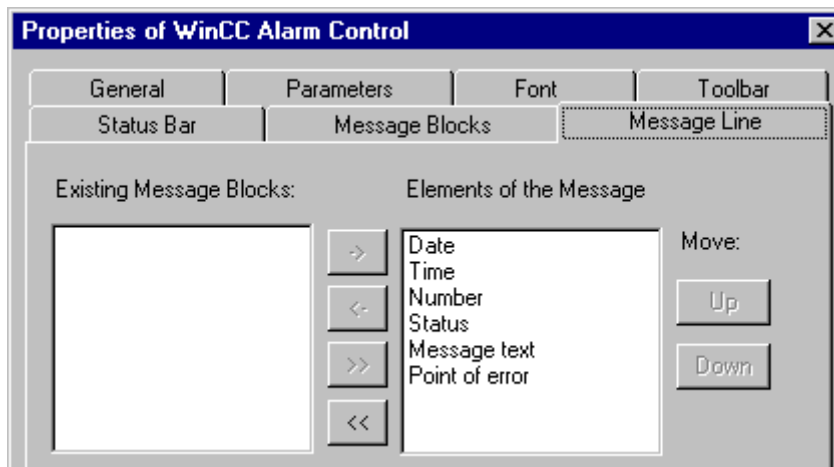
6. Save the picture.

7. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

See also

How to Check the WinCC Startup Parameters (Page 361)

How to Start Runtime (Page 362)

How to Configure a Tag in the System Info Channel (Page 351)

16.6.8 How to Display the Printer Status in a Status Display

Introduction

This example shows a possible evaluation of the "Printer monitoring" system information based on the "Printer status" display format. The printer or job status could also be evaluated by configuring messages that are triggered by single bits in the printer or job status.

Requirements

- Configure a "Sysinfo_Printerstate" tag with data type "Unsigned 32-bit value". This tag must be assigned to the "Printer monitoring" system information with the "Printer status" display format.

- The printers in use must support these status displays to be able to use this system information.

Procedure

1. Start Graphics Designer and open a picture.

2. Insert a status display into the picture. Select the "Status display" object from the object palette under "Smart Objects". The "Status Display Configuration" dialog is opened.
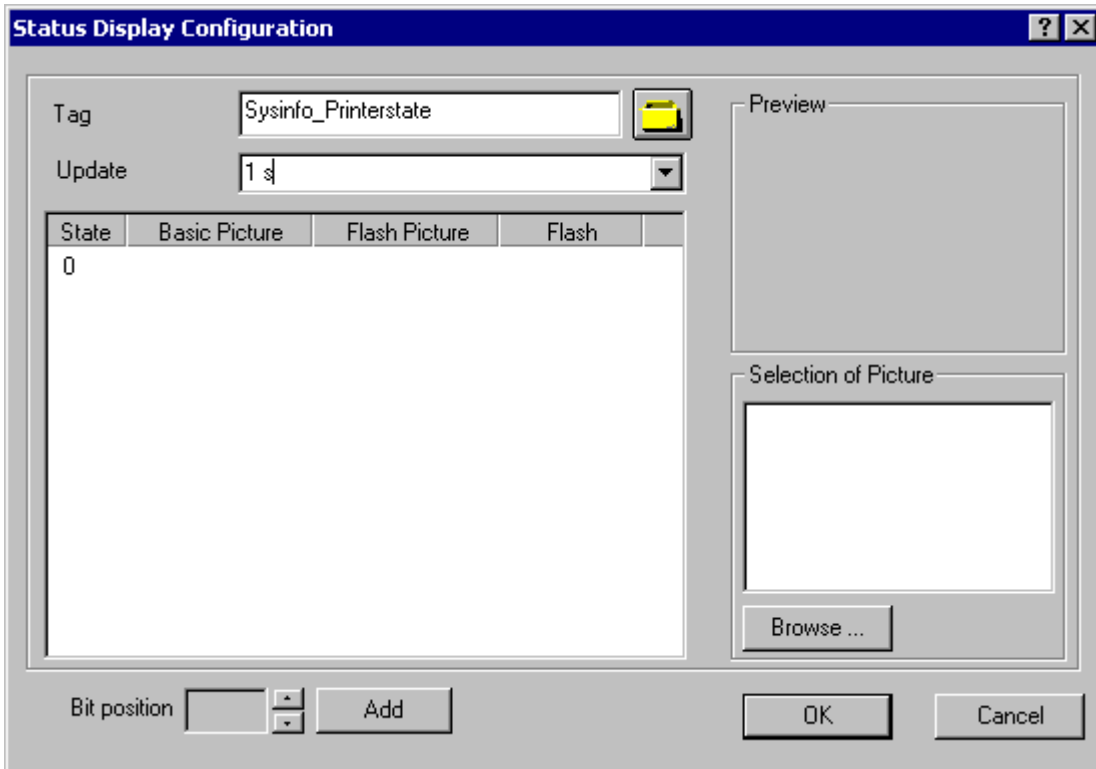
3. In the "Tag" field, enter the name "Sysinfo_Printerstate".

4. Set the update to "1 s".

5. Select the value "0" in the "Status" column. Assign this status an icon from the "Picture Selection" area which, for example, represents a printer. Select the desired icon, drag it to the "0" line with the mouse and drop it in the "Basic Picture" column. If no picture or a picture other than the desired picture is shown in the "Picture Selection" area, a selection dialog can be opened by clicking the "Browse..." button.

6. If you wish you can add additional bit positions with the "Add" button and assign another picture to these statuses.

7. Close the dialog and save the picture.

8. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

In Runtime, a printer ready for use is displayed with the icon assigned the status "0" in step 5. No picture is shown if there is an error in the printer or if you have not run step 6. If you assigned icons to the other bit positions in step 6, they are shown accordingly.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a time display every 250 ms is detrimental to the system performance.

See also

How to Configure a Tag in the System Info Channel (Page 351)

How to Start Runtime (Page 362)

16.6.9 How to Check the WinCC Startup Parameters

Procedure

1. In WinCC Explorer, select "Computer".

2. Open the shortcut menu and select "Properties".

3. The "Computer List Properties" dialog is opened. Click the "Properties" button.

4. The "Computer Properties" dialog is opened.

5. Click the "Startup" tab and check the entries. If necessary, activate or deactivate Runtime modules or add other applications to the startup list.

6. Close the open dialogs.

16.6.10 How to Insert a Bar Graph

Procedure

1. Start Graphics Designer and open a picture.

2. In the "Standard" object palette under "Smart Objects", select the object "Bar".

3. Insert a bar graph into the picture. To do this, point the mouse on the location in the working area where you want the bar graph to be placed. While keeping the mouse button pressed, drag the object to the desired size.

4. After you release the mouse button, the "Bar Configuration" dialog is opened. In this dialog, enter the name of a WinCC tag and set the update specifications and limit. Additionally, you can use the "Bar Direction" to set the orientation of the displayed bar.

5. Close the dialog.

16.6.11   How to Insert an I/O Field

Procedure

1. Start Graphics Designer and open a picture.

2. In the "Standard" object palette under "Smart Objects", select the object "I/O Field".

3. Insert the I/O field into the picture. To do this, position the mouse on the location in the working area where you want the I/O field to be placed. While keeping the mouse button pressed, drag the object to the desired size.

4. After you release the mouse button, the "I/O Field Configuration" dialog is opened. In this dialog, enter the name of a WinCC tag and set the update and field type settings. Additionally, you can also select the "Font" to be used to display the value.

5. Close the dialog.

16.6.12   How to Start Runtime

Requirements

A startup picture must be defined before Runtime is activated.

Procedure

1. Save and close all files that may be open in an editor.

2. Select WinCC Explorer.

3. Activate the project by clicking the "Activate" button in the toolbar or by selecting "Activate" in the "File" menu.

16.7 Special Functions

16.7.1 Use in Multi-User and Client Systems

16.7.1.1 Use in Multi-User and Client Systems

Introduction

In multi-user and client systems, the System Info channel can be used to process the system information from a server on a client system. In a client system, it is thus possible for a single WinCC client to monitor several servers.

See also

Monitoring the system information of several servers on a WinCC client (Page 363)

16.7.2 Example of monitoring system information from multiple servers

16.7.2.1 Monitoring the system information of several servers on a WinCC client

Introduction

In this example, two servers are monitored by a single WinCC client. The monitored system information such as available disk space and CPU load is displayed in a process picture on the WinCC client.

This requires the following configurations:

Configuration of first server

Configuration of second server

Import of tags on the WinCC client

Configuration of the process picture on the WinCC client

Activation of the project

Requirements

The server and the WinCC client must be connected through a Windows network.

See also

16.7.2.2 How to Configure the First Server

Introduction

This section presents the configuration of the first server, which is necessary for this example.

1. Configure the tags of the System Info channel to display available disk space and CPU load.

2. Generation of a package

Table of the Data Types Used

The tag names and formats used in this "System Info" channel example are listed in the following table.

Tag	Function	Data type	Format
Sysvar_1_Drive_C	Free disk capacity	Floating-point number 32-bit IEEE 754	0-100% (free in %)
Sysvar_1_CPU	CPU load	Floating-point number 32-bit IEEE 754	0-100% (total load in %)

Procedure

1. Create a multi-user project named "Testinfo_1" on the first server. Install the driver "System Info.chn" in the project.

2. In the shortcut menu of the associated "System Info" channel unit, select the entry "New Connection" and create a connection named "Connection1".

3. In the connection's shortcut menu, select "New Tag".

4. In the "Tag Properties" dialog, enter "Sysvar_1_Drive_C" as the name of the tag. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".

5. Click the "Select" button to open the "System Info" dialog. Set the "Function" field to "Free Disk Space", the "Drive" to "C" and "Format" to "Free capacity in %". Close all open dialogs.

6. In the connection's shortcut menu, select "New Tag".

7. In the "Tag Properties" dialog, enter "Sysvar_1_CPU" as the name of the tag. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".

8. Click the "Select" button to open the "System Info" dialog. Set the value in the "Function" field to "CPU Load" and the value in "Format" to "Total load in %". Close all open dialogs.

9. Create a package. Proceed by selecting "Server data" in the navigation window and opening the shortcut menu. Select the menu item "Create". Acknowledge the message stating that the package was created.

See also

How to Configure the Second Server (Page 365)

16.7.2.3 How to Configure the Second Server

Introduction

This section presents the configuration of the second server, which is necessary for this example.

1. Configure the tags of the System Info channel to display available disk space and CPU load.

2. Generation of a package.

Table of the Data Types Used

The tag names and formats used in this "System Info" channel example are listed in the following table.

Tag	Function	Data type	Format
Sysvar_2_Drive_C	Free disk capacity	Floating-point number 32-bit IEEE 754	0-100% (free in %)
Sysvar_2_CPU	CPU load	Floating-point number 32-bit IEEE 754	0-100% (total load in %)

Procedure

1. Create a multi-user project named "Testinfo_2" on the second server. Install the driver "System Info.chn" in the project.

2. In the shortcut menu for the associated "System Info" channel unit, select the entry "New Connection" and create a connection named "Connection2".

3. In the connection's shortcut menu, select "New Tag".

4. In the "Tag Properties" dialog, enter "Sysvar_2_Drive_C" as the name of the tag. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".

5. Click the "Select" button to open the "System Info" dialog. Set the "Function" field to "Free Disk Space", the "Drive" to "C" and "Format" to "Free capacity in %". Close all open dialogs.

6. In the connection shortcut menu, select "New Tag".

7. In the "Tag Properties" dialog, enter "Sysvar_2_CPU" as the name of the tag. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".

8. Click the "Select" button to open the "System Info" dialog. Set the value in the "Function" field to "CPU Load" and the value in "Format" to "Total load in %". Close all open dialogs.

9. Create a package. Proceed by selecting "Server data" in the navigation window and opening the shortcut menu. Select the menu item "Create". Acknowledge the message stating that the package was created.

See also

How to Import the Tags to the WinCC Client (Page 366)

16.7.2.4 How to Import the Tags to the WinCC Client

Introduction

This section presents the configuration of the WinCC client, which is necessary for this example.

