

SIEMENS

Ingenuity for life

Industry Online Support

Home

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

STEP 7 V16 / STEP 7 V5.6 / PUT / GET

<https://support.industry.siemens.com/cs/ww/en/view/82212115>

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) apply.

Security
informa-
tion

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. 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 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>.

Table of content

1	Introduction	5
2	Configuration of SIMATIC S7-1200	6
2.1	Create the Hardware Configuration	6
2.1.1	Set the IP address of the CPU	6
2.1.2	Set the IP address of the CP	7
2.1.3	Permit Access with PUT/GET Communication from Remote Partner	8
2.1.4	Enable S7 communication to the CPU	8
2.2	Install SIMATIC S7-1200 as Client	9
2.2.1	Configuration of the S7 Connection	9
2.2.2	Create Send and Receive Data Areas	13
2.2.3	Create Variables for Parameters of Function Blocks "PutData" and "GetData"	17
2.2.4	Create a User Program	20
2.3	Install SIMATIC S7-1200 as Server	27
2.3.1	Create Send and Receive Data Areas	27
2.4	Download the Hardware Configuration and User Program	33
3	Configuration of SIMATIC S7-300 in STEP 7 V16	37
3.1	Create the Hardware Configuration	37
3.1.1	Set the IP address of the CPU	37
3.1.2	Set the IP address of the CP	38
3.2	Install SIMATIC S7-300 as Client	39
3.2.1	Configuration of the S7 Connection	39
3.2.2	Create Send and Receive Data Areas	43
3.2.3	Create Variables for Parameters of Function Blocks "PutData" and "GetData"	47
3.2.4	Create a User Program	50
3.3	Install SIMATIC S7-300 as Server	57
3.3.1	Create Send and Receive Data Areas	57
3.4	Download the Hardware Configuration and User Program	61
4	Configuration of SIMATIC S7-300 in STEP 7 V5.6	65
4.1	Create the Hardware Configuration	65
4.1.1	Open the Hardware Configuration	65
4.1.2	Define IP Address and Assign Subnet to the PROFINET interface of the CPU	67
4.1.3	Define IP Address and Assign Subnet to the PROFINET interface of the CP	69
4.2	Configure the SIMATIC S7-300 as Client	71
4.2.1	Configuration of the S7 Connection	71
4.2.2	Create Send and Receive Data Areas	76
4.2.3	Create Variables for Parameters of Function Blocks "PutData" and "GetData"	79
4.2.4	Create a User Program	82
4.3	Configure the SIMATIC S7-300 as Server	90
4.3.1	Create a User Program	90
4.4	Download the Hardware Configuration and User Program	94
5	Error Handling	96
5.1	FB "PutData"	96
5.2	FB "GetData"	96
6	Operating and Monitoring	97
6.1	SIMATIC S7-1200 Writes and Reads Data as Client	97

6.1.1	Write Data to the S7-300.....	97
6.1.2	Read Data from the S7-300	99
6.2	SIMATIC S7-300 Writes and Reads Data as Client (STEP 7 V16)	101
6.2.1	Write Data to the S7-1200.....	101
6.2.2	Read Data from the S7-1200	103
6.3	SIMATIC S7-300 Writes and Reads Data as Client (STEP 7 V5.6)	105
6.3.1	Write Data to the S7-1200 CPU	105
6.3.2	Read Data from the S7-1200 CPU.....	107

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 in STEP 7 V16 (TIA Portal V16).

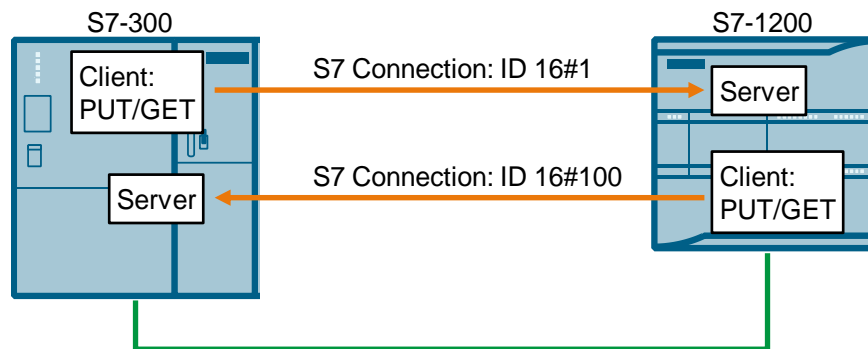
The SIMATIC S7-300 can be configured in STEP 7 V16 (TIA Portal V16) or in STEP 7 V5.6.

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 "GET" and "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



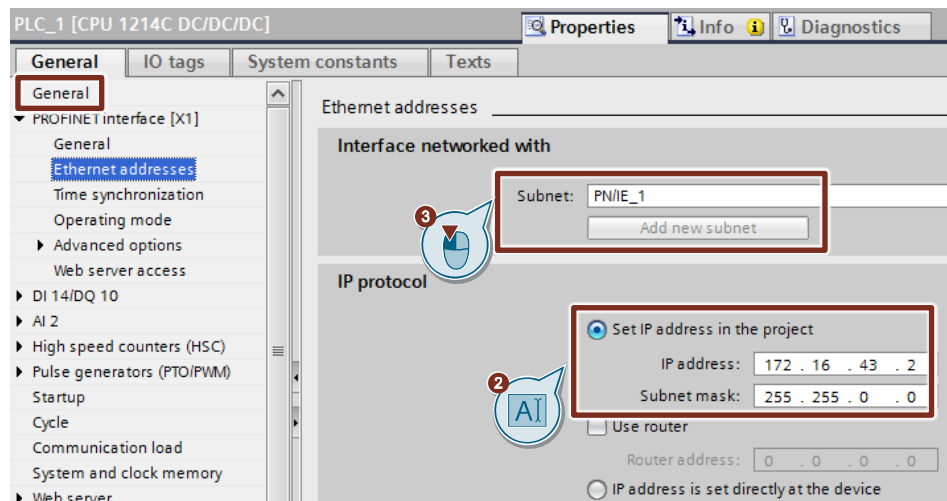
2 Configuration of SIMATIC S7-1200

1. Create a new project or open an existing project in STEP 7 V16 (TIA Portal V16).
2. Add the S7-1200 CPU and create the hardware configuration.
3. Install the S7-1200 CPU as client or server.
 - Client: Configure a new S7 connection and create the user program. In the user program you call the "PUT" instruction to write data to the S7-300 CPU. In the user program you call the "GET" instruction to read data from the S7-300 CPU. Add the data blocks for saving the Send and Receive data.
 - Server: Create the user program. In the user program you add data blocks in which the sent and received data is stored.

