

**SIEMENS**

*Ingenuity for life*

*Industry Online Support*

*Home*

# S7 Communication between SIMATIC S7-1200 and SIMATIC S7-300

STEP 7 V15.1 / STEP 7 V5.6

<https://support.industry.siemens.com/cs/ww/en/view/92269951>

Siemens  
Industry  
Online  
Support



---

This entry is from the Siemens Industry Online Support. The general terms of use ([http://www.siemens.com/terms\\_of\\_use](http://www.siemens.com/terms_of_use)) apply.

## **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 protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

The customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit <http://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase the customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under <http://www.siemens.com/industrialsecurity>.

---

# Contents

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
<b>2</b>	<b>Configuration and Programming of the SIMATIC S7-1200</b> .....	<b>5</b>
2.1	Create a Project .....	5
2.2	Configure the Hardware .....	7
2.2.1	Add a SIMATIC S7-1200 .....	7
2.2.2	Define IP Address and Assign Subnet .....	9
2.2.3	Permit Access with PUT/GET Communication from Remote Partner .....	10
2.2.4	Define Clock Memory .....	10
2.3	Configure the SIMATIC S7-1200 as Server .....	11
2.3.1	Create a User Program .....	11
2.4	Configure the SIMATIC S7-1200 as Client .....	17
2.4.1	Configure the S7 Connection .....	17
2.4.2	Create a User Program .....	21
2.5	Download the Hardware Configuration and User Program .....	26
<b>3</b>	<b>Configuration and Programming of the SIMATIC S7-300</b> .....	<b>30</b>
3.1	Create a STEP 7 project .....	30
3.2	Configure the Hardware .....	31
3.2.1	Open the Hardware Configuration .....	31
3.2.2	Define IP Address and Assign Subnet .....	32
3.2.3	Define the Clock Memory .....	33
3.3	Configure the SIMATIC S7-300 as Client .....	35
3.3.1	Configure the S7 Connection .....	35
3.3.2	Create a User Program .....	40
3.4	Configure the SIMATIC S7-300 as Server .....	45
3.4.1	Create a User Program .....	45
3.5	Download the Hardware Configuration and User Program .....	49
<b>4</b>	<b>Operating and Monitoring</b> .....	<b>51</b>
4.1	SIMATIC S7-300 Writes and Reads Data as Client .....	51
4.1.1	Write Data to the S7-1200 CPU .....	51
4.1.2	Read Data from the S7-1200 CPU .....	55
4.2	SIMATIC S7-1200 Writes and Reads Data as Client .....	58
4.2.1	Write Data to the S7-300 .....	58
4.2.2	Read Data from the S7-300 .....	62

# 1 Introduction

This document shows you how to exchange data between an S7-1200 and an S7-300 using S7 Communication.

The SIMATIC S7-1200 is configured and programmed in STEP 7 V15.1.

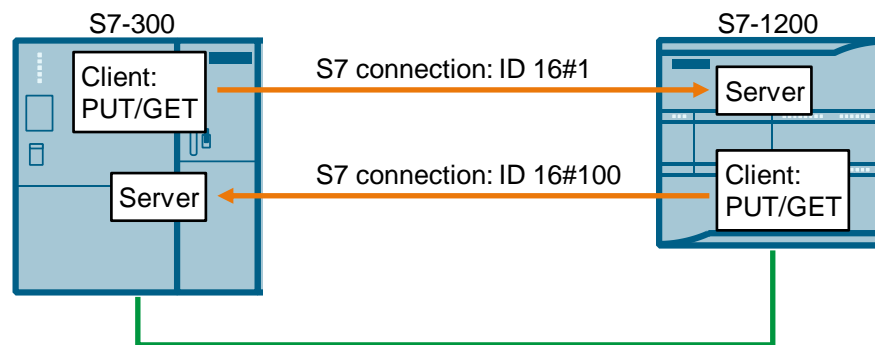
The SIMATIC S7-300 is configured and programmed in STEP 7 V5.6.

Two S7 connections are established.

For the first S7 connection the S7-300 acts as a client and the S7-1200 as server. This means that the S7-300 actively establishes the connection. The function blocks FB14 "GET" and FB 15 "PUT" are called in the user program of the S7-300 to read data from the S7-1200 and write data to the S7-1200.

For the second S7 connection the S7-1200 acts as a client and the S7-300 as server. This means that the S7-1200 actively establishes the connection. The instructions "GET" and "PUT" are called in the user program of the S7-1200 to read data from the S7-300 and write data to the S7-300.

Figure 1-1



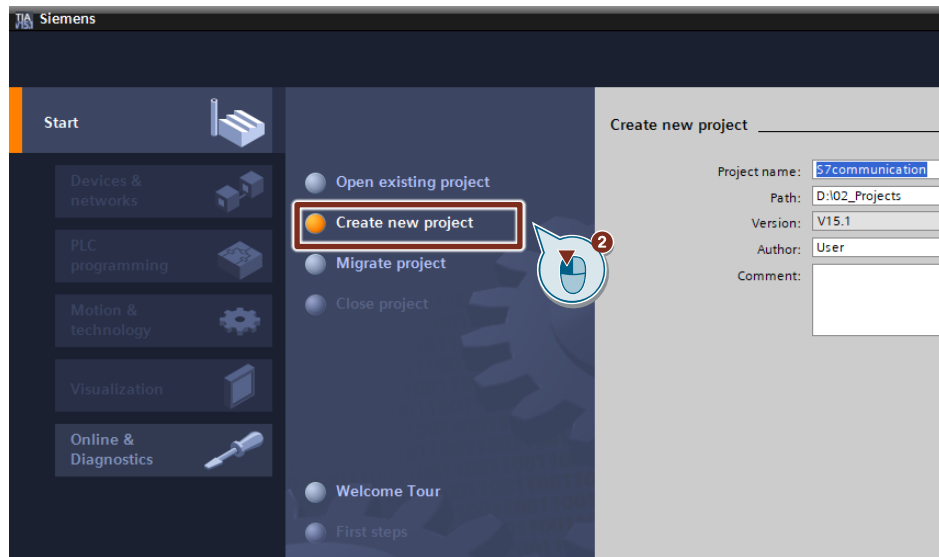
## 2 Configuration and Programming of the SIMATIC S7-1200

You configure and program the SIMATIC S7-1200 in STEP 7 V15.1 (TIA Portal V15.1).

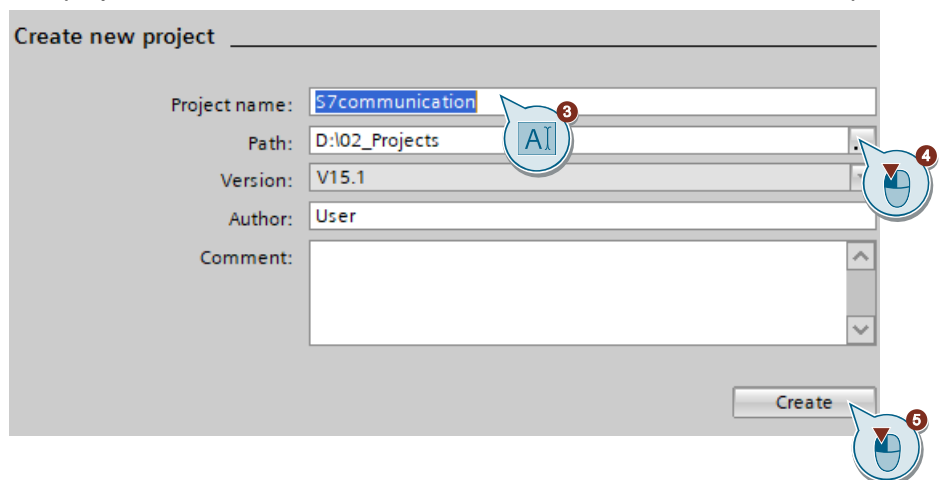
Then you create the user program and define which data is to be exchanged with the S7-300 via the S7 connection.

### 2.1 Create a Project

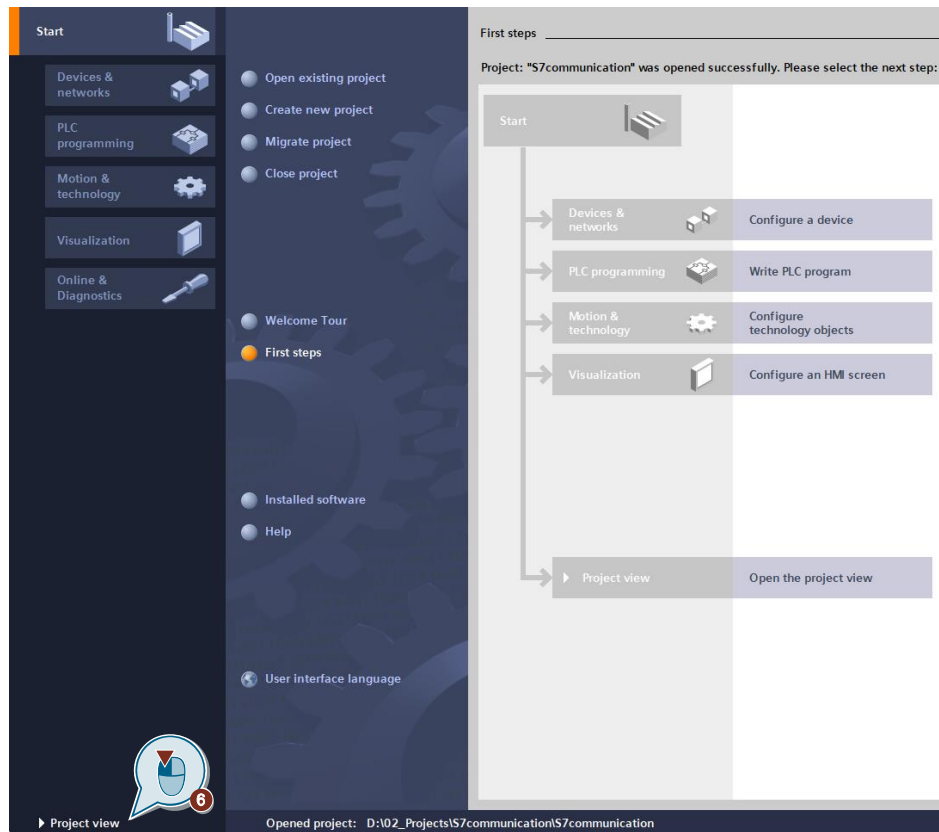
1. Open STEP 7 V15.1 (TIA Portal V15.1).
2. In the Portal View, select the "Create new project" action.



3. Enter a project name, "S7communication", for example.
4. Select a path in which the project is to be stored.
5. Click the "Create" button.  
The project with the name "S7communication" is stored in the selected path.



6. Click the "Project view" link to switch to the Project View.

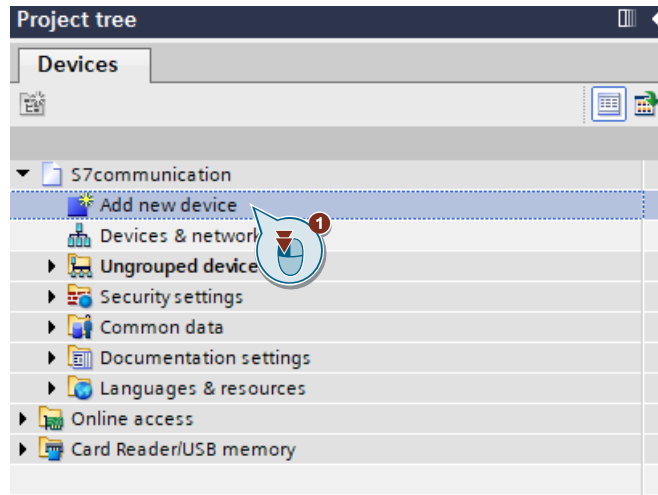




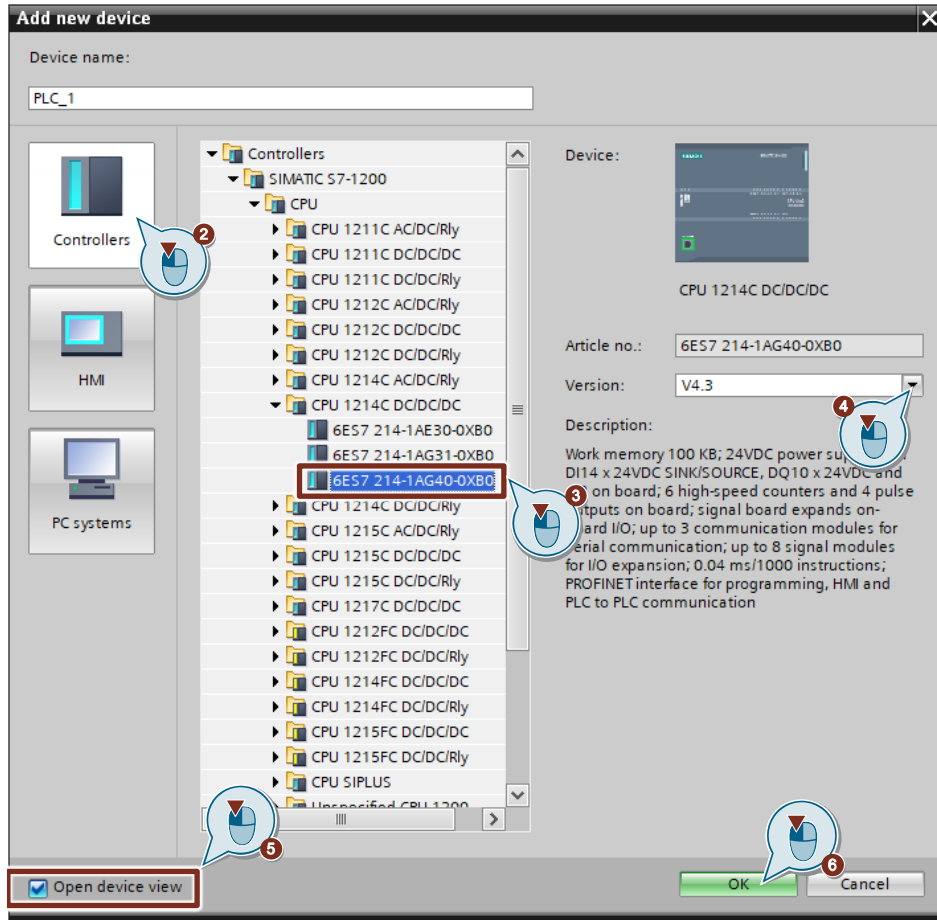
## 2.2 Configure the Hardware

### 2.2.1 Add a SIMATIC S7-1200

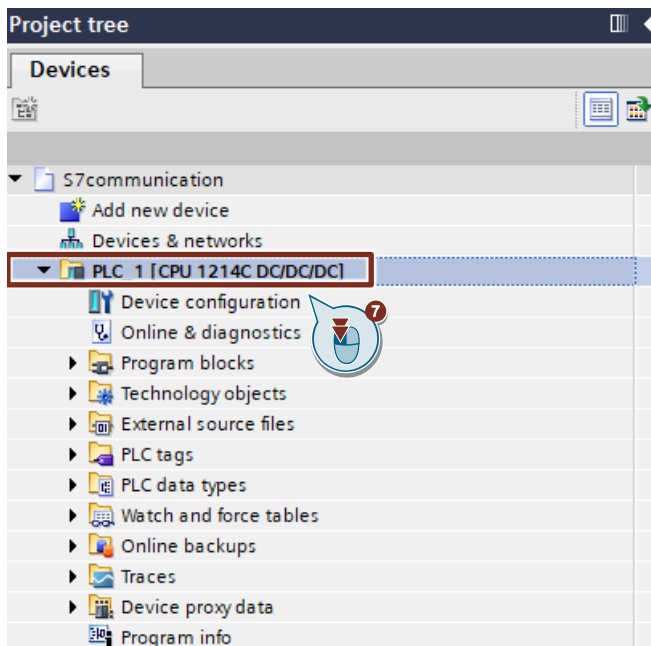
1. In the project tree you double-click the "Add new device" item. The "Add new device" dialog opens.



2. In the working area you click the "Controllers" button.
3. Under "Controllers > SIMATIC S7 1200 > CPU" you select the required S7-1200 CPU.
4. As required, you can change the firmware version that is to be used to configure the selected S7-1200 CPU.
5. Enable the "Open device view" function.
6. Click the "OK" button to apply the settings and close the dialog.
  - If the "Open device view" function is enabled, the device view of the S7-1200 CPU opens in the "Devices & networks" editor.
  - The device folder of the S7-1200 CPU is displayed in the project tree.



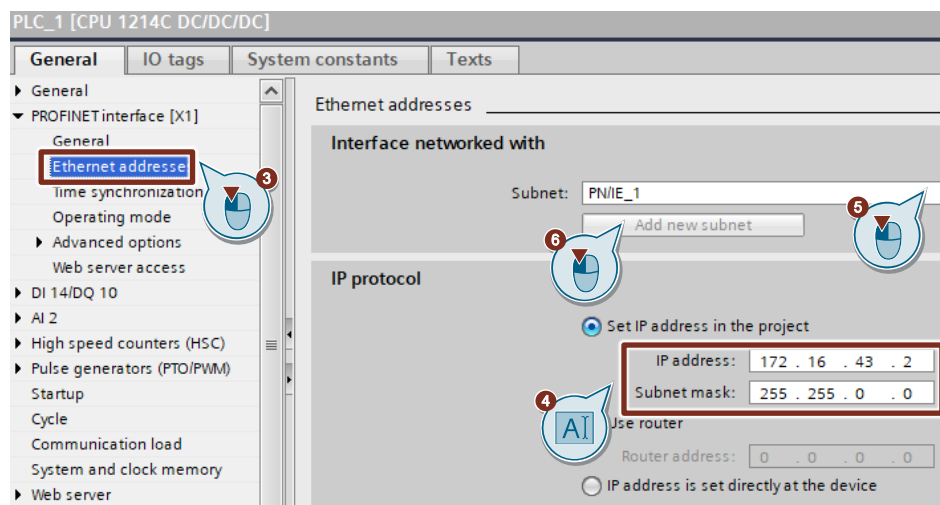
- Alternatively, you can open the device view of the S7-1200 CPU in the "Devices & networks" editor by opening the device folder of the S7-1200 CPU and double-clicking the "Device configuration" item.





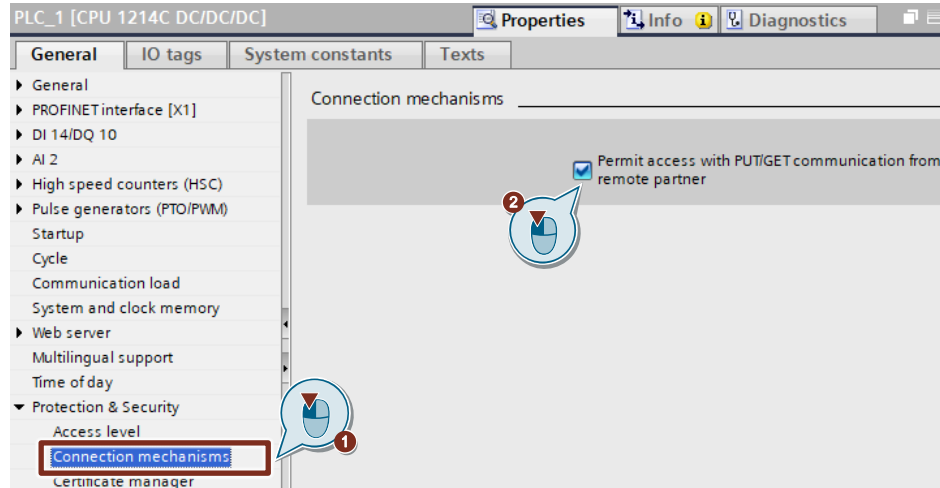
### 2.2.2 Define IP Address and Assign Subnet

1. In the device view or network view of the "Devices & networks" editor you mark the S7-1200 CPU.
2. In the inspector window you select the "Properties" tab.
3. In the "General" tab you select "PROFINET interface [X1] > Ethernet addresses".
4. In this example you enter the IP address and the subnet mask for the PROFINET interface of the S7-1200 CPU as follows:
  - IP address: 172.16.43.2
  - Subnet mask: 255.255.0.0
5. Assign a subnet to the PROFINET interface.
6. Click the "Add new subnet" button to create a new subnet.