1. Loading the package of the first sever project.

2. Loading the package of the second sever project.

Requirements

This example requires the use of two server project packages.

Server	Project	Package
1	Testinfo_1	Testinfo_1_<computer_name>
2	Testinfo_2	Testinfo_2_<computer_name>

Procedure

1. Create a client project named "mc_info" on the WinCC client.

2. In the server data shortcut menu, select "Load". The "Open" dialog is opened.

3. Select the computer on which the first server project "Testinfo_1" is located.

4. Select the package "Testinfo_1_<computer_name>.pck" in the "<project_name> \ <computer_name> \ Packages" directory.

5. Click the "Open" button and acknowledge the message after the package has opened.

6. Load the package "Testinfo_2_<computer_name>.pck" on the second server. For this purpose, repeat steps 2 to 5 with the appropriate settings and names for the second project from the "Requirements" table.

See also

How to Configure the Process Picture on the WinCC Client (Page 367)

16.7.2.5   How to Configure the Process Picture on the WinCC Client

Introduction

This section illustrates the configuration of the WinCC client, which is required in this example to display the server system information in a process picture on a WinCC client.

1. Configuration of the system information display of the first server

2. Configuration of the system information display of the second server

Requirements

This example requires that the server project packages are loaded in the client project.

Package	Project	Tag
Testinfo_1_<computer_name>	Testinfo_1	Sysvar_1_Drive_C
Testinfo_1_<computer_name>	Testinfo_1	Sysvar_1_CPU
Testinfo_2_<computer_name>	Testinfo_2	Sysvar_2_Drive_C
Testinfo_2_<computer_name>	Testinfo_2	Sysvar_2_CPU

Procedure

1. On the WinCC client, start Graphics Designer and create a picture named "p_serverinfo".

2. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.

3. Click the button for tag selection. The "Tags" dialog is opened.

4. Select the tag "Sysvar_1_Drive_C" of the first server project "Testinfo_1". For this purpose, open the directory structure under the package "Testinfo_1_<computer_name>". Close the dialog.

5. In the "I/O Field Configuration" dialog, set the update to "5 s".

6. Set the field type to "Output". Close the dialog.

7. Insert a second I/O field into the picture and configure it for the tag "Sysvar_1_CPU" of the same project. For this purpose, repeat the steps 2 to 6 with the appropriate settings taken from the "Requirements" table.

8. Repeat steps 2 to 7 to configure the tags of the second server project "Testinfo_2".

9. Close the dialogs and save the picture.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a date display every second is detrimental to the system performance.

See also

How to Activate the Project (Page 368)

16.7.2.6 How to Activate the Project

Introduction

This section shows how to activate the projects on the servers and the WinCC client.

1. Create a startup picture in the server project "Testinfo_1" and activate the project.

2. Create a startup picture in the server project "Testinfo_2" and activate the project.

3. Define the startup picture in the project "mc_info" on the WinCC client and activate the project.

Procedure

1. In the navigation window in the project "Testinfo_1" on server 1, select Graphics Designer and use the shortcut menu to create a new picture.

2. Set this picture as the startup picture. To do this, select "Set as startup picture" from the shortcut menu.

3. Click the "Activate" button in the toolbar to activate the project.

4. In the navigation window in the project "Testinfo_2" on server 2, select Graphics Designer and use the shortcut menu to create a new picture.

5. Set this picture as the startup picture. To do this, select "Set as startup picture" from the shortcut menu.

6. Click the "Activate" button in the toolbar to activate the project.

7. On the WinCC client in the navigation window in the project "mc_info", select Graphics Designer. The process picture "p_serverinfo" is displayed in the data window.

8. Set this picture as the startup picture. To do this, select "Set as startup screen" from the shortcut menu.

9. Click the "Activate" button in the toolbar to activate the project.

Communication - Diagnostics

17

17.1 Diagnosis of Channels and Tags

This section describes the diagnosis of channels and their tags, as well as of internal tags. This diagnosis can be used for example in the event of communication problems or unexpected tag values.

See also

Diagnostic Options for the "SIMATIC S5 PROFIBUS FDL" Channel (Page 395)

Quality Codes of Tags (Page 414)

Monitoring Tag Status Using Global Actions (Page 421)

Using the Tag Status to Monitor Connection Status (Page 420)

How to Check an Internal Tag (Page 422)

Channel diagnosis (Page 371)

General Information about Error Detection (Page 370)

Possibilities for Diagnosing the "OPC" Channel (Page 404)

"SIMATIC S7 Protocol Suite" Channel - Diagnostic Options (Page 384)

"System Info" Channel - Diagnostic Options (Page 379)

17.2 General Information about Error Detection

A fault or error in establishing a communication link is generally first detected in Runtime.

Objects dynamized using WinCC tags, which cannot be supplied with current process values, are displayed in the process picture as inactive. These could be e.g. I/O fields, slider objects or bar graphs.

If the fault does not affect some of a connection's WinCC tags, this indicates that one of the WinCC tags is the source of the trouble. In this case, you should for example check the addressing of the tags as well as their spelling when used in Graphics Designer.

If the fault affects all of a connection's WinCC tags, this indicates a fault in the connection itself.

The following sections describe which measures and means can be used to pinpoint the source of the error.

17.3 Channel Diagnosis

17.3.1 Channel diagnosis

The following functions are available to support the diagnosis of channels and their connections:

- "Status - Logical Connections" Function
- WinCC "Channel Diagnosis"

See also

How to Use the "Status - Logical Connections" Function to Check a Channel (Page 371)

"Status - Logical Connections" Function (Page 371)

Principle of Channel Diagnosis (Page 373)

17.3.2 "Status - Logical Connections" Function

With the "Status - Logical Connections" function, WinCC Explorer offers a simple way to display the current status of all configured connections. However, the status can only be displayed in Runtime.

The function is started in WinCC Explorer via the "Tools" menu.

Note

Detailed data regarding the connection status can be obtained with "WinCC Channel Diagnosis".

17.3.3 How to Use the "Status - Logical Connections" Function to Check a Channel

Introduction

This section describes how to use the "Status - Logical Connections" function to display the current status of all configured connections.

Requirements

- Configure a channel and create a tag in this channel.

Procedure

1. Click the appropriate button in the WinCC Explorer toolbar to activate Runtime.

2. In the WinCC Explorer "Tools", select option "Status of Driver Connections". The "Status - Logical Connections" dialog opens.
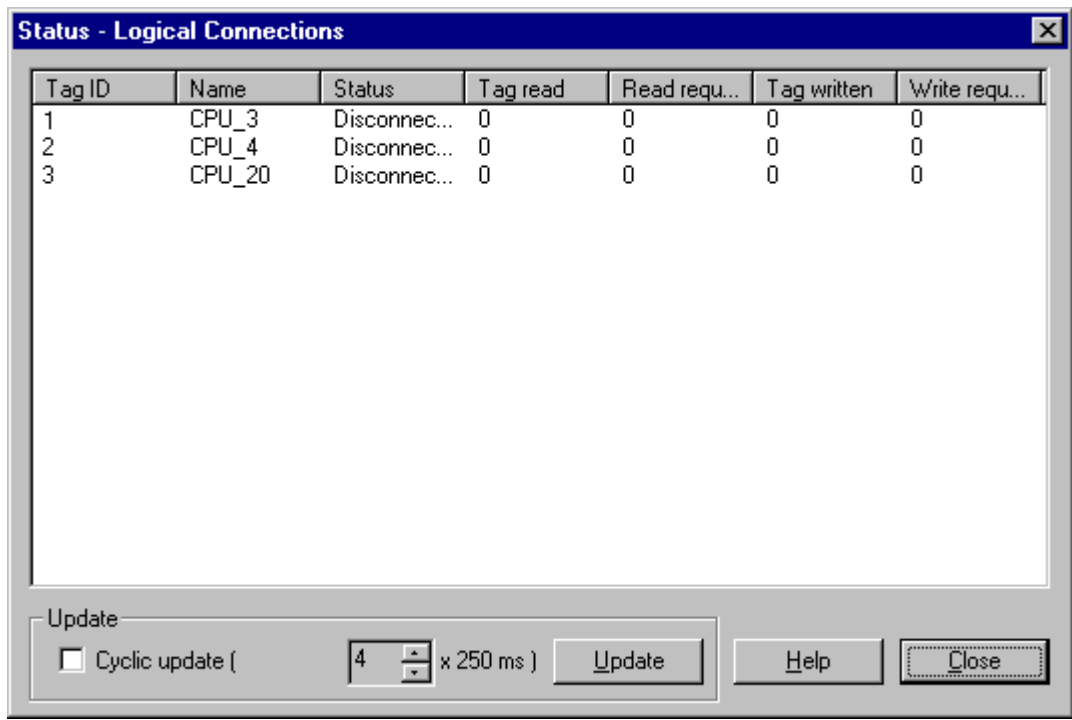
3. In the "Update" area, you can specify that an update is to be done cyclically. Otherwise, you can start an update of the display manually by clicking the "Update" button.

4. The configured connections are displayed in the "Name" column. The "Status" column displays the status of the respective connection.

5. Check the entries in the "Status" column. The "Disconnected" status indicates either a configuration or hardware error.
   Further information can be found under "Channel Diagnosis" for the associated channel.

17.3.4   Diagnosis of Channels with Channel Diagnosis

17.3.4.1   Principle of Channel Diagnosis

Introduction

The WinCC Channel Diagnosis enables WinCC users to gain a quick overview of the status of the active connections in Runtime. On the one hand, "Channel Diagnosis" provides status and diagnostic information regarding channel units and, on the other hand, it serves as a user interface for the configuration of the diagnostic output:

● The output of statistics or information on the status of the communication e.g. in a process picture

● Text output in a logbook file for fault analysis and correction by Service.

● Text output in a trace file to assist the Hotline in pinpointing the cause of communication problems.

The Diagnostic module can be inserted into a process picture as an ActiveX control or be started as an independent application from the Windows Start menu. The module only displays status information for channels that support diagnosis.

The diagnosis of a channel's tag can be found in the description of the channel-specific diagnosis.

Logbook file

Channel Diagnosis creates a logbook file named <Channel_name.log> for every configured WinCC channel. The system records important information and errors in these. The exact text content depends on the channel.

The creation of the file and the output text cannot be configured.

The logbook files contain information such as the start and end messages, version information and information regarding communication errors.

Each entry in the file consists of a data and time stamp, the flag name and a description. The file is always saved immediately after an entry to ensure that e.g., in the event of a voltage drop, all of the information possible is available.

Trace File

A trace file named <Channel_name.trc> can be created to hold error messages and any additional information for every configured WinCC channel. You can select in Runtime whether to use a trace file. When the trace function is activated, a message will be displayed warning that the link's propagation time is affected.

Each entry in a trace file has a time stamp followed by a flag name and description.

When the trace function is enabled all information recorded in the logbook is also written to the trace file.

The information recorded in a trace file is intended to assist the Hotline in pinpointing the cause of communication problems.

Note

The trace and logbook file entries are only recorded in English.

Both files are saved in the "Diagnostics" directory in the WinCC directory structure.

The current counter values are not recorded in these files.

See also

Channel Diagnosis with ActiveX Control (Page 374)

How to Check a Channel with Channel Diagnosis as an ActiveX Control (Page 374)

Diagnosing a Channel with "Channel Diagnosis" (Page 375)

How to Check a Channel with Channel Diagnosis (Page 376)

How to Configure the Trace Function of a Channel (Page 377)

How to Start Runtime (Page 378)

17.3.4.2   Channel Diagnosis with ActiveX Control

Introduction

The status information for a channel can also be displayed in a process picture by the "WinCC Channel Diagnosis Control" ActiveX control.

The ActiveX control is found in the "Controls" object palette in Graphics Designer and is simply inserted in a picture. The user can thus create e.g. a diagnostics process picture, in which he can view the status of the communication and other information in Runtime, without needing to reconfigure this arrangement every time.