2.1 Create the Hardware Configuration

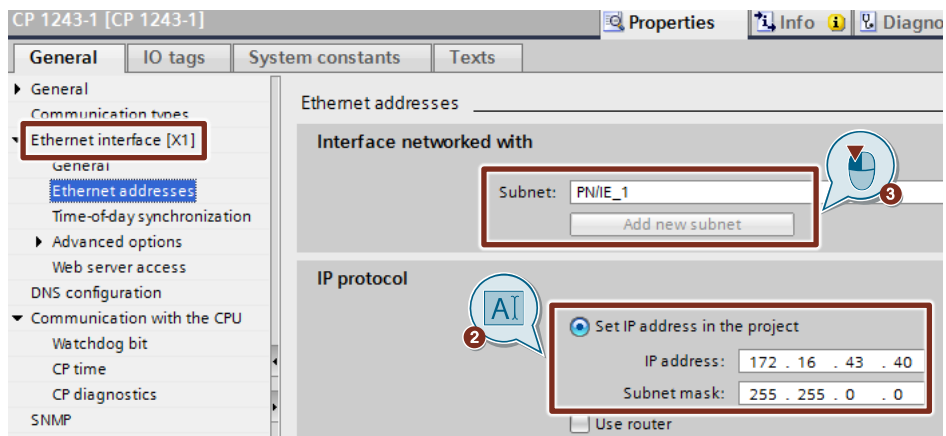
2.1.1 Set the IP address of the CPU

1. In the Device view or Network view of the "Devices & networks" editor you mark the S7-1200 CPU. The properties of the S7-1200 CPU are displayed in the inspector window.
2. In the "General" tab under "PROFINET interface [X1] > Ethernet addresses" you enter the IP address and the subnet mask of the S7-1200 CPU.
 - IP address: 172.16.43.2
 - Subnet mask: 255.255.0.0
3. Assign a subnet to the Ethernet interface of the S7-1200 CPU. You have two options:
 - Click the "Add new subnet" button to add a new subnet.
 - Select an existing subnet.



2.1.2 Set the IP address of the CP

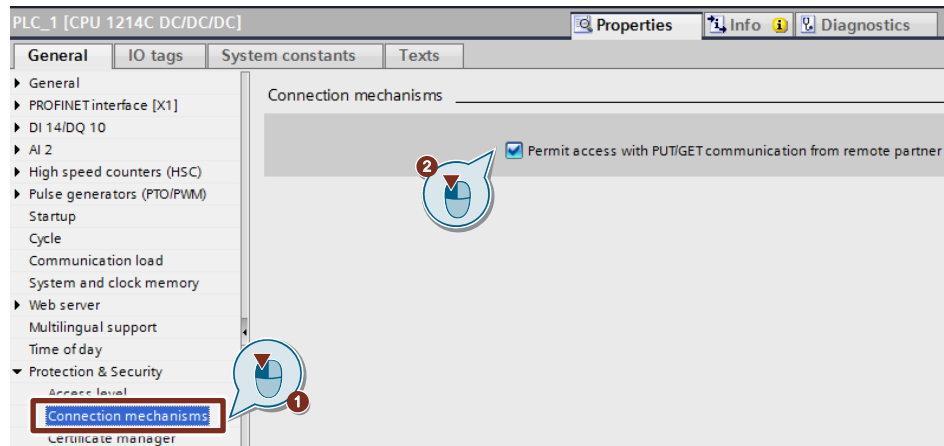
1. In the Device view or Network view of the "Devices & networks" editor you mark the CP in the S7-1200.
The properties of the CP are displayed in the inspector window.
2. In the "General" tab under "Ethernet interface [X1] > Ethernet addresses" you enter the IP address and the subnet mask of the CP.
 - IP address: 172.16.43.40
 - Subnet mask: 255.255.0.0
3. Assign a subnet to the Ethernet interface of the CP. You have two options:
 - Click the "Add new subnet" button to add a new subnet.
 - Select an existing subnet.



2.1.3 Permit Access with PUT/GET Communication from Remote Partner

In the S7-1200 CPU you enable the function "Permit access with PUT/GET communication from remote partner". This enables the partner CPU to access the data in the S7-1200 CPU using the "PUT" and "GET" instructions.

1. In the Device view or Network view of the "Devices & networks" editor you mark the S7-1200 CPU.
The properties of the S7-1200 CPU are displayed in the inspector window.
2. In the "General" tab, under "Protection & Security > Connection mechanisms" you enable the "Permit access with PUT/GET communication from remote partner" function.

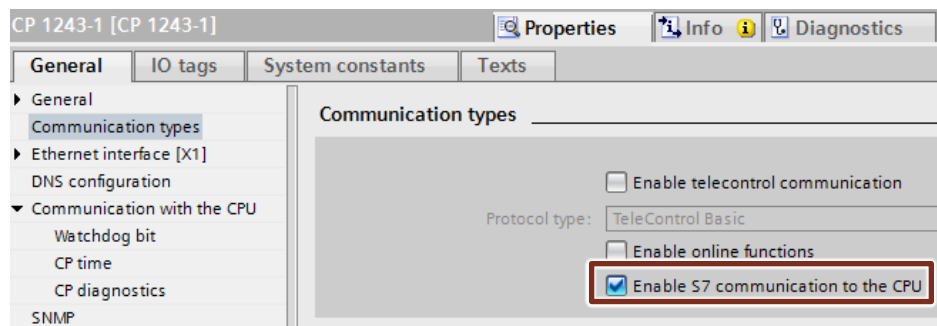


2.1.4 Enable S7 communication to the CPU

With the S7-1200, enable S7 communication to the CPU in the CP. In order to release the function of S7 communication with the assigned CPU and the S7 routing.

If you configure an S7 connection that runs via the CP of the S7-1200, then you have to activate this option.

