SIEMENS

Application Example• 09/2016

Exchange of large data volumes between S7-300/400 control system and WinCC with „BSEND/BRCV“

S7-300,S7-400, WinCC V7.4

https://support.industry.siemens.com/cs/ww/de/view/37873547

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These Application Examples do not absolve you from responsibility for safe and professional use, installation, operation and servicing. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e. g. catalogs – the contents of the other documents shall have priority.

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

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

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

Security information

Siemens provides products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines and networks.

In order to secure plants, systems, machines and networks against cyber threats it is necessary to implement (and to continuously maintain) a holistic, state-of-the-art Industrial Security concept. With this in mind, Siemens' products and solutions are only one part of such a concept.

It is the client's responsibility to prevent unauthorized access to his plants, systems, machines, and networks. Systems, machines, and components should only be connected to the company's network or the Internet if and to the extent to which this is required and the appropriate protective actions (for example, use of firewalls and network segmentation) have been taken.

In addition, Siemens' recommendations regarding appropriate protective action should be followed. For more information on Industrial Security, please visit http://www.siemens.com/industrialsecurit.

Siemens' products and solutions undergo continuous development to make them even more secure. Siemens explicitly recommends updating the software as soon as new updates are available and always using current product versions only. Continuing to use product versions which became obsolete or are no longer supported may increase customer's exposure to cyber threats.

In order to stay informed about product updates, please subscribe to the Siemens Industrial Security RSS Feed at http://www.siemens.com/industrialsecurity.

Table of Contents

1 Task

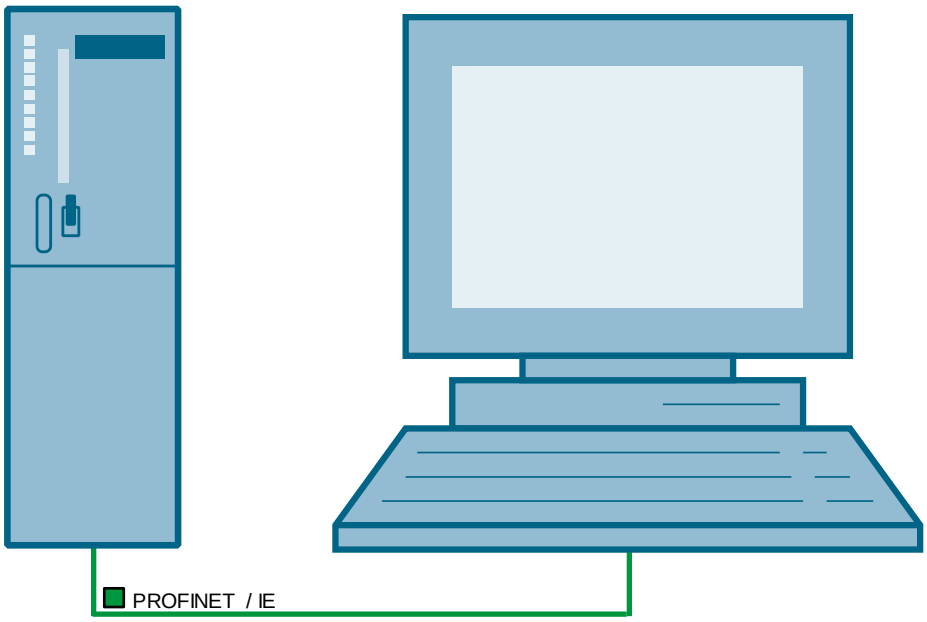
Description of the automation task

The present application example shows you how to transmit large data volumes from an automation control system to WinCC. This functionality will be explained using the example of the S7-300.

In our example, a „BSEND/BRCV“ communication is established to transmit large data volumes from the control system to WinCC / from WinCC to the control system, respectively. In this type of communication, the data volume is split into single segments which are sent individually to the related partners.

1.1 Overview

Figure 1-1



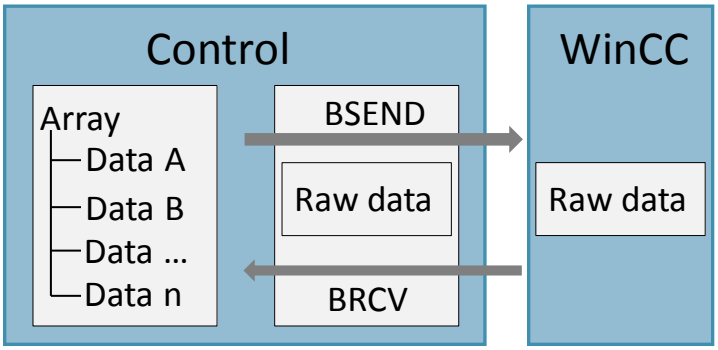
2 Solution

2.1 Configuration

A standard WinCC configuration is supposed. The present application example considers the connection from an automation control system to WinCC.

Diagram

Figure 2-1



A sender and a receiver are necessary for the bilateral data exchange. Both the sender and the receiver have a common connection ID assigned. The data blocks below are available for the bilateral data exchange:

- BSEND Send a data block
- BRCV Receive a data block

Advantage

The advantage is that a Raw Data Variable is licensed as external variable. As a raw data variable contains an array of “n” values, the latter will not be counted apart. Hence, one licensed external variable is enough to transmit 8000 byte values from the control system to WinCC, as an example.

Disadvantage

Extended time and effort are necessary to configure the raw data communication.

Required knowledge

To implement the solution described in the present application example, basic knowledge in the following branches is necessary:

- Automation technology

2.2 Hardware and software components

2.2.1 Applicability

This application example is valid for

- TIA Portal V13 SP1 Update 7
- WinCC V7.4

2.2.2 Components used

The application example has been created using the following components:

Hardware components

Table 2-1

Component	Qty	Article number	Note
SIMATIC S7-300	1x	6ES7 317-2EK14-0AB0	Firmware: V3.2 The control system is given as an example; other control systems may be used considering the software requirements.
Industrial PC SIMATIC IPC647D	1	6AG4112-2….-….	The IPC is given as an example; other IPCs may be used considering the software requirements.

Software components

Table 2-2

Component	Qty	Article number	Note
WinCC V7.4	1	6AV63.1-....7-4...	
STEP 7 Professional V13 SP1	1	6ES7822-1..03	

Example files and projects

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

Table 2-3

Component	Note
37873547_Rohdaten_WinCC_V74.zip	This zip file contains the WinCC project.
37873547_Rohdaten_TIA_S7-300.zip	This zip file contains the PLC project.
37873547_Rohdaten_WinCC_V74_TIA_S7-300_de.pdf	The present document.

3 Basic information

3.1 Building communication

To implement the functionality of the „BSEND“ and „BRCV“, a data memory must be created within the control system.

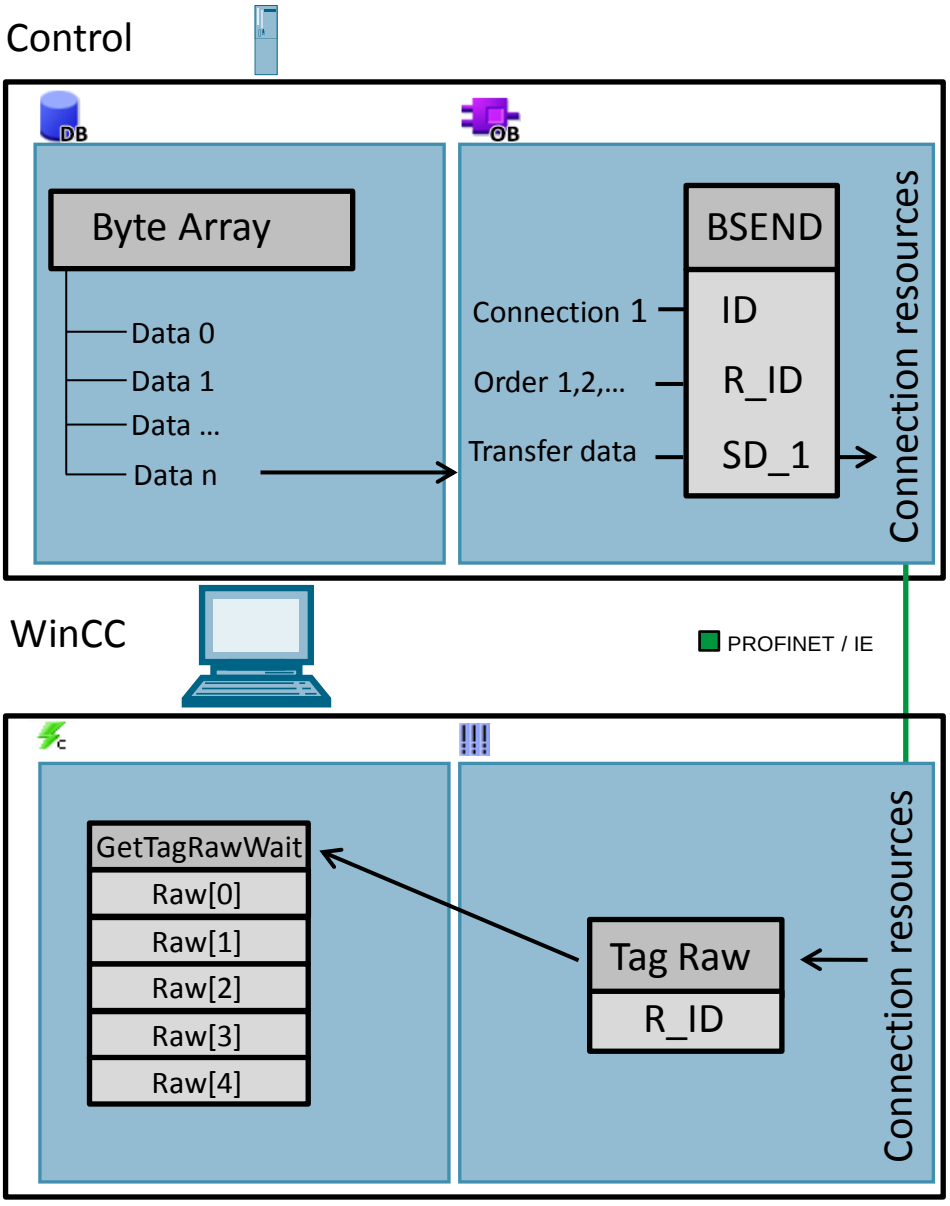
This data memory is used to save data in structured form as an array. That array is used as data basis in the control system. “BSEND” is used to write data from that array to the raw data tag of WinCC. WinCC in turn is able to interpret these data and read / display the data from this tag.