17.3.4.3   How to Check a Channel with Channel Diagnosis as an ActiveX Control

Introduction

This section shows how to configure the diagnosis of a channel using the "WinCC Channel Diagnosis Control" ActiveX control.

Requirements

● Configure a channel and create a tag in this channel.

Procedure

1. Start Graphics Designer and open a picture.

2. Insert the "WinCC Channel Diagnosis Control" ActiveX control into the picture. This is accomplished by selecting the ActiveX control from the "Controls" object palette, inserting it into the picture and dragging it to the desired size.

3. Save the picture.

4. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

5. Select the picture into which you have inserted the ActiveX control. The status information for the channels appears in the "Channel Diagnosis" application window on the "Channels/ Connection" tab.

6. Click the "Configuration" tab. Select one of the displayed channels and configure which error messages are to be recorded in the associated log file.

7. More information about the activation of the Trace function can be found under "Configuring a Channel's Trace Function".

Note

Channel Diagnosis only displays status information for channels that support channel diagnosis.

See also

How to Configure the Trace Function of a Channel (Page 377)

17.3.4.4   Diagnosing a Channel with "Channel Diagnosis"

Introduction

Independent of WinCC, Channel Diagnosis can also be started as an application from the Windows Start menu.

Channel Diagnosis is thus always available and not dependent on the selection of a process picture, as is the case with WinCC Channel Diagnosis Control. The status information is only displayed by Channel Diagnosis when WinCC is in Runtime.

17.3.4.5   How to Check a Channel with Channel Diagnosis

Introduction

This section describes how to start Channel Diagnosis as an application from the Windows Start menu.

Requirements

- Configure a channel and create a tag in this channel.

Procedure

1. In the Windows Start menu, select the entry "Channel Diagnosis" in the "Simatic/WinCC/ Tools" group.

2. The Channel Diagnosis application window opens. If no WinCC project is currently in Runtime, message "No connection to WinCC" is displayed.

3. Click the appropriate button in the WinCC Explorer toolbar to activate Runtime.

4. Select the picture into which you have inserted the ActiveX control. The status information for the channels appears in the Channel Diagnosis application window on the "Channels/ Connection" tab.

5. Click the "Configuration" tab. Select one of the displayed channels and configure which error messages are to be recorded in the associated log file.

6. More information about the activation of the Trace function can be found under "Configuring a Channel's Trace Function".

Note

Channel Diagnosis only displays status information for channels that support channel diagnosis.

See also

17.3.4.6   How to Configure the Trace Function of a Channel

Introduction

This section describes how to configure and activate a Trace function of a channel in Runtime. Additional information regarding errors and the status of the communication is recorded in the Trace file.

Note

The information recorded in a trace file is intended to assist the Hotline in pinpointing the cause of communication problems. For this reason, the evaluation of the data in the file is not further described here.

Requirements

- Configure a channel plus a connection and a tag.
- Activate the WinCC project.

Standard Flags - Overview

Flag	Description
Fatal Error	Serious error (requires user action)
Error	Error (frame error, etc.)
Warning	Warning (checksum error, etc.)
Information	Information (function call, etc.)
Success	Successful execution (completion of function call, etc.)
Check User Flags	Enables the "User Flags" check boxes

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.

2. On the "Configuration" tab, select the desired channel.

3. Under "Flags", activate the status and error messages to be recorded in the Trace file. A description of the standard flags can be found in the "Default Flags - Overview" table.

4. Select the "Check User Flags" check box, if the "User Flags" are to be recorded in the Trace file. The number and significance of the "User Flags" is depending on the channel.

5. Select the check boxes of the desired "User Flags". By clicking the "Set" or "Reset" buttons, you can set or reset all "User Flags".

6. In the "Trace File" section, select the "Enable" check box. This activates the other fields in this area.

7. In the "max. Files" field, enter the maximum number of Trace files.

8. In the "max. Size" field, set the size of the individual Trace files.

9. Activate the "Overwrite" field, if the channel's existing Trace files are to be overwritten - beginning with the oldest - after the maximum number of files and file size has been reached.

10. Click "Save" to save the settings and activate the changes.

17.3.4.7 How to Start Runtime

Requirements

A startup picture must be defined before Runtime is activated.

Procedure

1. Save and close all files that may be open in an editor.

2. Select WinCC Explorer.

3. Activate the project by clicking the "Activate" button in the toolbar or by selecting "Activate" in the "File" menu.

See also

How to Check a Channel with Channel Diagnosis (Page 376)

17.4 Diagnosis of "System Info" Channel

17.4.1 "System Info" Channel - Diagnostic Options

The following options for the diagnosis of the "System Info" channel or one of its tags are available:

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

How to Check a Tag (Page 382)

How to Check the Channel and the Connection (Page 380)

17.4.2 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. These entries can be used to analyze a communication problem.

Each entry in the file has a date and time stamp followed by a flag name and description.

Example of a logbook entry:

2000-03-10 12:00:21,050 INFO Log starting ...

2000-03-10 12:00:21,050 INFO | LogFileName : C:\Siemens\WinCC\Diagnose\SYSTEM_INFO_01.LOG

2000-03-10 12:00:21,050 INFO | LogFileCount : 3

2000-03-10 12:00:21,050 INFO | LogFileSize : 1400000

2000-03-10 12:00:21,050 INFO | TraceFlags : fa000001

2000-03-10 12:00:21,050 INFO start timer

2000-03-10 12:00:21,360 ERROR Illegal tag type! tag: "Format_0" correct type: "Text Tag 8-Bit Character Set"!

Entries for "INFO" Flag

Message text	Description
Log starting ...	Start message
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	Name of the log file with path
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : fa000001	Flags used by the channel in hexadecimal format
start timer	Start message

Entries for "ERROR" Flag

Message text	Description
Illegal tag type! tag: "tag" correct type: "data type"!	Incorrect data type of a tag tag= Name of tag with incorrect data type data type = Correct data type

17.4.3 Determining the Cause of Incorrect Tag Values

17.4.3.1 How to Determine the Cause of Incorrect Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Check the channel and connections

2. Check the tags of the channel

See also

How to Check a Tag (Page 382)

How to Check the Channel and the Connection (Page 380)

17.4.3.2 How to Check the Channel and the Connection

Introduction

This section describes how to check the "System Info" channel and its connection in Runtime.
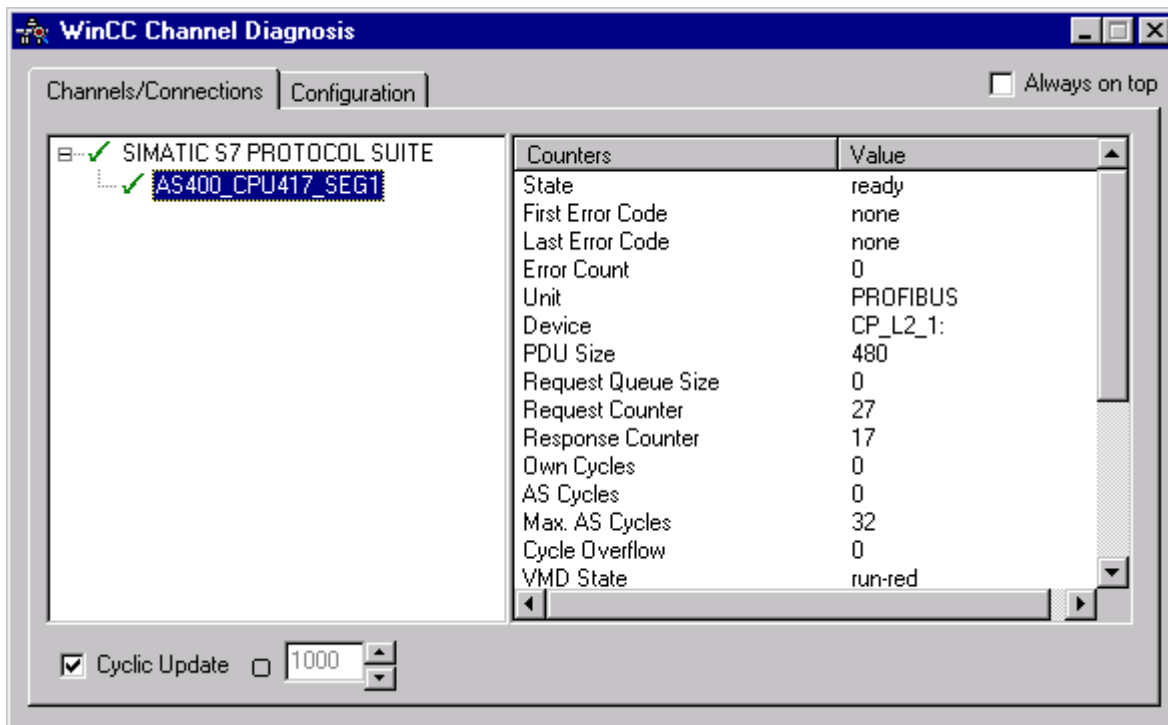
Requirements

- Configure a connection and tag for the "System Info" channel.
- Activate the WinCC project.

Overview of Status Messages

Icon	Description
✔	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
🔧	Channel / no statement possible regarding connection status
✖	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.

2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.

3. Check the icons in front of the channel named "System Info" and its connection. If the status of the channel and connection are OK, a green check mark is displayed in front of each respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.

4. If there is no green check mark in front of the channel's name and the connection, select the connection in the window on the left. In the window on the right, check the counter values for "Address Error", Size Error" and "Type Error". These values indicate the errors detected.

5. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, refer to "Description of Log File Entries".

6. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
   For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 377)

Description of Log File Entries (Page 379)

How to Check a Tag (Page 382)

17.4.3.3   How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- Configure a connection and tag for the "System Info" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in the tag management, select the "System Info" channel.

2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the data window.

3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.

4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag Quality Codes".

5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.

6. Check whether values have been configured for the upper or lower limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.

7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 414)

17.5 Diagnosis of the "SIMATIC S7 Protocol Suite" Channel

17.5.1 "SIMATIC S7 Protocol Suite" Channel - Diagnostic Options

The following options for the detection of errors and the diagnosis of the "SIMATIC S7 Protocol Suite" channel or one of its tags are available:

Checking the Communication Processor Configuration

Besides checking the access point, the communication processor can be tested with the "Set PG/PC Interface" application. The communication processor can be checked under SIMATIC NET in the same way.

Checking the Configuration of the Connection and Tags

There may be errors in the configuration of the system and connection parameters. Invalid tag values may also result from improperly addressing the tag in the AS.

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

How to Check a Tag (Page 393)

How to Check the Channel and the Connection (Page 391)

How to Check the Configuration of the Connection and Tags (Page 390)

Checking the Communication Processor under SIMATIC NET (Page 389)

How to Check the Configuration of the Communication Processor (Page 387)

17.5.2 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. These entries can be used to analyze a communication problem.

Each entry in the file has a date and time stamp followed by a flag name and description.

Example of a logbook entry:

1999-04-01 12:00:24,524 INFO Log starting ...

1999-04-01 12:00:24,524 INFO LogFileName : C:\Siemens\WinCC\Diagnose\SIMATIC_S7_Protocol_Suite_01.LOG

1999-04-01 12:00:24,524 INFO LogFileCount : 3

1999-04-01 12:00:24,524 INFO LogFileSize : 1400000

1999-04-01 12:00:24,524 INFO TraceFlags : c4000000

1999-04-01 12:00:24,524 INFO S7 channel DLL started!

1999-04-01 12:00:26,096 ERROR Illegal tag address "nCPU3_1"!

1999-04-01 12:00:27,428 INFO S7DOS release: @(#)TIS-Block Library DLL Version C5.0.17.3-REL5,0,17,47,3-BASIS

1999-04-01 12:00:27,428 INFO S7DOS version: V5.0 / 0

1999-04-01 12:00:27,428 INFO S7CHN version: V5.0 / Mar 1 1999 / 22:36:40

1999-04-01 12:00:27,428 INFO S7 channel unit "Industrial Ethernet" activated!