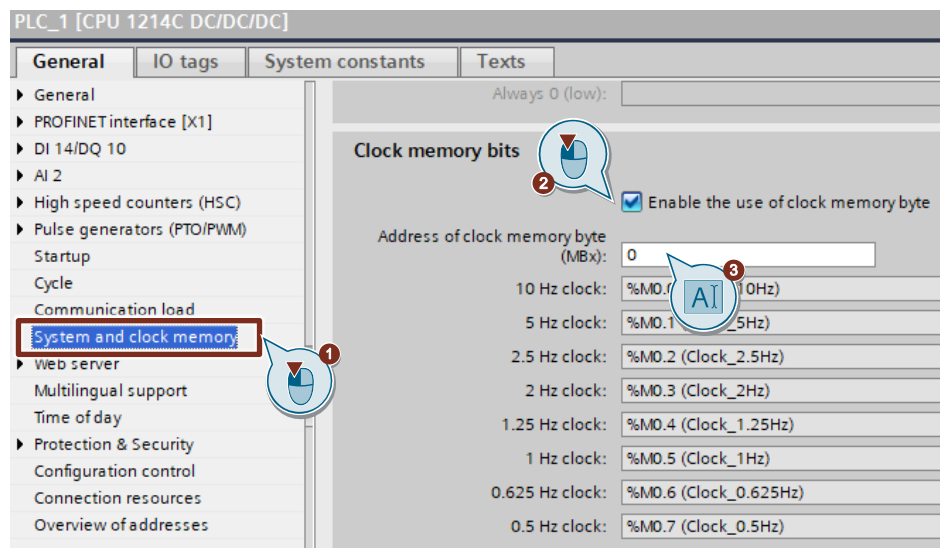
### 2.2.3 Permit Access with PUT/GET Communication from Remote Partner

1. In the "General" tab you select "Protection & Security > Connection mechanisms".
2. Enable the "Permit access with PUT/GET communication from remote partner" function.



### 2.2.4 Define Clock Memory

1. In the "General" tab you select "System and clock memory".
2. Enable the "Enable the use of clock memory byte" function.
3. Enter the address of the clock memory byte, "0", for example.



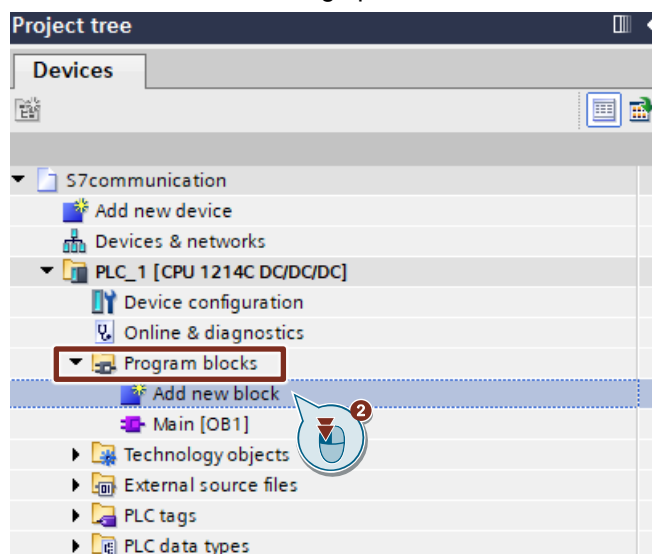
## 2.3 Configure the SIMATIC S7-1200 as Server

### 2.3.1 Create a User Program

In the user program of the S7-1200 CPU you add the data blocks for saving the Send and Receive data.

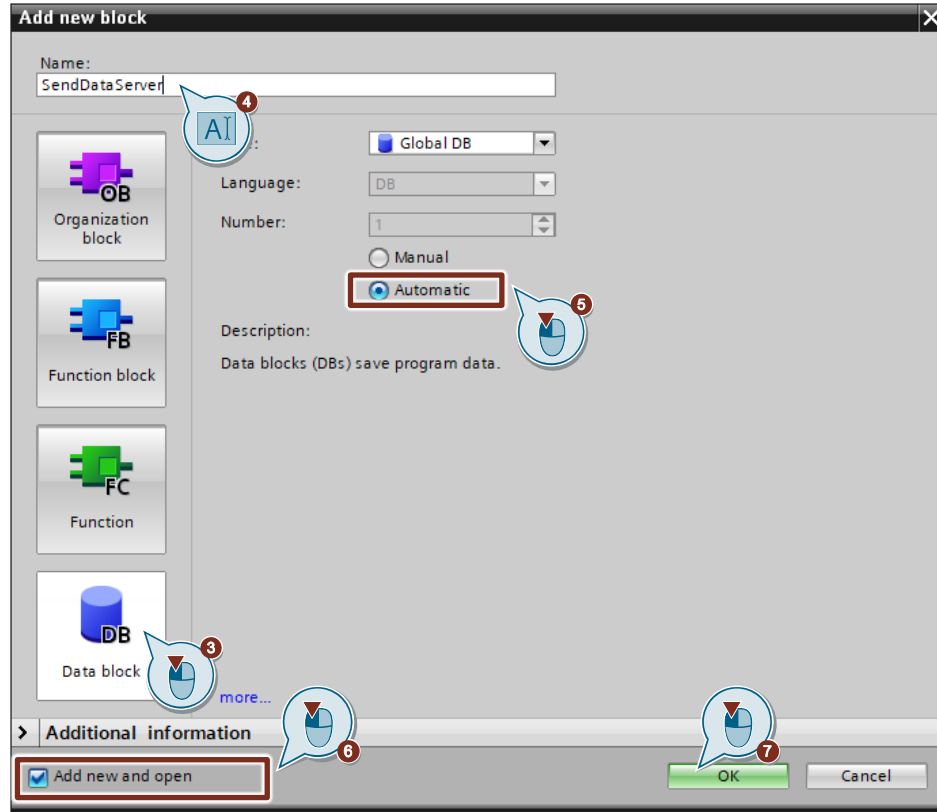
#### Add data block for saving the Send data

1. In the project tree you navigate to the device folder of the S7-1200 CPU. The device folder contains structured objects and actions that belong to the device.
2. In the device folder you navigate to the "Program blocks" subfolder and double-click the "Add new block" action. The "Add new block" dialog opens.

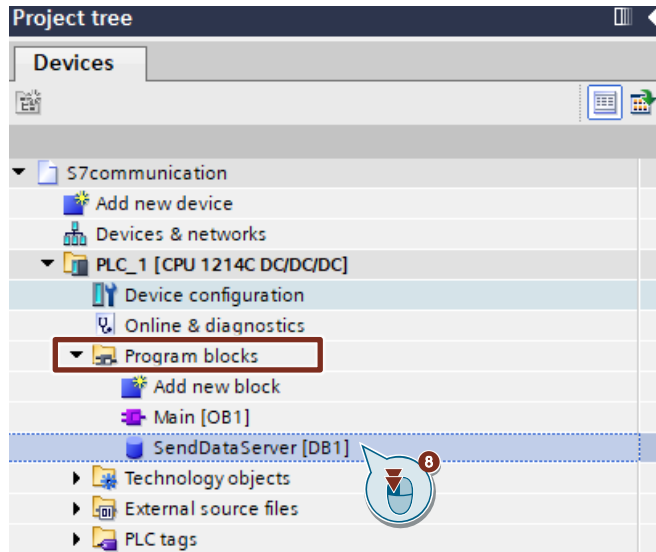


3. Click the "Data block (DB)" button.
4. Enter the name of the data block (DB), "SendDataServer", for example.
5. Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
6. Enable the "Add new and open" function.

7. Click the "OK" button to apply the settings and close the dialog.
  - In the project tree the "SendDataServer" DB is added to the "Program blocks" folder of the S7-1200 CPU.
  - If the "Add new and open" function is enabled, the "SendDataServer" DB opens in the working area of STEP 7 V15.1.



8. Alternatively, you can open the "SendDataServer" DB in the working area of STEP 7 V15.1 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "SendDataServer" DB.

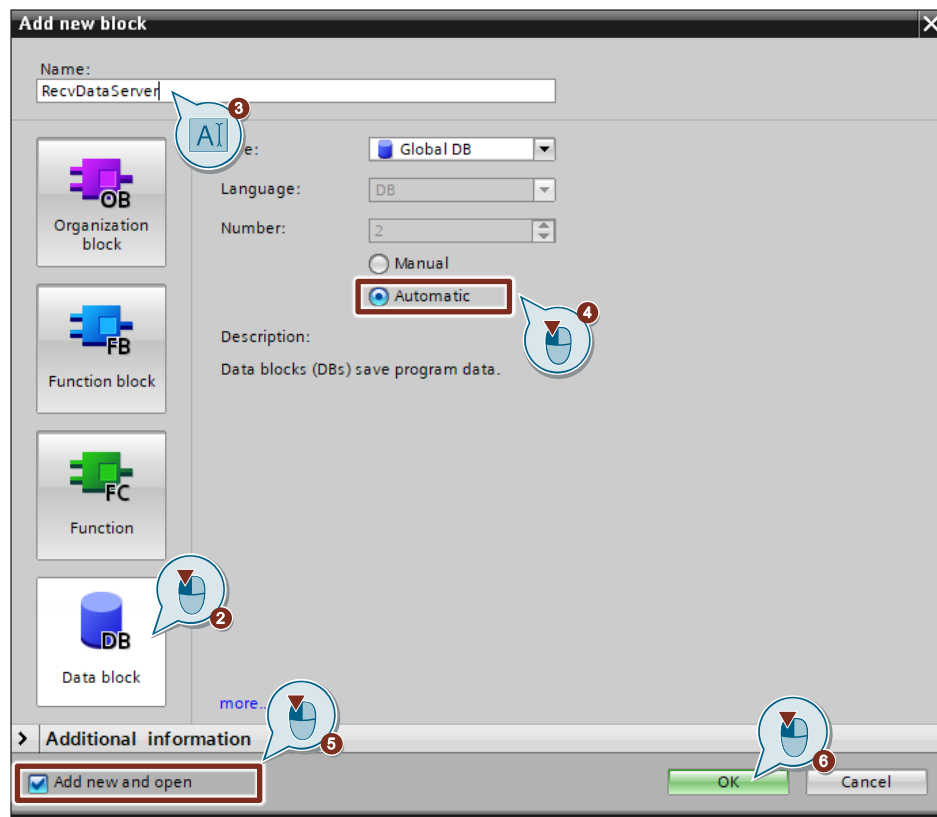


9. In the "SendDataServer" DB you define the static variable "sendData" of the data type Array[0..99] of Byte.

SendDataServer				
	Name	Data type	Start value	Retain
1	Static			<input type="checkbox"/>
2	sendData	Array[0..99] of Byte		<input type="checkbox"/>
3	<Add new>			<input type="checkbox"/>

### Add data block for saving the Receive data

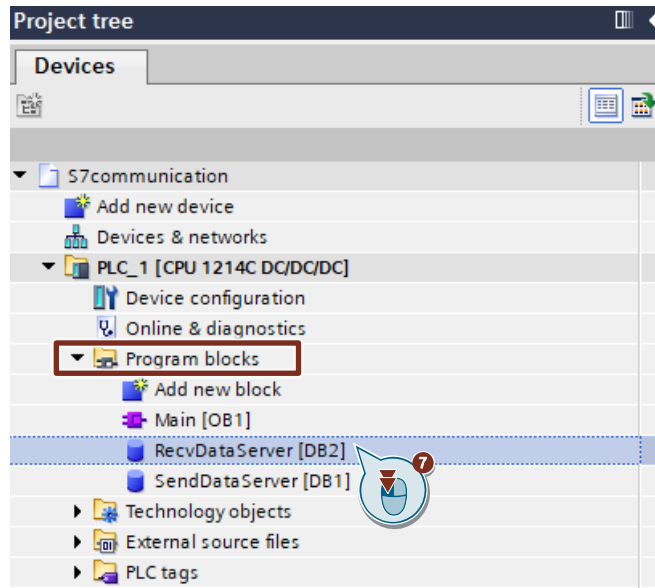
1. Add another DB for saving the Receive data.
2. In the "Add new block" dialog you click the "Data block (DB)" button.
3. Enter the name of the DB, "RecvDataServer", for example.
4. Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
5. Enable the "Add new and open" function.
6. Click the "OK" button to apply the settings and close the dialog.
  - In the project tree the "RecvDataServer" DB is added to the "Program blocks" folder of the S7-1200 CPU.
  - If the "Add new and open" function is enabled, the "RecvDataServer" DB opens in the working area of STEP 7 V15.1.





- Alternatively, you can open the "RecvDataServer" DB in the working area of STEP 7 V15.1 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "RecvDataServer" DB.

Figure 2-1



- In the "RecvDataServer" DB you define the static variable "recvData" of the data type Array[0..99] of Byte.

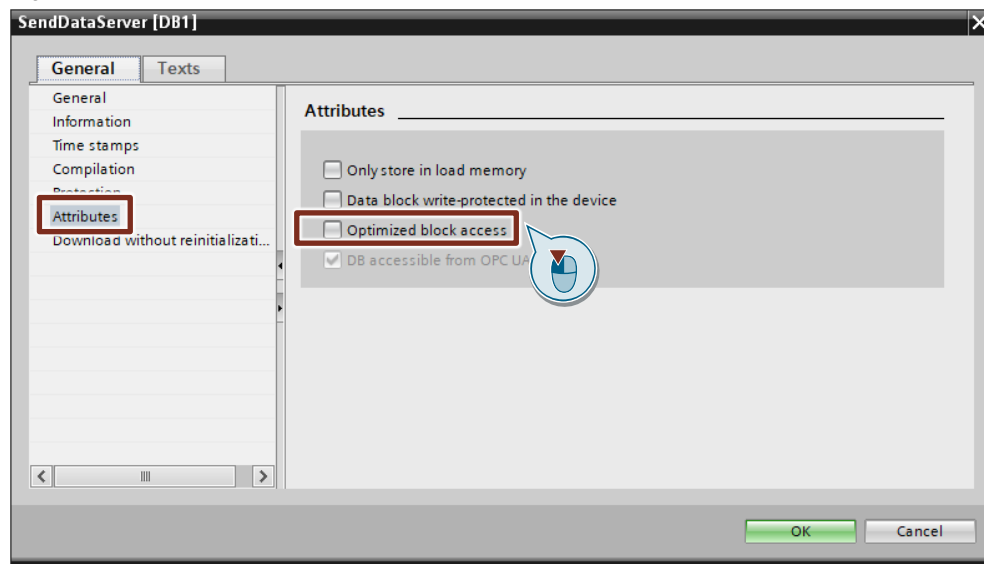
RecvDataServer				
	Name	Data type	Start value	Retain
1	Static			<input type="checkbox"/>
2	recvData	Array[0..99] of Byte		<input type="checkbox"/>
3	<Add new>			<input type="checkbox"/>

### Disable the "Optimized block access" option

Because the communication partner, SIMATIC S7-300, for example, does not support DBs with optimized block access, you have to disable the "Optimized block access" option under "Attributes" in the Properties of the DB.

DBs with standard access have a fixed structure. The data elements in the declaration include both a symbolic name and a fixed address in the block. The address is displayed in the "Offset" column. You can address the variables in this block both symbolically and absolutely.

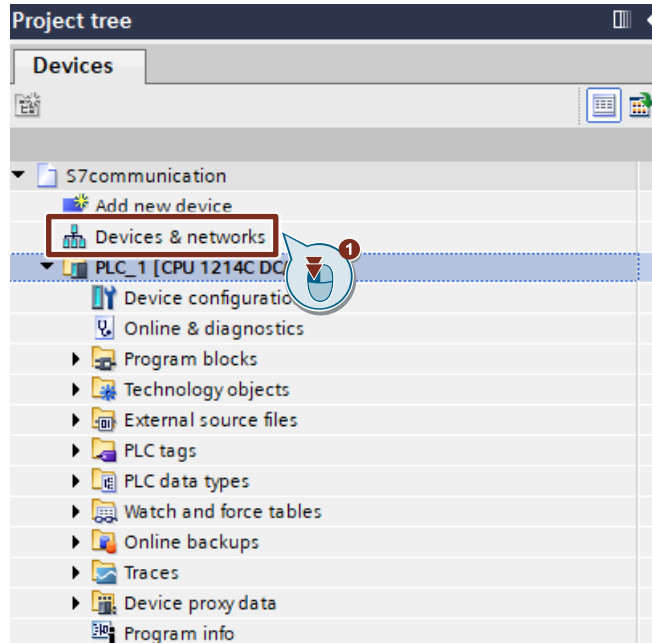
Figure 2-2



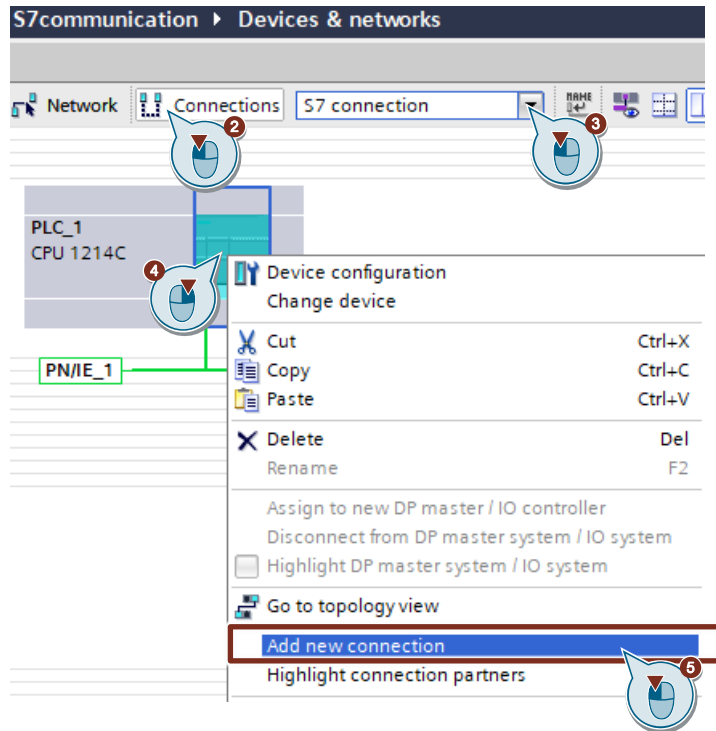
## 2.4 Configure the SIMATIC S7-1200 as Client

### 2.4.1 Configure the S7 Connection

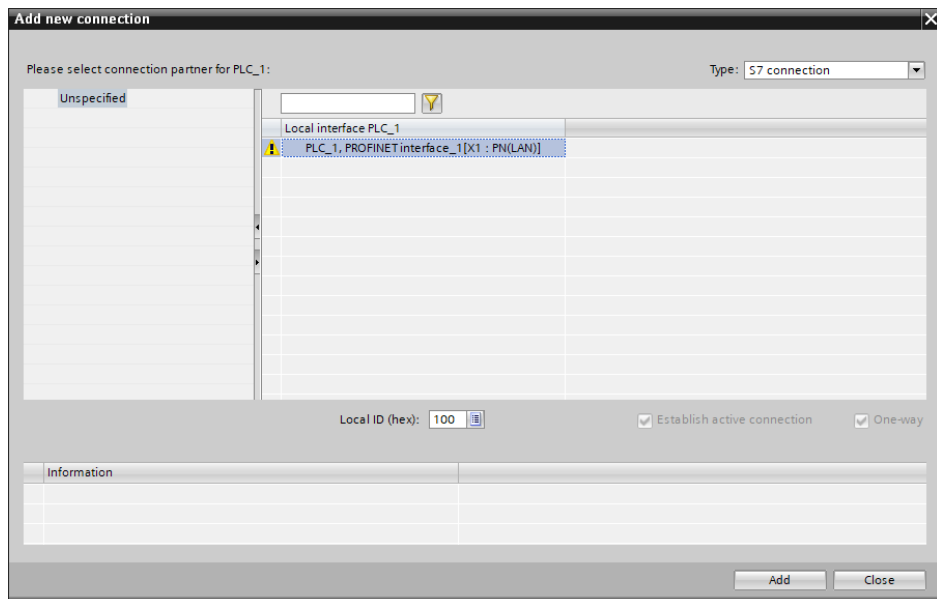
1. In the project tree you double-click the "Devices & networks" item. The network view opens in the "Devices & networks" editor.



2. Click the "Connections" icon in the toolbar.
3. Select "S7 connection" as the connection type in the adjacent drop-down list box. The S7-1200 CPU is displayed highlighted in color in the network view.
4. Right-click the S7-1200 CPU. The pop-up menu opens.
5. Select the "Add new connection" item. The "Add new connection" dialog opens.