The procedure in the other direction also works in the same way. The WinCC functionality allows the modification of the raw data tags. These data are now transmitted to the control system, evaluated by means of “BRCV”, and written into the data block array. The data within the array modified by this procedure are then re-transmitted to WinCC on the next “BSEND” operation.

The figures below show the data exchange between a control system and a WinCC station with the help of the „BSEND“ and „BRCV“ data blocks.

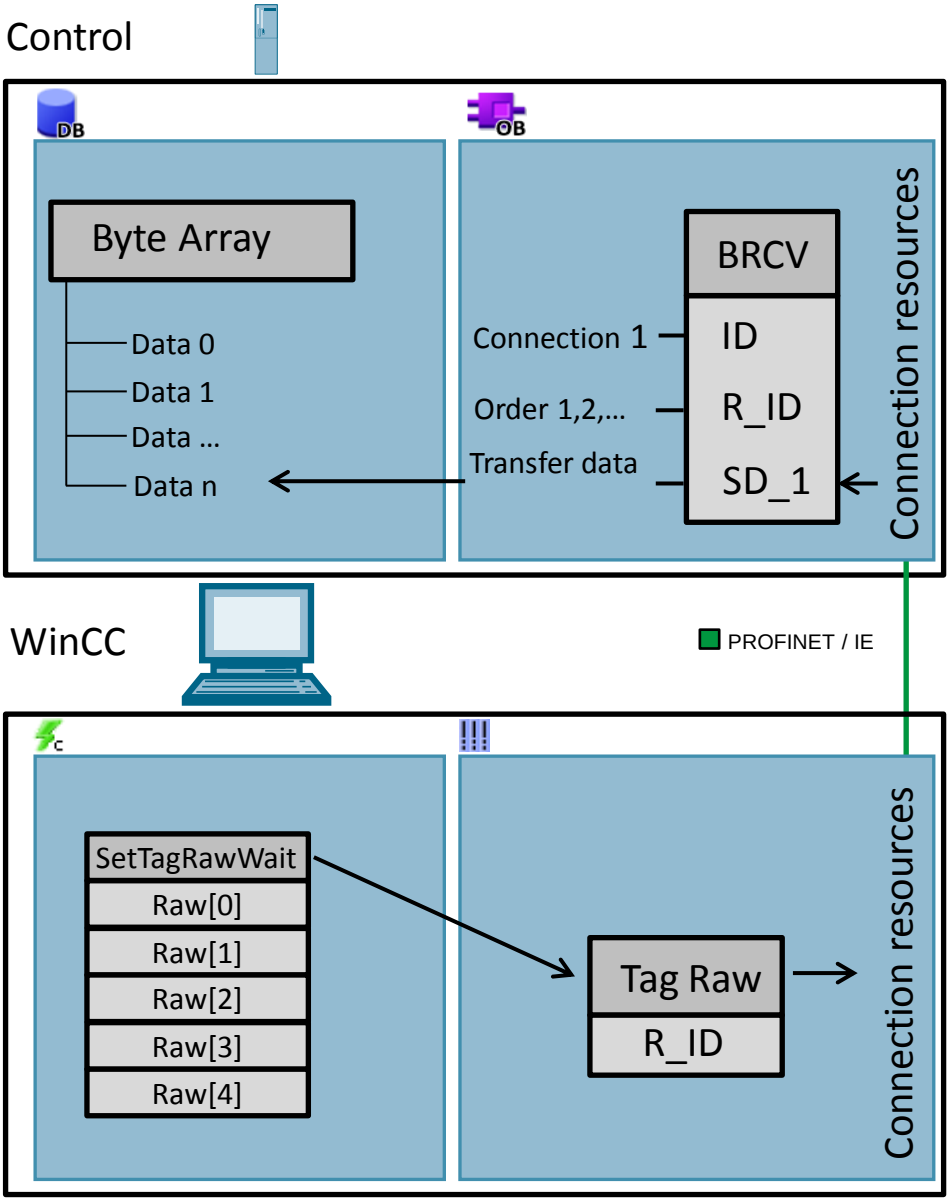
Sending data from the control system to WinCC with BSEND

Figure 3-1



Sending data from WinCC to the control system with BRCV

Figure 3-2

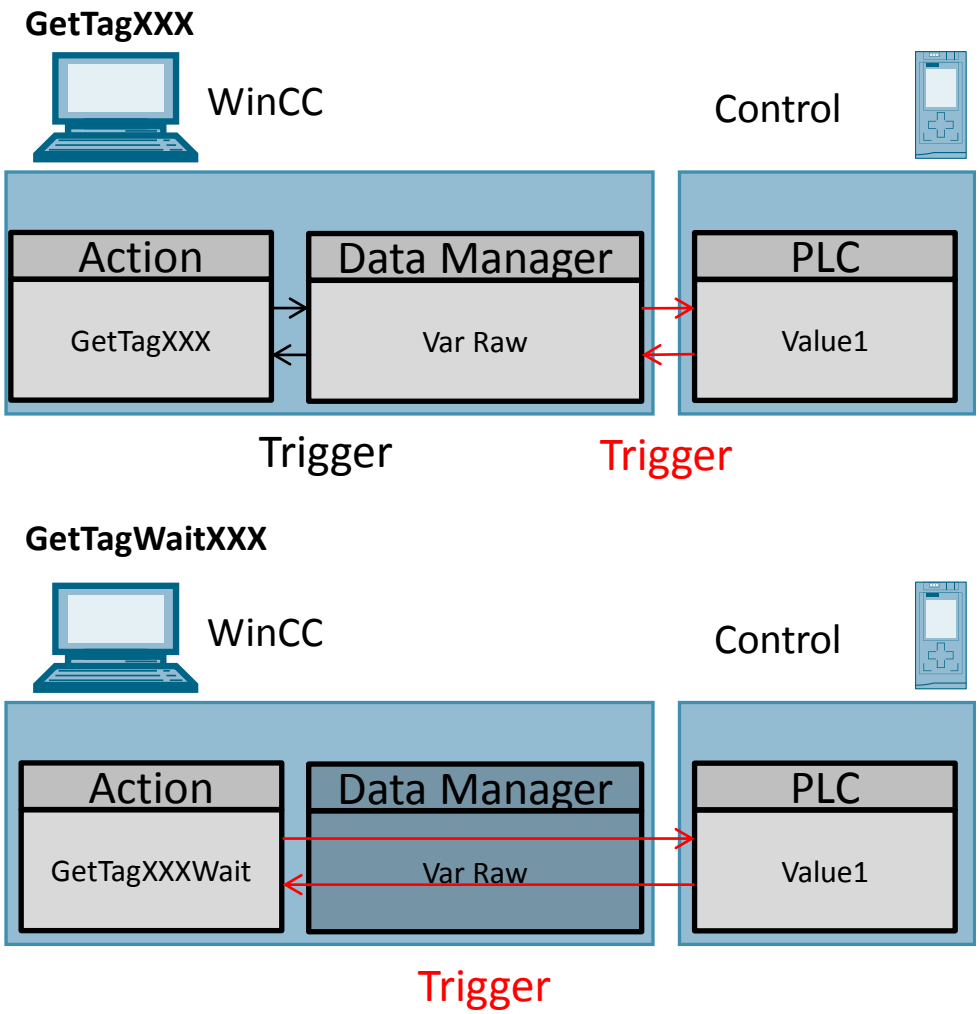


3.2 Difference between the functions GetTagRawWait and GetTagRaw

To read variables from / write variables to the AS, the C scripts provide the TAG object function.

Reading the variables only when they are necessary in the script is an error that we can often discover. As a consequence, the variable is only cyclically registered in the variables map of the data manager with 1 second standard cycle on it's first reading and thus will rise the basic load. The GetTagXXXWait function will remedy this problem. This function is used to bypass the data manager and the variable is not registered in this case.

Figure 3-3



The GetTagWait call is necessary in the following cases:

- Synchronizing quick write/read processes
- Reading explicitly one value out of the automation device
- By-passing deliberately the registering

The GetTagWait call shall be avoided in cyclical C actions.

3.3 Quality Code

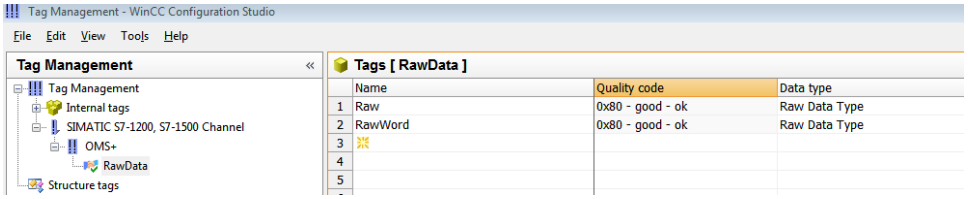
The Quality Code is used to check state and quality of a tag. This information is completed by quality-related information regarding the partners which acquire and process the tags. Potential partners are the following:

- Automation systems
- Automation systems with field devices
- OPC servers
- OPC servers with lower-level automation systems

Access “Tag Management” to display the Quality Code of the tags.

The following requirements must be met:

- The WinCC project is activated
- The “Quality Code” column is visible among others in the “Tag Management” data area.