1999-04-01 12:00:27,468 ERROR Cannot connect to "CPU_4": Errorcode 0xFFDF 42C2!

1999-04-01 12:00:27,538 INFO S7 channel unit "MPI" activated!

Description of the Most Important Entries for the "INFO" Flag

Message text	Description
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	Name of the log file with path
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : c4000000	Displays the flags used by the Trace function as a hexadecimal number
S7 channel DLL started!	Start message
S7 channel DLL terminated!	End message
S7 channel unit "unitname" activated!	Channel unit activated
S7 channel unit "unitname" deactivated!	Channel unit deactivated
S7DOS version: versionsstring	Version information
S7CHN version: versionsstring	Version information

Description of the Most Important Entries for the "ERROR" Flag

Message text	Description
Cannot connect to "connectionname": Errorcode 0xhhhh ffff!	Communication error Could not establish a connection to the AS immediately after activating WinCC. If the connection could be established without error at least once, the following message is output in the event of a later error. nnn = Number of disconnects for this connection connectionname = Name of the connection hhh = 1st error code in hex S7DOS / SAPI-S7 ffff = 2nd error code in hex S7DOS / SAPI-S7
Cannot connect to "connectionname": Errorcode 0xhhhh ffff!	Communication error Could not establish a connection to the AS immediately after activating WinCC. The connection was established at least once without error.
Channel API error: errorstring	Channel API error The channel passed the error string 'errorstring' to WinCC Explorer. Depending on the significance of the error, the error string may or may not be displayed in a notice box. For a description of the error strings, please see the API Error Text.
Max. count of API errors reached - API logbook deactivated	Channel API error Depending on the error and function, errors can occur cyclically on the API. To avoid filling the logbook file with these error messages, a maximum of 32 messages are output for an API error.
Cannot write storage data! Cannot read storage data / use default data Storage data illegal or destroyed / use default data! No storage data / use default data!	General Channel Error Messages
Devicename in unit "unitname" changed from "old devicename" to "new devicename"	Initialization message
Max. logbooksize reached - Logbook deactivated	Message sent when log file has exceeded its maximum length. The logbook output is monitored for length. If the specified length is reached, the logbook is deactivated. The message is only output, when message output causes the max. file length to be exceeded. No message be output, if the file length is changed with an editor or the maximum file length is reduced in the INI file!

17.5.3 Determining the Cause of Incorrect Tag Values

17.5.3.1 How to Determine the Cause of Incorrect Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Checking the Configuration of the Communication Processor
2. Checking the Communication Processor under SIMATIC NET
3. Checking the Configuration of the Connection and Tags
4. Check the channel and connections
5. Check the tags of the channel

See also

How to Check a Tag (Page 393)

How to Check the Channel and the Connection (Page 391)

How to Check the Configuration of the Connection and Tags (Page 390)

Checking the Communication Processor under SIMATIC NET (Page 389)

How to Check the Configuration of the Communication Processor (Page 387)

17.5.3.2 How to Check the Configuration of the Communication Processor

Introduction

This section describes how to use the "PG/PC Port" program to check a communication processor. In this example, a "CP 5613" type processor is used for the PROFIBUS communication.

Requirements

- Install the CP 5613.
- Install the associated communication driver.
- Configure the CP 5613.

Procedure

1. On the Control Panel, click the "Set PG/PC Port" icon. The "Set PG/PC Port" dialog is opened.

2. Check the entry for the access point. The access point "CP_L2_1:" for Profibus connection is automatically added when a CP 5613 is installed. Select the entry for this access point. Click "Properties" to open the "Properties - CP5613_5614(PROFIBUS)" dialog.

3. Check the entry on the "PROFIBUS" tab.

4. Click the "Operational State" tab. Click the "Test" button to execute a function test on the CP 5613. The test results is shown in the output field below. Depending on the test results, you can click the "Restart" button to perform a reset and a complete restart of the CP 5613.

5. Click "OK" to close all open dialogs.

See also

How to Check the Configuration of the Connection and Tags (Page 390)

Checking the Communication Processor under SIMATIC NET (Page 389)

17.5.3.3   Checking the Communication Processor under SIMATIC NET

Introduction

This section explains how to check a communication processor using the "Set PC Station" program in the SIMATIC NET software.

In this example, the "CP 5613" for the PROFIBUS communication is used to the "SIMATIC S7 Protocol Suite" channel.

Requirements

- Install the CP 5613.

- Install the SIMATIC NET software. For Windows 2000, a version from V6.0 SP5 is required, with Windows XP a version from V6.1 must be installed.

- Configure the CP 5613 under SIMATIC NET.

Procedure

1. Activate the Windows Start menu, open "SIMATIC / SIMATIC NET / Settings" and select menu item "Set PC Station". The "Configurations Console PC Station" dialog opens.

2. Check the entry for the access point. Select the "Access Point" directory in the navigation window. The existing access points are listed in the data window. Upon installation of the CP 5613, access point "CP_L2_1:" for PROFIBUS communication is added automatically. Select this access point in the data window. Use the "Properties" menu item from the shortcut menu to open the "Properties of CP_L2_1:" dialog.

3. Check the entry in the "Assigned Interface Parameters" field. In the case of a CP5613 in a PROFIBUS network, select entry "CP5613_5614(PROFIBUS)".

4. Open the navigation window, select the "Components" directory and then the "CP5613/ CP5614" subdirectory.

5. Select the "Network Diagnosis" directory. Click the "Test" button to execute a function test on the CP 5613. The result is displayed in the output window. Depending on the test results, click the "Restart" button in the "General" directory to perform a reset and then a complete restart of the CP 5613.

6. Check the list of participants connected to PROFIBUS in the list in the "Bus Participants" directory. Based on the display, it is possible to determine the function and status of your own station as well as other participants which are connected.

7. Close the dialog.

8. If a fault is detected in the configuration of the communication processor, modifications can only be made to the configuration using SIMATIC NET tools. Further information is available under SIMATIC NET.

See also

How to Check the Configuration of the Connection and Tags (Page 390)

## 17.5.3.4 How to Check the Configuration of the Connection and Tags

Introduction

This section describes how to check the system parameters and the configuration of the connection and tags. In this example, a "CP 5613" communication processor is used for the PROFIBUS communication.
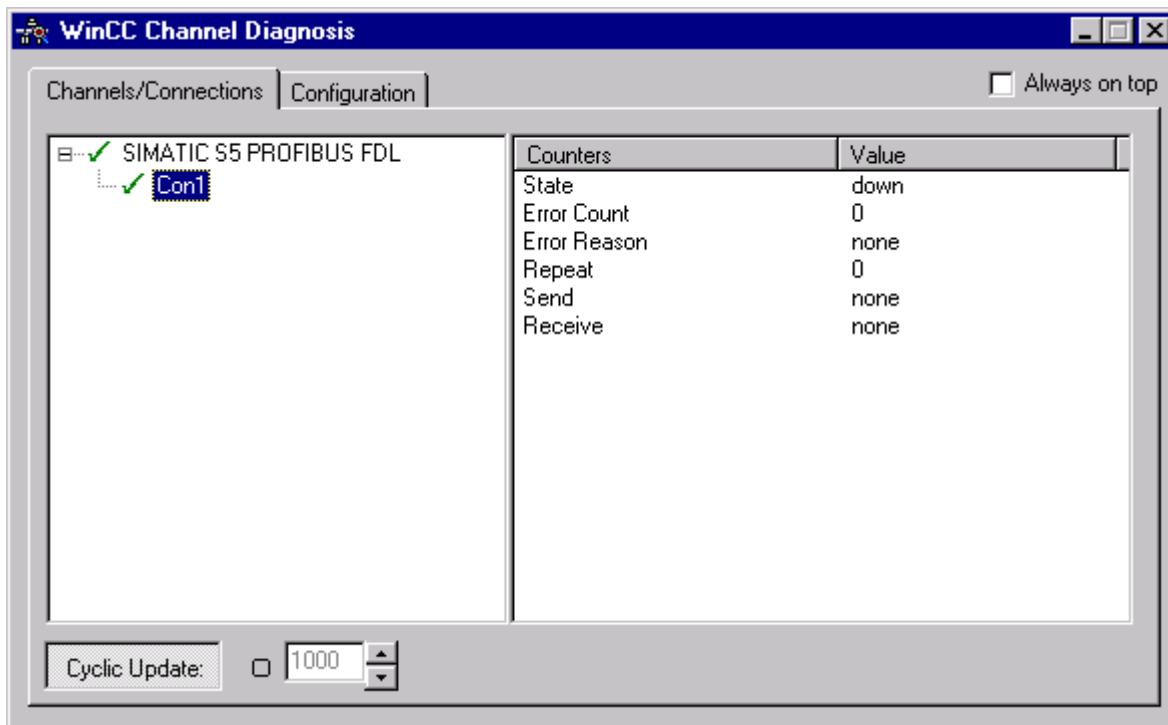
Requirements

- Install the CP 5613.

- Install the associated communication driver.

- Configure the CP 5613.

- Configure a connection and tag for the "SIMATIC S7 Protocol Suite" channel.

- Activate the WinCC project.

Procedure

1. In WinCC Explorer in the tag management, select the "SIMATIC S7 Protocol Suite" channel. In the data window, select the "PROFIBUS" channel unit. In the channel unit shortcut menu, click "System Parameters". The "System Parameters - PROFIBUS" dialog opens.

2. On the "Unit" tab, check the entry in the "Logical Device Name" field. By default, this is set to the access point "CP_L2_1:" . The access point is assigned during installation of the communication processor in the CP 5613 system. Close the dialog.
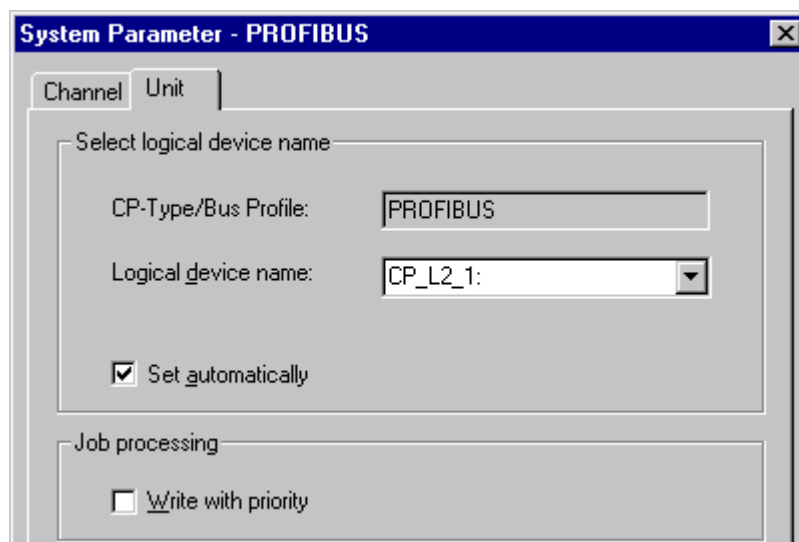
3. In the tag management navigation window, select the "PROFIBUS" channel unit. In the data window, select the connection to be checked. In the shortcut menu, click "Properties" to open the "Connection Properties" dialog.

4. Click the "Properties" button to open the "Connection Parameters - PROFIBUS" dialog.

5. Check the settings on the "Connection" tab. Close the open dialogs.

6. In the navigation window, select the checked connection. In the data window, select the tag to be checked. In the shortcut menu, click "Properties" to open the "Tag Properties" dialog. Check the values in the "Type Conversion" and "Data Type" fields.

7. Click the "Select" button to open the "Address properties" dialog. Check the settings for addressing the tag in the AS.

8. Click "OK" to close all open dialogs.

See also

How to Check the Channel and the Connection (Page 391)

17.5.3.5   How to Check the Channel and the Connection

Introduction

This section describes how to check the "SIMATIC S7 Protocol Suite" channel and its connection in Runtime.

Requirements

- Install a communication processor in a WinCC computer, for example the CP 5613 for PROFIBUS communication.

- Install the associated communication driver.

- Configure the CP 5613.