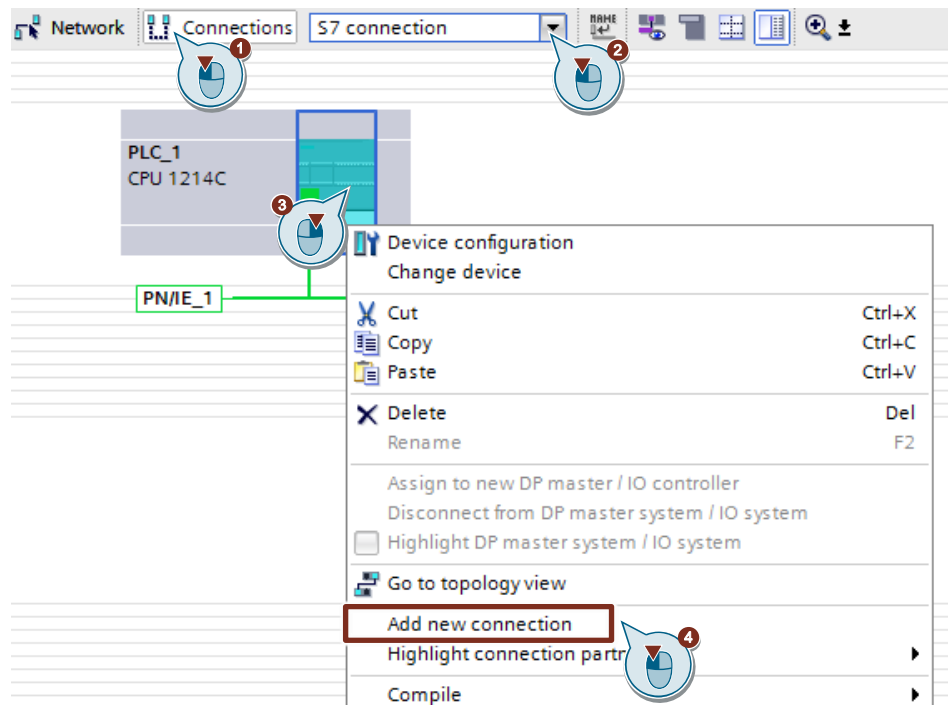
1. In the Device view or Network view of the "Devices & networks" editor you mark the CP in the S7-1200.
2. In the "General" tab, under "Communication types" you enable the "Enable S7 communication to the CPU" option.



2.2 Install SIMATIC S7-1200 as Client

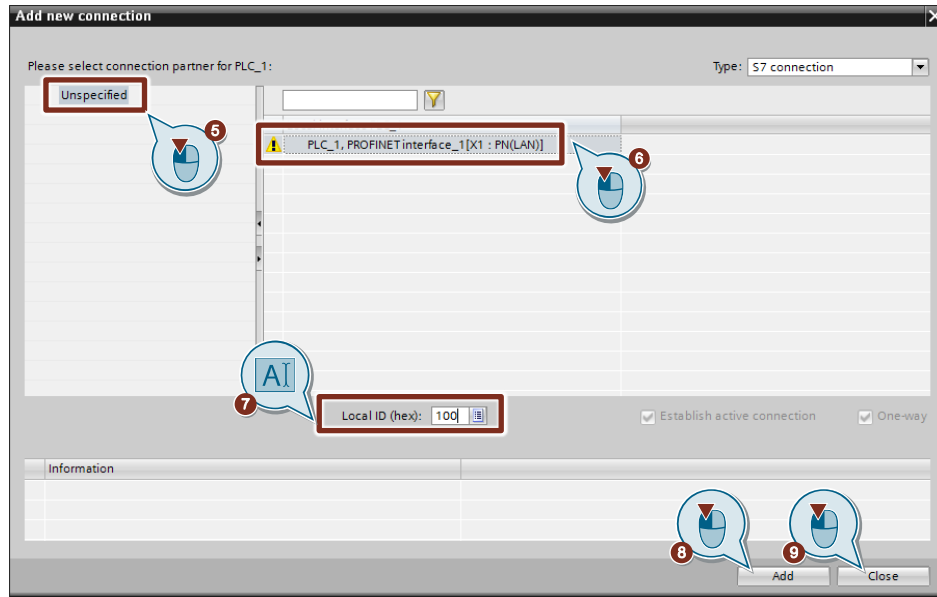
2.2.1 Configuration of the S7 Connection

1. In the Network view you click the "Connections" icon to enable the Connection mode.
2. Select "S7 connection" as the connection type in the adjacent drop-down list box.
In the Network view, all devices that are available for an S7 connection are highlighted in color.
3. Right-click the S7-1200 CPU. The pop-up menu opens.
4. Select the "Add new connection" item.
The "Add new connection" dialog opens.



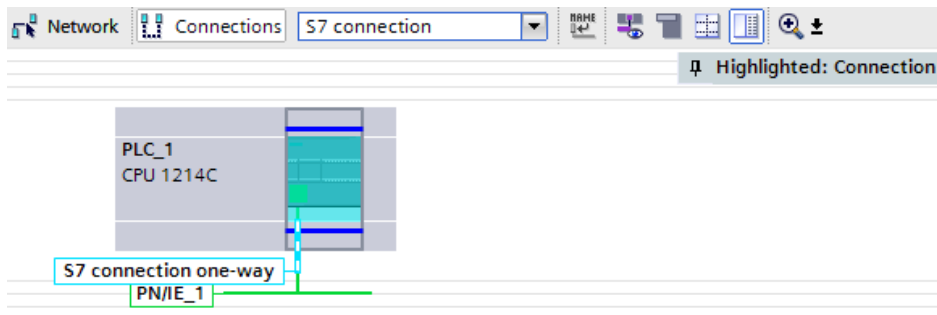
5. Specify the following connection partner: "Unspecified".
6. Select the local interface. In this example, the integrated interface of the S7-1200 CPU is used. If the S7 connection is to be established via a CP, select the CP.
7. Enter the local ID for the S7 connection, 100 (hex), for example.
8. Click the "Add" button to add the unspecified S7 connection.
9. Click the "Close" button to end the dialog.

2 Configuration of SIMATIC S7-1200



Result

- An unspecified S7 connection is created.
- The connection path is displayed highlighted in the graphical area of the Network view.

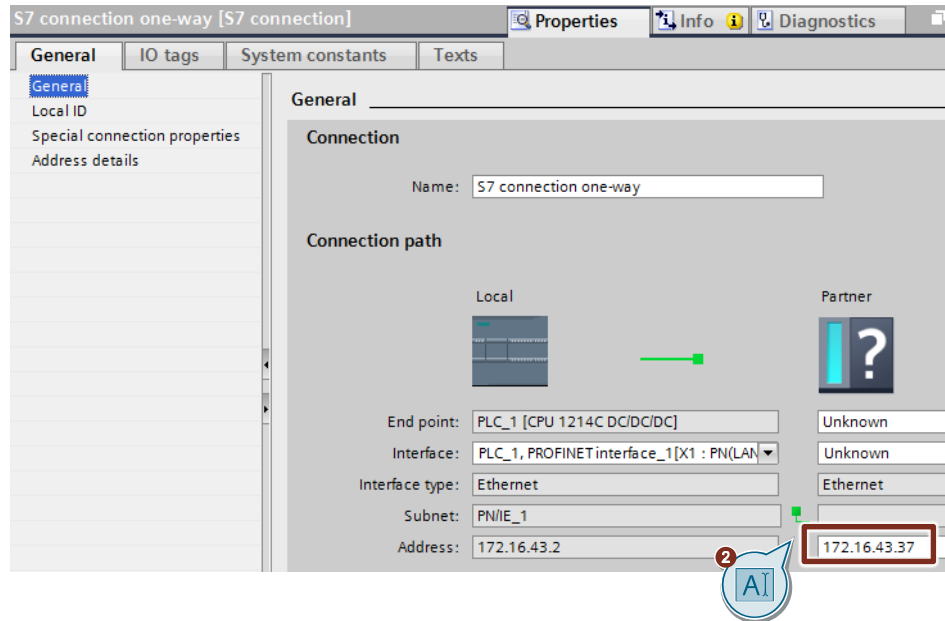


- The S7 connection is entered in the "Connections" table in the table area of the Network view.