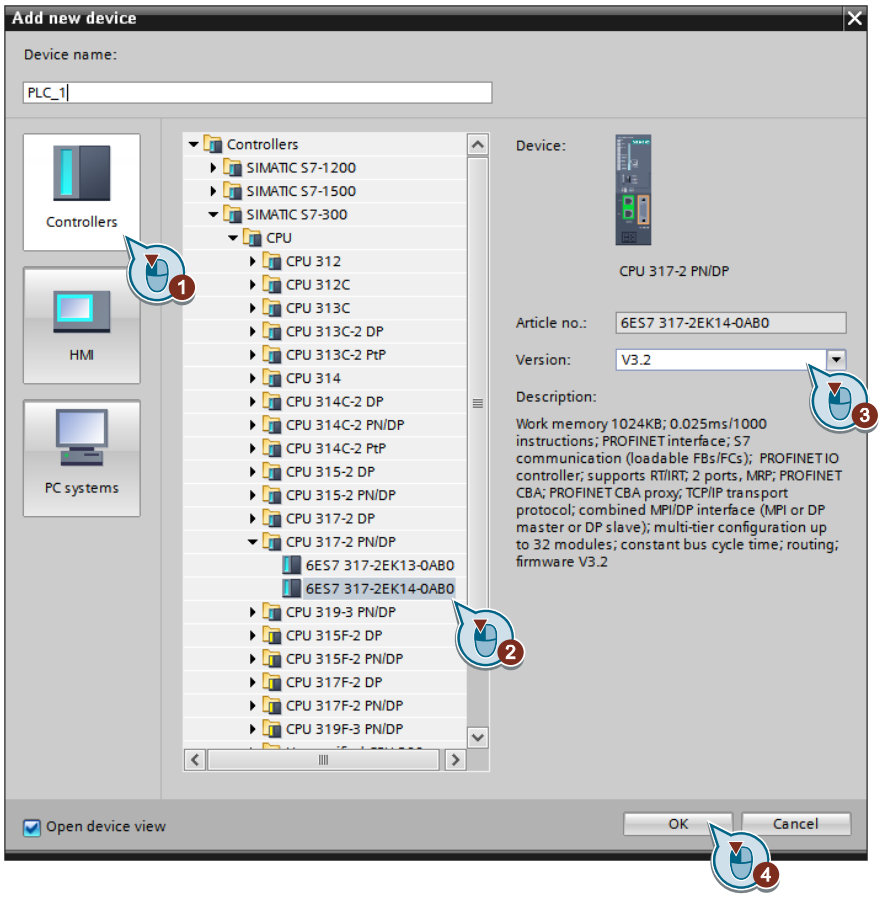
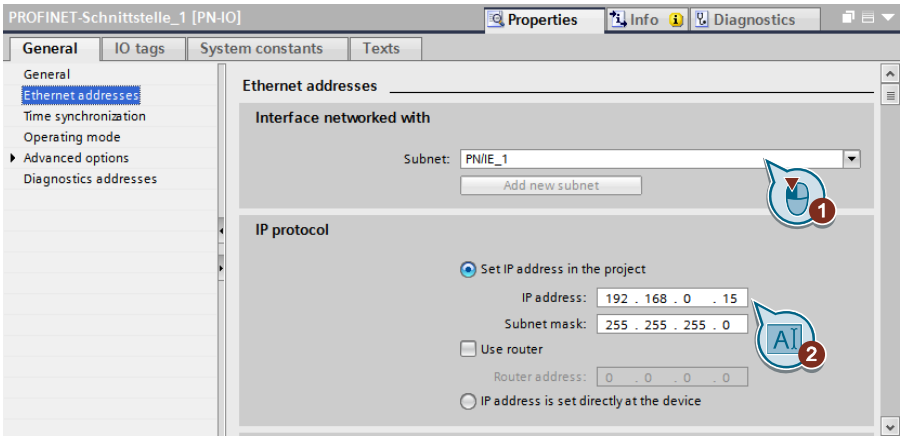
The Quality Code “good” for example means that this tag may be used. For detailed information on the Quality Code elements, please refer to the table in WinCC Handbuch Kommunikation. The table starts with the worst Quality Code and ends with the best one.

Note Owing to the asynchronous read operation, the Quality Code is always “bad” when a raw data tag is read for the first time. The status will switch to “good” for any further read operation.

4 PLC Configuration and Design

4.1 Creating and configuring the CPU

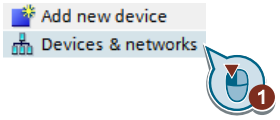
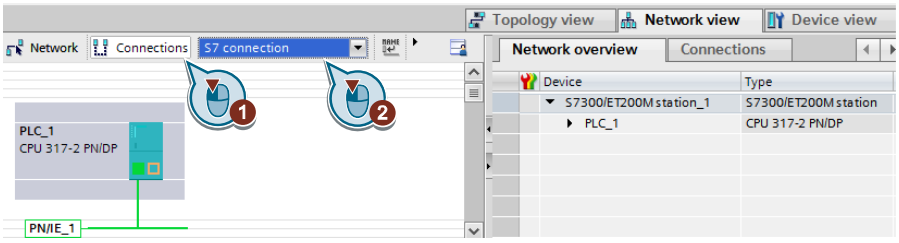
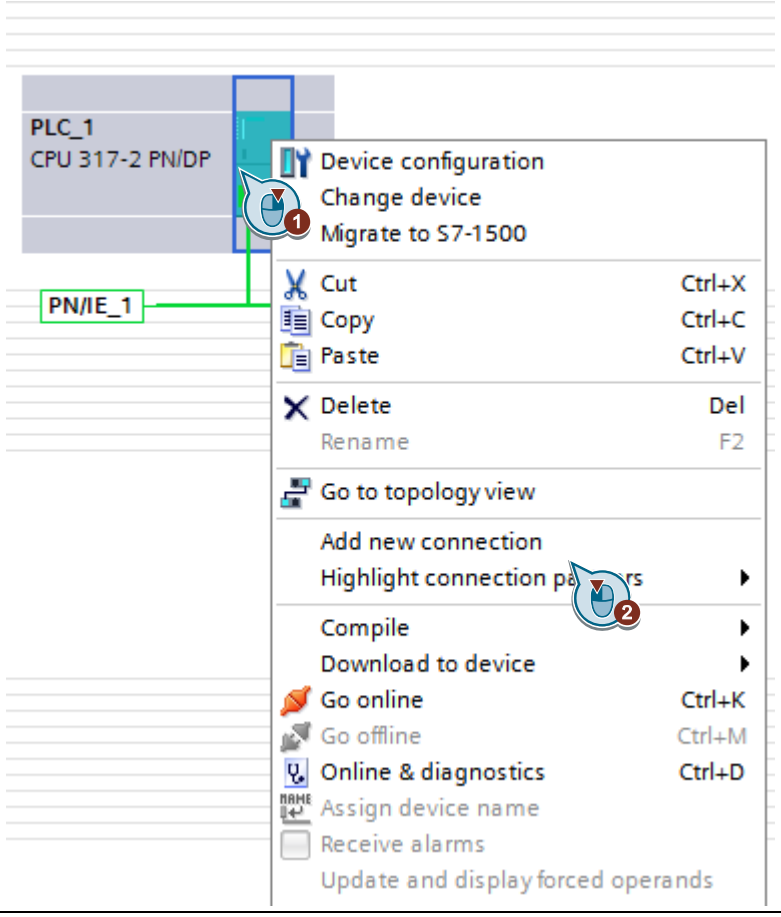
Table 4-1

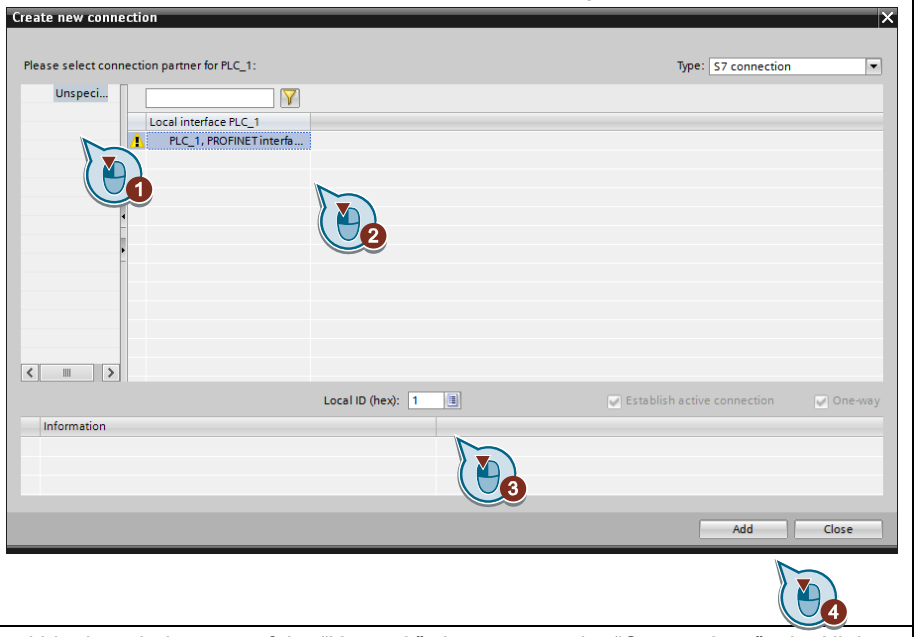
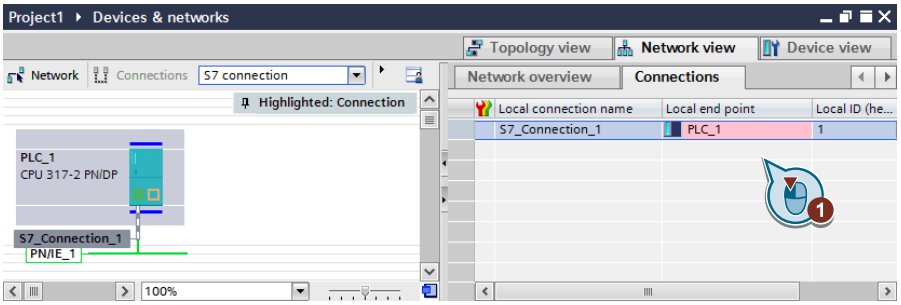
Item	Action
1.	Insert a new S7-300 control system into your project. 
2.	Assign an IP address and a subnet to the new control system. 