- Create a STEP7 project.

- Configure a connection and tag for the "SIMATIC S7 Protocol Suite" channel.

- Activate the WinCC project.

Overview of Status Messages

Icon	Description
✔	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
?	Channel / no statement possible regarding connection status
✘	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.

2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.

3. Check the icon in front of the channel named "SIMATIC S7 Protocol Suite" and the connection. If the channel and connection are OK, a green check mark is displayed in front of each respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.

4. If there is no green check mark in front of the channel's name and the connection, select the connection in the window on the left. In the window on the right, check the entries for the counters "First Error Code" and "Last Error Code". These values indicate the errors detected. You can access Direct Help by clicking the shortcut menu of the value display.

5. On the "Configuration" tab, select the status and error messages, which are to be output to the log file. This is done by selecting "SIMATIC S7 Protocol Suite" and configuring the error display. For more information, please refer to "Configuring the Log File of a Channel".

6. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, please see "Description of Log File Entries".

7. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 377)

Description of Log File Entries (Page 384)

How to Check a Tag (Page 393)

17.5.3.6   How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

In a "SIMATIC S7 Protocol Suite" channel, you can also use connection-specific internal tags. This procedure can also be used to check these tags.

Requirements

- Install a communication module on a WinCC computer, for example the CP 5613 for MPI communication.
- Install the associated communication driver.
- Configure the CP 5613.
- Create a STEP7 project.
- Configure a connection and tag for the "SIMATIC S7 Protocol Suite" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in the tag management, select the "SIMATIC S7 Protocol Suite" channel.

2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the data window.

3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.

4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag Quality Codes".

5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.

6. Check whether values have been configured for the upper or lower limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.

7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

In Runtime, the current values of the connection-specific internal tags can be viewed in detail with "WinCC Channel Diagnosis". When the main connection is selected, the tags is shown in the "Counter" column.

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 414)

17.6 Diagnosis of the "SIMATIC S5 Profibus FDL" Channel

17.6.1 Diagnostic Options for the "SIMATIC S5 PROFIBUS FDL" Channel

The following options for the detection of errors and the diagnosis of the "SIMATIC S5 PROFIBUS FDL" channel or one of its tags:

Checking the Communication Processor Configuration

Besides checking the access point, the communication processor can be tested with the "Set PG/PC Interface" application. The communication processor can be checked under SIMATIC NET in the same way.

Checking the Configuration of the Connection and Tags

There may be errors in the configuration of the system and connection parameters. Invalid tag values may also result from improperly addressing the tag in the AS.

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

17.6.2 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. These entries can be used to analyze a communications problem.

Each entry in the file has a date and time stamp followed by a flag name and description.

Example of a logbook entry:

2000-05-03 14:43:48,733 INFO Log starting ...

2000-05-03 14:43:48,733 INFO | LogFileName : d:\Siemens\WinCC\Diagnose \SIMATIC_S5_PROFIBUS_FDL_01.LOG

2000-05-03 14:43:48,733 INFO | LogFileCount : 3

2000-05-03 14:43:48,733 INFO | LogFileSize : 1400000

2000-05-03 14:43:48,733 INFO | TraceFlags : fa017fff

Description of the "INFO" Flag Entries

Message text	Description
Log starting ...	Start message
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	Name of the log file with path
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : fa017fff	Displays the flags used by the Trace function as a hexadecimal number

17.6.3 Determining the Cause of Incorrect Tag Values

17.6.3.1 How to Determine the Cause of Incorrect Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Check the configuration of the communication processor

2. Check the communication processor under SIMATIC NET

3. Check the configuration of the connection and tags

4. Check the channel and connections

5. Check the tags of the channel

See also

How to Check the Configuration of the Connection and Tags (Page 399)

How to Check a Tag (Page 403)

How to Check the Channel and the Connection (Page 401)

Checking the Communication Processor under SIMATIC NET (Page 398)

How to Check the Configuration of the Communication Processor (Page 397)

17.6.3.2   How to Check the Configuration of the Communication Processor

Introduction

This section describes how to use the "PG/PC Port" program to check a communication processor. In this example, a "CP 5613" type processor is used for the PROFIBUS communication.

Requirements

- Install the CP 5613.
- Install the associated communication driver.
- Configure the CP 5613.

Procedure

1. On the Control Panel, click the "Set PG/PC Port" icon. The "Set PG/PC Port" dialog is opened.

2. Check the entry for the access point. The access point "CP_L2_1:/SCP" is automatically added for the PROFIBUS connection when a CP 5613 is installed. Select the entry for this access point. Click "Properties" to open the "Properties - CP5613_5614(PROFIBUS)" dialog.

3. Check the entry on the "PROFIBUS" tab.

4. Click the "Operational State" tab. Click the "Test" button to execute a function test on the CP 5613. The test results is shown in the output field below. Depending on the test results, you can click the "Restart" button to perform a reset and a complete restart of the CP 5613.

5. Click "OK" to close all open dialogs.

See also

How to Check the Configuration of the Connection and Tags (Page 399)

Checking the Communication Processor under SIMATIC NET (Page 398)

17.6.3.3   Checking the Communication Processor under SIMATIC NET

Introduction

This section explains how to check a communication processor using the "Set PC Station" program in the SIMATIC NET software.

In this example, the "CP 5613" type is used for the PROFIBUS communication to channel "SIMATIC S5 PROFIBUS FDL".

Requirements

* Install the CP 5613.

* Install the SIMATIC NET software. For Windows 2000, a version from V6.0 SP5 is required, with Windows XP a version from V6.1 must be installed.

* Configure the CP 5613 under SIMATIC NET.

Procedure

1. Activate the Windows Start menu, open "SIMATIC / SIMATIC NET / Settings" and select menu item "Set PC Station". The "Configurations Console PC Station" dialog opens.

2. Check the entry for the access point. Select the "Access Point" directory in the navigation window. The existing access points are listed in the data window. Upon installation of the CP 5613, access point "CP_L2_1:" for PROFIBUS communication is added automatically. Select this access point in the data window. Use the "Properties" menu item from the shortcut menu to open the "Properties of CP_L2_1:" dialog.

3. Check the entry in the "Assigned Interface Parameters" field. In the case of a CP5613 in a PROFIBUS network, select entry "CP5613_5614(PROFIBUS)".

4. Open the navigation window, select the "Components" directory and then the "CP5613/ CP5614" subdirectory.

5. Select the "Network Diagnosis" directory. Click the "Test" button to execute a function test on the CP 5613. The result is displayed in the output window. Depending on the test results, click the "Restart" button in the "General" directory to perform a reset and then a complete restart of the CP 5613.

6. Check the list of participants connected to PROFIBUS in the list in the "Bus Participants" directory. Based on the display, it is possible to determine the function and status of your own station as well as other participants which are connected.

7. Close the dialog.

8. If a fault is detected in the configuration of the communication processor, modifications can only be made to the configuration using SIMATIC NET tools. Further information is available under SIMATIC NET.

See also

How to Check the Configuration of the Connection and Tags (Page 399)

17.6.3.4 How to Check the Configuration of the Connection and Tags

Introduction

This section describes how to check the system parameters and the configuration of the connection and tags.

Requirements

- Install the CP 5613.
- Install the communication driver.
- Configure the CP 5613.
- Configure a connection and tag for the "SIMATIC S5 PROFIBUS FDL" channel.
- Activate the WinCC project.

Procedure

1. Click the plus sign in front of the "SIMATIC S5 PROFIBUS FDL" icon in WinCC Explorer navigation window. In the context menu of the "FDL(CP5412/A2-1)" icon, click "System Parameters". The "System Parameters" dialog opens.
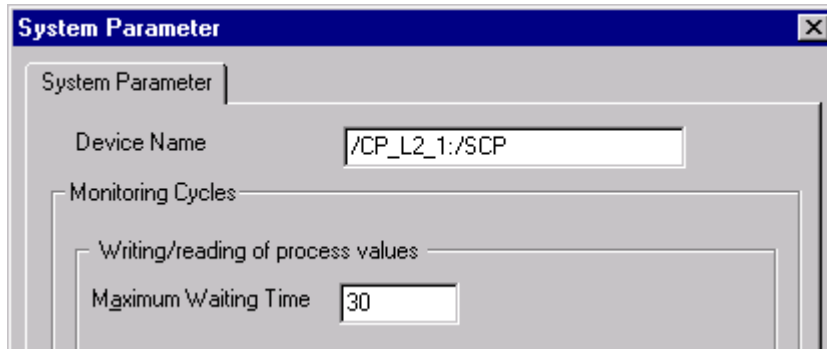
2. Check the entry in the "Device Name" field. By default, this is set to access point "CP_L2_1:/ SCP". The access point is assigned during installation of the communication processor in the CP 5613 system. Close the dialog.

3. Click the plus sign in front of the FDL(CP5412/A2-1)" icon. In the context menu of the tag to be tested, select the "Properties" entry. The "Connection Properties" dialog is opened.

4. In the "Connection Properties" dialog, click the "Properties" button. The "Connection Parameters" dialog opens.
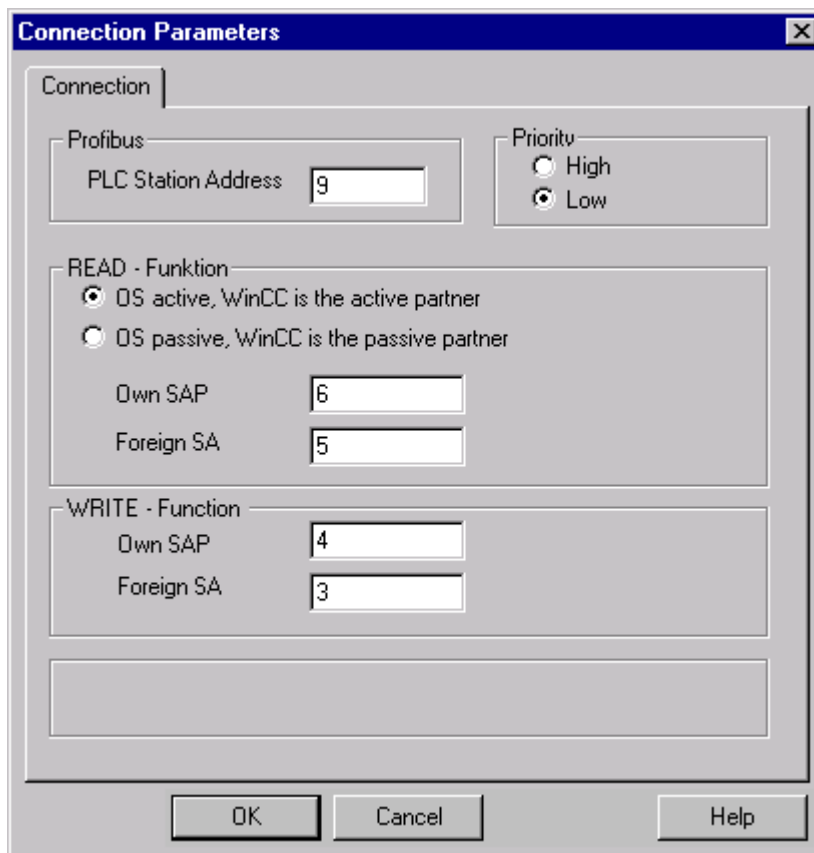
5. Check the settings on the "Connection" tab. Close the open dialogs.

6. Click the plus sign in front of the icon of the connection. In the context menu of the tag to be tested, click the "Properties" entry. The "Tag Properties" dialog opens. Check the entries in the "Type Conversion" and "Data Type" fields.

7. In the "Tag Properties" dialog, click the "Select" button. The "Address Properties" dialog opens. Check the settings.

8. Click "OK" to close all open dialogs.

See also

How to Check the Channel and the Connection (Page 401)

17.6.3.5   How to Check the Channel and the Connection

Introduction

This section describes how to check the "SIMATIC S5 PROFIBUS FDL" channel and its connection in Runtime.

Requirements

- Install the CP 5613.

- Install the communication driver.