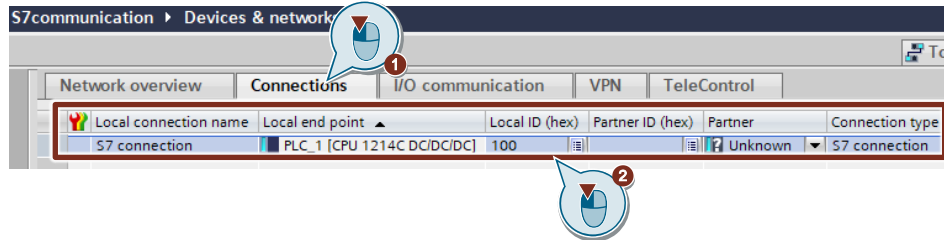


6. Click the "Add" button. An S7 connection is added.
7. Click the "Close" button to close the dialog.

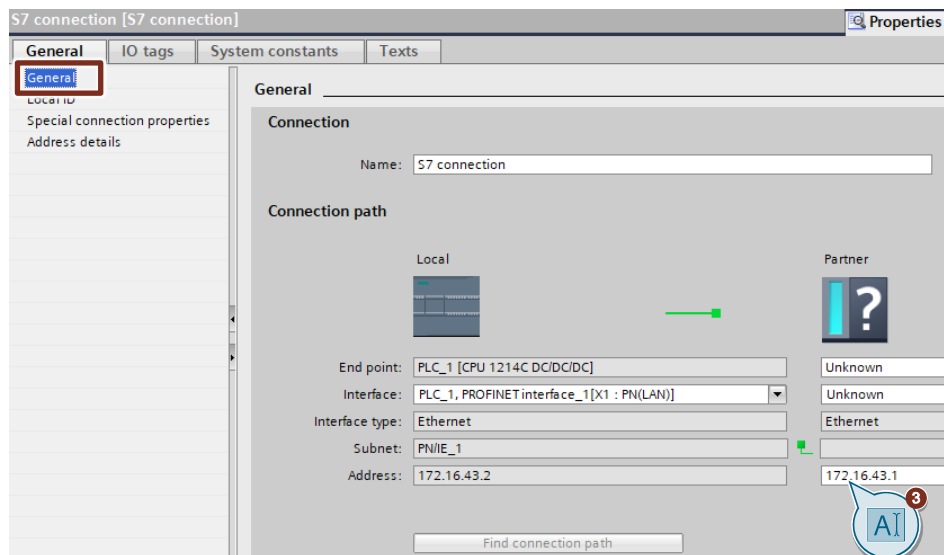


### Define Connection Parameters

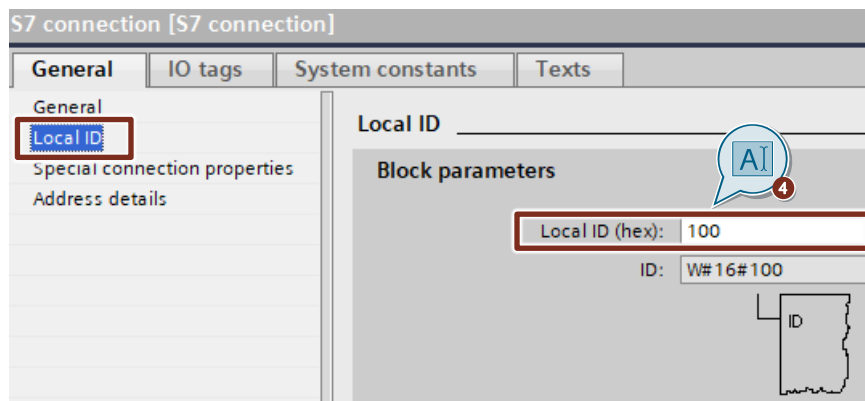
1. In the table area of the network view you open the "Connections" tab.
2. Mark the S7 connection. The properties of the S7 connection are displayed in the inspector window.



3. In the "General" tab you select the "General" item. Enter the IP address of the partner CPU. In this example you enter the IP address of the S7-300 CPU: 172.16.43.1.

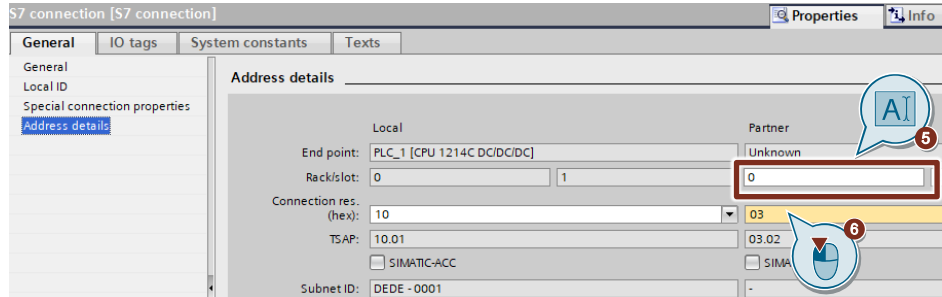


4. In the "General" tab you select the "Local ID" item. You specify the local ID of the connection in the user program at the input parameter "ID" of the instructions "GET" and "PUT".

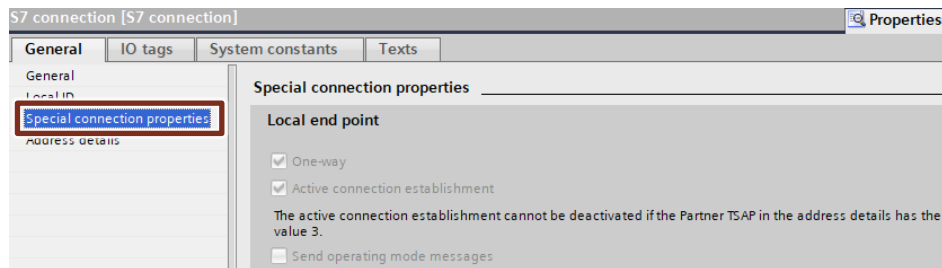


## 2 Configuration and Programming of the SIMATIC S7-1200

5. In the "General" tab you select the "Address details" item. Enter the rack and slot of the partner CPU. The S7-300 CPU uses Rack 0 and Slot 2.
6. Select the connection resource 03(hex) for the partner CPU because the S7 connection is only configured one-sided in the S7-1200 CPU. With these settings the TSAP has the value 3 in the partner CPU.



7. In the "General" tab you select the "Special connection properties" item. If the TSAP has the value 3 in the partner CPU, the "Active connection establishment" function is enabled automatically and cannot be disabled.



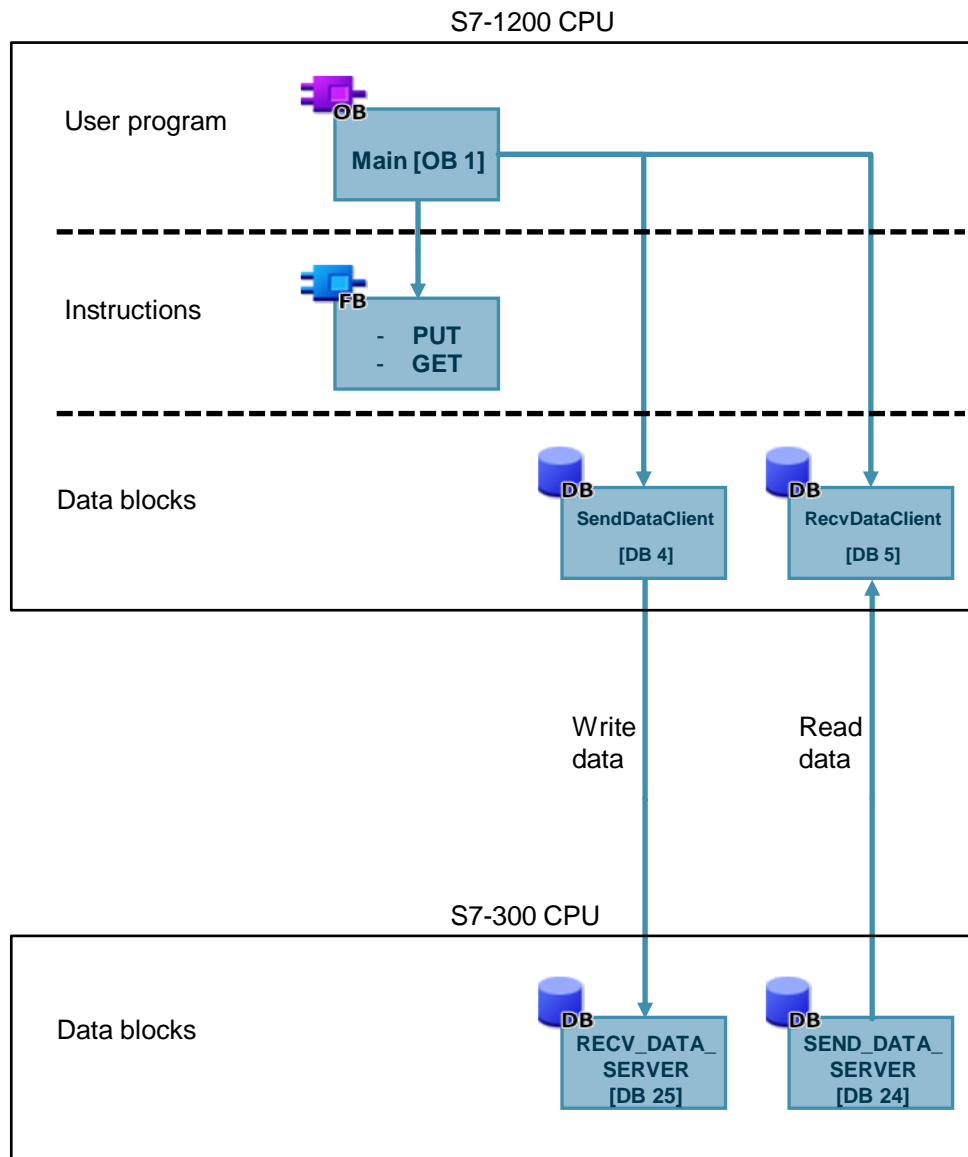


### 2.4.2 Create a User Program

In the user program of the S7-1200 CPU you call the instructions "GET" and "PUT" for data transfer. The instructions are in the "Instructions" task card under "Communication > S7 Communication". Add the instructions by drag-and-drop to the "Program blocks" folder of your S7-1200 CPU.

The following figure shows the block calls in the S7-1200 and S7-300 CPUs.

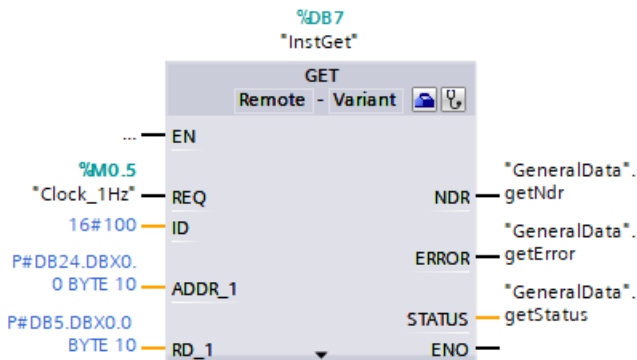
Figure 2-3



### "GET" instruction

You use the "GET" instruction to read data from the S7-300 CPU.

Figure 2-4



The "GET" instruction has the following input parameters.

Table 2-1

Input parameters	Data type	Description
REQ	BOOLEAN	The "REQ" control parameter enables data transfer on a rising edge. In this example the job to read the data is enabled via the clock memory M0.5 "Clock_1Hz".
ID	WORD	Reference to the local connection description (preset by the connection configuration in STEP 7). In this example the ID = w#16#1 is preset by the configuration connection.
ADDR_1	ANY	Pointer to the area to be read in the partner CPU (Send data area in the partner CPU, here S7-300 CPU).
RD_1	ANY	Pointer to the area in your own CPU in which the read data is stored (Receive data area in your own CPU, here S7-1200 CPU).

The "GET" instruction has the following output parameters.

Table 2-2

Output parameters	Data type	Description
NDR	BOOL	State parameter NDR: 0 = Job not yet started or is still being executed 1 = Job executed error-free
ERROR	BOOL	State parameters ERROR and STATUS: <ul style="list-style-type: none"> <li>ERROR = 0:                             <ul style="list-style-type: none"> <li>STATUS = 0000(hex): neither warning nor error</li> <li>STATUS &lt;&gt; 0000(hex): warning, STATUS provides detailed information</li> </ul> </li> <li>ERROR = 1: A fault has occurred. STATUS provides detailed information about the type of error.</li> </ul>
STATUS	WORD	

### Receive data area in the S7-1200 CPU

In the S7-1200 CPU the Receive data read from the S7-300 CPU is stored in data block DB5 "RecvDataClient".

In this example 10 bytes of data starting at address 0 are stored in DB5 "RecvDataClient".

### Send data area in the S7-300 CPU

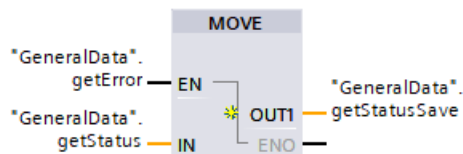
In the S7-300 CPU the data read from the S7-1200 CPU is stored in data block DB24 "SEND\_DATA\_SERVER".

In this example 10 bytes of data starting at address 0 are read from the data block DB24 "SEND\_DATA\_SERVER" of the S7-300 CPU.

### Error evaluation of the Read job

If the Read job does not terminate successfully, in other words ERROR =1, the value of the output parameter STATUS is stored in a global data block.

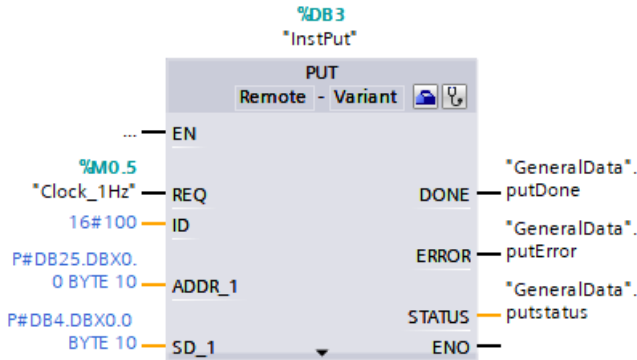
Figure 2-5



### "PUT" instruction

You use the "PUT" instruction to write data to the S7-300 CPU.

Figure 2-6



The "PUT" instruction has the following input parameters.

Table 2-3

Input parameters	Data type	Description
REQ	BOOL	The "REQ" control parameter enables data transfer on a rising edge. In this example the job to write the data is enabled via the clock memory M0.5 "Clock_1Hz".
ID	WORD	Reference to the local connection description (preset by the connection configuration in STEP 7). In this example the ID = w#16#1 is preset by the configuration connection.
ADDR_1	ANY	Pointer to the area to be written to in the partner CPU (Receive data area in the partner CPU, here S7-300 CPU).
SD_1	ANY	Pointer to the area in your own CPU which contains the data to be sent (Send data area in your own CPU, here S7-1200 CPU).

The "PUT" instruction has the following output parameters.

Table 2-4

Input parameters	Data type	Description
DONE	BOOL	State parameter DONE: 0 = Job not yet started or is still being executed 1 = Job executed error-free
ERROR	BOOL	State parameters ERROR and STATUS: <ul style="list-style-type: none"> <li>ERROR = 0:                             <ul style="list-style-type: none"> <li>STATUS = 0000(hex): neither warning nor error</li> <li>STATUS &lt;&gt; 0000(hex): warning, STATUS provides detailed information</li> </ul> </li> <li>ERROR = 1: A fault has occurred. STATUS provides detailed information about the type of error.</li> </ul>
STATUS	WORD	

**Send data area in the S7-1200 CPU**

In the S7-1200 CPU the Send data written to the S7-300 CPU is stored in data block DB4 "SendDataClient".

In this example 10 bytes of data starting at address 0 are written from DB4 "SendDataClient" to the S7-300 CPU.

**Receive data area in the S7-300 CPU**

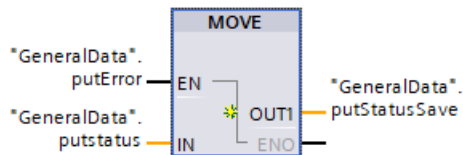
In the S7-300 CPU the received data is stored in data block DB25 "RECV\_DATA\_SERVER".

In this example 10 bytes of data starting at address 0 are stored in DB25 "RECV\_DATA\_SERVER".

**Error evaluation of the Write job**

If the Write job does not terminate successfully, in other words ERROR =1, the value of the output parameter STATUS is stored in a global data block.

Figure 2-7



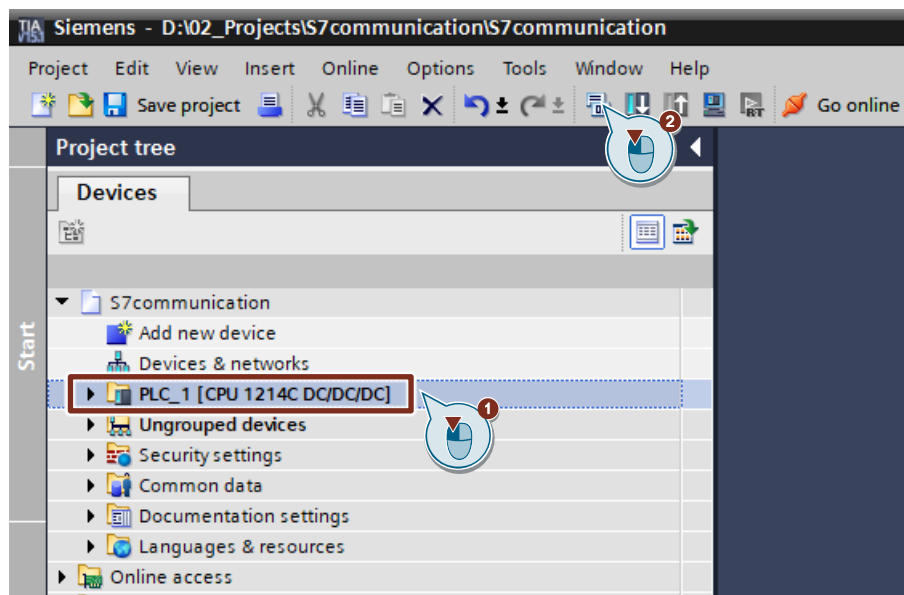
## 2.5 Download the Hardware Configuration and User Program

### Requirements

You have already assigned the configured IP address and subnet mask to the S7-1200 CPU.

### Compile

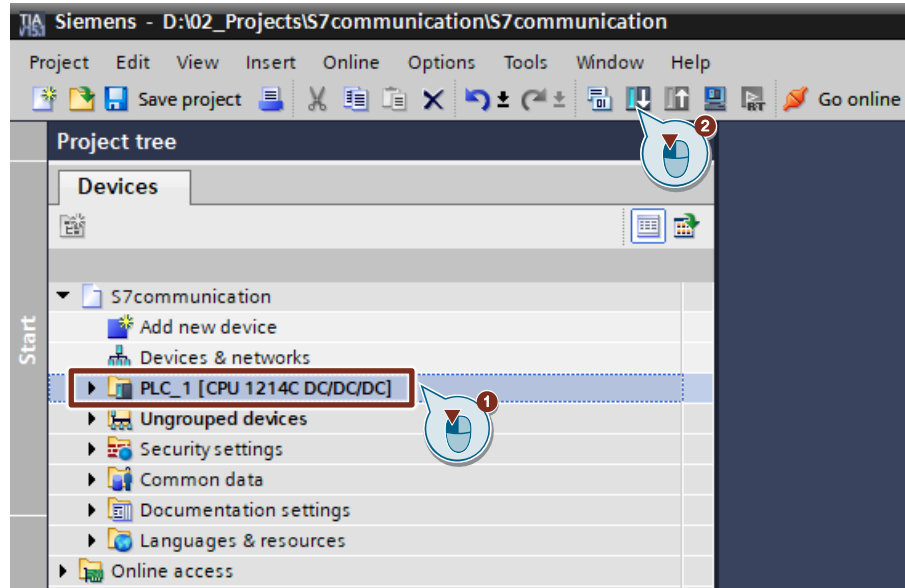
1. In the project tree you mark the device folder of the S7-1200 CPU.
2. Click the "Compile" button in the toolbar. The hardware configuration and the software of the S7-1200 are compiled.