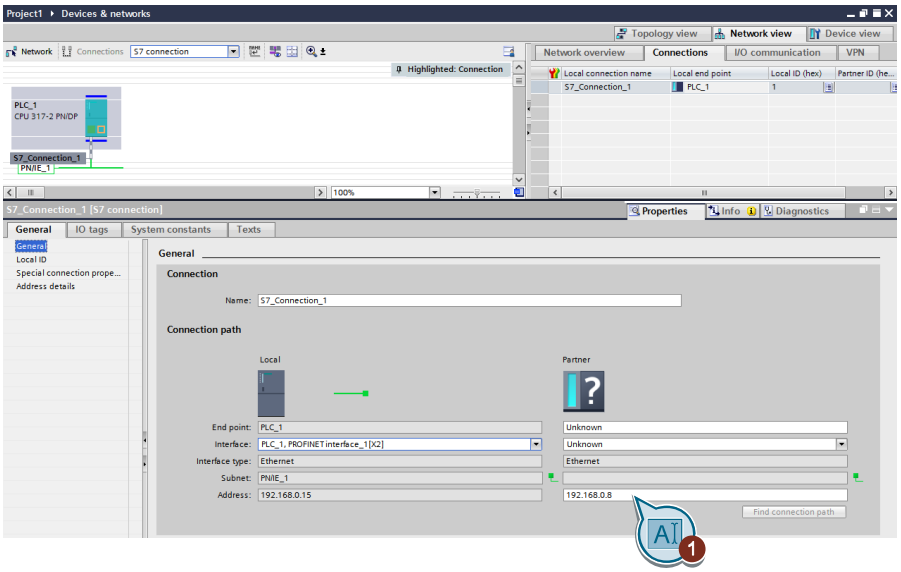
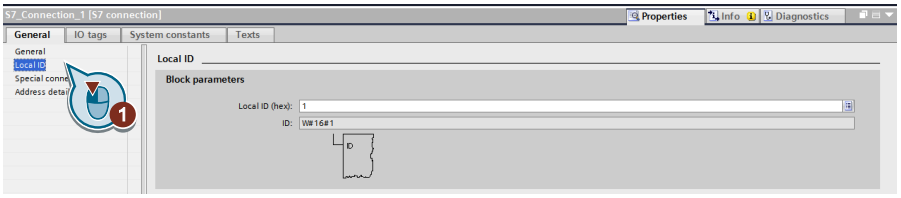
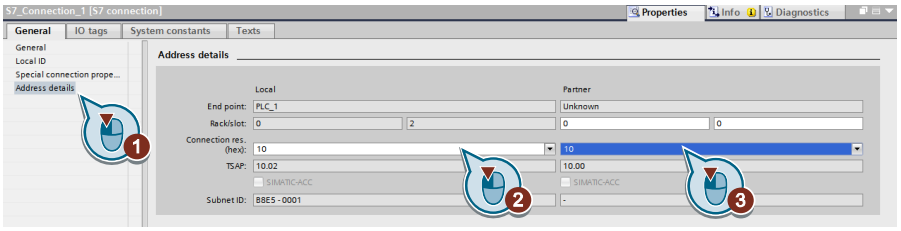
4.2 Designing an S7-connection

In TIA Portal, the S7 connections are designed in the “Network” view.

Table 4-2

Item	Action
1.	Open the “Network” view” within the “Devices & Networks” editor. 
2.	Click the “Connections” button. Select an S7 connection. 
3.	Right-click the CPU and select “Add new connection” from the CPU’s context menu. 

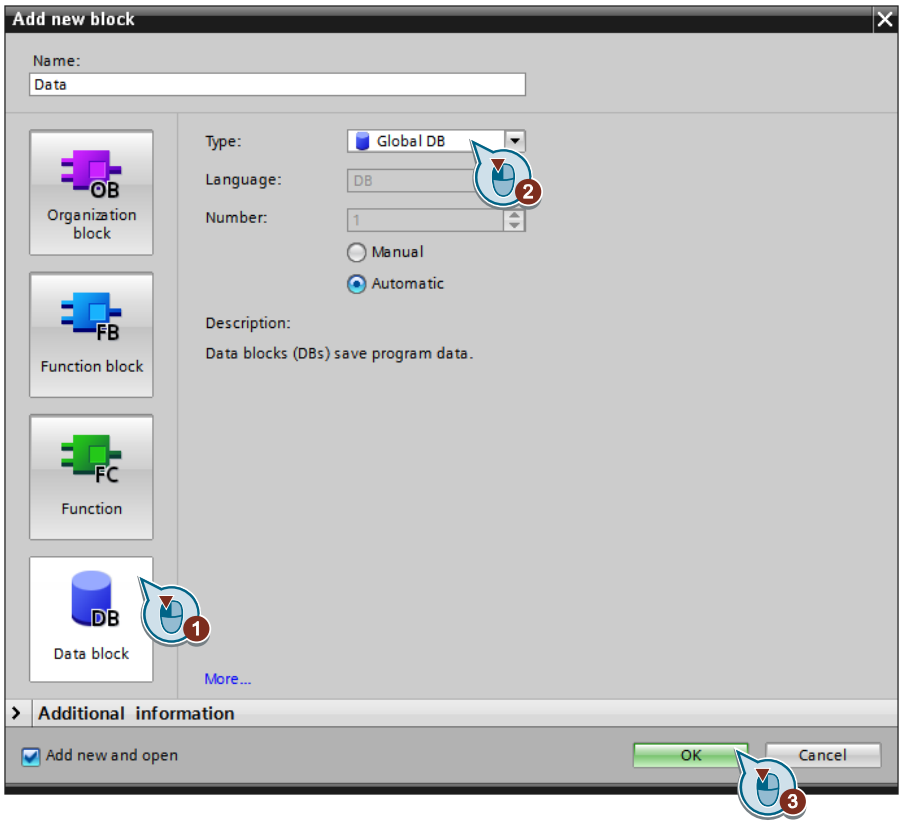
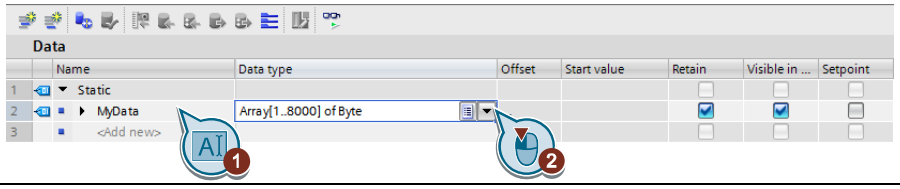
Item	Action
4.	In the left part of the dialog box, select the connection partner, e.g. “Unspecified”. In the right part of the dialog box, select the local interface to be used for the communication. Click the “Add” button. Close the dialog. 
5.	Within the tabular area of the “Network” view, access the “Connections” tab. All the connections configured will be displayed here. A connection is highlighted red if, for example, the IP address of the partner has not been configured.  Select the connection not yet configured.

Item	Action
6.	In the “General” tab, enter the IP address of the communication partner. 
7.	In the “Local ID” tab, determine the connection ID. You will need this ID later for the blocks „BSEND“ and „BRCV“. 
8.	In the “Address details” tab, establish the same connection resource between partner and CPU. 
9.	Access the “Special connection properties” tab. Untick the “Active connection establishment” option. 

4.3 Creating the data block

The “Data” data block is used to transmit the data to the BSEND and BRCV communication blocks in the form of an array.

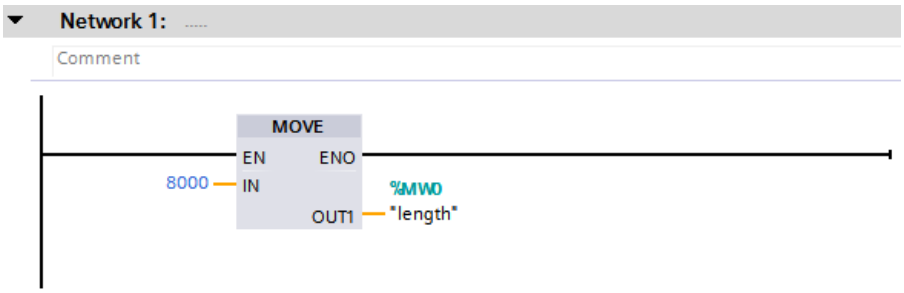
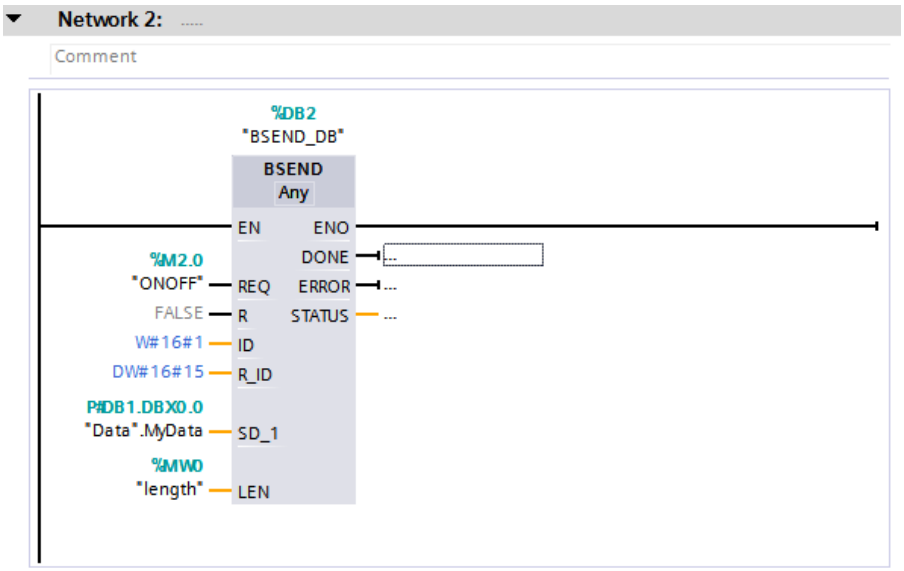
Table 4-3

Item	Action
1.	Create a new “Data” data block. 
2.	Create a byte array with 8000 values, as an example. These values will be sent later as raw data to WinCC. 

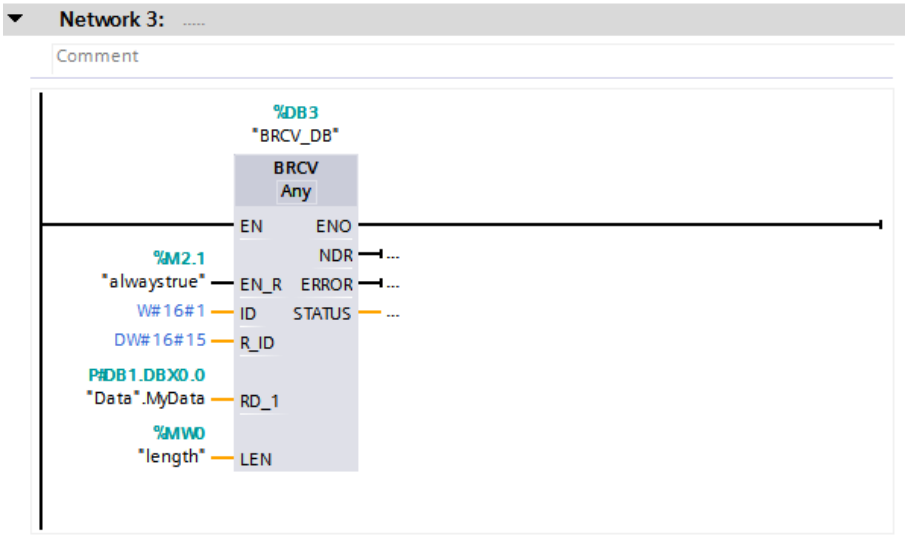
4.4 Creating the data blocks BSEND and BRCV

The communication blocks “BSEND” and “BRCV” are used to exchange data between the PLC and the WinCC station.