- Configure the CP 5613.

- Create a STEP5 project.

- Configure a connection and tag for the "SIMATIC S5 PROFIBUS FDL" channel.

- Activate the WinCC project.

Overview of Status Messages

Icon	Description
✔	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
?	Channel / no statement possible regarding connection status
✘	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.

2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.

3. Check the icon in front of the channel name "SIMATIC S5 PROFIBUS FDL" and the connection. If the status of the channel and connection are OK, a green check mark is displayed in front of each respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.

4. If there is no green check mark in front of the channel's name and the connection, select the connection in the window on the left. In the window on the right, check the entries for the counters "State", "Error Count", "Error Reason", "Send" and "Receive". These values indicate the errors detected.

5. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, please see "Description of Log File Entries".

6. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 377)

Description of Log File Entries (Page 395)

How to Check a Tag (Page 403)

17.6.3.6 How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- Install the CP 5613.
- Install the communication driver.
- Configure the CP 5613.
- Create a STEP5 project.
- Configure a connection and tag for the "SIMATIC S5 PROFIBUS FDL" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in tag management, select the "SIMATIC S5 PROFIBUS FDL" channel.

2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the data window.

3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.

4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag Quality Codes".

5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.

6. Check whether values have been configured for the upper or lower limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.

7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 414)

17.7 Diagnosis of the "OPC" Channel

17.7.1 Possibilities for Diagnosing the "OPC" Channel

There are the following possibilities for detecting errors and diagnosing the "OPC" channel or one of its tags:

Checking the Configuration of the Connection and Tags

There may be errors in the configuration of the system and connection parameters. Invalid tag values may also result from improperly addressing the tag in the AS.

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

17.7.2 Description of Log File Entries

17.7.2.1 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. The following sections cover only the most important entries. These entries can be used to analyze a communications problem.

A distinction must be made between two types of entries:

- INFO
- ERROR

Structure of an Entry

Date/Time Stamp	Flag Name	Description

Examples of entries in a logbook

2000-03-24 10:43:18,756 INFO Log starting ...

2000-03-24 10:43:18,756 INFO | LogFileName : C:\Siemens\WinCC\Diagnose\OPC.LOG

2000-03-24 10:43:18,756 INFO | LogFileCount : 3

2000-03-24 10:43:18,756 INFO | LogFileSize : 1400000

2000-03-24 10:43:18,756 INFO | TraceFlags : fa000007

000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC

2000-03-23 10:46:18,756 INFO Process detached at 2000-03-2410:46:18,746UTC

2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba

2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit #1)]![OPC_No_Machine]: CoCreateInstance for server "OPCServer.WinCC" on machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba - RPC_S_SERVER_UNAVAILABLE (Der RPC-Server ist nicht verfügbar.))")

See also

Entries for "ERROR" Flag (Page 406)

Entries for "INFO" Flag (Page 405)

17.7.2.2   Entries for "INFO" Flag

Introduction

Each entry in the file has a date and time stamp followed by a flag name and description.

Date/Time Stamp	Flag Name	Description

Examples of entries in a logbook

2000-03-24 10:43:18,756 INFO Log starting ...

2000-03-24 10:43:18,756 INFO | LogFileName : C:\Siemens\WinCC\Diagnose\OPC.LOG

2000-03-24 10:43:18,756 INFO | LogFileCount : 3

2000-03-24 10:43:18,756 INFO | LogFileSize : 1400000

2000-03-24 10:43:18,756 INFO | TraceFlags : fa000007

000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC

2000-03-23 10:46:18,756 INFO Process detached at 2000-03-2410:46:18,746UTC

Description of the Mst Iportant Logbook Entries

Message text	Description
Log starting ...	Start message
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	Name of the log file with path
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : fa000007	Displays the flags used by the Trace function as a hexadecimal number
Process attached at 2000-03-24 09:43:18,746 UTC	The channel was loaded by the WinCC Data Manager.
Process detached at 2000-03-2410:46:18,746 UTC	The channel was unloaded by the WinCC Data Manager.
IOPCChnShutdown::ShutdownRequest was called... Reason: system going down" IOPCChnShutdown::ShutdownRequest	The WinCC OPC Server WinCC project was deactivated. The WinCC OPC clients are requested to disconnect from the WinCC OPC server.

17.7.2.3   Entries for "ERROR" Flag

Introduction

Each entry in the file has a date and time stamp followed by a flag name and description. In the case of the "Error" flag, the description consists of a message text, error code and the text of the error message. Some error codes do not have text for an error message.

Date/Time Stamp	Flag Name	Description Message text + error code + error message text

Examples of entries in a logbook

2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba

2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit #1)]![OPC_No_Machine]: CoCreateInstance for server "OPCServer.WinCC" on machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba - RPC_S_SERVER_UNAVAILABLE (RPC server not available.))")

Description of the Most Important Logbook Entries

Error Code	Error Message Text	Possible Causes
c0040004	Conversion between the "canonicalDatatype" and the "requestedDatatype" is not supported by the server.	Access to the WinCC tag on the OPC server failed. Conversion is possible but failed. The WinCC tag is not on the server or the configured data type does not match.
c0040007	The name does not exist in the name space of the server.	The error code is always returned by the server if the OPC client is accessed with a tag name that does not exist in the name space of the server. Examples: Browse, read tag, write tag, insert tag in a subscription.
00000001	AddItems	Access to the WinCC tag on the OPC server failed. The WinCC tag is not on the server or the configured data type does not match. Data Type WinCC Tag OPC Server = Data Type WinCC Tag OPC Client.
80004005	Could not resolve Server Name	The computer that is used as the WinCC OPC Server is not available in the network. The WinCC OPC server, which was accessed by the "OPC" channel, was not available.
80040154	Class not registered	The WinCC OPC Server is not properly registered in the system. The WinCC OPC Server's WinCC project is not activated.
80070057	Parameter wrong	The WinCC tag is not on the OPC Server, or the configured data type does not match.
800706ba	The RPC Server is not available.	The computer on which the OPC Server is to be started could not be found in the network.

17.7.3   Determining the Cause of Incorrect Tag Values

17.7.3.1   How to Determine the Cause of Invalid Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Check the configuration data
2. Check connection
3. Check the tags of the channel

See also

17.7.3.2 How to Check the Configuration Data

Requirements

- A computer as WinCC OPC Client with a WinCC Project.
- The "OPC" channel must be integrated in the OPC client's WinCC project.
- Configure a WinCC tag in the OPC server's WinCC project.
- On the OPC client configure a connection and a WinCC tag, which communicates with the created server tags.
- Activate the WinCC project on the OPC Server and Client.

Procedure

1. Click the plus sign in front of the "OPC" icon in WinCC Explorer navigation window on the OPC client. Click the "plus sign in front of the icon "OPC Groups (OPCHN Unit#1)".

2. In the shortcut menu of the tag to be tested, select the "Properties" entry. The "Connection Properties" dialog is opened. Click the "OPC Group" tab.

3. In the field "OPC Server Name", check the ProgID of the OPC server.
   In the case of a connection to a server with WinCC V 5.0 or higher, you must enter "OPCServer.WinCC".
   In the case of a server running WinCC V4.x, you must enter "OE.Groups".

4. Enter the name of the computer to serve as the OPC Server in the "Start the server on this computer" field. Click the "Test Server" button to test the connection to the OPC Server. Close the dialog.

5. Click the plus sign in front of the icon of the connection. In the context menu of the tag to be tested, click the "Properties" entry. The "Tag Properties" dialog opens.

6. The same "Data Type" must be entered for this tag as for the tag on the OPC Server.

7. In the "Tag Properties" dialog, click the "Select" button. The "Address Properties" dialog opens.

8. Check the entries in the fields "Item Name" and "Data Type". The "Item Name" must match the tag name on the OPC Server. The "Data Type" must match the data type of the tag on the OPC Server.

9. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, refer to "Description of Log File Entries".

10. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
    For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 377)

Description of Log File Entries (Page 404)

How to Check the Channel and the Connection (Page 409)

17.7.3.3   How to Check the Channel and the Connection

Introduction

This section describes how to check the "OPC" channel and its connection in Runtime.

Requirements

- A computer as WinCC OPC Client with a WinCC Project.
- The "OPC" channel must be integrated in the OPC client's WinCC project.
- Configure a WinCC tag in the OPC server's WinCC project.
- On the OPC client configure a connection and a WinCC tag, which communicates with the created server tags.
- Activate the WinCC project on the OPC Server and Client.

Overview of Status Messages

Icon	Description
✔	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
❓	Channel / no statement possible regarding connection status
✗	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.

2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.

3. Check the icons in front of the OPC connection. If the status of the connection is OK, a green check mark is displayed in front of the respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.

4. If there is no green check mark in front of the name of the connection, select the connection in the window on the left. In the window on the right, check the entries for the counters "AddItemFailures", "Server Status", "Last Error" and "Last Error Name". These values indicate the errors detected.

5. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, refer to "Description of Log File Entries".

6. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 377)

Description of Log File Entries (Page 404)

How to Check a Tag (Page 411)

17.7.3.4 How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- A computer as WinCC OPC Client with a WinCC Project.
- The "OPC" channel must be integrated in the OPC client's WinCC project.
- Configure a WinCC tag in the OPC server's WinCC project.
- On the OPC client configure a connection and a WinCC tag, which communicates with the created server tags.
- Activate the WinCC project on the OPC Server and Client.

Procedure

1. In WinCC Explorer in tag management, select the "OPC" channel.
2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the data window.
3. Move the mouse pointer over the tag to be checked. A tooltip window opens with the current tag value, the quality value and the time of the most recent change.
4. Check the quality code. If the value "C0" is displayed, the tag value is OK. A description of the other values can be found under "Tag Quality Codes".
5. If the quality code is not equal to "C0", select the tag in tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.
6. Check whether values have been configured for the upper or lower limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.
7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 414)

17.8 Quality of Tags

17.8.1 Quality of Tags

Introduction

In WinCC, there are two quality indicators that allow you to evaluate the quality of tags. These two indicators are tag status and quality code.

The tag status is formed in WinCC and informs of the quality of configuration settings within the OS. The tag status informs additionally of the connection status to the WinCC communication partner. This may be an automated system or the server computer.

The quality code contains the same information as the tag status. In addition to this information, the quality status contains quality statements on partners which assess or process tags. Possible partners are:

● Automation systems

● Automation systems with field devices

● OPC server

● OPC server with subordinate automation systems

Therein the quality code is forwarded within the processing chain. If at one point in the processing chain several quality codes are pending for a tag, the worst code is forwarded.

AS:	Automation system (e.g. SIMATIC S7)
QC:	Quality Code of the tag
WinCC:	HMI System WinCC
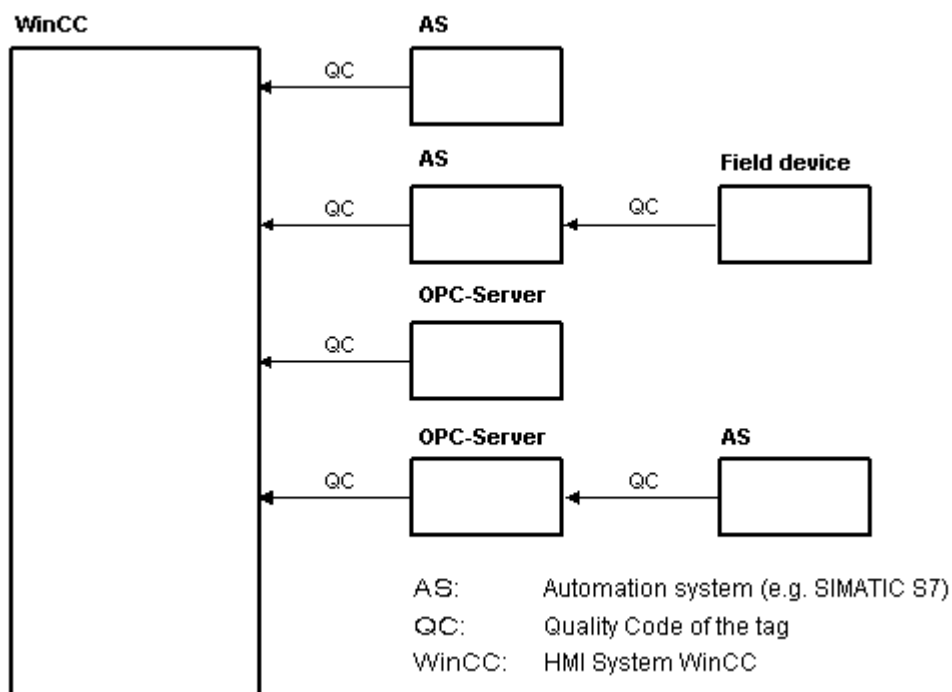
The quality code informs of the quality of a tag independent of where this code was formed.