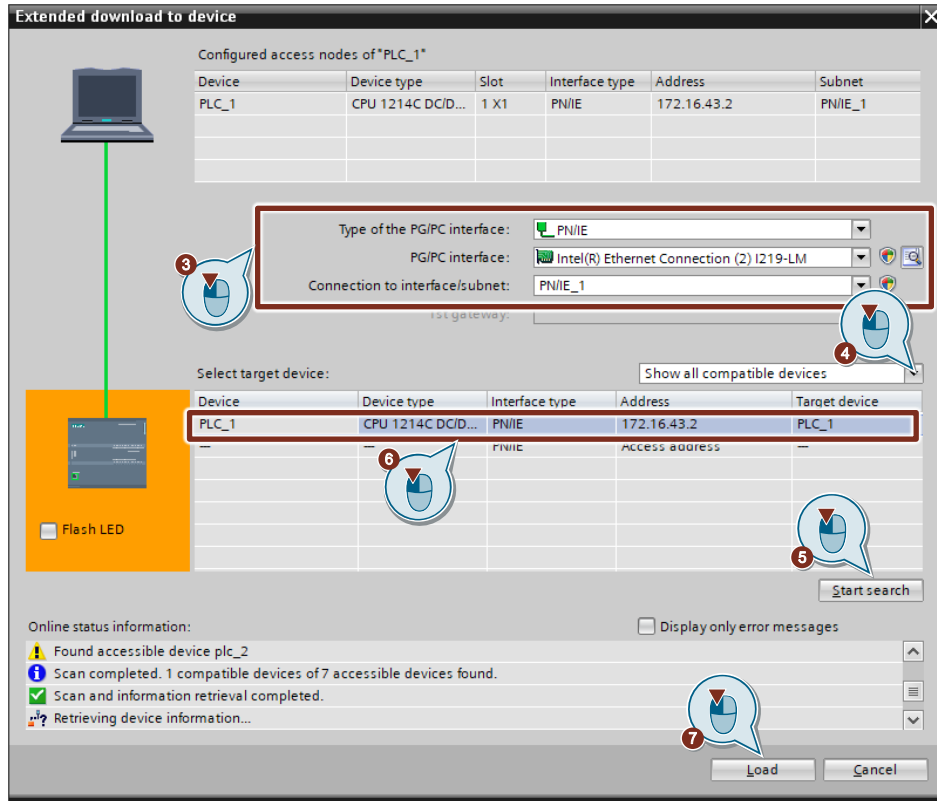


### Download

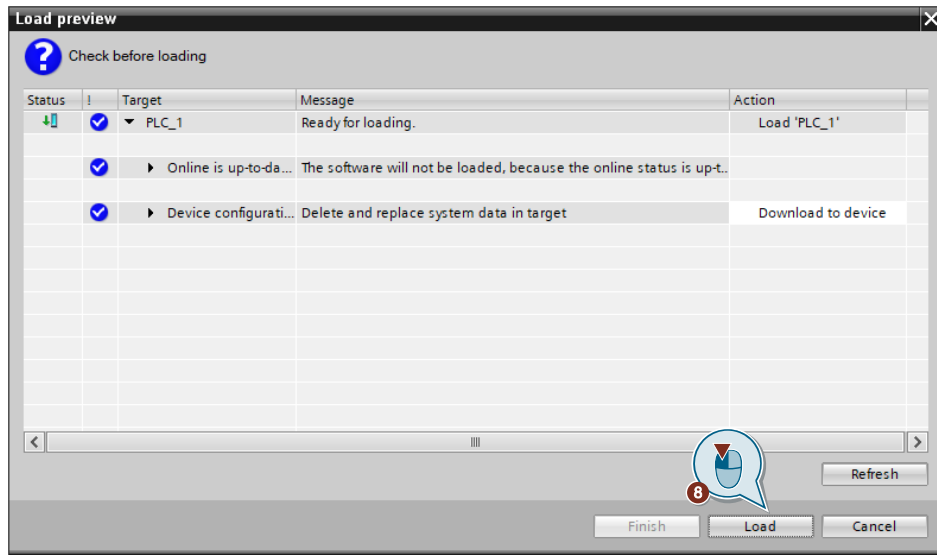
1. In the project tree you mark the device folder of the S7-1200 CPU.
2. Click the "Download to device" button in the toolbar. The "Extended download to device" or "Load preview" dialog opens automatically.



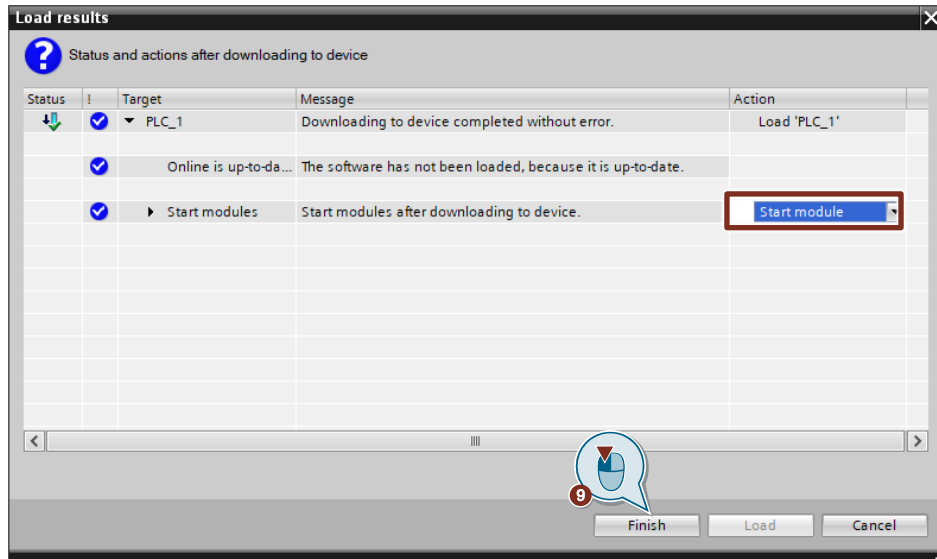
3. The "Extended download to device" dialog opens automatically only if the access path from the PG/PC to the S7-1200 CPU has to be set. Make the following settings:
  - Type of the PG/PC interface: PN/IE
  - PG/PC interface: Network card of the PG/PC
  - Connection to interface/subnet: Subnet to which the S7-1200 CPU is connected
4. In the drop-down list box, you select "Show all compatible devices".
5. Click "Start Search".
6. The S7-1200 CPU is displayed in the "Select target device:" list. Mark the S7-1200 CPU.
7. Click the "Load" button.



- In the "Load preview" dialog you click the "Load" button to start the loading procedure.



9. In the "Load results" dialog you enable the "Start module" action and click the "Finish" button to terminate the loading procedure.



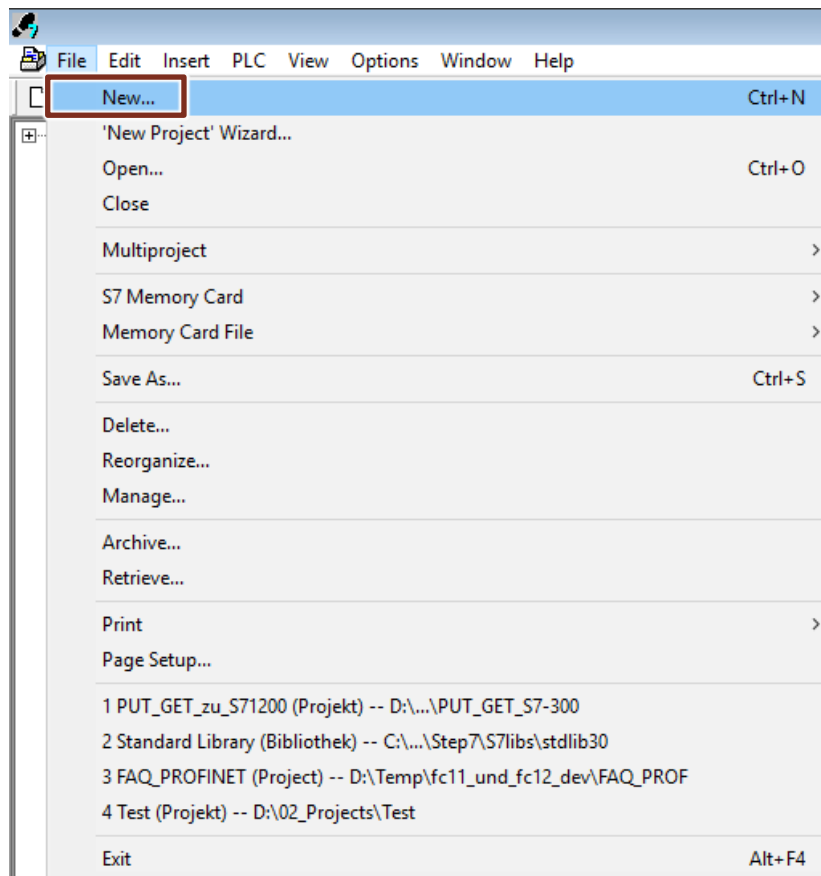
## 3 Configuration and Programming of the SIMATIC S7-300

You configure and program the SIMATIC S7-300 in STEP 7 V5.6.

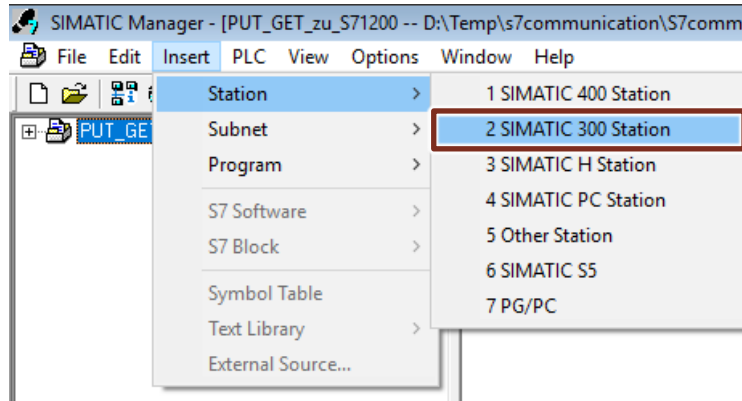
Then you create the user program and define which data is to be exchanged with the S7-1200 via the S7 connection.

### 3.1 Create a STEP 7 project

1. Open the SIMATIC Manager.
2. In the SIMATIC Manager, you create a new STEP 7 project via the menu "File > New".



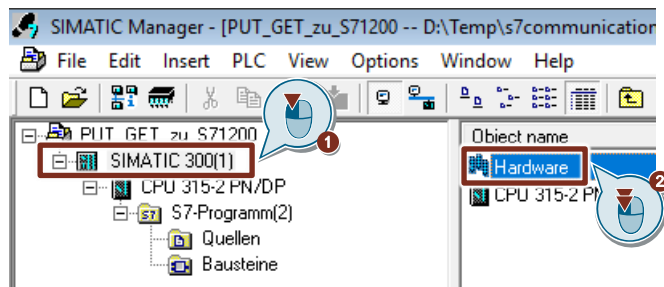
- In the SIMATIC Manager, you add a SIMATIC PC station with the menu "Insert > Station > SIMATIC S7-300 Station".



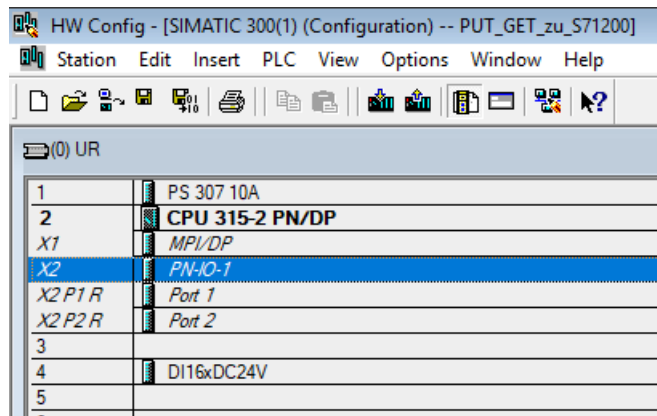
## 3.2 Configure the Hardware

### 3.2.1 Open the Hardware Configuration

- In the SIMATIC Manager you mark the SIMATIC S7-300 station that you have added to your STEP 7 project.
- Double-click the "Hardware" item. The Hardware Configuration opens.

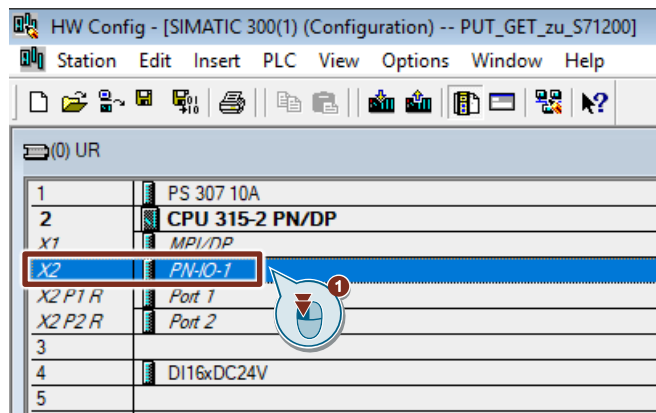


- Configure the S7-300 station by dragging-and-dropping the relevant modules like Power Supply, CPU etc. from the hardware catalog into the S7-300 profile channel.

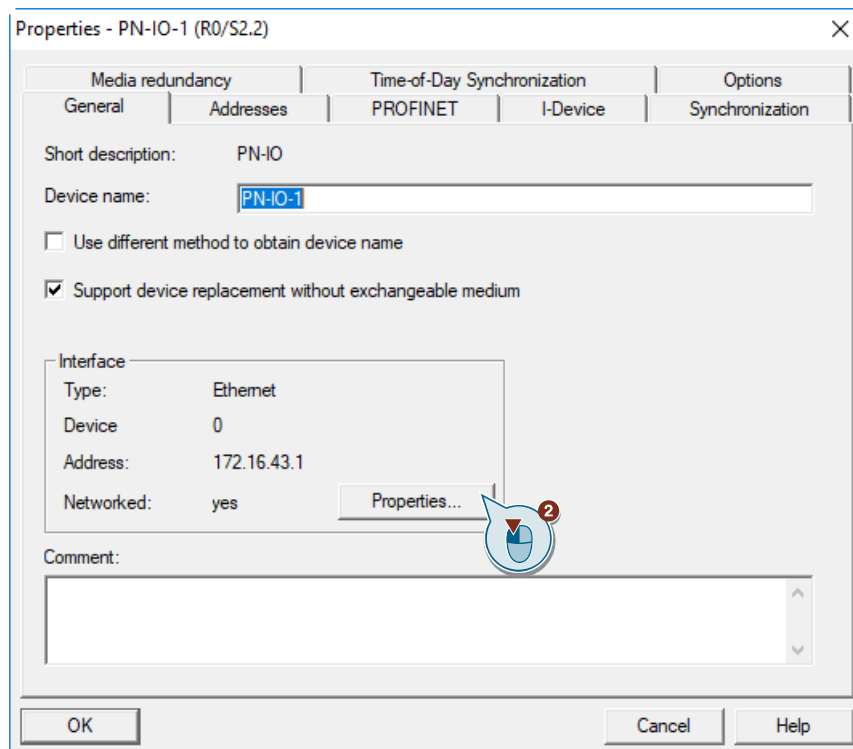


### 3.2.2 Define IP Address and Assign Subnet

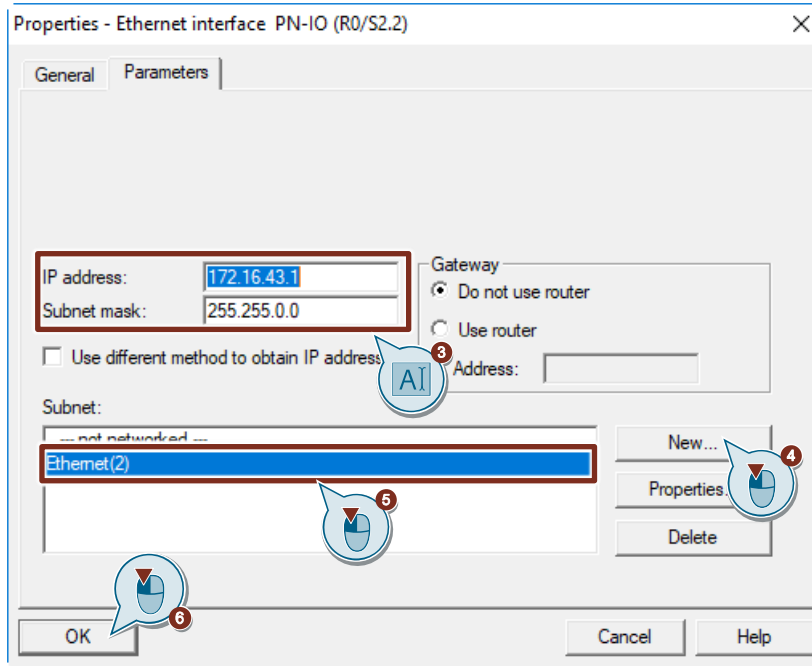
1. Double-click the PROFINET interface of the S7-300 CPU. The Properties dialog of the PROFINET interface opens.



2. Click the "Properties" button to change the IP address and subnet mask and assign the subnet.

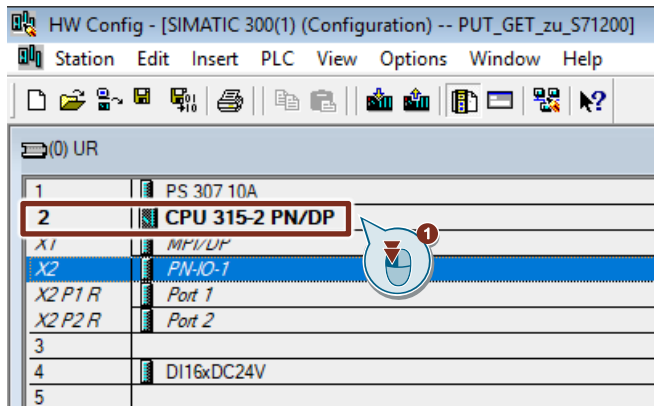


3. Enter the IP address and subnet mask, for example:
  - IP address: 172.16.43.1
  - Subnet mask: 255.255.0.0
4. Click the "New..." button to create a new subnet.
5. Select the new subnet or another existing subnet.
6. Click the "OK" button to apply the IP address and subnet mask and assign the selected subnet to the PROFINET interface of the S7-300 CPU.



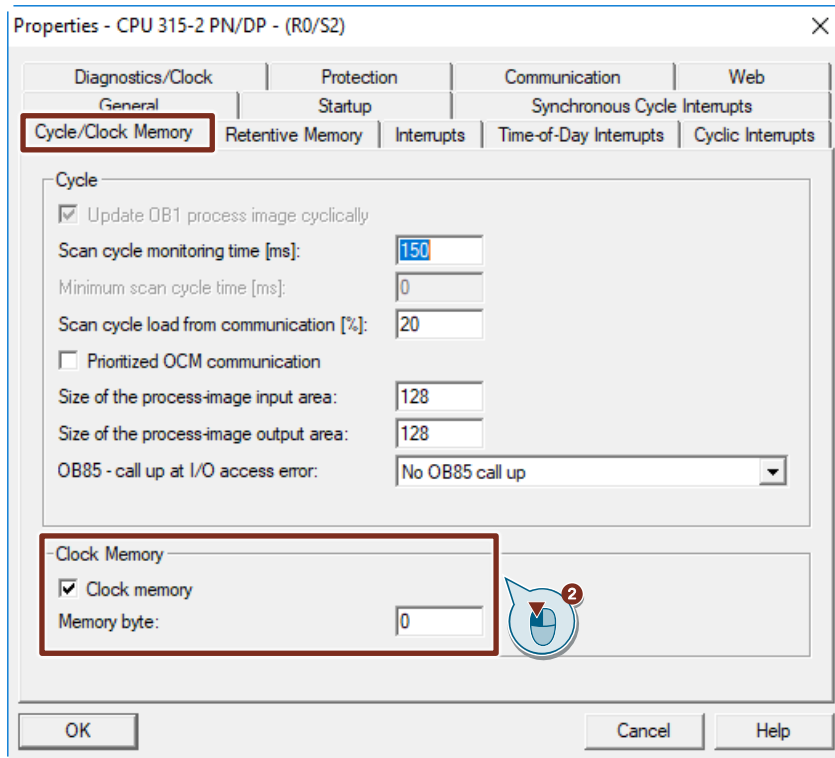
### 3.2.3 Define the Clock Memory

1. Double-click the S7-300 CPU. The Properties dialog of the S7-300 CPU opens.

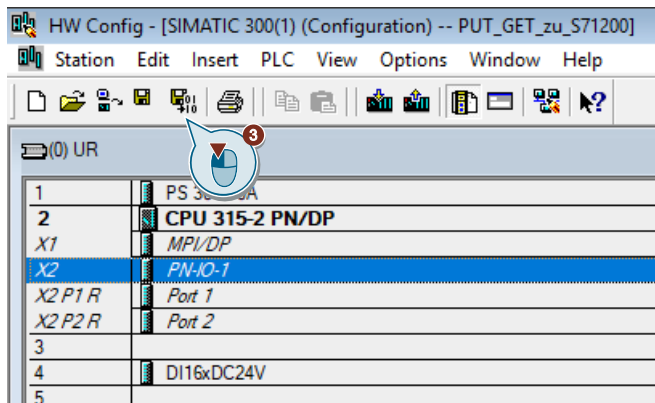


2. Switch to the "Cycle/Clock Memory" tab and enable, for example, the memory byte 0 as clock memory.

### 3 Configuration and Programming of the SIMATIC S7-300



3. Click the "Save and Compile" button to save and compile the configuration of the SIMATIC S7-300 station.





### 3.3 Configure the SIMATIC S7-300 as Client

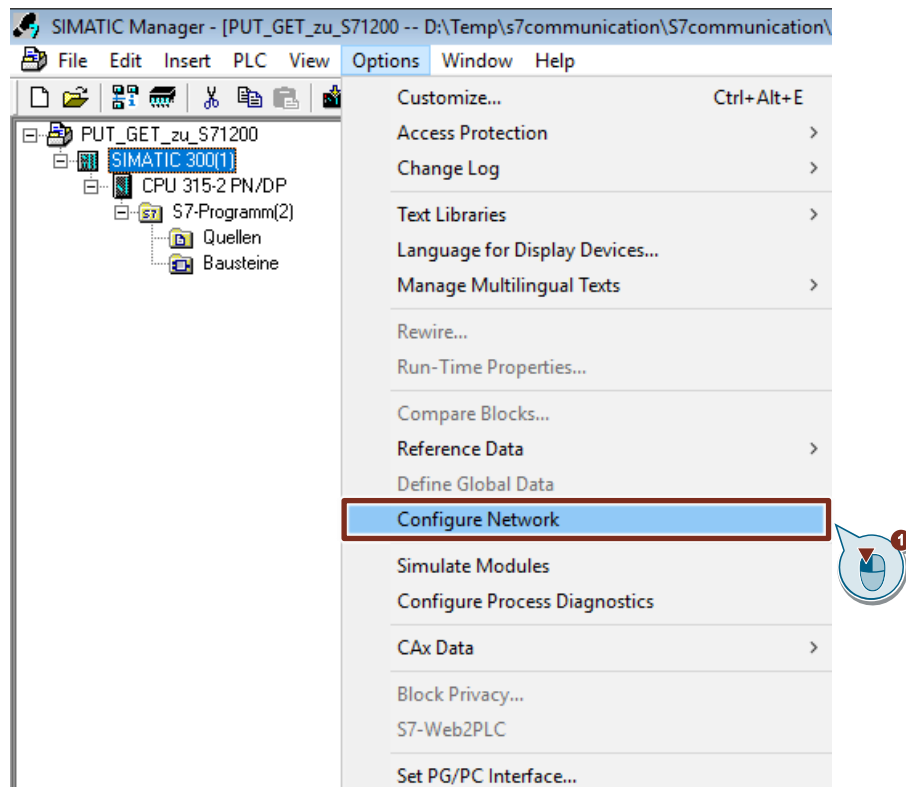
The S7-300 station actively sets up the S7 connection. In this case you must configure an S7 connection for the S7-300 CPU and call the following function blocks (FBs) in the user program:

- FB14 "GET": You use FB14 "GET" to read data from a partner CPU. With an S7-400 CPU you use the system function block SFB14 "GET".
- FB15 "PUT": You use FB15 "PUT" to write data to a partner CPU. With an S7-400 CPU you use the system function block SFB15 "PUT".