Table 4-4

<table>
<tr><th>Item</th><th>Action</th></tr>
<tr><td>1.</td><td>Open OB1. Move one MOVE block into your network 1. Both the block “BSEND” and the block “BRCV” need a fixed length to send and receive data.
 
</td></tr>
<tr><td>2.</td><td>Add one “BSEND” block to your network 2. The “BSEND” instruction is used to send data from a data block to a communication partner with fixed target indication.
 

<table>
<tr><th>Parameters</th><th>Data type</th><th>Description</th></tr>
<tr><td>REQ</td><td>BOOLEAN</td><td>The data exchange starts on positive edge at the parameter.</td></tr>
<tr><td>R</td><td>BOOLEAN</td><td>The data exchange stops on positive edge at the parameter.</td></tr>
<tr><td>ID</td><td>WORD</td><td>Assign the “ID” parameter with the connection ID defined in STEP7 in the connection table for both the local and the partner devices. See Chapter 4.2.</td></tr>
<tr><td>R_ID</td><td>DWORD</td><td>The parameter R_ID is used to define a free but unique correlation between SEND and RECEIVE instructions. This ID must be the same for the sender and the receiver. You can also use several</td></tr>
</table></td></tr>
</table>

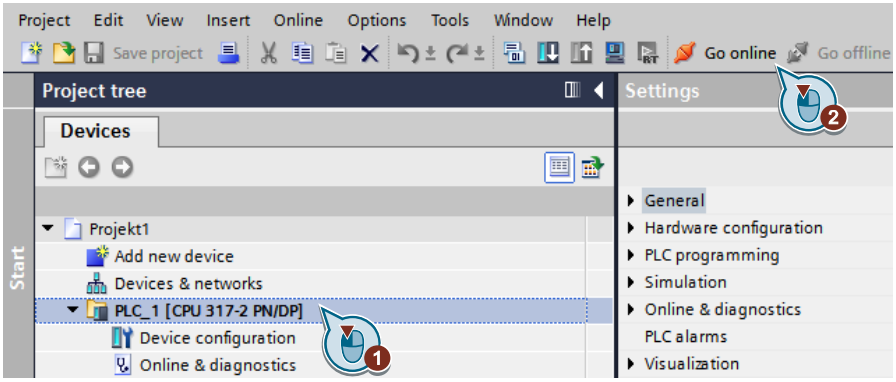
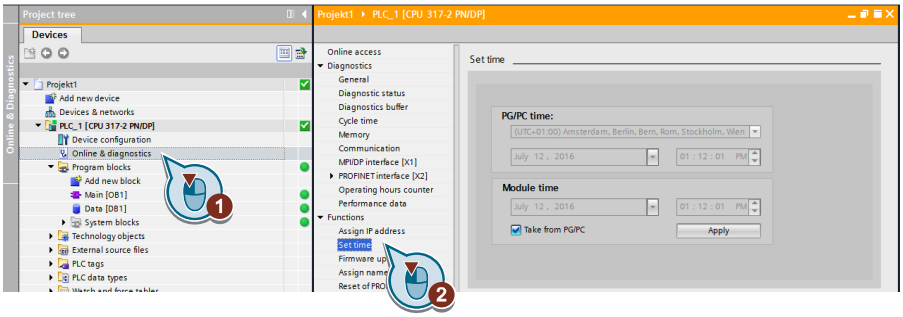
<table>
<tr><th>Item</th><th>Action</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td>“BSEND” and “BRCV” couples on one connection (ID) that differ in the order R_ID.</td></tr>
<tr><td>SD_1</td><td>ANY</td><td>Indicate the array variable contained in the data block at parameter RD_1.</td></tr>
<tr><td>LEN</td><td>WORD</td><td>Indicate the number of bytes to send at parameter LEN.</td></tr>
</table>
</td></tr>
<tr><td>3.</td><td>
Add one “BRCV” block to your network 3. The “BRCV” data block is used to receive data from a communication partner and transmit them to a data block.


<table>
<tr><th>Parameters</th><th>Data type</th><th>Description</th></tr>
<tr><td>EN_R</td><td>BOOLEAN</td><td>The parameter EN_R outputs the ready-to-receive signal if the input has been set.</td></tr>
<tr><td>ID</td><td>WORD</td><td>Assign the “ID” parameter with the connection ID defined in STEP7 in the connection table for both the local and the partner devices. See Chapter 4.2.</td></tr>
<tr><td>R_ID</td><td>DWORD</td><td>The parameter R_ID is used to define a free but unique correlation between SEND and RECEIVE instructions. This ID must be the same for the sender and the receiver. You can also use several “BSEND” / “BRCV” couples on one connection (ID) that differ in the order R_ID.</td></tr>
<tr><td>RD_1</td><td>ANY</td><td>Indicate the array variable contained in the data block at parameter RD_1.</td></tr>
<tr><td>LEN</td><td>WORD</td><td>Indicate the number of bytes to send at parameter LEN.</td></tr>
</table>
</td></tr>
</table>

4.5 Setting the clock time

Make sure that the CPU and WinCC are set to the same clock time (UTC or local time).

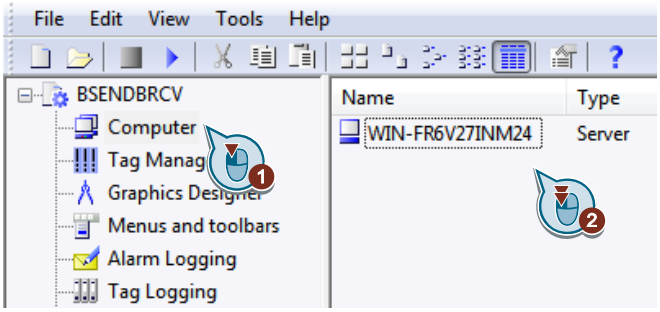
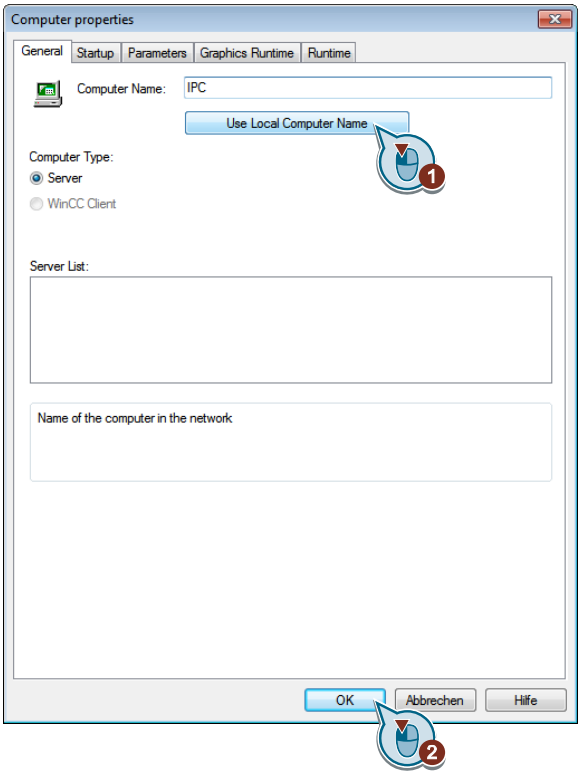
Table 4-5

Item	Action
1.	Access your control system and go “online”. 
2.	In the project navigation of your CPU, select “Online & Diagnostics”. Adjust your system to the same clock time as your WinCC station. 

5 HMI Configuration and Design

5.1 Preparing the design environment in SIMATIC WinCC

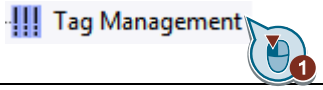
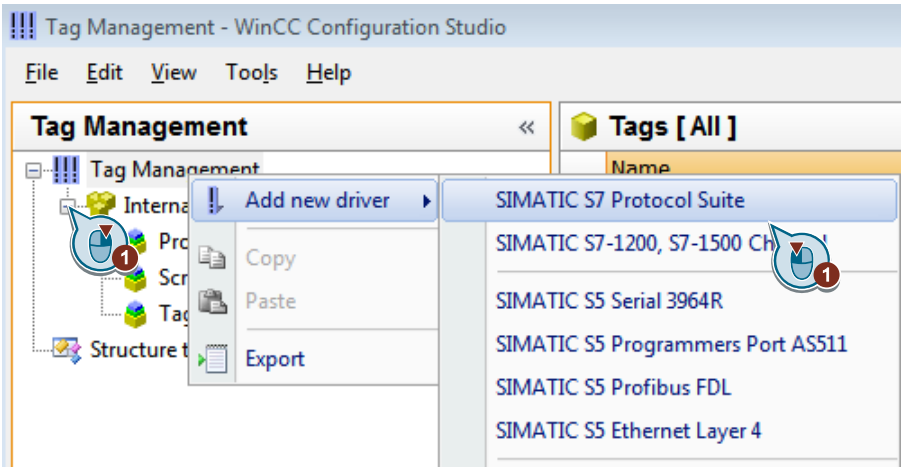
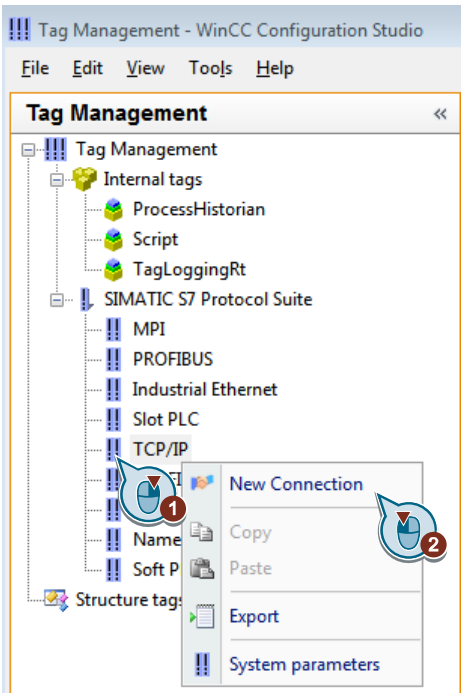
Table 5-1