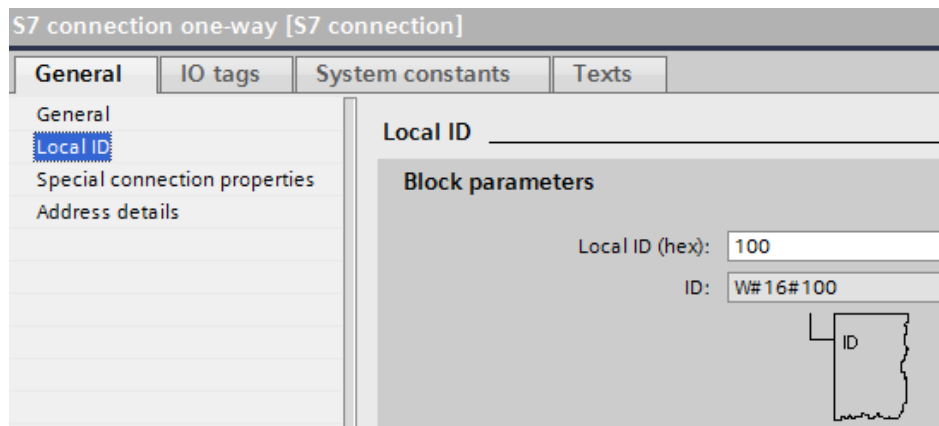
Network overview	Connections	I/O communication	VPN	TeleControl		
	Local connection name	Local end point	Local ID (hex)	Partner ID (hex)	Partner	Connection type ▲
	S7 connection one-way	PLC_1 [CPU 1214C DC/DC/DC]	100		Unknown	S7 connection

Setting the connection parameters

1. In the “Connections” table, mark the unspecified S7 connections.
The Properties of the S7 connection are displayed in the inspector window.
2. Under "General" the "General" tab displays the general connection parameters that identify the local endpoint. Under Partner you enter the IP address of the S7-300 CPU, for example: 172.16.43.37. If you use a CP as partner instead of the integrated interface of the S7-300 CPU, enter the IP address of the CP.

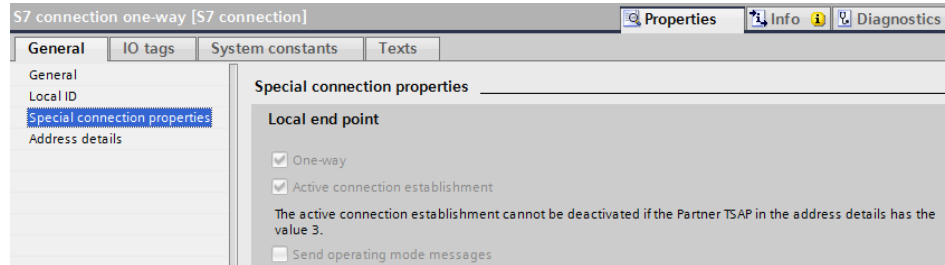


3. In the “General” tab under “Local ID” you see the local ID of the module from which the connection is viewed (local partner). You can change the local ID. You specify the local ID later at the "ID" input parameter of the "PUT" or "GET" instruction.
In this example we use the local ID with the value 100 (hex).



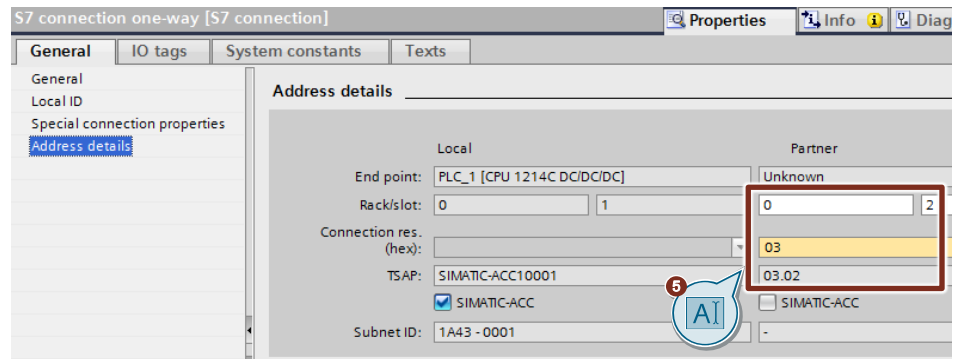
2 Configuration of SIMATIC S7-1200

- In the "General" under "Special connection properties" you see that the S7 connection is configured unilaterally. Unilaterally means that the connection partner is server for this connection and cannot actively send or receive. The S7 CPU that is used as the local endpoint for the S7 connection, actively establishes the S7 connection as a client and actively sends and receives.



- In the "General" tab under "Address details" you enter the partner TSAP. Since the S7 connection is configured unilaterally to an unspecified partner, the partner TSAP is composed as follows:
03 <slot of CPU or CP >, for example:

- S7-300 CPU in slot 2: 03.02

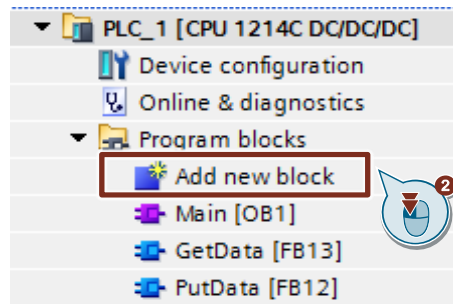


2.2.2 Create Send and Receive Data Areas

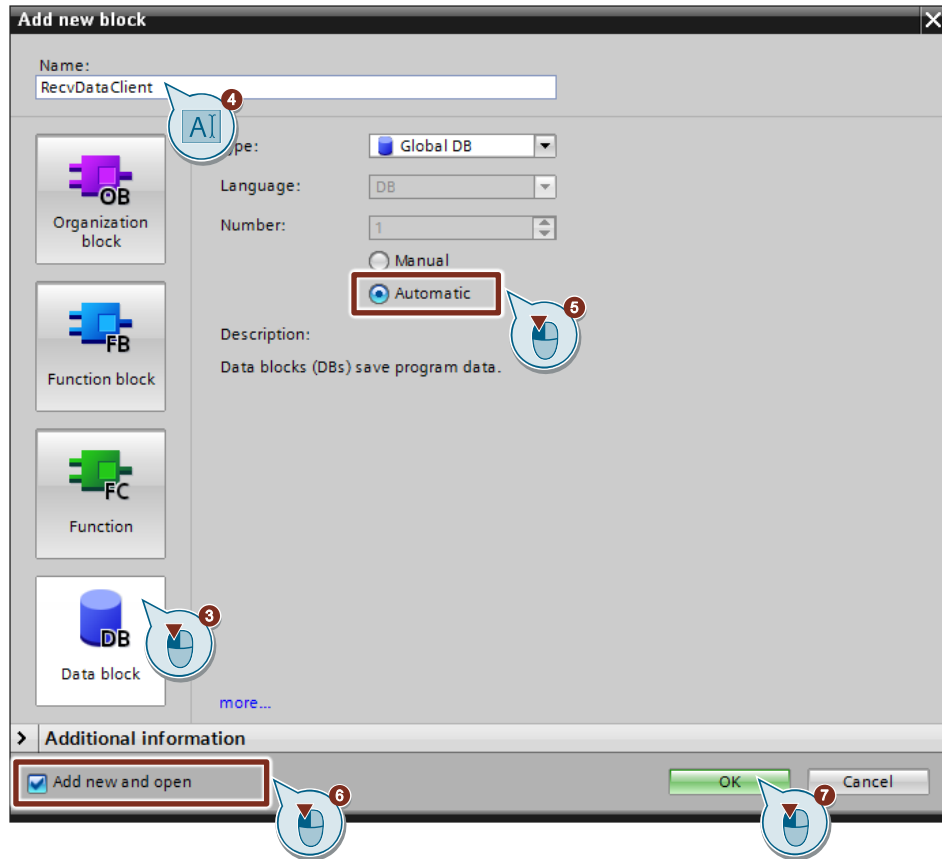
Add data block for saving the Receive data

The data received from the S7-300 CPU is stored in data block DB1 "RecvDataClient" of the S7-1200 CPU.

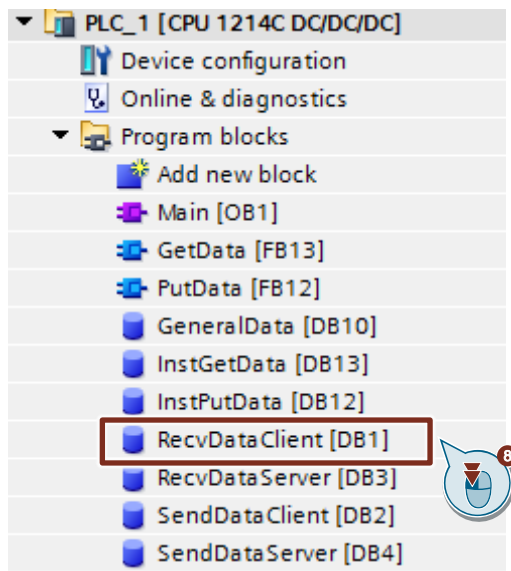
1. In the project tree you navigate to the device folder of the S7-1200 CPU. The device folder contains structured objects and actions that belong to the device.
2. In the device folder you navigate to the "Program blocks" subfolder and double-click the "Add new block" action. The "Add new block" dialog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



3. Click the "Data block (DB)" button.
4. Enter the name of the data block (DB), "RecvDataClient", 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 "RecvDataClient" DB is added to the "Program blocks" folder of the S7-1200 CPU.
 - If the "Add new and open" function is enabled, the "RecvDataClient" DB opens in the working area of STEP 7 V16.



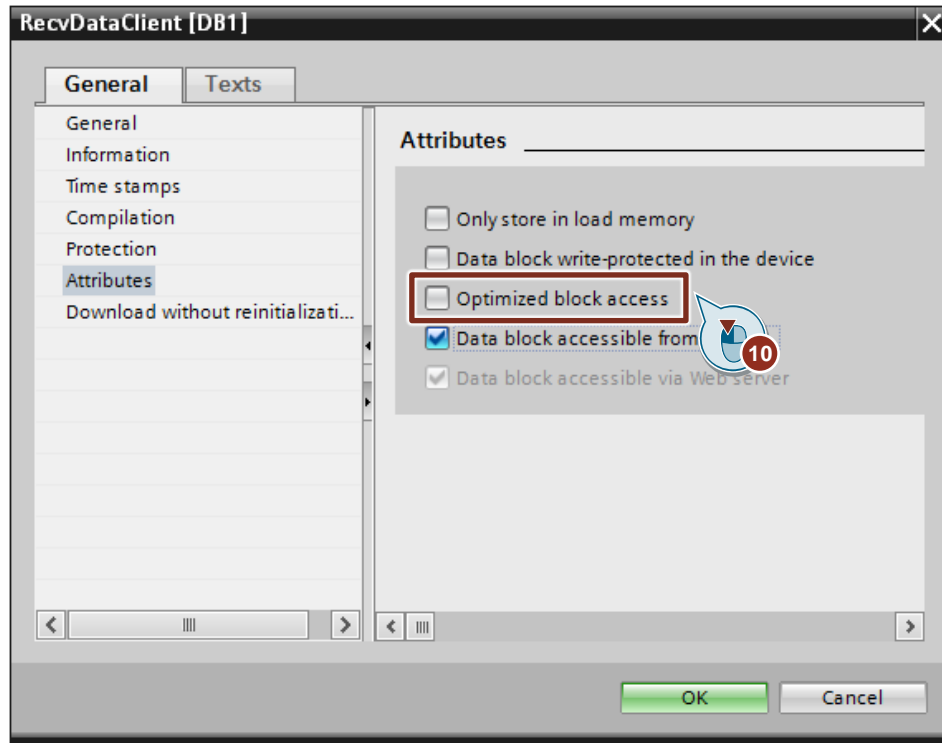
8. Alternatively, you can open the "RecvDataClient" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "RecvDataClient" DB.



- In the DB "RecvDataClient" you define the variable "data" of the data type Array[0..199] of Byte.