#### 3.3.1 Configure the S7 Connection

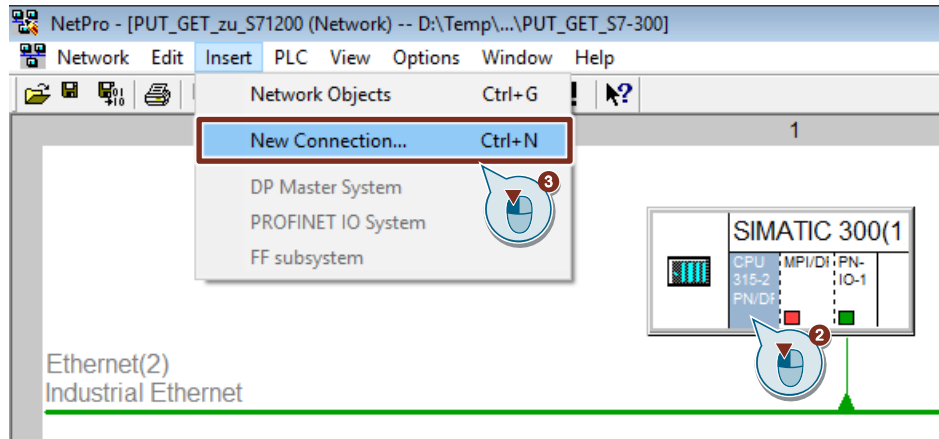
##### Add an S7 Connection

1. In the SIMATIC Manager you open the "NetPro" tool via the menu "Options > Configure Network". In "NetPro" you configure the S7 connection for the S7-300 CPU.

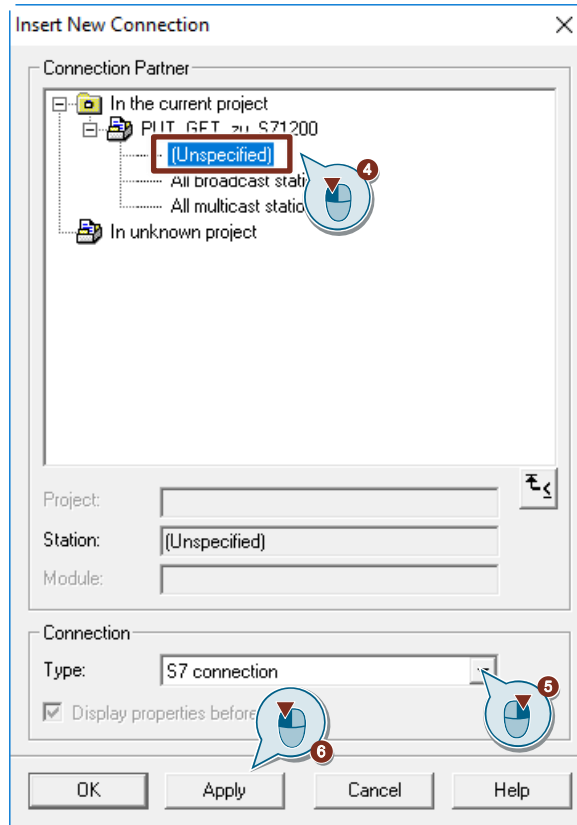


2. Mark the CPU in the SIMATIC S7-300 station.
3. Open the "Insert New Connection" dialog via the menu "Insert > New Connection...".

### 3 Configuration and Programming of the SIMATIC S7-300



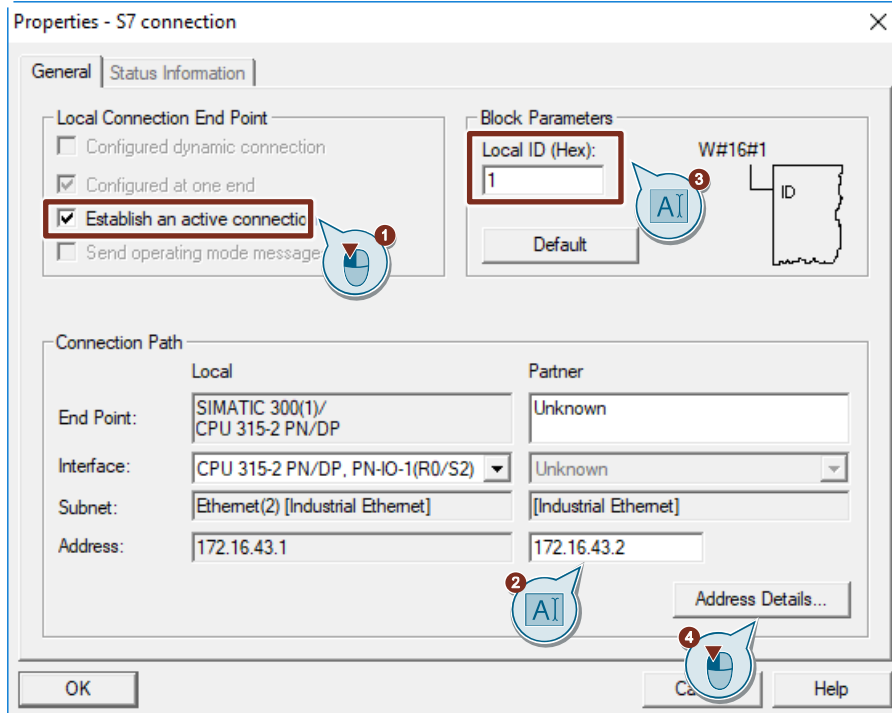
4. Select the "Unspecified" item under "Connection Partner".
5. Select "S7 connection" as the connection type.
6. Click "Apply". The Properties dialog of the S7 connection opens.



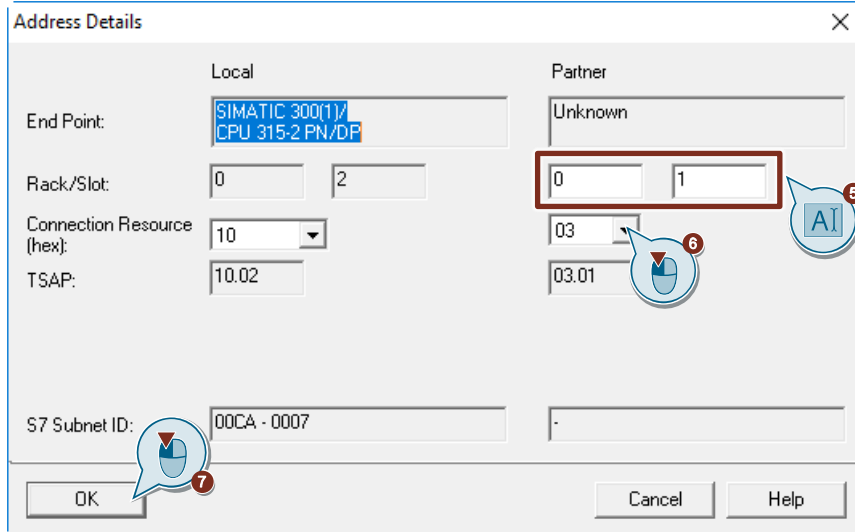
#### Define connection parameters

In the Properties dialog of the S7 connection you define the connection parameters.

1. Enable the "Establish an active connection" function.
2. Enter the IP address of the partner CPU. In this example you enter the IP address of the S7-1200 CPU: 172.16.43.2.
3. You specify the local ID of the connection in the user program at the input parameter "ID" of the function blocks FB14 "GET" and FB15 "PUT".
4. Click the "Address Details..." button. The "Address Details" dialog opens.



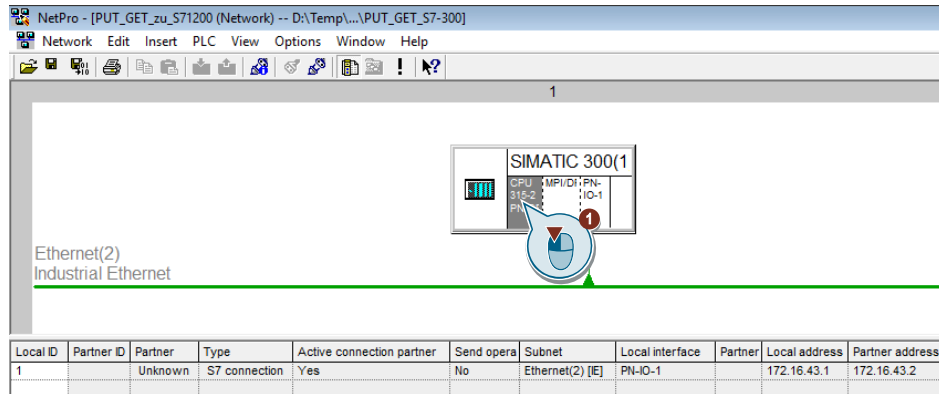
5. Enter the rack and slot of the partner CPU. The S7-1200 CPU uses Rack 0 and Slot 1.
6. Select the connection resource 03(hex) for the partner CPU because the S7 connection is only configured unilaterally in the S7-300 CPU. With these settings the TSAP has the value 03.01 in the partner CPU.
7. Apply the settings with "OK".



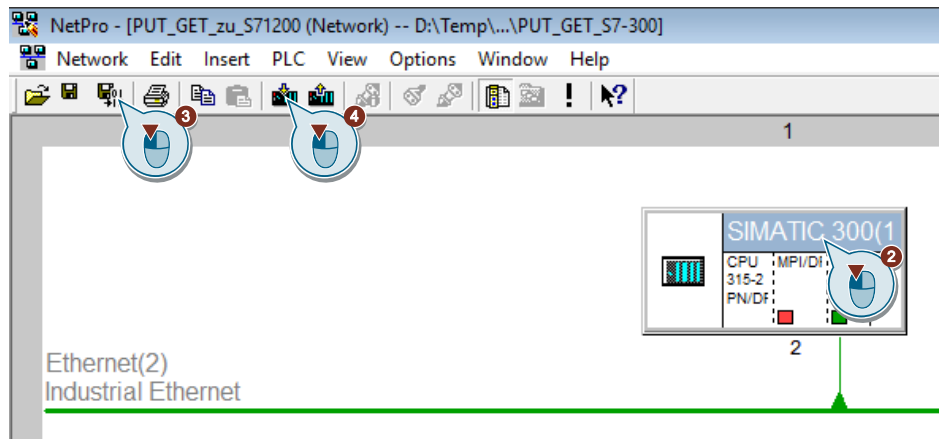
8. Likewise apply the settings in the Properties dialog of the S7 connection with "OK".
9. In the "Insert New Connection" dialog you click the "OK" button to close the dialog. The S7 connection is inserted in "NetPro".

**Load the S7 Connection**

1. In "NetPro" you mark the CPU in the SIMATIC S7-300 station.  
The connection table shows all the connections configured for the CPU.



2. In "NetPro" you mark the SIMATIC S7-300 station.
3. In the toolbar you click the "Save and Compile" button to save and compile the connection configuration.
4. In the toolbar you click the "Load marked station(s)" button to load the connection configuration into the CPU.



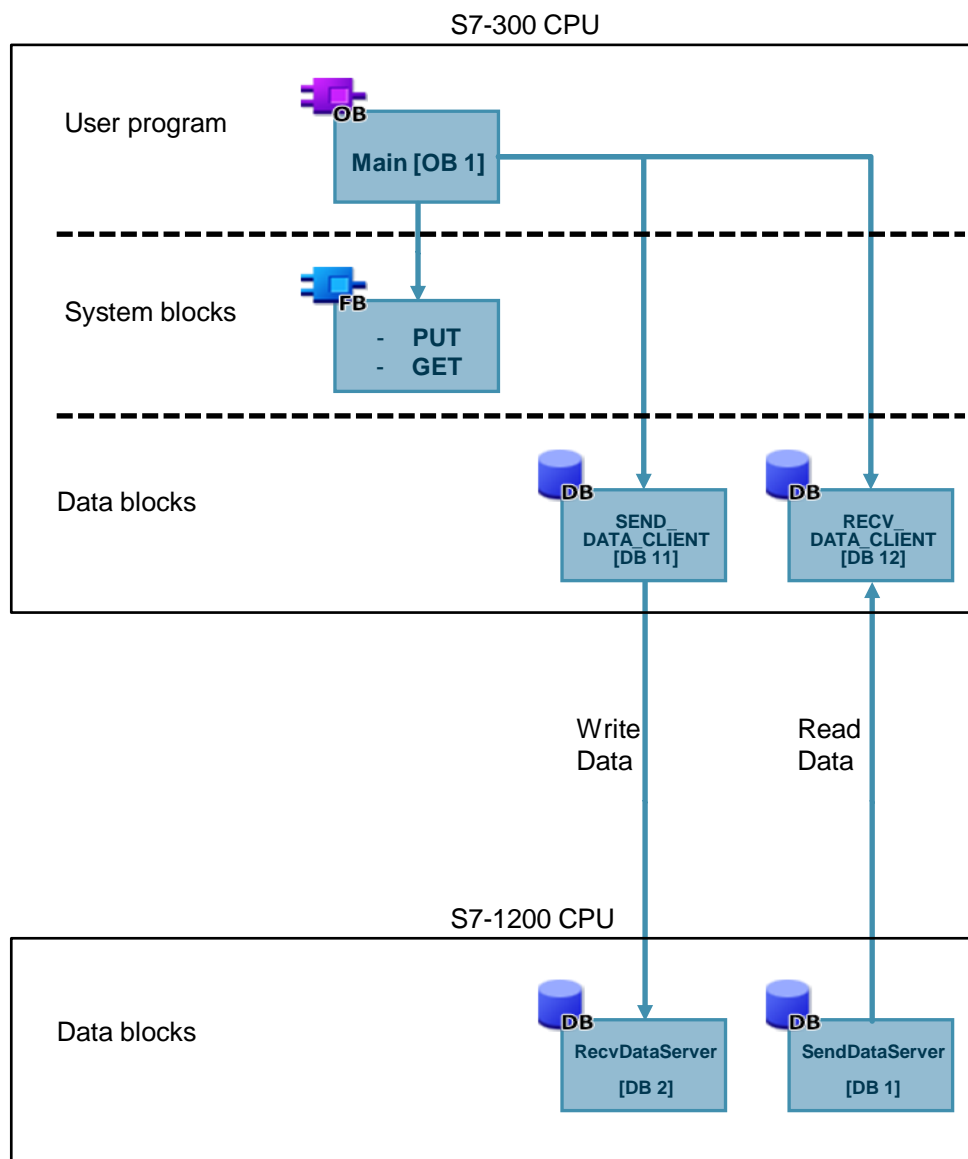
### 3.3.2 Create a User Program

In the user program of the S7-300 CPU you call the function blocks FB14 "GET" and FB15 "PUT" for data transfer. You will find the FBs in the Standard Library under "Communication Blocks". Copy the FBs from the Standard Library and insert them in your STEP 7 project.

**Note** The system function blocks SFB14 "GET" and SFB15 "PUT" for the S7-400 are located in the Standard Library under "System Function Blocks".

The following figure shows the block calls in the S7-300 and S7-1200 CPUs.

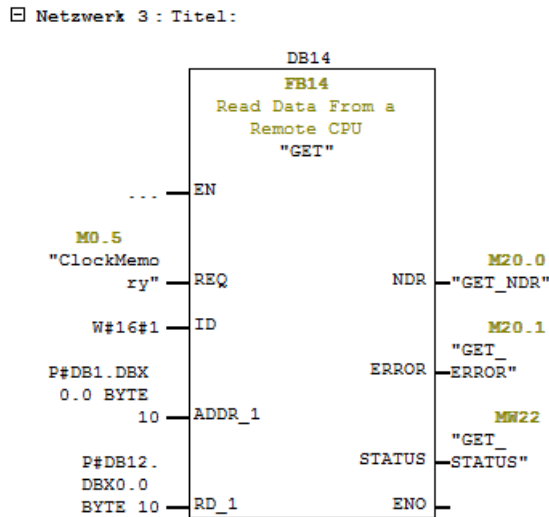
Figure 3-1



### FB14 "GET"

You use FB14 "GET" to read data from the S7-1200 CPU.

Figure 3-2



FB14 "GET" has the following input parameters.

Table 3-1

Input parameters	Data type	Description
REQ	BOOL	The "REQ" control parameter enables data transfer on a rising edge. In this example the job to read the data is enabled via the clock memory M0.5 "ClockMemory".
ID	WORD	Reference to the local connection description (preset by the connection configuration in "NetPro"). In this example the ID = w#16#1 is preset by the configuration connection.
ADDR_1	ANY	Pointer to the area to be read in the partner CPU (Send data area in the partner CPU, here S7-1200 CPU).
RD_1	ANY	Pointer to the area in your own CPU in which the read data is stored (Receive data area in your own CPU, here S7-300 CPU).

FB14 "GET" has the following output parameters.

Table 3-2

Output parameters	Data type	Description
NDR	BOOL	State parameter NDR: 0 = Job not yet started or is still being executed 1 = Job executed error-free
ERROR	BOOL	State parameters ERROR and STATUS: <ul style="list-style-type: none"> <li>ERROR = 0:                             <ul style="list-style-type: none"> <li>STATUS = 0000(hex): neither warning nor error</li> <li>STATUS &lt;&gt; 0000(hex): warning, STATUS provides detailed information</li> </ul> </li> <li>ERROR = 1:                             <ul style="list-style-type: none"> <li>A fault has occurred. STATUS provides detailed information about the type of error.</li> </ul> </li> </ul>
STATUS	WORD	

### Receive data area in the S7-300 CPU

In the S7-300 CPU the Receive data read from the S7-1200 CPU is stored in data block DB12 "RECV\_DATA\_CLIENT".

In this example, 10 bytes of data starting at address 0 are stored in DB12 "RECV\_DATA\_CLIENT".

### Send data area in the S7-1200 CPU

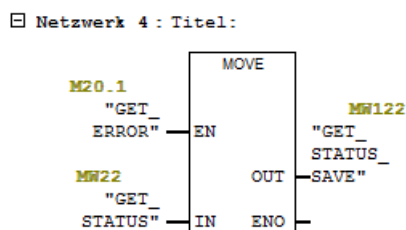
In the S7-1200 CPU the data read from the S7-300 is stored in data block DB1 "SendDataServer".

In this example, 10 bytes of data starting at address 0 are read from DB1 "SendDataServer" of the S7-1200 CPU.

### Error evaluation of the Read job

If the Read job does not terminate successfully, in other words ERROR =1, the value of the output parameter STATUS is stored in MW122 "GET\_STATUS\_SAVE".

Figure 3-3



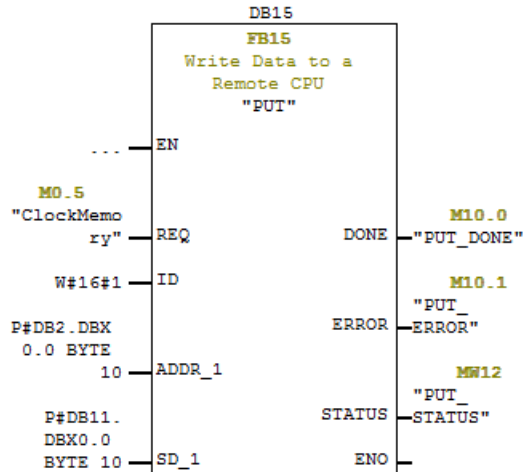


### FB15 "PUT"

You use FB15 "PUT" to write data to the S7-1200 CPU.

Figure 3-4

Netzwerk 1: Titel:



FB15 "PUT" has the following input parameters.