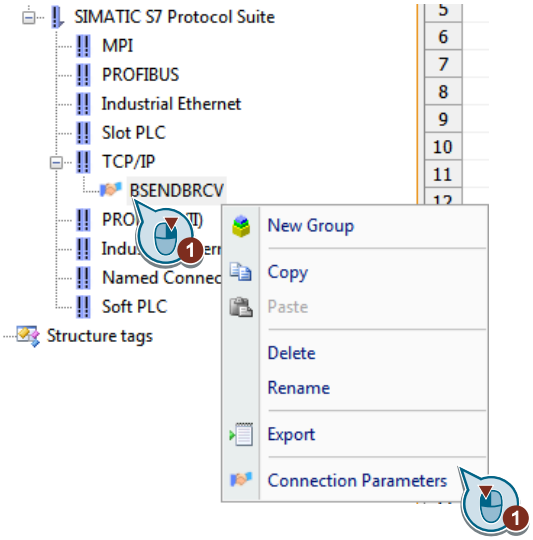
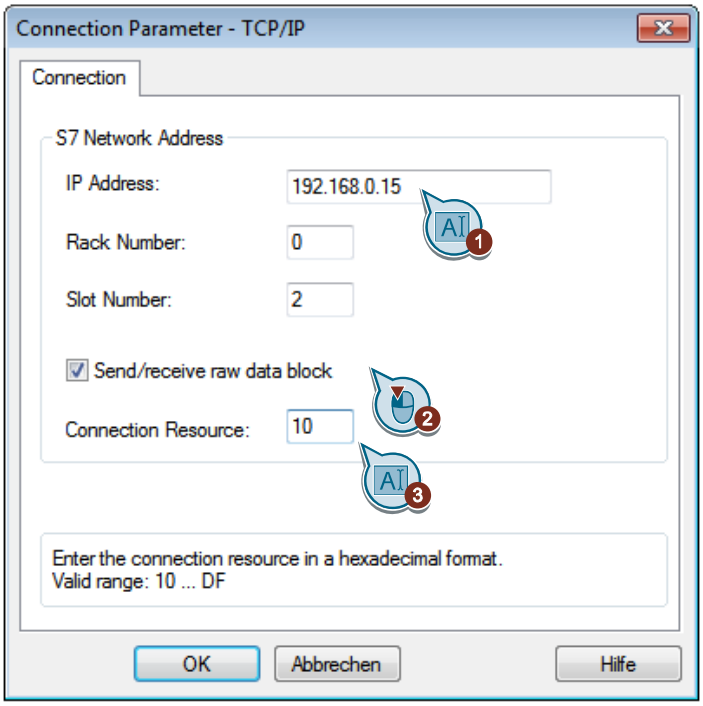
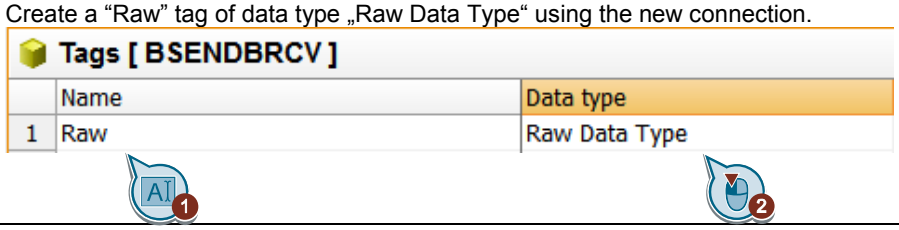
Item	Action
1.	Start the SIMATIC WinCC explorer.
2.	In the WinCC explorer, click the “Computer” icon. Then click the computer name. 
3.	In “Computer properties → General” click “Use local computer name” and confirm with “OK”. 

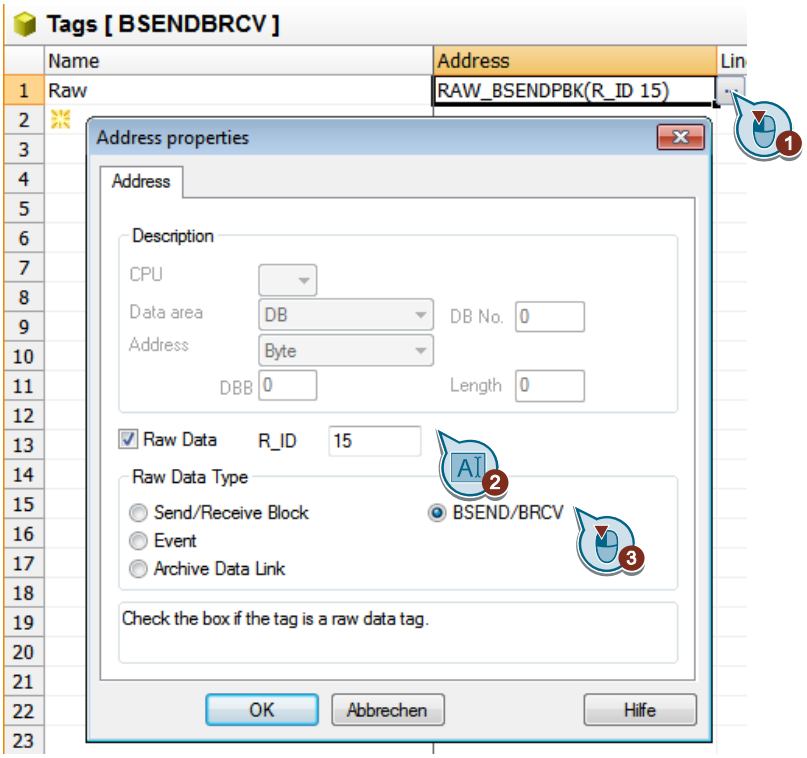
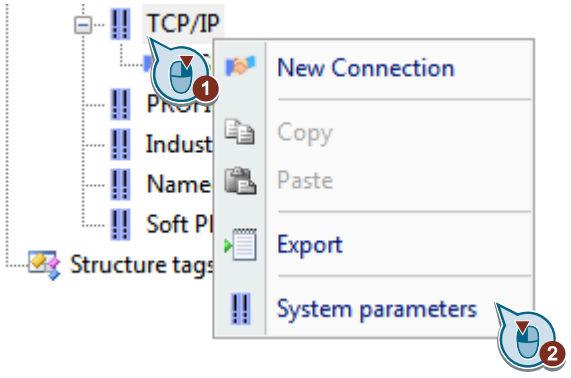
5.2 Establishing a connection to the S7-300 control system

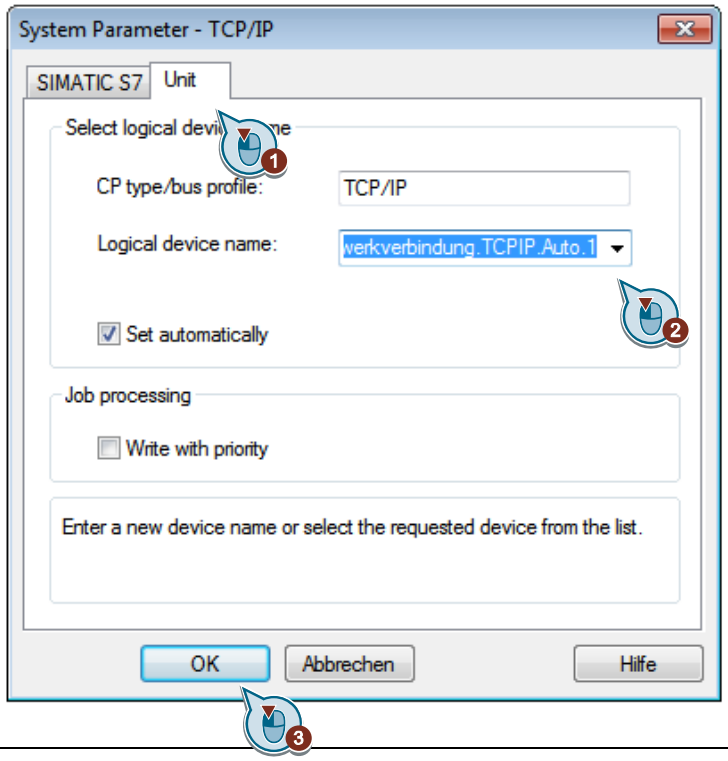
Make sure that the interface conforms to your Programming device / PC interface in Windows.

Table 5-2

Item	Action
1.	Open the “Tag Management” in WinCC. 
2.	In the Tag Management”, select “Add new driver” → SIMATIC S7 Protocol Suite“. 
3.	In the project tree view of the SIMATIC S7 Protocol Suite, select TCP/IP. Create a new „BSENDBRCV“ connection. 

Item	Action
4.	Open the Connection Parameters of the new connection. 
5.	Type the parameters of your CPU in the fields “IP address”, “Rack number” and “Slot number”. Tick the “Send/receive raw data block” option. Type the same value as in chapter 4.2 in the parameter “Connection resource”. Confirm the dialog with OK. 
6.	Create a “Raw” tag of data type „Raw Data Type“ using the new connection. 

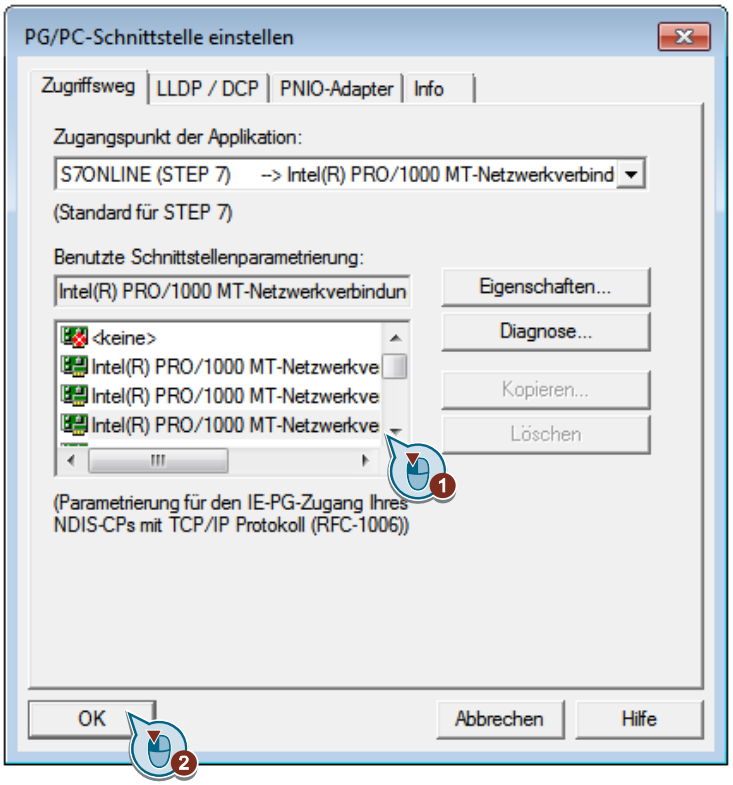
Item	Action
7.	In the “Address properties” window, tick the “BSEND/BRCV” option and type the R_ID from chapter 4.4. Confirm the dialog with OK. 
8.	In the project tree view of the WinCC explorer, tick again the “TCP/IP” entry and select “System parameters” from the context menu. 