RecvDataClient			
Name	Data type	Offset	Start value
Static			
data	Array[0..199] of Byte	0.0	
<Add new>			

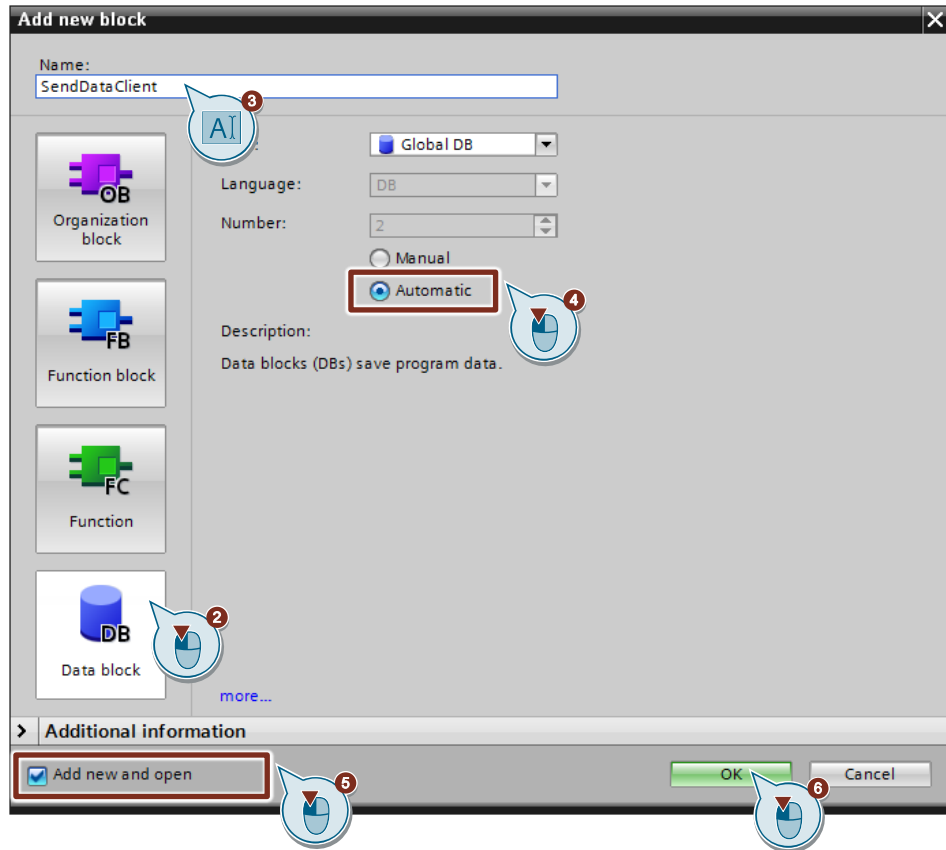
- In the Properties of the DB1 "RecvDataClient", under "Attributes" you disable the "Optimized block access" function.



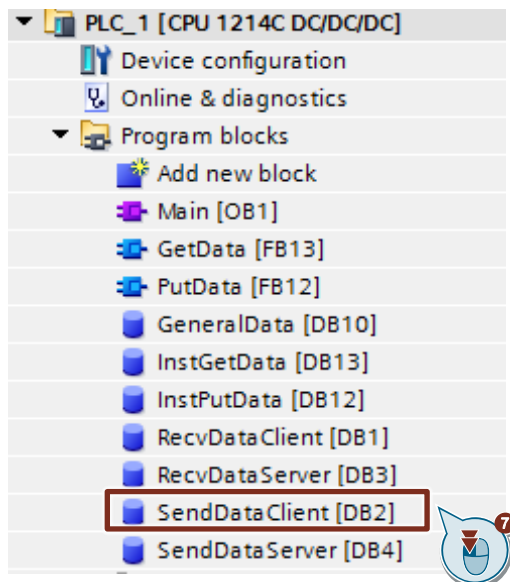
Add data block for saving the Send data

The data that is transferred to the S7-300 CPU is stored in data block DB2 "SendDataClient" of the S7-1200 CPU.

- Add another DB for saving the Send data.
- In the "Add new block" dialog you click the "Data block (DB)" button.
- Enter the name of the DB, "SendDataClient", for example.
- Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
- Enable the "Add new and open" function.
- Click the "OK" button to apply the settings and close the dialog.
 - In the project tree the "SendDataClient" DB is added to the "Program blocks" folder of the S7-1200 CPU.
 - If the "Add new and open" function is enabled, the "SendDataClient" DB opens in the working area of STEP 7 V16.



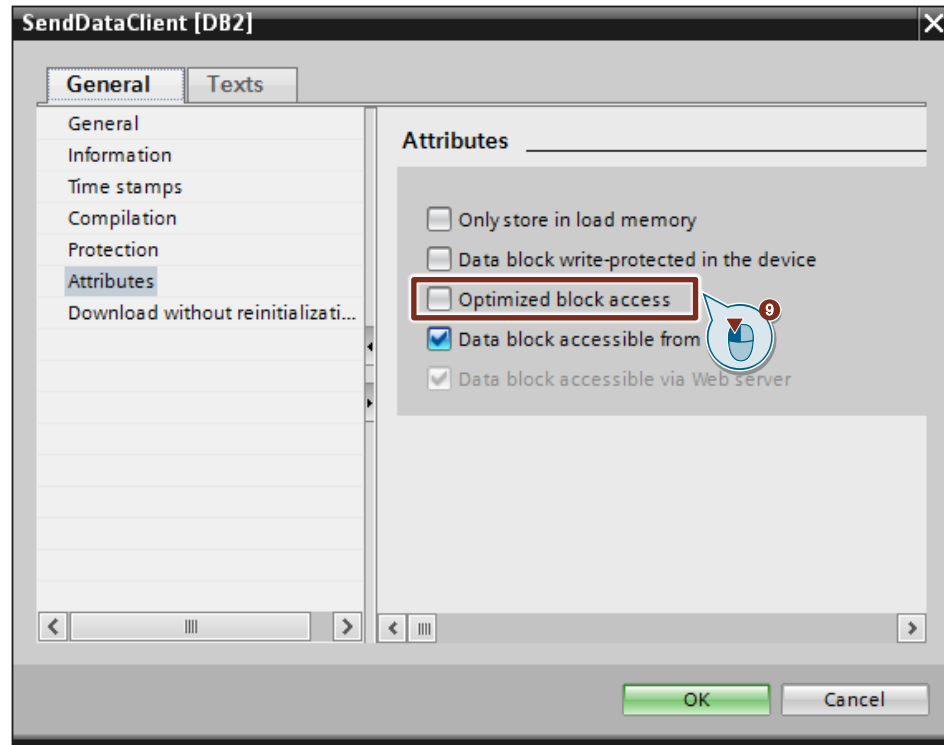
7. Alternatively, you can open the "SendDataClient" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "SendDataClient" DB.



- In the DB "SendDataClient" you define the variable "data" of the data type "Array[0..199] of Byte".