Table 3-3

Input parameters	Data type	Description
REQ	BOOL	The "REQ" control parameter enables data transfer on a rising edge. In this example the job to write the data is enabled via the clock memory M0.5 "ClockMemory".
ID	WORD	Reference to the local connection description (preset by the connection configuration in "NetPro"). In this example the ID = w#16#1 is preset by the configuration connection.
ADDR_1	ANY	Pointer to the area to be written to in the partner CPU (Receive data area in the partner CPU, here S7-1200 CPU).
SD_1	ANY	Pointer to the area in your own CPU which contains the data to be sent (Send data area in your own CPU, here S7-300 CPU).

FB15 "PUT" has the following output parameters.

Table 3-4

Input parameters	Data type	Description
DONE	BOOL	State parameter DONE: 0 = Job not yet started or is still being executed 1 = Job executed error-free
ERROR	BOOL	State parameters ERROR and STATUS: <ul style="list-style-type: none"> <li>ERROR = 0:                             <ul style="list-style-type: none"> <li>STATUS = 0000(hex): neither warning nor error</li> <li>STATUS &lt;&gt; 0000(hex): warning, STATUS provides detailed information</li> </ul> </li> <li>ERROR = 1: A fault has occurred. STATUS provides detailed information about the type of error.</li> </ul>
STATUS	WORD	

### Send data area in the S7-300 CPU

In the S7-300 CPU the Send data written to the S7-1200 CPU is stored in data block DB11 "SEND\_DATA\_CLIENT".

In this example 10 bytes of data starting at address 0 are written from DB11 "SEND\_DATA\_CLIENT" to the S7-1200 CPU.

### Receive data area in the S7-1200 CPU

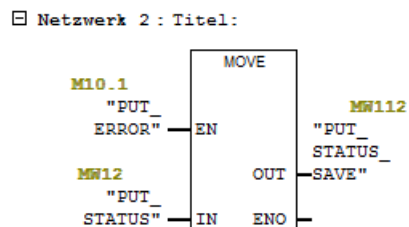
In the S7-1200 CPU the received data is stored in data block DB2 "RcvDataServer".

In this example 10 bytes of data starting at address 0 are stored in DB2 "RcvDataServer".

### Error evaluation of the Write job

If the Write job does not terminate successfully, in other words ERROR =1, the value of the output parameter STATUS is stored in MW112 "PUT\_STATUS\_SAVE".

Figure 3-5



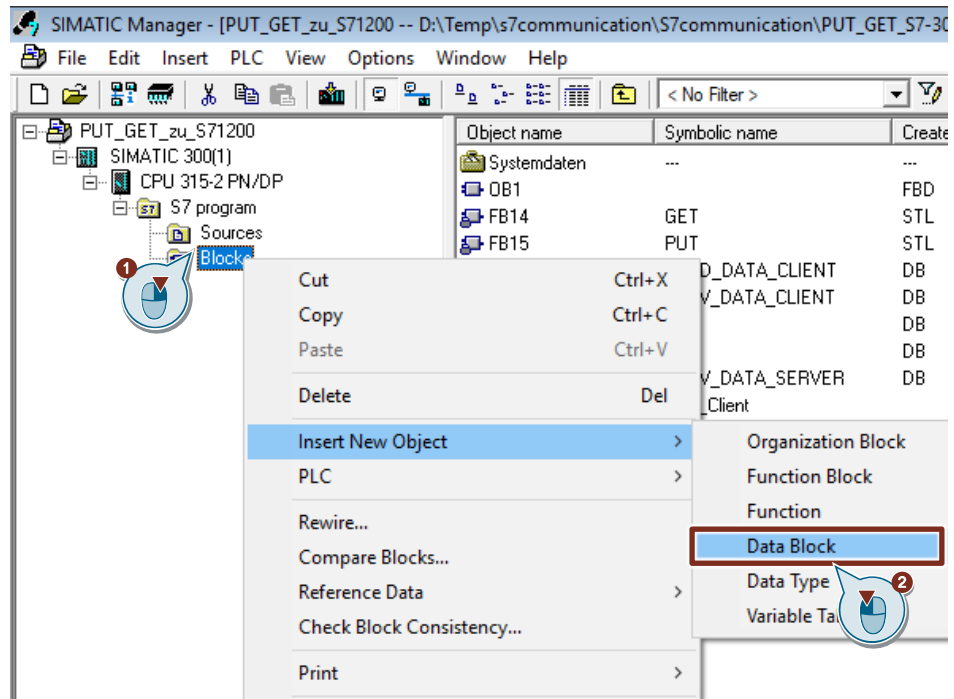
## 3.4 Configure the SIMATIC S7-300 as Server

### 3.4.1 Create a User Program

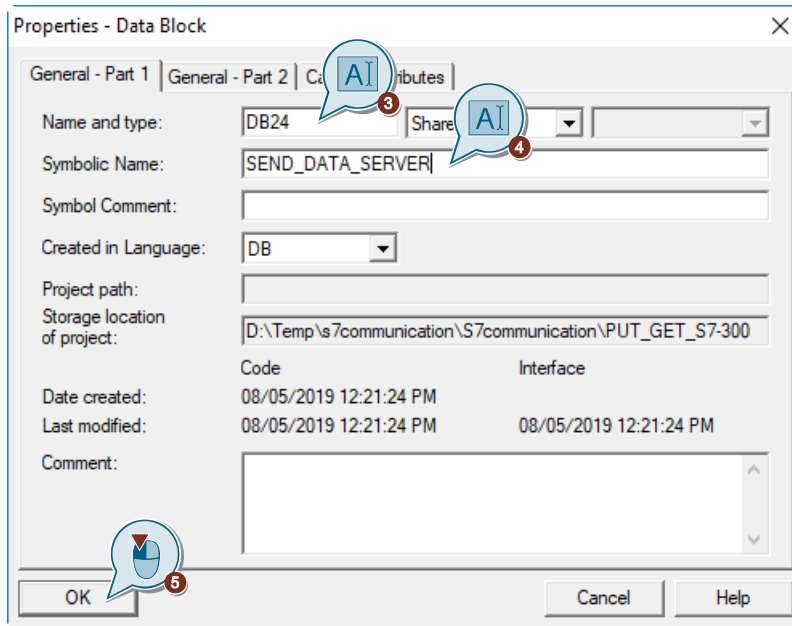
In the user program of the S7-300 CPU you add the data blocks for saving the Send and Receive data.

#### Add data block for saving the Send data

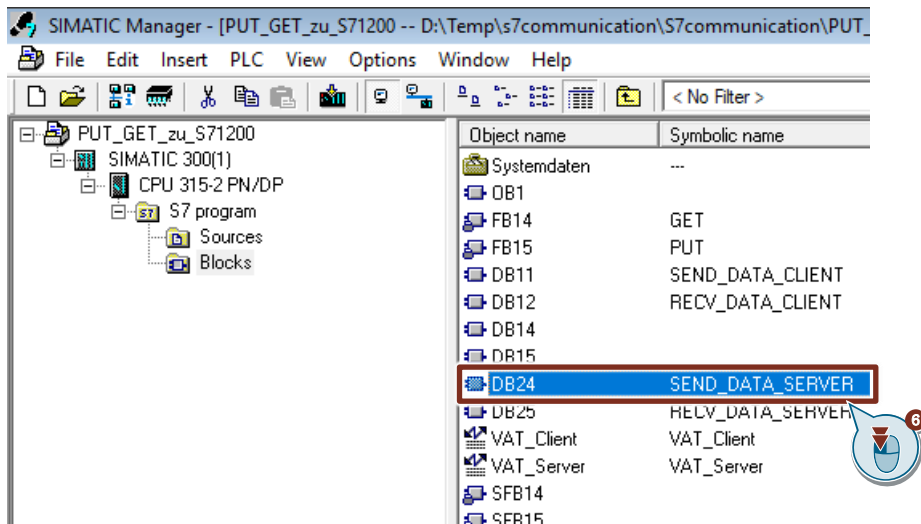
1. In your STEP 7 project you right-click the "Blocks" folder. The pop-up menu opens.
2. You select the menu "Insert New Object > Data Block". The Properties dialog of the DB opens.



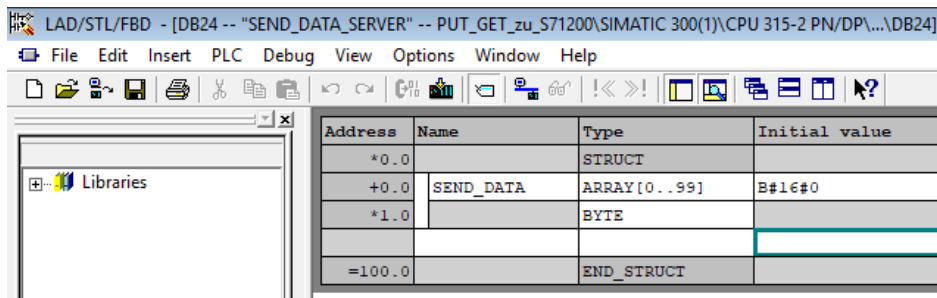
3. You enter the name and type of block that is to be created, DB24, for example.
4. Enter a symbolic name for the block, "SEND\_DATA\_SERVER", for example.
5. Apply the settings with "OK".  
Block DB24 "SEND\_DATA\_SERVER" is inserted in the "Blocks" folder.



6. Double-click DB24 "SEND\_DATA\_SERVER" in the "Blocks" folder. The DB opens.

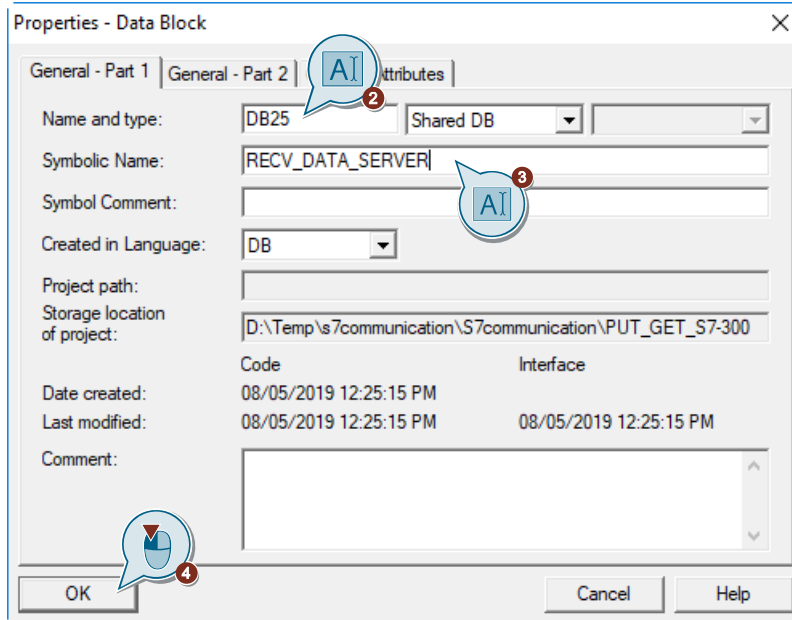


7. In DB24 "SEND\_DATA\_SERVER" you define the variable "SEND\_DATA" of the data type Array[0..99] of Byte.

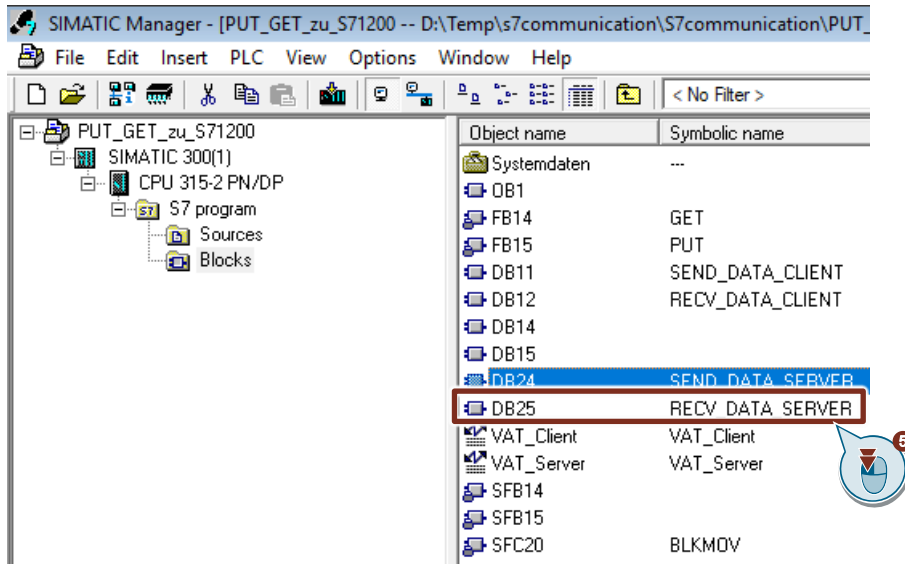


### Add data block for saving the Receive data

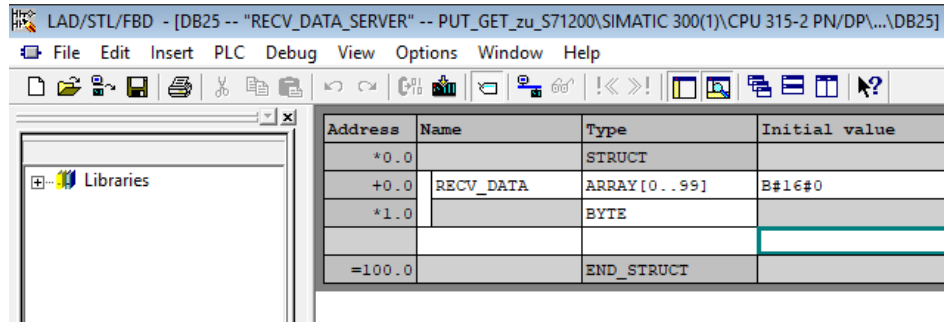
1. Add another DB for saving the Receive data.
2. You enter the name and type of block that is to be created, DB25, for example.
3. Enter a symbolic name for the block, "RECV\_DATA\_SERVER", for example.
4. Apply the settings with "OK".  
Block DB25 "RECV\_DATA\_SERVER" is inserted in the "Blocks" folder.



5. Double-click DB25 "RECV\_DATA\_SERVER" in the "Blocks" folder. The DB opens.



6. In DB25 "RECV\_DATA\_SERVER" you define the variable "RECV\_DATA" of the data type Array[0..99] of Byte.



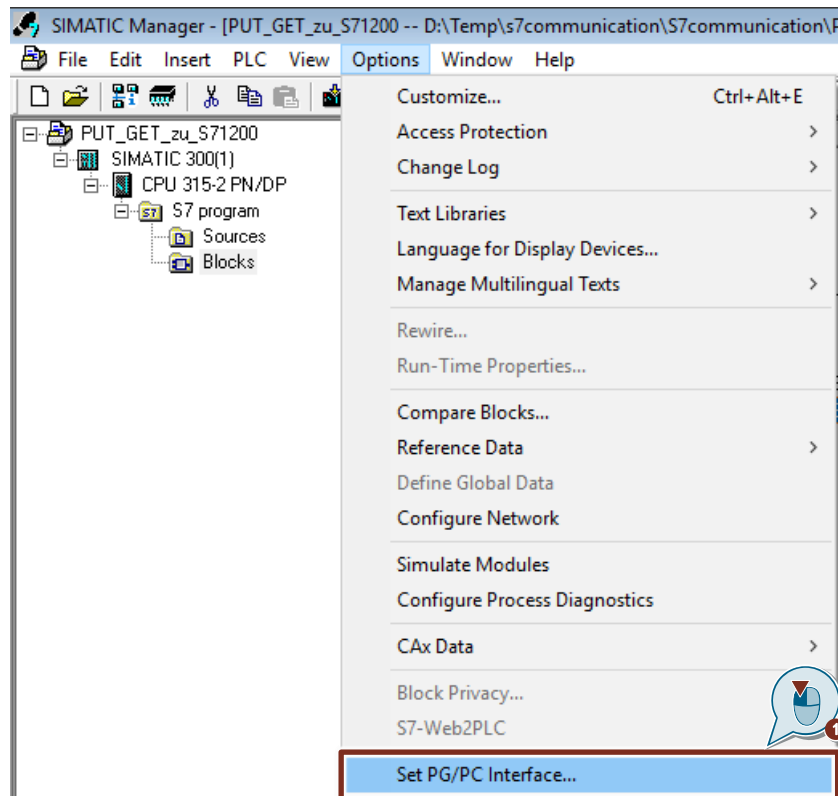
## 3.5 Download the Hardware Configuration and User Program

### Requirements

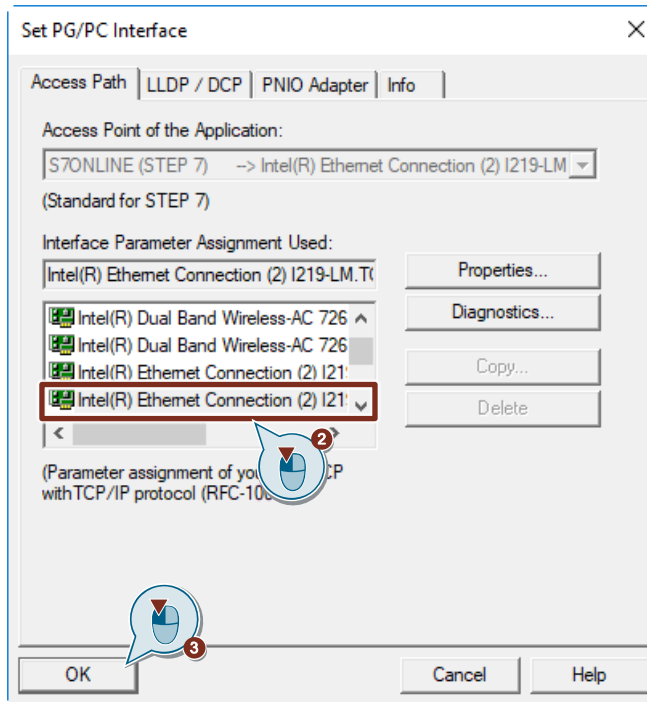
You have already assigned the configured IP address and subnet mask to the S7-300 CPU.

### Instructions

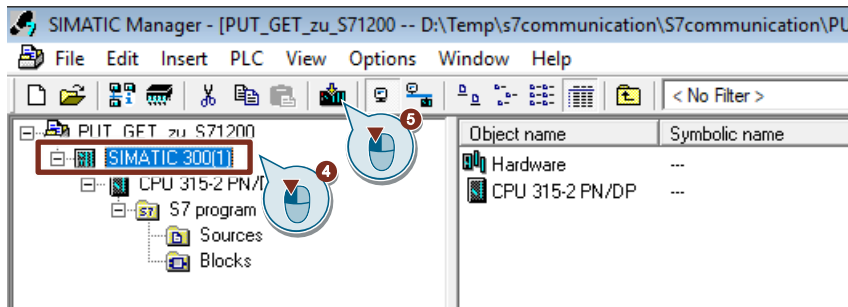
1. In the SIMATIC Manager, you select the menu "Options > Set PG/PC Interface...". The "Set PG/PC Interface" dialog opens.



2. As access point for the application you set the network card with TCP/IP protocol via which the PC is connected to the S7-300 CPU and via which you access the S7-300 CPU.
3. Apply the settings with "OK".



4. In the SIMATIC Manager you mark the S7-300 station.
5. Click the "Load" button in the toolbar. The hardware configuration and the user program are downloaded to the S7-300 CPU.





## 4 Operating and Monitoring

Operating and monitoring is done via a variable table in STEP 7 V5.6 and STEP 7 V15.1.

In STEP 7 V5.6 you can monitor and control the Send and Receive data of the S7-300 CPU in the variable table.

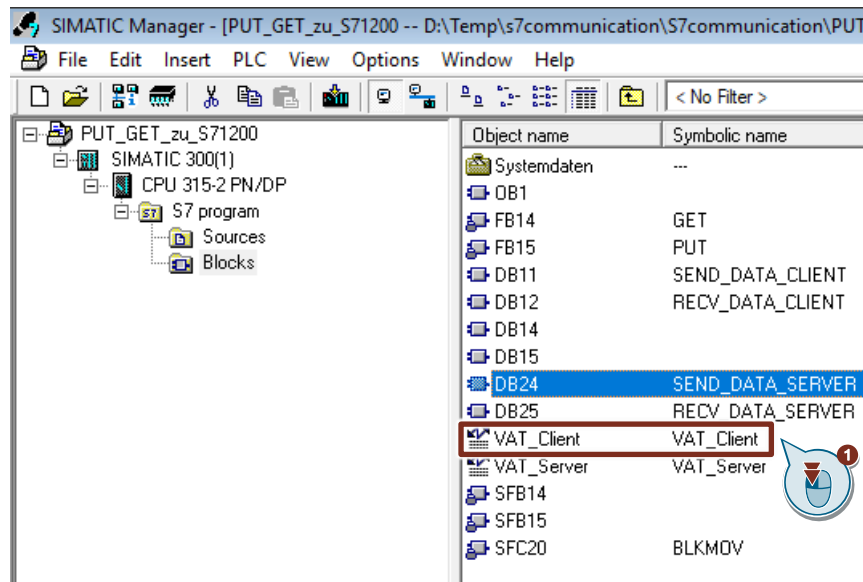
In STEP 7 V15.1 you can monitor and control the Send and Receive data of the S7-1200 CPU in the variable table.