Item	Action
9.	Within the “System parameters” window, access the “Unit” tab and select your interface in the “Logical device name” field. Confirm the dialog with OK. 

5.3 Setting the PG/PC interface

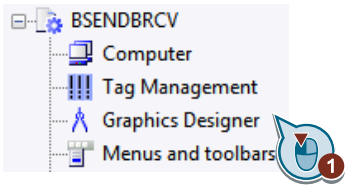
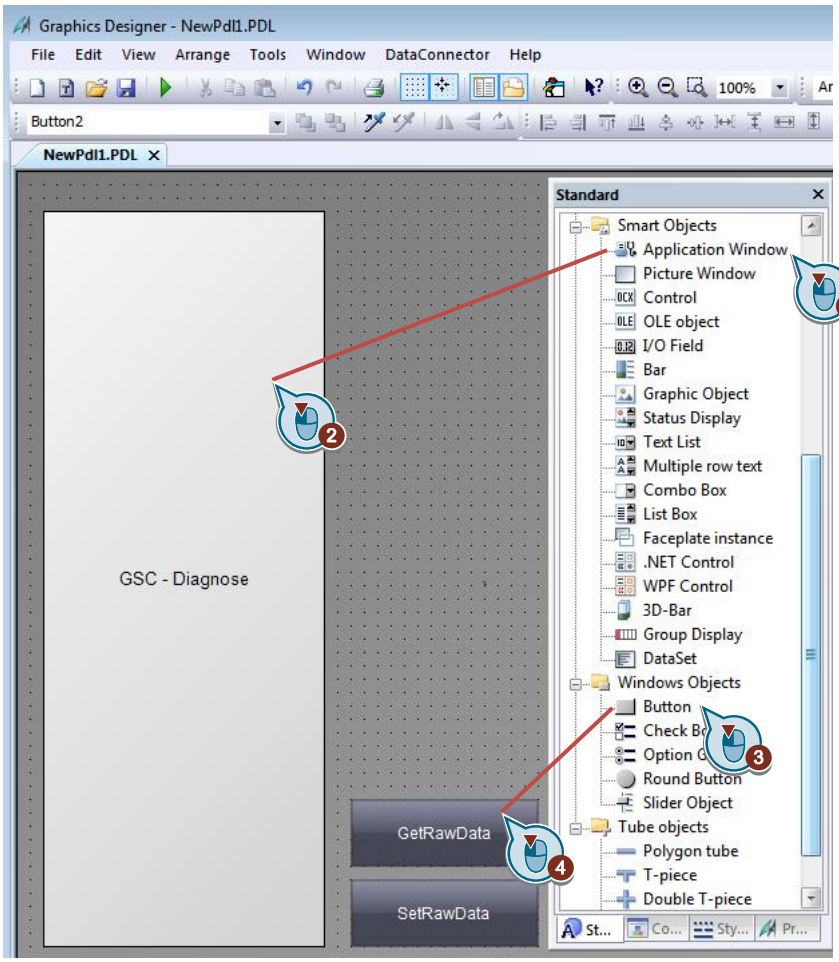
Further information on the PG/PC interface is available in entry ID: 79689088.

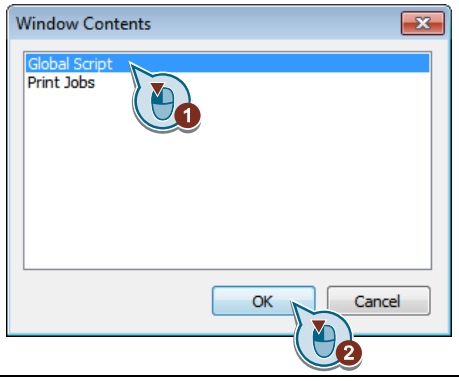
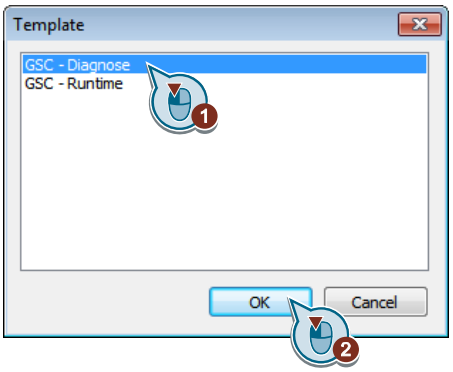
Table 5-3

Item	Action
1.	In Windows, select the PG/PC Interface entry in “Start → Control Panel → PG/PC Interface Setting”.
2.	In the “PG/PC Interface Setting” window, select the desired interface in the “Use interface parameterizing” option.  Confirm the dialog with OK.

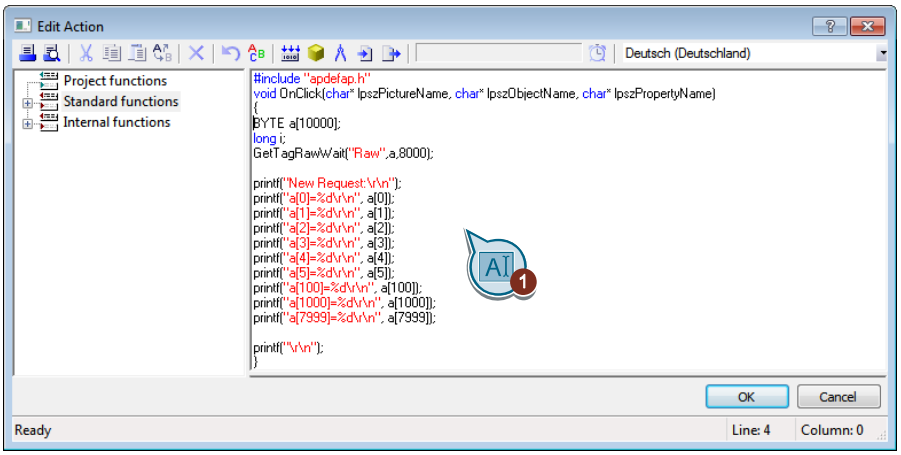
5.4 Creating C scripts

Table 5-4

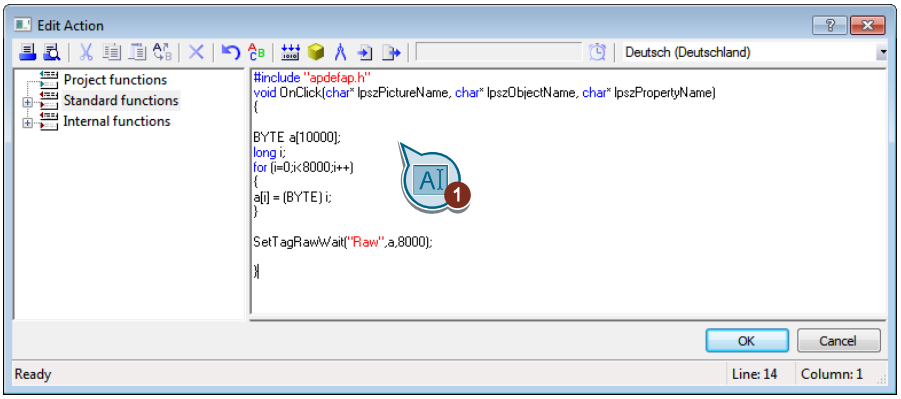
Item	Action
1.	Open the Graphics Designer. 
2.	Insert the „GetRawData“ and „SetRawData“ buttons into your screen using drag&drop. Insert an application window into your screen. 

Item	Action
3.	Once the application window inserted the pop-up window below opens. Select “Global Script” here. 
4.	In the next pop-up window, select “GSC-Diagnose”. 

Item	Action

5.	Add the script below to the „GetRawData“ button.
	 
	
	A byte array is being created within the script. The values from the “Raw” tag (chapter 5.2) are written into the array created by means of the „GetTagRawWait“ order. The single elements of the array are output using the “printf” function.
	
	BYTE a[10000];
	long i;
	GetTagRawWait("Raw",a,8000);
	
	printf("New Request:\r\n");
	printf("a[0]=%d\r\n", a[0]);
	printf("a[1]=%d\r\n", a[1]);
	printf("a[2]=%d\r\n", a[2]);
	printf("a[3]=%d\r\n", a[3]);
	printf("a[4]=%d\r\n", a[4]);
	printf("a[5]=%d\r\n", a[5]);
	printf("a[100]=%d\r\n", a[100]);
	printf("a[1000]=%d\r\n", a[1000]);
	printf("a[7999]=%d\r\n", a[7999]);
	
	printf("\r\n");

Item	Action

6.	Insert the script below into the „SetRawData“ button.
	 
	
	A byte array is being created within the script. A “For” loop is used to fill the array with values. The values from the array created are written into the tag variable “Raw” using the „SetTagRawWait“ order.
	
	BYTE a[10000];
	long i;
	for (i=0;i<8000;i++)
	{
	a[i] = (BYTE) i;
	}
	
	SetTagRawWait("Raw",a,8000);

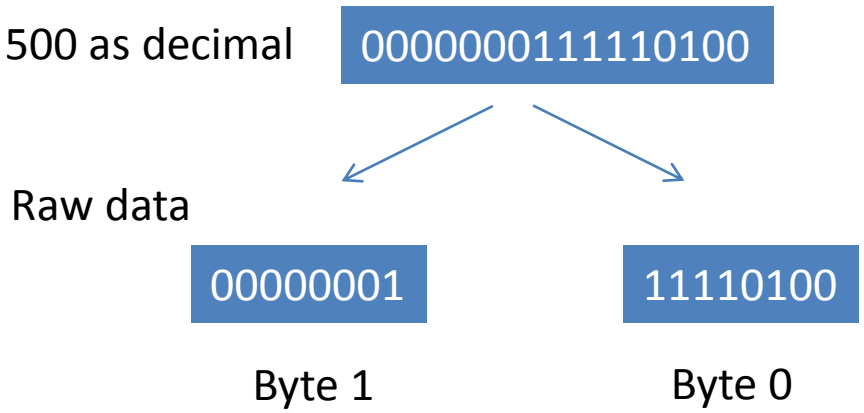
6 Converting raw data to other data types

6.1 Example from Byte to Word

Raw data are available in byte format in WinCC. To integrate other data types, like for example, Word, some single bytes must be pooled to build the data type Word.