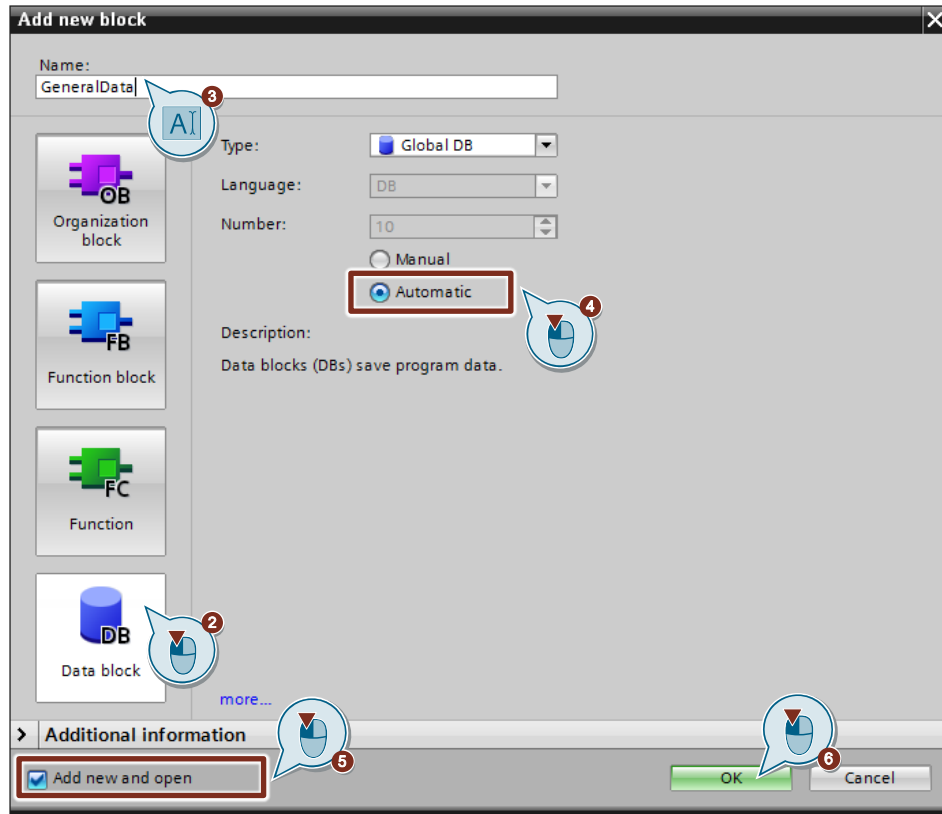
SendDataClient			
Name	Data type	Offset	Start value
Static			
data	Array[0..199] of Byte	0.0	
<Add new>			

- In the Properties of DB1 "SendDataClient", under "Attributes" you disable the "Optimized block access" function.

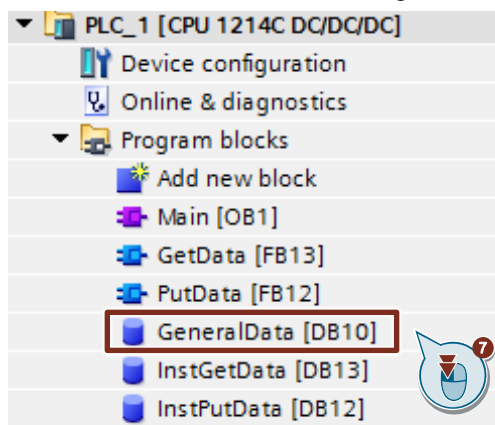


2.2.3 Create Variables for Parameters of Function Blocks "PutData" and "GetData"

- Add another DB to create variables for the parameters of function blocks "PutData" and "GetData".
- In the "Add new block" dialog you click the "Data block (DB)" button.
- Enter the name of the DB, "GeneralData", for example.
- Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
- Enable the "Add new and open" function.
- Click the "OK" button to apply the settings and close the dialog.
 - In the project tree the "GeneralData" DB is added to the "Program blocks" folder of the S7-1200 CPU.
 - If the "Add new and open" function is enabled, the "GeneralData" DB opens in the working area of STEP 7 V16.



- Alternatively, you can open the "GeneralData" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "GeneralData" DB.



- Create the following variables to assign the input and output parameters of the FBs "PutData" and "GetData". PLC data types are used as data type.

Variable	PLC data type	Description
put	typePut	Variables for parameterizing FB12 "PutData".
get	typeGet	Variables for parameterizing FB13 "GetData".
diagnostic	typeDiagnostic	Variables to store the status of FBs "PutData" and "GetData" in case of error.

PLC Data Type "typePut"

The following table shows the structure of "typePut" data type.

Table 2-1

Parameter	Data type	Start value	Description
execute	Boolean	false	Control parameter for FB12 "PutData".
connectionId	Word	16#0	Addressing parameter for specifying the connection to the communication partner.
done	Boolean	false	Status parameters
busy	Boolean	false	Status parameters
error	Boolean	false	Status parameters
status	Word	16#0	Status parameters

PLC Data Type "typeGet"

The following table shows the structure of the PLC data type "typeGet".

Table 2-2

Parameter	Data type	Start value	Description
execute	Boolean	false	Control parameter for FB13 "GetData".
connectionId	Word	16#0	Addressing parameter for specifying the connection to the communication partner.
done	Boolean	false	Status parameters
busy	Boolean	false	Status parameters
error	Boolean	false	Status parameters
status	Word	16#0	Status parameters

PLC Data Type "typeDiagnostic"

The following table shows the structure of the "typeDiagnostic" data type.