### 4.1 SIMATIC S7-300 Writes and Reads Data as Client

#### 4.1.1 Write Data to the S7-1200 CPU

##### Using the variable table in STEP 7 V5.6

1. In the "Blocks" folder you double-click the variable table "VAT\_Client". The variable table opens.



2. Click the "Monitor variable" button in the toolbar. The "Status value" column displays the current values in the Send data area (DB11) and Receive data area (DB12).

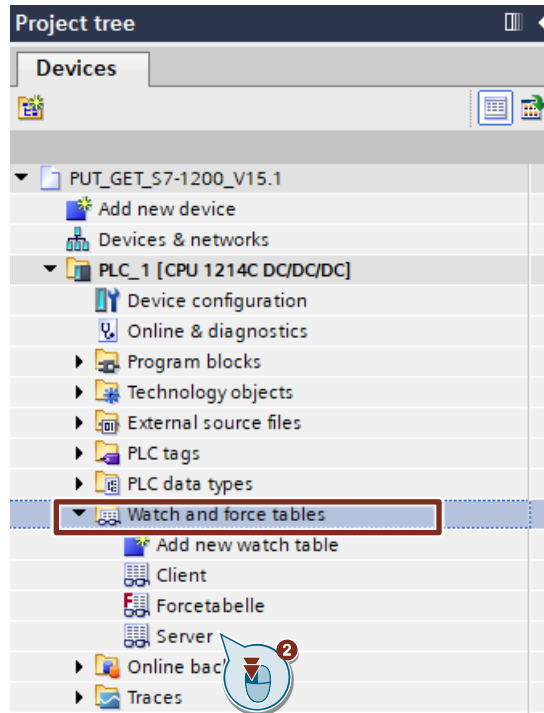
	Address	Symbol	Display format	Status value	Modify value
1		//SEND DATA			
2	DB11.DBB 0	"SEND_DATA_CLIENT".SEND_DATA[0]	HEX	B#16#11	B#16#11
3	DB11.DBB 1	"SEND_DATA_CLIENT".SEND_DATA[1]	HEX	B#16#12	B#16#12
4	DB11.DBB 2	"SEND_DATA_CLIENT".SEND_DATA[2]	HEX	B#16#13	B#16#13
5	DB11.DBB 3	"SEND_DATA_CLIENT".SEND_DATA[3]	HEX	B#16#14	B#16#14
6	DB11.DBB 4	"SEND_DATA_CLIENT".SEND_DATA[4]	HEX	B#16#15	B#16#15
7	DB11.DBB 5	"SEND_DATA_CLIENT".SEND_DATA[5]	HEX	B#16#16	B#16#16
8	DB11.DBB 6	"SEND_DATA_CLIENT".SEND_DATA[6]	HEX	B#16#17	B#16#17
9	DB11.DBB 7	"SEND_DATA_CLIENT".SEND_DATA[7]	HEX	B#16#18	B#16#18
10	DB11.DBB 8	"SEND_DATA_CLIENT".SEND_DATA[8]	HEX	B#16#19	B#16#19
11	DB11.DBB 9	"SEND_DATA_CLIENT".SEND_DATA[9]	HEX	B#16#20	B#16#20
12					
13		//RECEIVE DATA			
14	DB12.DBB 0	"RECV_DATA_CLIENT".RECV_DATA[0]	HEX	B#16#26	
15	DB12.DBB 1	"RECV_DATA_CLIENT".RECV_DATA[1]	HEX	B#16#09	
16	DB12.DBB 2	"RECV_DATA_CLIENT".RECV_DATA[2]	HEX	B#16#19	
17	DB12.DBB 3	"RECV_DATA_CLIENT".RECV_DATA[3]	HEX	B#16#81	
18	DB12.DBB 4	"RECV_DATA_CLIENT".RECV_DATA[4]	HEX	B#16#30	
19	DB12.DBB 5	"RECV_DATA_CLIENT".RECV_DATA[5]	HEX	B#16#06	
20	DB12.DBB 6	"RECV_DATA_CLIENT".RECV_DATA[6]	HEX	B#16#19	
21	DB12.DBB 7	"RECV_DATA_CLIENT".RECV_DATA[7]	HEX	B#16#84	
22	DB12.DBB 8	"RECV_DATA_CLIENT".RECV_DATA[8]	HEX	B#16#03	
23	DB12.DBB 9	"RECV_DATA_CLIENT".RECV_DATA[9]	HEX	B#16#11	
24					

3. In the "Modify value" column you enter the values that are to be put in the Send data area and written to the S7-1200 CPU.
4. Click the "Modify variable" button in the toolbar.  
The modified values are put in the Send data area (DB1) of the S7-300 CPU and written to the S7-1200 CPU.

	Address	Symbol	Display format	Status value	Modify value
1		//SEND DATA			
2	DB11.DBB 0	"SEND_DATA_CLIENT".SEND_DATA[0]	HEX	B#16#11	B#16#11
3	DB11.DBB 1	"SEND_DATA_CLIENT".SEND_DATA[1]	HEX	B#16#12	B#16#12
4	DB11.DBB 2	"SEND_DATA_CLIENT".SEND_DATA[2]	HEX	B#16#13	B#16#13
5	DB11.DBB 3	"SEND_DATA_CLIENT".SEND_DATA[3]	HEX	B#16#14	B#16#14
6	DB11.DBB 4	"SEND_DATA_CLIENT".SEND_DATA[4]	HEX	B#16#15	B#16#15
7	DB11.DBB 5	"SEND_DATA_CLIENT".SEND_DATA[5]	HEX	B#16#16	B#16#16
8	DB11.DBB 6	"SEND_DATA_CLIENT".SEND_DATA[6]	HEX	B#16#17	B#16#17
9	DB11.DBB 7	"SEND_DATA_CLIENT".SEND_DATA[7]	HEX	B#16#18	B#16#18
10	DB11.DBB 8	"SEND_DATA_CLIENT".SEND_DATA[8]	HEX	B#16#19	B#16#19
11	DB11.DBB 9	"SEND_DATA_CLIENT".SEND_DATA[9]	HEX	B#16#20	B#16#20
12					
13		//RECEIVE DATA			
14	DB12.DBB 0	"RECV_DATA_CLIENT".RECV_DATA[0]	HEX	B#16#26	
15	DB12.DBB 1	"RECV_DATA_CLIENT".RECV_DATA[1]	HEX	B#16#09	
16	DB12.DBB 2	"RECV_DATA_CLIENT".RECV_DATA[2]	HEX	B#16#19	
17	DB12.DBB 3	"RECV_DATA_CLIENT".RECV_DATA[3]	HEX	B#16#81	
18	DB12.DBB 4	"RECV_DATA_CLIENT".RECV_DATA[4]	HEX	B#16#30	
19	DB12.DBB 5	"RECV_DATA_CLIENT".RECV_DATA[5]	HEX	B#16#06	
20	DB12.DBB 6	"RECV_DATA_CLIENT".RECV_DATA[6]	HEX	B#16#19	
21	DB12.DBB 7	"RECV_DATA_CLIENT".RECV_DATA[7]	HEX	B#16#84	
22	DB12.DBB 8	"RECV_DATA_CLIENT".RECV_DATA[8]	HEX	B#16#03	
23	DB12.DBB 9	"RECV_DATA_CLIENT".RECV_DATA[9]	HEX	B#16#11	
24					

**Using the variable table in STEP 7 V15.1**

1. In the project tree you open the device folder of the S7-1200 CPU.
2. In the "Watch and force tables" folder you double-click the watch table "Server". The watch table opens in the working area of STEP 7 V15.1.



3. Click the "Monitor all" button in the toolbar.  
The "Monitor value" column displays the current values stored in the Send data area (DB1) and Receive data area (DB2). This permits you to check whether the data written from the S7-300 CPU has arrived in the Receive data area (DB2) of the S7-1200 CPU.

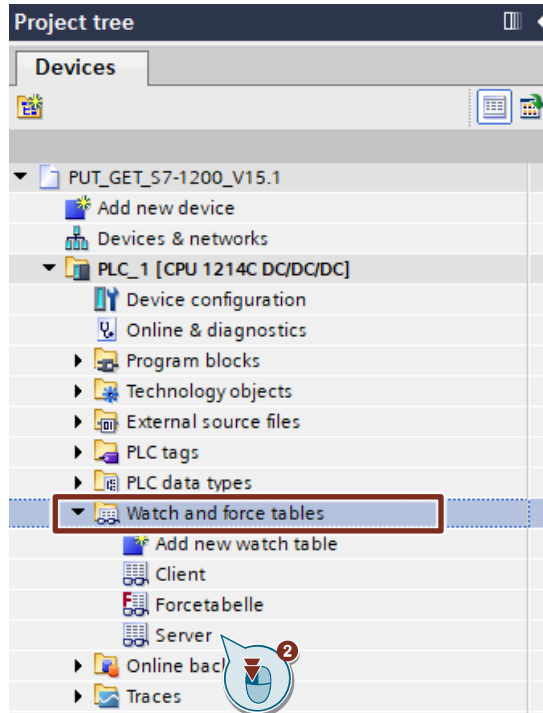
PUT\_GET\_S7-1200\_V15.1 ▶ PLC\_1 [CPU 1214C DC/DC/DC] ▶ Watch and force tables ▶ Server

	Name	Address	Display format	Monitor value	Modify value
1	// Receive Data				
2	"RecvDataServer".recvData[0]	%DB2.DBB0	Hex	16#11	
3	"RecvDataServer".recvData[1]	%DB2.DBB1	Hex	16#12	
4	"RecvDataServer".recvData[2]	%DB2.DBB2	Hex	16#13	
5	"RecvDataServer".recvData[3]	%DB2.DBB3	Hex	16#14	
6	"RecvDataServer".recvData[4]	%DB2.DBB4	Hex	16#15	
7	"RecvDataServer".recvData[5]	%DB2.DBB5	Hex	16#16	
8	"RecvDataServer".recvData[6]	%DB2.DBB6	Hex	16#17	
9	"RecvDataServer".recvData[7]	%DB2.DBB7	Hex	16#18	
10	"RecvDataServer".recvData[8]	%DB2.DBB8	Hex	16#19	
11	"RecvDataServer".recvData[9]	%DB2.DBB9	Hex	16#20	
12					
13	// Send Data				
14	"SendDataServer".sendData[0]	%DB1.DBB0	Hex	16#26	16#26
15	"SendDataServer".sendData[1]	%DB1.DBB1	Hex	16#09	16#09
16	"SendDataServer".sendData[2]	%DB1.DBB2	Hex	16#19	16#19
17	"SendDataServer".sendData[3]	%DB1.DBB3	Hex	16#81	16#81
18	"SendDataServer".sendData[4]	%DB1.DBB4	Hex	16#30	16#30
19	"SendDataServer".sendData[5]	%DB1.DBB5	Hex	16#06	16#06
20	"SendDataServer".sendData[6]	%DB1.DBB6	Hex	16#19	16#19
21	"SendDataServer".sendData[7]	%DB1.DBB7	Hex	16#84	16#84
22	"SendDataServer".sendData[8]	%DB1.DBB8	Hex	16#03	16#03
23	"SendDataServer".sendData[9]	%DB1.DBB9	Hex	16#11	16#11
24	<Add new>				

### 4.1.2 Read Data from the S7-1200 CPU

#### Using the variable table in STEP 7 V15.1

1. In the project tree you open the device folder of the S7-1200 CPU.
2. In the "Watch and force tables" folder you double-click the watch table "Server". The watch table opens in the working area of STEP 7 V15.1.



3. Click the "Monitor all" button in the toolbar. The "Monitor value" column displays the current values stored in the Send data area (DB1) and Receive data area (DB2).

PUT\_GET\_S7-1200\_V15.1 ▶ PLC\_1 [CPU 1214C DC/DC/DC] ▶ Watch and force tables ▶ Server

	Name	Address	Display format	Monitor value	Modify value
1	// Receive Data				
2	"RecvDataServer".recvData[0]	%DB2.DBB0	Hex	16#11	
3	"RecvDataServer".recvData[1]	%DB2.DBB1	Hex	16#12	
4	"RecvDataServer".recvData[2]	%DB2.DBB2	Hex	16#13	
5	"RecvDataServer".recvData[3]	%DB2.DBB3	Hex	16#14	
6	"RecvDataServer".recvData[4]	%DB2.DBB4	Hex	16#15	
7	"RecvDataServer".recvData[5]	%DB2.DBB5	Hex	16#16	
8	"RecvDataServer".recvData[6]	%DB2.DBB6	Hex	16#17	
9	"RecvDataServer".recvData[7]	%DB2.DBB7	Hex	16#18	
10	"RecvDataServer".recvData[8]	%DB2.DBB8	Hex	16#19	
11	"RecvDataServer".recvData[9]	%DB2.DBB9	Hex	16#20	
12					
13	// Send Data				
14	"SendDataServer".sendData[0]	%DB1.DBB0	Hex	16#26	16#26
15	"SendDataServer".sendData[1]	%DB1.DBB1	Hex	16#09	16#09
16	"SendDataServer".sendData[2]	%DB1.DBB2	Hex	16#19	16#19
17	"SendDataServer".sendData[3]	%DB1.DBB3	Hex	16#81	16#81
18	"SendDataServer".sendData[4]	%DB1.DBB4	Hex	16#30	16#30
19	"SendDataServer".sendData[5]	%DB1.DBB5	Hex	16#06	16#06
20	"SendDataServer".sendData[6]	%DB1.DBB6	Hex	16#19	16#19
21	"SendDataServer".sendData[7]	%DB1.DBB7	Hex	16#84	16#84
22	"SendDataServer".sendData[8]	%DB1.DBB8	Hex	16#03	16#03
23	"SendDataServer".sendData[9]	%DB1.DBB9	Hex	16#11	16#11
24	<Add new>				

4. In the "Modify value" column you enter the values that are to be put in the Send data area (DB1) of the S7-1200 CPU and read from the S7-300 CPU. The data read is stored in the Receive data area (DB12) of the S7-300.
5. In the toolbar you click the "Modify all enabled values once and immediately" button. The modified values are put in the Send data area (DB1) of the S7-1200 CPU.

PUT\_GET\_S7-1200\_V15.1 ▶ PLC\_1 [CPU 1214C DC/DC/DC] ▶ Watch and force tables ▶ Server

	Name	Address	Display format	Monitor value	Modify value
1	// Receive Data				
2	"RecvDataServer".recvData[0]	%DB2.DBB0	Hex	16#11	
3	"RecvDataServer".recvData[1]	%DB2.DBB1	Hex	16#12	
4	"RecvDataServer".recvData[2]	%DB2.DBB2	Hex	16#13	
5	"RecvDataServer".recvData[3]	%DB2.DBB3	Hex	16#14	
6	"RecvDataServer".recvData[4]	%DB2.DBB4	Hex	16#15	
7	"RecvDataServer".recvData[5]	%DB2.DBB5	Hex	16#16	
8	"RecvDataServer".recvData[6]	%DB2.DBB6	Hex	16#17	
9	"RecvDataServer".recvData[7]	%DB2.DBB7	Hex	16#18	
10	"RecvDataServer".recvData[8]	%DB2.DBB8	Hex	16#19	
11	"RecvDataServer".recvData[9]	%DB2.DBB9	Hex	16#20	
12					
13	// Send Data				
14	"SendDataServer".sendData[0]	%DB1.DBB0	Hex	16#26	16#26
15	"SendDataServer".sendData[1]	%DB1.DBB1	Hex	16#09	16#09
16	"SendDataServer".sendData[2]	%DB1.DBB2	Hex	16#19	16#19
17	"SendDataServer".sendData[3]	%DB1.DBB3	Hex	16#81	16#81
18	"SendDataServer".sendData[4]	%DB1.DBB4	Hex	16#30	16#30
19	"SendDataServer".sendData[5]	%DB1.DBB5	Hex	16#06	16#06
20	"SendDataServer".sendData[6]	%DB1.DBB6	Hex	16#19	16#19
21	"SendDataServer".sendData[7]	%DB1.DBB7	Hex	16#84	16#84
22	"SendDataServer".sendData[8]	%DB1.DBB8	Hex	16#03	16#03
23	"SendDataServer".sendData[9]	%DB1.DBB9	Hex	16#11	16#11
24	<Add new>				

### Using the variable table in STEP 7 V5.6

1. In the "Blocks" folder you double-click the variable table "VAT\_Client". The variable table opens.

SIMATIC Manager - [PUT\_GET\_zu\_S71200 -- D:\Temp\s7communication\S7communication\PUT\_

File Edit Insert PLC View Options Window Help

< No Filter >

Object name	Symbolic name
Systemdaten	---
OB1	
FB14	GET
FB15	PUT
DB11	SEND_DATA_CLIENT
DB12	RCV_DATA_CLIENT
DB14	
DB15	
DB24	SEND_DATA_SERVER
DB25	RCV_DATA_SERVER
VAT_Client	VAT_Client
VAT_Server	VAT_Server
SFB14	
SFB15	
SFC20	BLKMOV

2. Click the "Monitor variable" button in the toolbar. The "Status value" column displays the current values in the Send data area (DB11) and Receive data area (DB12). This permits you to check whether the data read from the S7-1200 CPU has arrived in the Receive data area (DB12) of the S7-300 CPU.

	Address	Symbol	Display format	Status value	Memory value
1		//SEND DATA			
2	DB11.DBB 0	"SEND_DATA_CLIENT".SEND_DATA[0]	HEX	B#16#11	B#16#11
3	DB11.DBB 1	"SEND_DATA_CLIENT".SEND_DATA[1]	HEX	B#16#12	B#16#12
4	DB11.DBB 2	"SEND_DATA_CLIENT".SEND_DATA[2]	HEX	B#16#13	B#16#13
5	DB11.DBB 3	"SEND_DATA_CLIENT".SEND_DATA[3]	HEX	B#16#14	B#16#14
6	DB11.DBB 4	"SEND_DATA_CLIENT".SEND_DATA[4]	HEX	B#16#15	B#16#15
7	DB11.DBB 5	"SEND_DATA_CLIENT".SEND_DATA[5]	HEX	B#16#16	B#16#16
8	DB11.DBB 6	"SEND_DATA_CLIENT".SEND_DATA[6]	HEX	B#16#17	B#16#17
9	DB11.DBB 7	"SEND_DATA_CLIENT".SEND_DATA[7]	HEX	B#16#18	B#16#18
10	DB11.DBB 8	"SEND_DATA_CLIENT".SEND_DATA[8]	HEX	B#16#19	B#16#19
11	DB11.DBB 9	"SEND_DATA_CLIENT".SEND_DATA[9]	HEX	B#16#20	B#16#20
12					
13		//RECEIVE DATA			
14	DB12.DBB 0	"RCV_DATA_CLIENT".RCV_DATA[0]	HEX		B#16#26
15	DB12.DBB 1	"RCV_DATA_CLIENT".RCV_DATA[1]	HEX		B#16#09
16	DB12.DBB 2	"RCV_DATA_CLIENT".RCV_DATA[2]	HEX		B#16#19
17	DB12.DBB 3	"RCV_DATA_CLIENT".RCV_DATA[3]	HEX		B#16#81
18	DB12.DBB 4	"RCV_DATA_CLIENT".RCV_DATA[4]	HEX		B#16#30
19	DB12.DBB 5	"RCV_DATA_CLIENT".RCV_DATA[5]	HEX		B#16#06
20	DB12.DBB 6	"RCV_DATA_CLIENT".RCV_DATA[6]	HEX		B#16#19
21	DB12.DBB 7	"RCV_DATA_CLIENT".RCV_DATA[7]	HEX		B#16#84
22	DB12.DBB 8	"RCV_DATA_CLIENT".RCV_DATA[8]	HEX		B#16#03
23	DB12.DBB 9	"RCV_DATA_CLIENT".RCV_DATA[9]	HEX		B#16#11

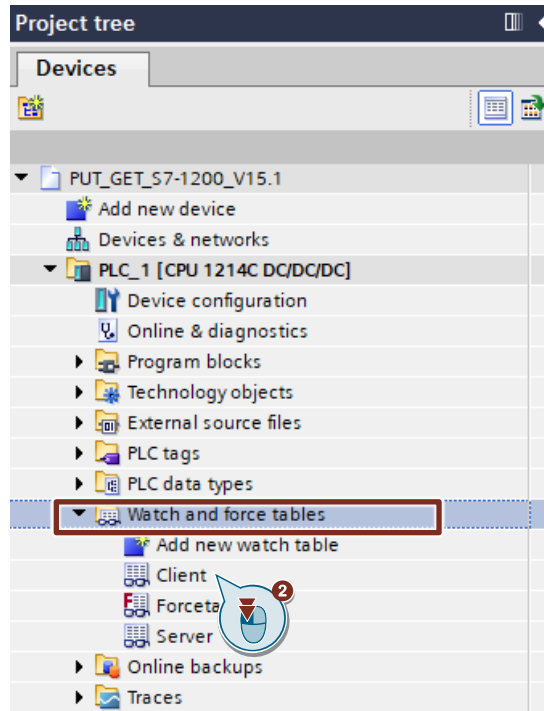
## 4.2 SIMATIC S7-1200 Writes and Reads Data as Client

### 4.2.1 Write Data to the S7-300

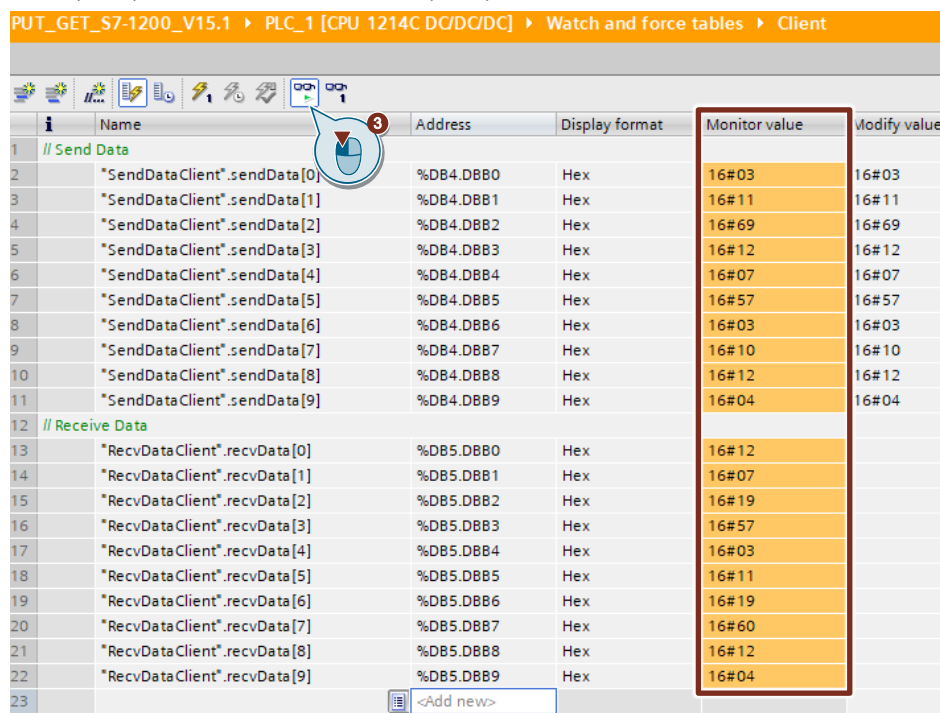
#### Using the variable table in STEP 7 V15.1

1. In the project tree you open the device folder of the S7-1200 CPU.
2. In the "Watch and force tables" folder you double-click the watch table "Client". The watch table opens in the working area of STEP 7 V15.1.