Example: The figure 500 is saved as Word

Figure 6-1

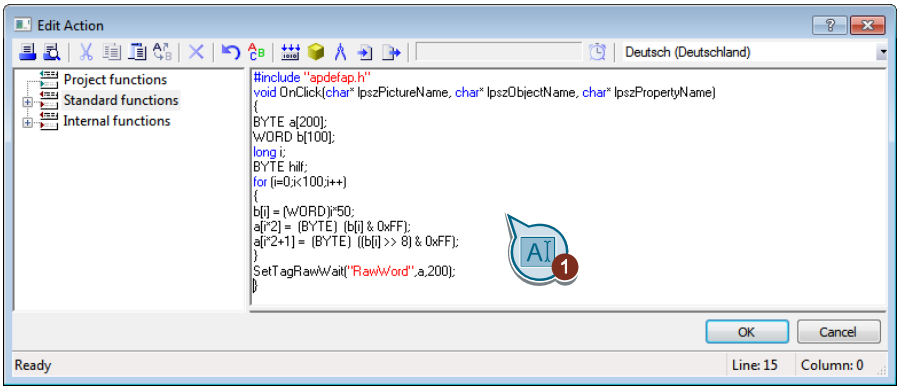


You must use a script to execute the separating and merging process.

6.1 Example from Byte to Word

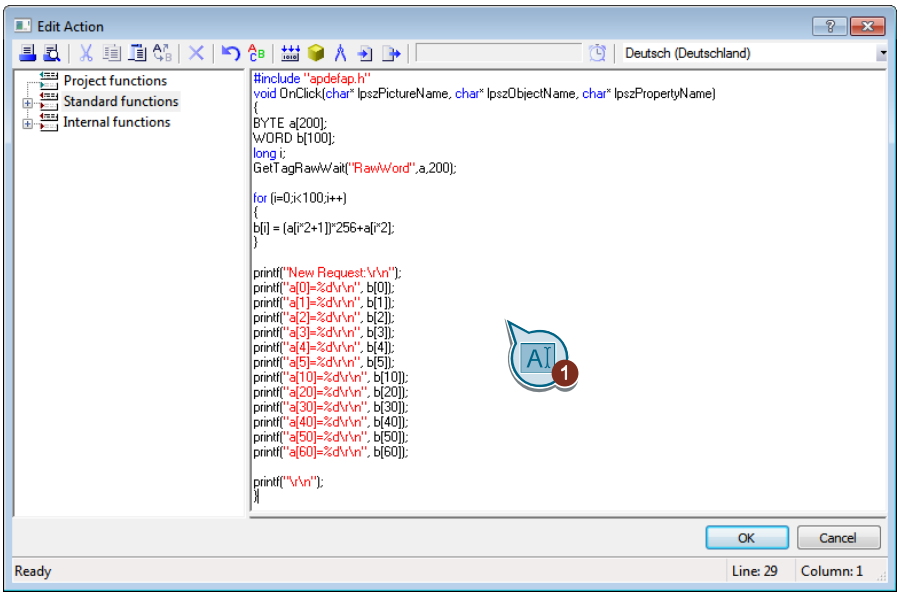
Table 6-1

Item	Action
1.	Add a new data block. Create a byte array like in chapter 4.3.
2.	Insert a „BSEND“ and „BRCV“ block like in chapter 4.4. Pay attention to assign a new R_ID.
3.	Establish a new connection to the data block in the Tag Management, following the instructions in chapter 5.2.

4.	Insert the following script into your project for sending data.
	 
	
	The script generates 100 tags in 50 steps with the help of a “for” loop. Byte 0 and byte 1 are masked out and written into a new array. This array is written as raw data to the control system using the „SetTagRawWait“ command.
	
	BYTE a[200];
	WORD b[100];
	long i;
	BYTE hilf;
	for (i=0;i<100;i++)
	{
	b[i] = (WORD)i*50;
	a[i*2] = (BYTE) (b[i] & 0xFF);
	a[i*2+1] = (BYTE) ((b[i] >> 8) & 0xFF);
	
	}
	
	SetTagRawWait("RawWord",a,200);

6.1 Example from Byte to Word

Item	Action
5.	Insert the following script into your project for receiving data.



The function „GetTagRawWait“ is used to read the raw data. Byte[0] and byte[1] are merged with the help of a “for” loop and output afterward.

```
BYTE a[200];
WORD b[100];
long i;
GetTagRawWait("RawWord",a,200);

for (i=0;i<100;i++)
{
b[i] = (a[i*2+1])*256+a[i*2];
}

printf("New Request:\r\n");
printf("a[0]=%d\r\n", b[0]);
printf("a[1]=%d\r\n", b[1]);
printf("a[2]=%d\r\n", b[2]);
printf("a[3]=%d\r\n", b[3]);
printf("a[4]=%d\r\n", b[4]);
printf("a[5]=%d\r\n", b[5]);
printf("a[10]=%d\r\n", b[10]);
printf("a[20]=%d\r\n", b[20]);
printf("a[30]=%d\r\n", b[30]);
printf("a[40]=%d\r\n", b[40]);
printf("a[50]=%d\r\n", b[50]);
printf("a[60]=%d\r\n", b[60]);

printf("\r\n");
```

7 Using the Application

Make sure that the CPU and WinCC are set to the same clock time (UTC or local time) before you start the application project for the first time (see chapter 4.5).

7.1 Commission the example project

Table 7-1

Item	Action
1.	Unzip the file „37873547_Rohdaten_TIA_S7-300.zip“
2.	Start the TIA Portal.
3.	Retrieve the „Rohdaten.zap13“ project.
4.	Download the project to the control system.
5.	Unzip the file „37873547_Rohdaten_WinCC_V74.zip“.
6.	Start the SIMATIC WinCC explorer.
7.	Open the file “BSENDBRCV.MCP“.
8.	Make fit the IP addresses of your PC station and your control system.
9.	Start Runtime.

7.2 Using the example project

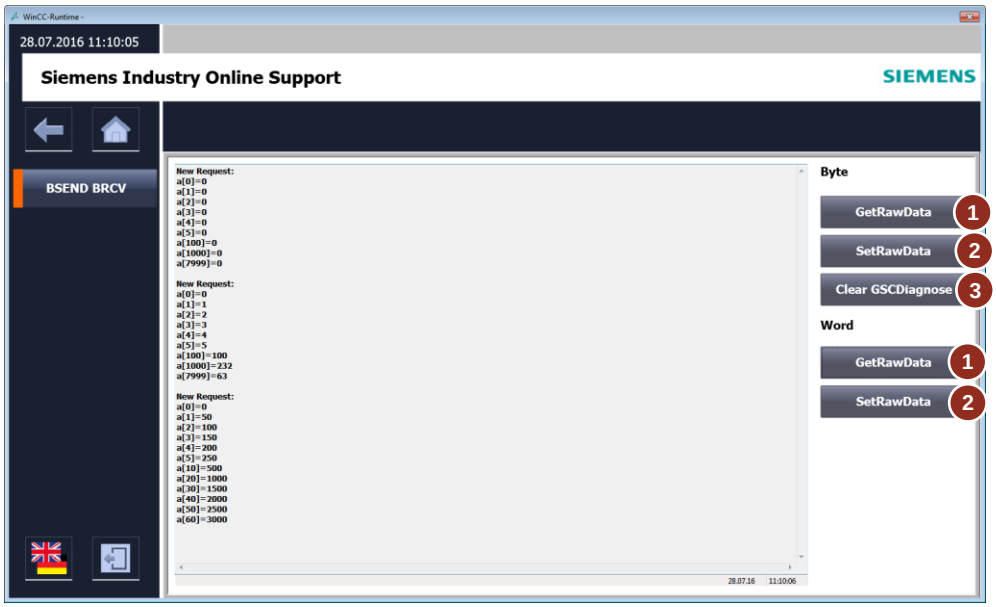


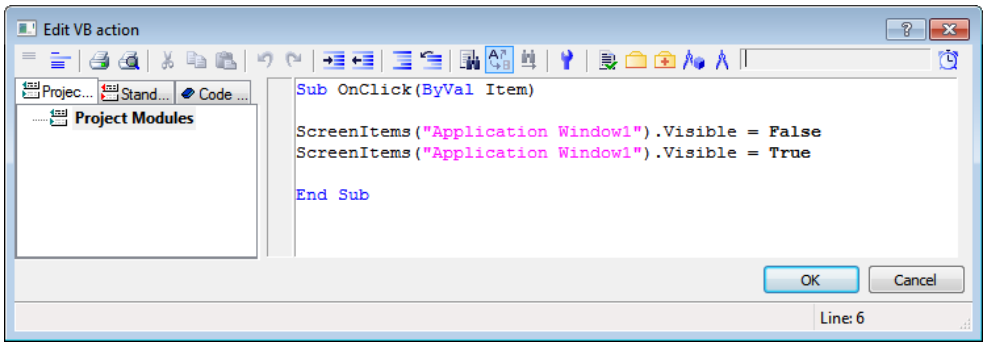
Table 7-2

Item	Action
1.	Use the „GetRawData“ button to receive raw data. Please, consider that the input “REQ” of the “BSEND” block must be triggered to “1”.
2.	Use the „SetRawData“ button to send raw data. Please, consider that the input “EN_R” of the “BRCV” block must be triggered to “1”.
3.	Use the “Clear GSCDiagnose” button to erase the contents of GSCDiagnostics.

8 Further Notes, Tips & Tricks, etc.

Erasing the GSC Diagnostics contents

GSC Diagnostics does not provide a button to erase text from the printf function in the control system. As a result, the continuous printf orders build a scroll bar in GSC Diagnostics and are filled with contents. If you wish to erase the contents, you can do it by showing/hiding for a short time the GSC Diagnostics window by means of a script. For this purpose insert the VB script below to a button in your project.



9 Links & References

Table 9-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/de/view/37873547

10 History

Table 10-1

Version	Date	Modifications
V1.0	08/2009	First version
V2.0	09/2016	Complete revision