Cascading of Quality Code

By using the example of an automation system with field device connected, the cascading of quality codes shall be outlined.

The automation system reads the quality codes generated by the field device. Using an analysis logic, quality codes pending for the same tag concurrently are evaluated by priority. The quality code with the worst status is assigned to the tag. This quality code must be saved in a data block directly behind the associated tag value.

You may initiate the analysis logic using the channel modules of the PCS7 Library. If the PCS7 Library is not available to you, you must configure the analysis logic in the automation system yourself.
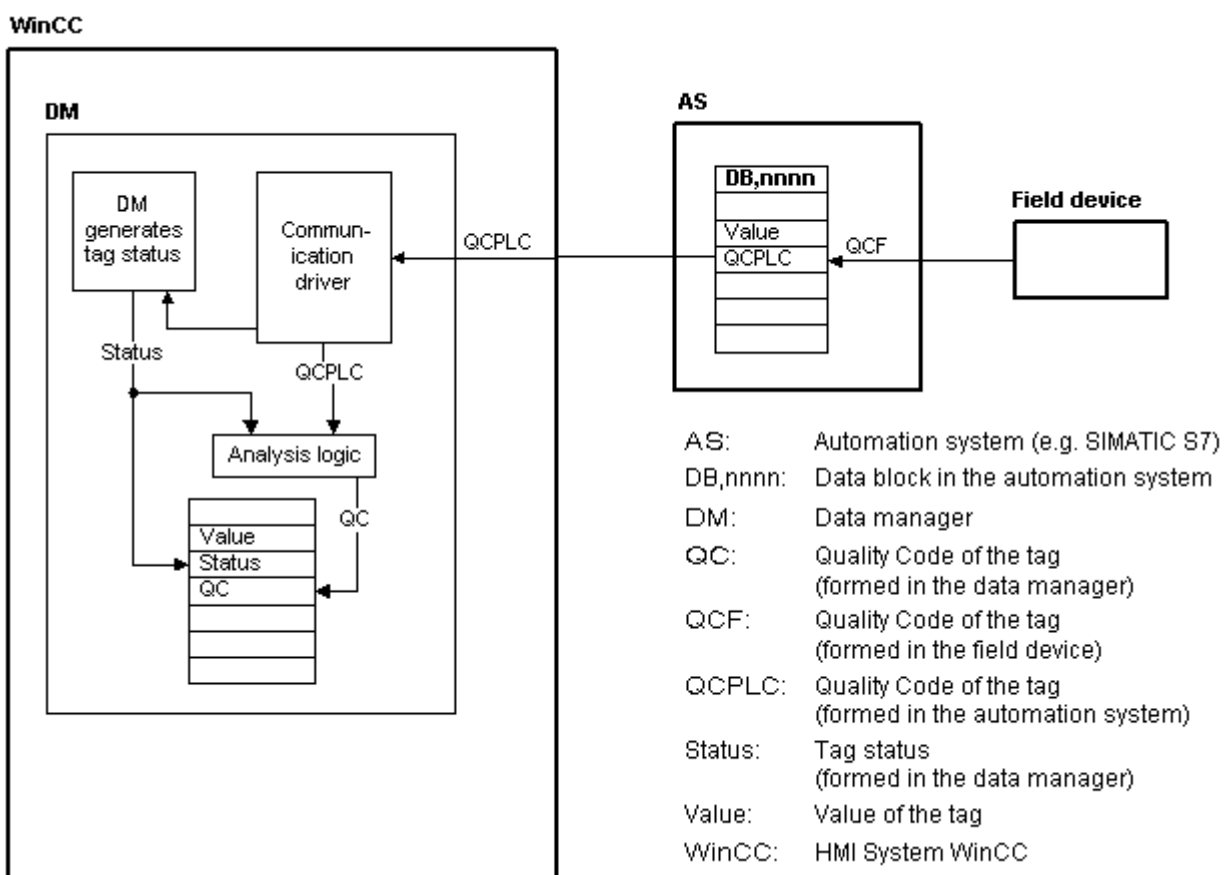
Using one of the communication drivers, WinCC reads the tags from the automation device in Runtime, including the associated quality codes. For each tag, the tag status is formed in the data manager. It contains, for example, violations of configured measurement range limits as well as the status of linkage between WinCC and the automation device.

Using the analysis logic in the data manager, the quality code is generated from the tag status of the data manager and the quality code of the automation device. Here too, the code with the worst status is passed on and saved as quality code by WinCC. For tags that do not have a quality code in the automation system, the quality code is always identical with the tag status.

17.8.2 Quality Codes of Tags

Introduction

The quality code is needed to check status and quality of a tag. The displayed quality code summarizes the quality of the entire value transmission and value processing for the respective tag. Thus with the quality code you can for example see whether the current value is a start value or a substitute value.

The quality codes are prioritized. If several codes occur at the same time, the code with the worst status is displayed.

Evaluation of Quality Codes

Quality codes can be evaluated in a number of different ways:

- Evaluation with VB scripts
- Evaluation with C scripts
- Evaluation through the dynamic dialog
- Evaluation of the "Quality Code Change Tag" result of an I/O field

---

Note

In order to include the entire value transfer and value processing in the quality code for a process tag, the connected automation system must support the quality code. When configuring the tags in the AS, make sure there is enough memory space for the quality code. In an AS from the S7 family, for example, the quality code needs an additional byte that is appended to the process value. To prevent errors, this byte must be taken into account when configuring a tag, for instance at the end of a data block.
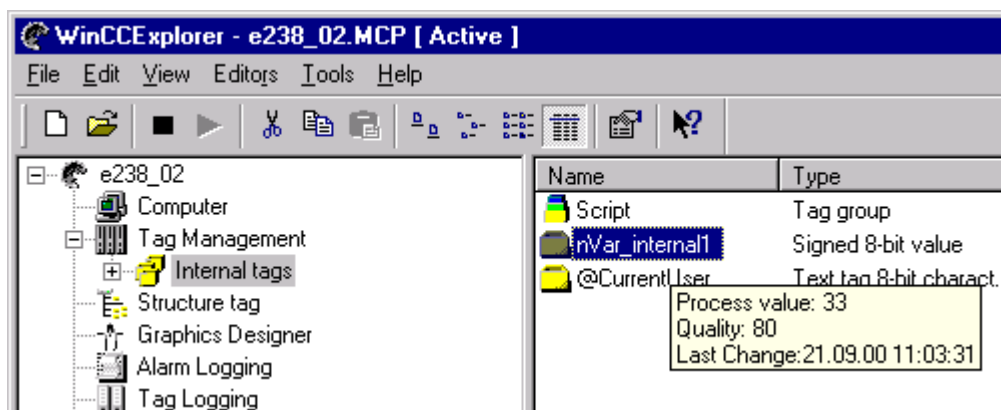
Display of Quality Codes in Process Pictures

For the display of tag values in graphic objects with process connection, the quality code may affect the display. If the quality code has a value of 0x80 (good) or 0x4C (initial value), the display of the tag value is not grayed out. For all other values, the display is grayed out. In

addition, a yellow warning triangle is displayed for the following objects depending on the set WinCC design:

- I/O field

- Bar, 3D bar

- Check box, radio box

- Group display, status display

- Slider object

Structure

The quality code has the following binary structure:

QQSSSSLL

Q: Quality

S: Substatus of the quality

L: Limits. This value is optional.

Note

The quality codes shown in the "Quality" table are basic values of the quality stages. Making use of the substatus and limit elements gives rise to intermediate values over and above the quality stage concerned.

Quality

The first two digits specify the quality of the tag.

	Q	Q	S	S	S	S	L	L
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bad - The value is not useful	0	0	-	-	-	-	-	-
Uncertain - The quality of the value is less than normal, but the value may still be useful.	0	1	-	-	-	-	-	-
Good (non-cascade) - The quality of the value is good. Possible alarm conditions may be indicated by the sub-status.	1	0	-	-	-	-	-	-
Good (cascade) - The value may be used in control.	1	1	-	-	-	-	-	-

Substatus

The quality alone is not enough. Individual qualities are divided into substatuses. The quality code is binary coded. In order to analyze quality codes their values must be converted into their hexadecimal representation.

Quality Codes of Tags

Possible quality codes are listed in the following table. At the top of the list, you find the poorest quality code, while the best quality code is shown at the bottom of the list. The best quality code is assigned the lowest priority, while the poorest quality code is assigned the highest priority. If several statuses occur for one tag in the process, the poorest code is passed on.

Code (Hex)	Quality		Q	Q	S	S	S	S	L	L
0x23	Bad	Device passivated - Diagnostic alerts inhibited	0	0	1	0	0	0	1	1
0x3F	Bad	Function check - Local override	0	0	1	1	1	1	1	1
0x1C	Bad	Out of Service - The value is not reliable because the block is not being evaluated, and may be under construction by a configuration planner. Set if the block mode is O/S.	0	0	0	1	1	1	-	-
0x73	Uncertain	Simulated value - Start	0	1	1	1	0	0	1	1
0x74	Uncertain	Simulated value - End	0	1	1	1	0	1	-	-
0x84	Good (non-cascade)	Active Update event - Set if the value is good and the block has an active Update event.	1	0	0	0	0	1	-	-
0x24	Bad	Maintenance alarm - More diagnostics available.	0	0	1	0	0	1	-	-
0x18	Bad	No Communication, with no usable value - Set if there has never been any communication with this value since it was last "Out of Service".	0	0	0	1	1	0	-	-
0x14	Bad	No Communication, with last usable value - Set if this value had been set by communication, which has now failed.	0	0	0	1	0	1	-	-
0x0C	Bad	Device failure - Set if the source of the value is affected by a device failure.	0	0	0	0	1	1	-	-
0x10	Bad	Sensor failure	0	0	0	1	0	0	-	-
0x08	Bad	Not Connected - Set if this input is required to be connected and is not connected.	0	0	0	0	1	0	-	-
0x04	Bad	Configuration error - Set if the value is not useful because there is some inconsistency regarding the parameterization or configuration, depending on what a specific manufacturer can detect.	0	0	0	0	0	1	-	-
0x00	Bad	Non-specific - There is no specific reason why the value is bad. Used for propagation.	0	0	0	0	0	0	-	-
0x28	Bad	Process related - Substitute value	0	0	1	0	1	0	-	-
0x2B	Bad	Process related - No maintenance	0	0	1	0	1	0	1	1
0x68	Uncertain	Maintenance demanded	0	1	1	0	1	0	-	-
0x60	Uncertain	Simulated value - Set when the process value is written by the operator while the block is in manual mode.	0	1	1	0	0	0	-	-
0x64	Uncertain	Sensor calibration	0	1	1	0	0	1	-	-
0x5C	Uncertain	Configuration error	0	1	0	1	1	1	-	-
0x58	Uncertain	Subnormal	0	1	0	1	1	0	-	-
0x54	Uncertain	Engineering unit range violation - Set if the value lies outside of the set of values defined for this parameter. The limits define which direction has been exceeded.	0	1	0	1	0	1	-	-
0x50	Uncertain	Sensor conversion not accurate	0	1	0	1	0	0	-	-
0x4B	Uncertain	Substitute (constant)	0	1	0	0	1	0	1	1