3. Click the "Monitor all" button in the toolbar. The "Monitor value" column displays the current values stored in the Send data area (DB4) and Receive data area (DB5).



4. In the "Modify value" column you enter the values that are to be put in the Send data area (DB4) of the S7-1200 CPU and written to the S7-300 CPU.
5. In the toolbar you click the "Modify all enabled values once and immediately" button.

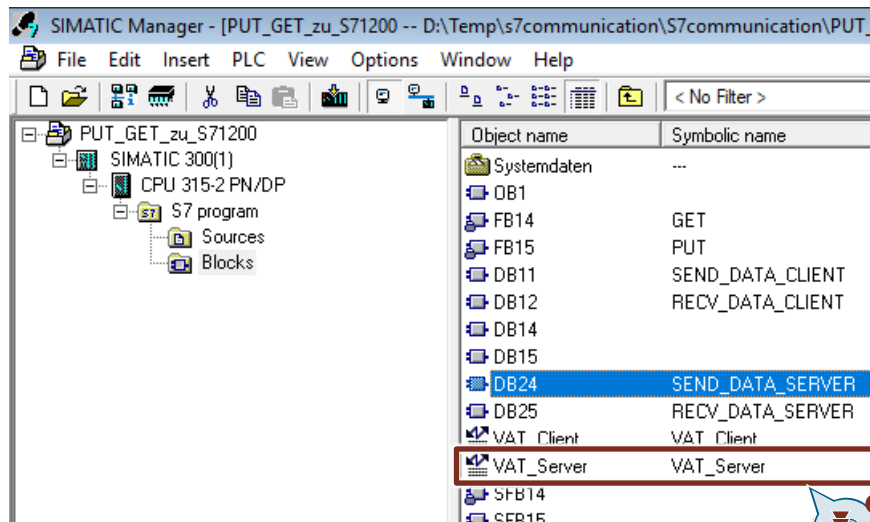
The modified values are put in the Send data area (DB4) of the S7-1200 CPU and written to the S7-300 CPU.

PUT\_GET\_S7-1200\_V15.1 ▶ PLC\_1 [CPU 1214C DC/DC/DC] ▶ Watch and force tables ▶ Client

	Name	Address	Display format	Monitor value	Modify value
1	// Send Data				
2	"SendDataClient".sendData[0]	%DB4.DBB0	Hex	16#03	16#03
3	"SendDataClient".sendData[1]	%DB4.DBB1	Hex	16#11	16#11
4	"SendDataClient".sendData[2]	%DB4.DBB2	Hex	16#69	16#69
5	"SendDataClient".sendData[3]	%DB4.DBB3	Hex	16#12	16#12
6	"SendDataClient".sendData[4]	%DB4.DBB4	Hex	16#07	16#07
7	"SendDataClient".sendData[5]	%DB4.DBB5	Hex	16#57	16#57
8	"SendDataClient".sendData[6]	%DB4.DBB6	Hex	16#03	16#03
9	"SendDataClient".sendData[7]	%DB4.DBB7	Hex	16#10	16#10
10	"SendDataClient".sendData[8]	%DB4.DBB8	Hex	16#12	16#12
11	"SendDataClient".sendData[9]	%DB4.DBB9	Hex	16#04	16#04
12	// Receive Data				
13	"RecvDataClient".recvData[0]	%DB5.DBB0	Hex	16#12	
14	"RecvDataClient".recvData[1]	%DB5.DBB1	Hex	16#07	
15	"RecvDataClient".recvData[2]	%DB5.DBB2	Hex	16#19	
16	"RecvDataClient".recvData[3]	%DB5.DBB3	Hex	16#57	
17	"RecvDataClient".recvData[4]	%DB5.DBB4	Hex	16#03	
18	"RecvDataClient".recvData[5]	%DB5.DBB5	Hex	16#11	
19	"RecvDataClient".recvData[6]	%DB5.DBB6	Hex	16#19	
20	"RecvDataClient".recvData[7]	%DB5.DBB7	Hex	16#60	
21	"RecvDataClient".recvData[8]	%DB5.DBB8	Hex	16#12	
22	"RecvDataClient".recvData[9]	%DB5.DBB9	Hex	16#04	
23		<Add new>			

### Using the variable table in STEP 7 V5.6

1. In the "Blocks" folder you double-click the variable table "VAT\_Server". The variable table opens.



2. Click the "Monitor variable" button in the toolbar. The "Status value" column displays the current values in the Send data area (DB24) and Receive data area (DB25). This permits you to check whether the data written from the S7-1200 CPU has arrived in the Receive data area (DB25) of the S7-300 CPU.

## 4 Operating and Monitoring

Var - VAT\_Server

Table Edit Insert PLC Variable View Options Window Help

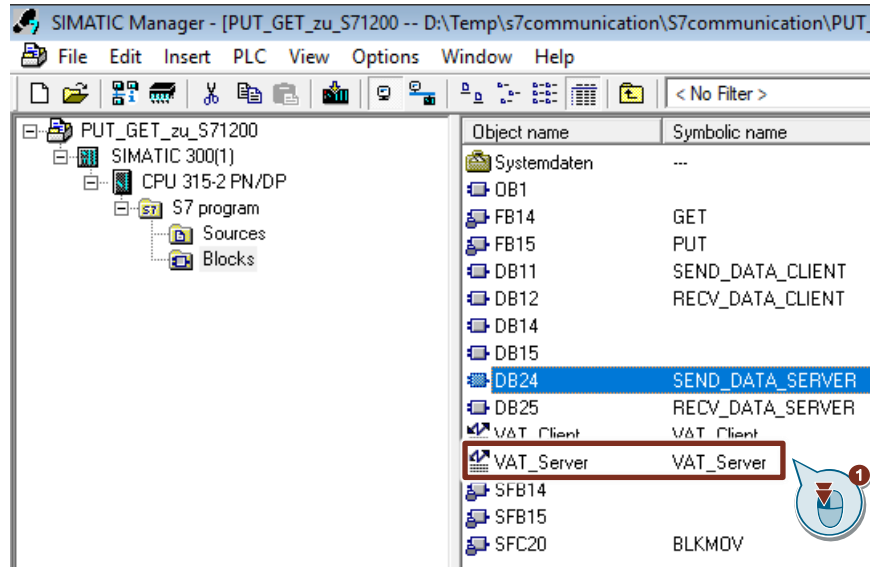
Var - VAT\_Server -- @PUT\_GET\_zu\_S71200\SIMATIC 300(1)\CPU 315-2 PN/DP\S7 program ONL

	Address	Symbol	Display format	Status value	Initial value
1	<b>//Receive Data</b>				
2	DB25.DBB 0	"RECV_DATA_SERVER".RECV_DATA[0]	HEX	B#16#03	
3	DB25.DBB 1	"RECV_DATA_SERVER".RECV_DATA[1]	HEX	B#16#11	
4	DB25.DBB 2	"RECV_DATA_SERVER".RECV_DATA[2]	HEX	B#16#69	
5	DB25.DBB 3	"RECV_DATA_SERVER".RECV_DATA[3]	HEX	B#16#12	
6	DB25.DBB 4	"RECV_DATA_SERVER".RECV_DATA[4]	HEX	B#16#07	
7	DB25.DBB 5	"RECV_DATA_SERVER".RECV_DATA[5]	HEX	B#16#57	
8	DB25.DBB 6	"RECV_DATA_SERVER".RECV_DATA[6]	HEX	B#16#03	
9	DB25.DBB 7	"RECV_DATA_SERVER".RECV_DATA[7]	HEX	B#16#10	
10	DB25.DBB 8	"RECV_DATA_SERVER".RECV_DATA[8]	HEX	B#16#12	
11	DB25.DBB 9	"RECV_DATA_SERVER".RECV_DATA[9]	HEX	B#16#04	
12	<b>//Send Data</b>				
13	DB24.DBB 0	"SEND_DATA_SERVER".SEND_DATA[0]	HEX	B#16#12	B#16#12
14	DB24.DBB 1	"SEND_DATA_SERVER".SEND_DATA[1]	HEX	B#16#07	B#16#07
15	DB24.DBB 2	"SEND_DATA_SERVER".SEND_DATA[2]	HEX	B#16#19	B#16#19
16	DB24.DBB 3	"SEND_DATA_SERVER".SEND_DATA[3]	HEX	B#16#57	B#16#57
17	DB24.DBB 4	"SEND_DATA_SERVER".SEND_DATA[4]	HEX	B#16#03	B#16#03
18	DB24.DBB 5	"SEND_DATA_SERVER".SEND_DATA[5]	HEX	B#16#11	B#16#11
19	DB24.DBB 6	"SEND_DATA_SERVER".SEND_DATA[6]	HEX	B#16#19	B#16#19
20	DB24.DBB 7	"SEND_DATA_SERVER".SEND_DATA[7]	HEX	B#16#60	B#16#60
21	DB24.DBB 8	"SEND_DATA_SERVER".SEND_DATA[8]	HEX	B#16#12	B#16#12
22	DB24.DBB 9	"SEND_DATA_SERVER".SEND_DATA[9]	HEX	B#16#04	B#16#04
23					

### 4.2.2 Read Data from the S7-300

#### Using the variable table in STEP 7 V5.6

1. In the "Blocks" folder you double-click the variable table "VAT\_Client". The variable table opens.



2. Click the "Monitor variable" button in the toolbar. The "Status value" column displays the current values in the Send data area (DB24) and Receive data area (DB25).

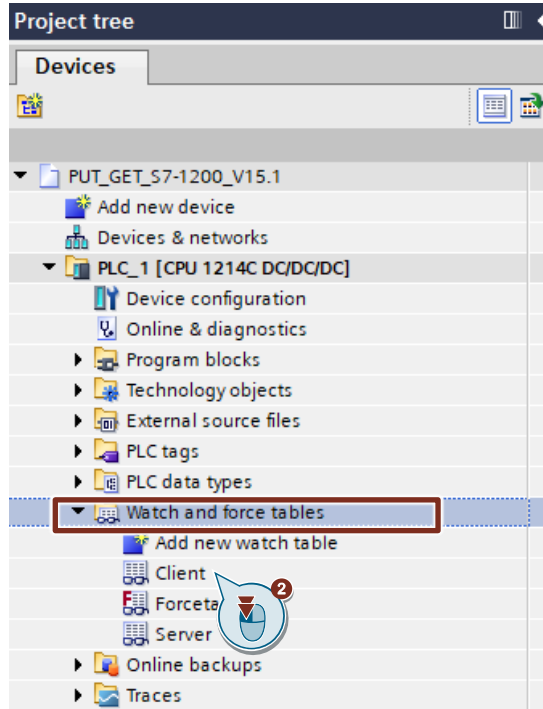
	Address	Symbol	Display format	Status value	Modify value
1	<b>//Receive Data</b>				
2	DB25.DBB 0	"RECV_DATA_SERVER".RECV_DATA[0]	HEX	B#16#03	
3	DB25.DBB 1	"RECV_DATA_SERVER".RECV_DATA[1]	HEX	B#16#11	
4	DB25.DBB 2	"RECV_DATA_SERVER".RECV_DATA[2]	HEX	B#16#69	
5	DB25.DBB 3	"RECV_DATA_SERVER".RECV_DATA[3]	HEX	B#16#12	
6	DB25.DBB 4	"RECV_DATA_SERVER".RECV_DATA[4]	HEX	B#16#07	
7	DB25.DBB 5	"RECV_DATA_SERVER".RECV_DATA[5]	HEX	B#16#57	
8	DB25.DBB 6	"RECV_DATA_SERVER".RECV_DATA[6]	HEX	B#16#03	
9	DB25.DBB 7	"RECV_DATA_SERVER".RECV_DATA[7]	HEX	B#16#10	
10	DB25.DBB 8	"RECV_DATA_SERVER".RECV_DATA[8]	HEX	B#16#12	
11	DB25.DBB 9	"RECV_DATA_SERVER".RECV_DATA[9]	HEX	B#16#04	
12	<b>//Send Data</b>				
13	DB24.DBB 0	"SEND_DATA_SERVER".SEND_DATA[0]	HEX	B#16#12	B#16#12
14	DB24.DBB 1	"SEND_DATA_SERVER".SEND_DATA[1]	HEX	B#16#07	B#16#07
15	DB24.DBB 2	"SEND_DATA_SERVER".SEND_DATA[2]	HEX	B#16#19	B#16#19
16	DB24.DBB 3	"SEND_DATA_SERVER".SEND_DATA[3]	HEX	B#16#57	B#16#57
17	DB24.DBB 4	"SEND_DATA_SERVER".SEND_DATA[4]	HEX	B#16#03	B#16#03
18	DB24.DBB 5	"SEND_DATA_SERVER".SEND_DATA[5]	HEX	B#16#11	B#16#11
19	DB24.DBB 6	"SEND_DATA_SERVER".SEND_DATA[6]	HEX	B#16#19	B#16#19
20	DB24.DBB 7	"SEND_DATA_SERVER".SEND_DATA[7]	HEX	B#16#60	B#16#60
21	DB24.DBB 8	"SEND_DATA_SERVER".SEND_DATA[8]	HEX	B#16#12	B#16#12
22	DB24.DBB 9	"SEND_DATA_SERVER".SEND_DATA[9]	HEX	B#16#04	B#16#04
23					

3. In the "Modify value" column you enter the values that are to be put in the Send data area (DB24) of the S7-300 CPU and read from the S7-1200 CPU. The data read is stored in the Receive data area (DB5) of the S7-1200.
4. Click the "Activate modify values" button in the toolbar. The modified values are put in the Send data area (DB24) of the S7-300 CPU.

	Address	Symbol	Display format	Status value	Modify
1	<b>//Receive Data</b>				
2	DB25.DBB 0	"RECV_DATA_SERVER".RECV_DATA[0]	HEX	B#16#03	
3	DB25.DBB 1	"RECV_DATA_SERVER".RECV_DATA[1]	HEX	B#16#11	
4	DB25.DBB 2	"RECV_DATA_SERVER".RECV_DATA[2]	HEX	B#16#69	
5	DB25.DBB 3	"RECV_DATA_SERVER".RECV_DATA[3]	HEX	B#16#12	
6	DB25.DBB 4	"RECV_DATA_SERVER".RECV_DATA[4]	HEX	B#16#07	
7	DB25.DBB 5	"RECV_DATA_SERVER".RECV_DATA[5]	HEX	B#16#57	
8	DB25.DBB 6	"RECV_DATA_SERVER".RECV_DATA[6]	HEX	B#16#03	
9	DB25.DBB 7	"RECV_DATA_SERVER".RECV_DATA[7]	HEX	B#16#10	
10	DB25.DBB 8	"RECV_DATA_SERVER".RECV_DATA[8]	HEX	B#16#12	
11	DB25.DBB 9	"RECV_DATA_SERVER".RECV_DATA[9]	HEX	B#16#04	
12	<b>//Send Data</b>				
13	DB24.DBB 0	"SEND_DATA_SERVER".SEND_DATA[0]	HEX	B#16#12	B#16#12
14	DB24.DBB 1	"SEND_DATA_SERVER".SEND_DATA[1]	HEX	B#16#07	B#16#07
15	DB24.DBB 2	"SEND_DATA_SERVER".SEND_DATA[2]	HEX	B#16#19	B#16#19
16	DB24.DBB 3	"SEND_DATA_SERVER".SEND_DATA[3]	HEX	B#16#57	B#16#57
17	DB24.DBB 4	"SEND_DATA_SERVER".SEND_DATA[4]	HEX	B#16#03	B#16#03
18	DB24.DBB 5	"SEND_DATA_SERVER".SEND_DATA[5]	HEX	B#16#11	B#16#11
19	DB24.DBB 6	"SEND_DATA_SERVER".SEND_DATA[6]	HEX	B#16#19	B#16#19
20	DB24.DBB 7	"SEND_DATA_SERVER".SEND_DATA[7]	HEX	B#16#60	B#16#60
21	DB24.DBB 8	"SEND_DATA_SERVER".SEND_DATA[8]	HEX	B#16#12	B#16#12
22	DB24.DBB 9	"SEND_DATA_SERVER".SEND_DATA[9]	HEX	B#16#04	B#16#04
23					

**Using the variable table in STEP 7 V15.1**

1. In the project tree you open the device folder of the S7-1200 CPU.
2. In the "Watch and force tables" folder you double-click the watch table "Client". The watch table opens in the working area of STEP 7 V15.1.



3. Click the "Monitor all" button in the toolbar.  
The "Monitor value" column displays the current values stored in the Send data area (DB4) and Receive data area (DB5). This permits you to check whether the data read from the S7-300 CPU has arrived in the Receive data area (DB5) of the S7-1200 CPU.

PUT\_GET\_S7-1200\_V15.1 ▶ PLC\_1 [CPU 1214C DC/DC/DC] ▶ Watch and force tables ▶ Client

	Name	Address	Display format	Monitor value	Modify value
1	// Send Data				
2	"SendDataClient".sendData[0]	%DB4.DBB0	Hex	16#03	16#03
3	"SendDataClient".sendData[1]	%DB4.DBB1	Hex	16#11	16#11
4	"SendDataClient".sendData[2]	%DB4.DBB2	Hex	16#69	16#69
5	"SendDataClient".sendData[3]	%DB4.DBB3	Hex	16#12	16#12
6	"SendDataClient".sendData[4]	%DB4.DBB4	Hex	16#07	16#07
7	"SendDataClient".sendData[5]	%DB4.DBB5	Hex	16#57	16#57
8	"SendDataClient".sendData[6]	%DB4.DBB6	Hex	16#03	16#03
9	"SendDataClient".sendData[7]	%DB4.DBB7	Hex	16#10	16#10
10	"SendDataClient".sendData[8]	%DB4.DBB8	Hex	16#12	16#12
11	"SendDataClient".sendData[9]	%DB4.DBB9	Hex	16#04	16#04
12	// Receive Data				
13	"RecvDataClient".recvData[0]	%DB5.DBB0	Hex	16#12	
14	"RecvDataClient".recvData[1]	%DB5.DBB1	Hex	16#07	
15	"RecvDataClient".recvData[2]	%DB5.DBB2	Hex	16#19	
16	"RecvDataClient".recvData[3]	%DB5.DBB3	Hex	16#57	
17	"RecvDataClient".recvData[4]	%DB5.DBB4	Hex	16#03	
18	"RecvDataClient".recvData[5]	%DB5.DBB5	Hex	16#11	
19	"RecvDataClient".recvData[6]	%DB5.DBB6	Hex	16#19	
20	"RecvDataClient".recvData[7]	%DB5.DBB7	Hex	16#60	
21	"RecvDataClient".recvData[8]	%DB5.DBB8	Hex	16#12	
22	"RecvDataClient".recvData[9]	%DB5.DBB9	Hex	16#04	
23					