Code (Hex)	Quality		Q	Q	S	S	S	S	L	L
0x78	Uncertain	Process related - No maintenance	0	1	1	1	1	0	-	-
0x4C	Uncertain	Initial value - Value of volatile parameters during and after reset of the device or of a parameter.	0	1	0	0	1	1	-	-
0x48	Uncertain	Substitute value - Predefined value is used instead of the calculated one. This is used for fail safe handling.	0	1	0	0	1	0	-	-
0x44	Uncertain	Last usable value - Whatever was writing this value has stopped doing so. This is used for fail safe handling.	0	1	0	0	0	1	-	-
0x40	Uncertain	Non-specific - There is no specific reason why the value is uncertain.	0	1	0	0	0	0	-	-
0xE0	Good (cascade)	Initiate fail safe (IFS) - The value is from a block that wants its downstream output block (e.g. AO) to go to fail safe.	1	1	1	0	0	0	-	-
0xD8	Good (cascade)	Local override (LO) - The value is from a block that has been locked out by a local key switch or is a Complex AO/DO with interlock logic active. The failure of normal control must be propagated to a function running in a host system for alarm and display purposes. This also implies "Not Invited".	1	1	0	1	1	0	-	-
0xD4	Good (cascade)	Do not select (DNS) - The value is from a block which should not be selected, due to conditions in or above the block.	1	1	0	1	0	1	-	-
0xCC	Good (cascade)	Not invited (NI) - The value is from a block which does not have a target mode that would use this input.	1	1	0	0	1	1	-	-
0xC8	Good (cascade)	Initialization request (IR) - The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or the mode is wrong.	1	1	0	0	1	0	-	-
0xC4	Good (cascade)	Initialization acknowledge (IA) - The value is an initialized value from a source (cascade input, remote-cascade in, and remote-output in parameters).	1	1	0	0	0	1	-	-
0xC0	Good (cascade)	OK - No error or special condition is associated with this value.	1	1	0	0	0	0	-	-
0xA0	Good (non-cascade)	Initiate fail safe	1	0	1	0	0	0	-	-
0x98	Good (non-cascade)	Unacknowledged critical alarm - Set if the value is good and the block has an unacknowledged alarm with a priority greater than or equal to 8.	1	0	0	1	1	0	-	-
0x94	Good (non-cascade)	Unacknowledged advisory alarm - Set if the value is good and the block has an unacknowledged alarm with a priority less than 8.	1	0	0	1	0	1	-	-
0x90	Good (non-cascade)	Unacknowledged update event - Set if the value is good and the block has an unacknowledged update event.	1	0	0	1	0	0	-	-
0x8C	Good (non-cascade)	Active critical alarm - Set if the value is good and the block has an active alarm with a priority greater than or equal to 8.	1	0	0	0	1	1	-	-
0x88	Good (non-cascade)	Active advisory alarm - Set if the value is good and the block has an active alarm with a priority less than 8.	1	0	0	0	1	0	-	-
0xA8	Good (non-cascade)	Maintenance demanded	1	0	1	0	1	0	-	-

Code (Hex)	Quality		Q	Q	S	S	S	S	L	L
0xA4	Good (non-cascade)	Maintenance required	1	0	1	0	0	1	-	-
0xBC	Good (non-cascade)	Function check - Local override	1	0	1	1	1	1	-	-
0x80	Good (non-cascade)	OK - No error or special condition is associated with this value.	1	0	0	0	0	0	-	-

Limit

Quality codes can be further subdivided by limits. Limits are optional.

	Q	Q	S	S	S	S	L	L
O.K. - The value is free to move.	-	-	-	-	-	-	0	0
Low limited - The value has acceded its low limits.	-	-	-	-	-	-	0	1
High limited - The value has acceded its high limits.	-	-	-	-	-	-	1	0
Constant (high and low limited) - The value cannot move, no matter what the process does.	-	-	-	-	-	-	1	1

Quality Codes in Communication with OPC

In the communication via the "OPC" channel, the quality codes that the OPC does not support are converted.

Quality Code in WinCC	Quality Code in OPC
0x48	0x40
0x4C	0x40
0x5C	0x40
0x60	0x40
0x80...0xD4	0xC0
0xD8	0xC0

17.8.3 Tag Status

Introduction

The tag status of individual WinCC tags can be monitored in Runtime. The tag status contains, among other information, data regarding violations of the configured measurement range limits as well as the status of linkage between WinCC and automation device.

The quality code informs of the quality of a tag independent of where this code was formed. Thereby, the status of the entire value transfer and value processing are taken into consideration.

For example, if a violation occurs of the measurement range at the lower limit, quality code "0x55" is communicated. This violation of the measurement range might have occurred in the WinCC data manager or in the field device. The tag status allows you to find out if this measurement range violation occurred in WinCC or prior to passing the value to WinCC.

For example, if the tag status reports a limit violation with code 0x0010, it indicates that the values remained below the lower range limit configured in WinCC. If the tag status does not report any limit violation, the quality code passed on to WinCC already contained the limit violation.

Evaluation of Quality Codes

Quality codes can be evaluated in a number of different ways:

● Evaluation with C scripts

● Evaluation through the dynamic dialog

● Evaluation of the "Quality Code Change Tag" result of an I/O field

WinCC Status Flags

Possible tag statuses are contained in the following table.

Name of flag	Value	Description
	0x0000	No error
DM_VARSTATE_NOT_ESTABLISHED	0x0001	Connection to partner not established
DM_VARSTATE_HANDSHAKE_ERROR	0x0002	Handshake error
DM_VARSTATE_HARDWARE_ERROR	0x0004	Network module defective
DM_VARSTATE_MAX_LIMIT	0x0008	Configured upper limit exceeded
DM_VARSTATE_MIN_LIMIT	0x0010	Configured lower limit exceeded
DM_VARSTATE_MAX_RANGE	0x0020	Format upper limit exceeded
DM_VARSTATE_MIN_RANGE	0x0040	Format lower limit exceeded
DM_VARSTATE_CONVERSION_ERROR	0x0080	Display conversion error (in connection with format limit xxx exceeded)
DM_VARSTATE_STARTUP_VALUE	0x0100	Tag initialization value
DM_VARSTATE_DEFAULT_VALUE	0x0200	Tag replacement value

Name of flag	Value	Description
DM_VARSTATE_ADDRESS_ERROR	0x0400	Channel addressing error
DM_VARSTATE_INVALID_KEY	0x0800	Tag not found / not available
DM_VARSTATE_ACCESS_FAULT	0x1000	Access to tag not permitted
DM_VARSTATE_TIMEOUT	0x2000	Timeout / no check-back message from the channel
DM_VARSTATE_SERVERDOWN	0x4000	Server not available.

17.8.4   Using the Tag Status to Monitor Connection Status

Introduction

The status of individual WinCC tags can be monitored in Runtime, providing information about the status of the associated connection.

The monitoring is configured in Graphics Designer as an object property. One possibility for monitoring would be to use the "Text" property of a static text.

In the shortcut menu of the "Dynamic" dialog for the desired property, select "Dynamic Dialog" to open the "Dynamic value ranges" dialog.

The following settings are specified here:

- Tag to be monitored

- For tag value: assignment of validity range and status display

- The activation of the tag status evaluation

- For the status: assignment of validity range and corresponding status text

In Runtime, one of the entered status texts, which corresponds to the tag's current status, is displayed in the configured object.

17.8.5   Monitoring Tag Status Using Global Actions

Introduction

One way to monitor the status of a tag is to make use the internal functions "GetTagState" and "GetTagStateWait" in the Global Script editor. In contrast to the "GetTag" and "GetTagWait"

functions, these not only return the tag's value but also its status. This status value can be evaluated and then used to trigger various events. It can also be used to assess the status of the associated connection.

In the global action, the status value of the monitored tag is determined using the "GetTagState" function for this tag type. There is such a function for each tag type. The status value "0" indicates a good connection with no errors. This status can now be evaluated as desired.

Example:

This example illustrates the monitoring of a WinCC tag of the type "Signed 16-Bit Value". The "GetTagSWordState" function is used to determine the status of this tag. The first function parameter is the name of the WinCC tag to be monitored. The second parameter gives where the returned status value is to be written.

```c
#include "apdefap.h"

int gscAction( void )
{
        DWORD dwState = 0;

        GetTagSWordState("Variable_01",&dwState);

        if ( dwState == 0 )
        {
                //Connection OK
                SetTagBit("BINi_E_CONNECTION",FALSE);
        }
        else
        {
                //Connection Error
                SetTagBit("BINi_E_CONNECTION",TRUE);
        }

        return 0;

}
```

The tag status is output in the internal tag BINi_E_CONNECTION. In the event of an error, the value of this tag is set to TRUE. In the error handling, the tag can, for example, be used to trigger an alarm or display an error message.

17.8.6   How to Check an Internal Tag

Introduction

If an internal tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- Configure an internal tag.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in tag management, select the entry "Internal Tags".

2. In the data window, select the tag that you wish to check. To do this, open the directory structure until the tag is displayed in the data window.

3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.

4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag Quality Codes".

5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.

6. Check whether values have been configured for the upper or lower limits, or start he value on the "Limits/Reporting" tab. These values can affect the display.

7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 414)

Index

as ActiveX Control, 374
Testing, 371

"

"MPI" channel unit, 217
   Configuring the connection, 217
"Slot PLC" channel unit, 224
   Configuring the connection, 224
"System Info" channel, 379
   Diagnostic option, 379

5

505-Ethernet (CP 1413-x)
   Channel Unit, 305

A

Adding a tag, 81, 88
Adding new drivers, 160
Address properties, 92, 107, 163, 165, 166, 171
   Example for S5 Profibus FDL, 181
Address type
   Mitsubishi Ethernet, 64
Addresses of tags, 309
   TI Ethernet Layer 4, 309
AR_SENDAR_SEND-FunctioninWinCC , 242
AR_SEND-Function in WinCC
   Configuring Variants for an Archive Tag, 269
   Configuring Variants for Multiple Archive Tags, 272
   Data Block,Structure and Parameters, 243
   Example for Data Block Structure, 252, 253, 254, 259, 261, 263, 265, 268
   Overview of the properties of the variants, 248
   Variants for an Archive Tag, 250
   Variants for Multiple Archive Tags, 255
   Variants for Multiple Archive Tags (optimized), 267
AS data type, 17
   Type conversion, 28
AS511, 185
Automation device, 158

B

Bar
   Example of System Info, 354
BinWrite Mechanism, 36
   General Procedure, 36
Bit by bit access, 36, 138, 230
   CCS7PlusChannel, 300
   S5 AS511, 192
   S5 Ethernet Layer 4, 138
   S5 Profibus FDL, 163
   S5 Serial 3964R, 201
   S7 Protocol Suite, 230
   TI Ethernet Layer 4, 310
   TI Serial, 322
Bit/Byte access, 36
Bit-/Byte-tag, 38
Block, 178
   Cyclic block, 178
Byte array, 171
Byte by byte access, 36, 140, 230
   S5 AS511, 193
   S5 Ethernet Layer 4, 140
   S5 Profibus FDL, 165
   S5 Serial 3964R, 202
   S7 Protocol Suite, 232
   SIMATIC S7-1200, S7-1500 Channel, 301
   TI Ethernet Layer 4, 311
   TI Serial, 323

C

CCS7PlusChannel
   Configuring a tag with bit by bit access, 300
Channel, 15, 133, 195, 206, 409
   check, 380, 391, 401
   Configuring SIMATIC S5 PROFIBUS FDL, 160
   Diagnosis of channels, 371
   Example for S5 PROFIBUS FDL, 178
   Mitsubishi Ethernet, (See Mitsubishi Ethernet)
   PROFIBUS FMS, 123
   SIMATIC S5 Ethernet Layer 4, 133
   SIMATIC S5 Programmers Port AS511, 185
   SIMATIC S5 Serial 3964R, 195
   SIMATIC S7 Protocol Suite, 206
   SIMATIC TI Ethernet Layer 4, 305
   Status - Logical Connections function , 371