Table 2-3

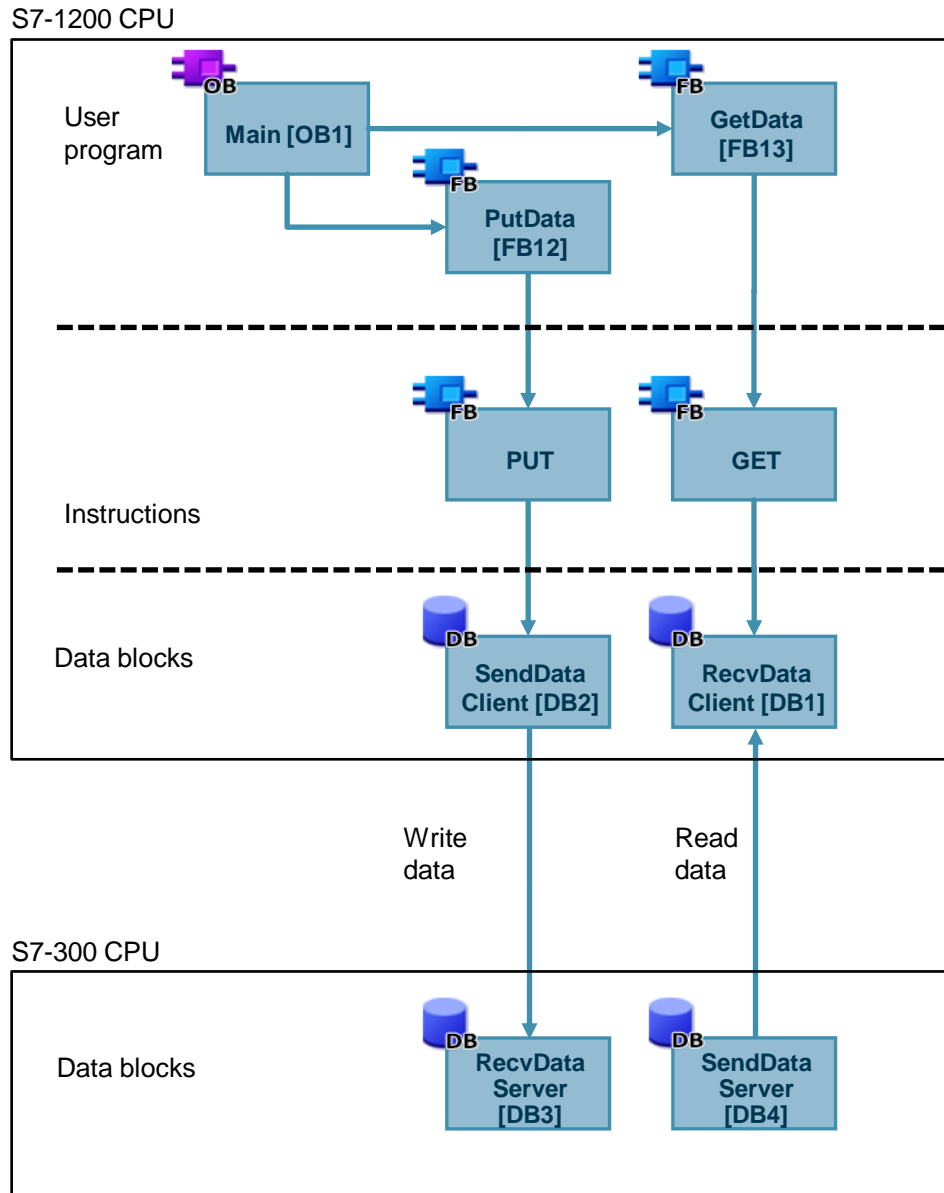
Parameter	Data type	Start value	Description
statusPut	Word	16#0	Parameter to store the status of FB12 "PutData".
statusGet	Word	16#0	Parameter to store the status of FB13 "GetData".

2.2.4 Create a User Program

Overview

The following figure shows an overview of the user program in the S7-1200 CPU and S7-300 CPU.

Figure 2-1



FB12 "PutData"

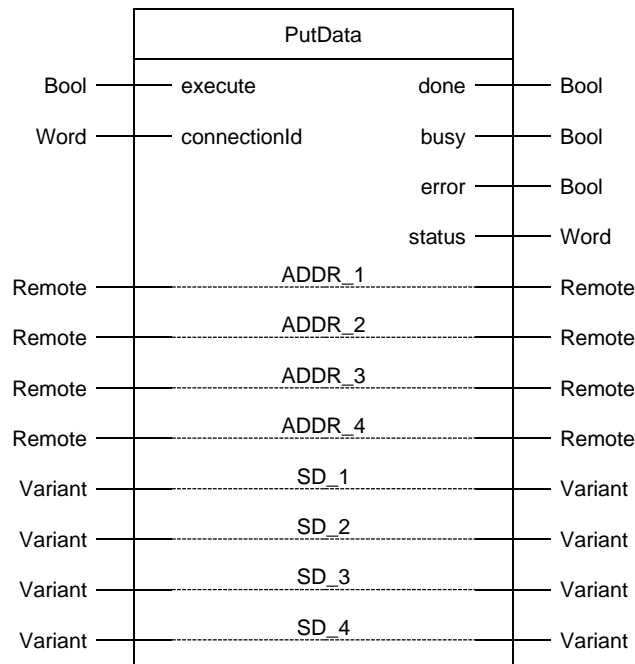
The FB12 "PutData" calls the "PUT" instruction to execute the following functions:

- Send data to the partner via the configured connection as soon as the input "execute" recognizes a positive edge. When the Write job is running, it is not possible to trigger a new Write job.
- Output the status of the FB and data transmission at the "status" output.

FB12 "PutData" is called cyclically in OB1.

The following figure shows the call of the FB12 "PutData" in OB1.

Figure 2-2



The following table shows the parameters of FB12 "PutData".

Table 2-4

Name	P type	Data type	Comment
execute	IN	Bool	Control parameter: Enables the Write job on a rising edge.
connectionId	IN	Word	Local ID: Addressing parameter for specifying the connection to the partner CPU. Note The local ID is available in the Properties of the configured S7 connection. The local ID 100 (hex) is used in this example.
done	OUT	Bool	TRUE: The Write job was executed error-free.
busy	OUT	Bool	TRUE: The Write job is running.
error	OUT	Bool	Status parameters "error" and "status":
status	OUT	Word	<ul style="list-style-type: none"> • "error" = 0: <ul style="list-style-type: none"> - "status" = 0000 (hex): neither warning nor error - "status" <> 0000 (hex): Warning The "status" parameter provides detailed information. • "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.
ADDR_1	IN_OUT	Remote	Pointer to the area to be written to in the partner CPU.
ADDR_2	IN_OUT	Remote	Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> • "ADDR_2" • "ADDR_3" • "ADDR_4" The following minimum user data size for the "PUT" instruction with 1 – 4 variables is guaranteed: 160 bytes
ADDR_3	IN_OUT	Remote	
ADDR_4	IN_OUT	Remote	
SD_1	IN_OUT	Variant	
SD_2	IN_OUT	Variant	Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> • "SD_2" • "SD_3" • "SD_4" The following minimum user data size for the "PUT" instruction with 1 – 4 variables is guaranteed: 160 bytes
SD_3	IN_OUT	Variant	
SD_4	IN_OUT	Variant	

Assign the variables created in section [2.2.3](#) to the parameters of FB12 "PutData".
 The following table shows the assignment of the parameters of the "put" variable to the parameters of FB12 "PutData".

Table 2-5

Parameters of FB12 "PutData"	Parameters of the "put" variable	Start value	Note
execute	execute	false	Set the "execute" parameter to the value "TRUE" to start the Write job. The Write job is started on a rising edge. Reset the "execute" parameter to the value "FALSE" if the Write job is completed with "done" = true or "error" = true.
connectionId	connectionId	16#100	Enter the local ID of the connection that you defined during the connection configuration.
done	done	false	-
busy	busy	false	-
error	error	false	-
status	status	16#0	-
ADDR_1	-	P#DB1.DBX0.0 BYTE 100.	Only absolute addressing is permitted.
SD_1	-	P#DB2.DBX0.0 BYTE 100	-

FB13 "GetData"

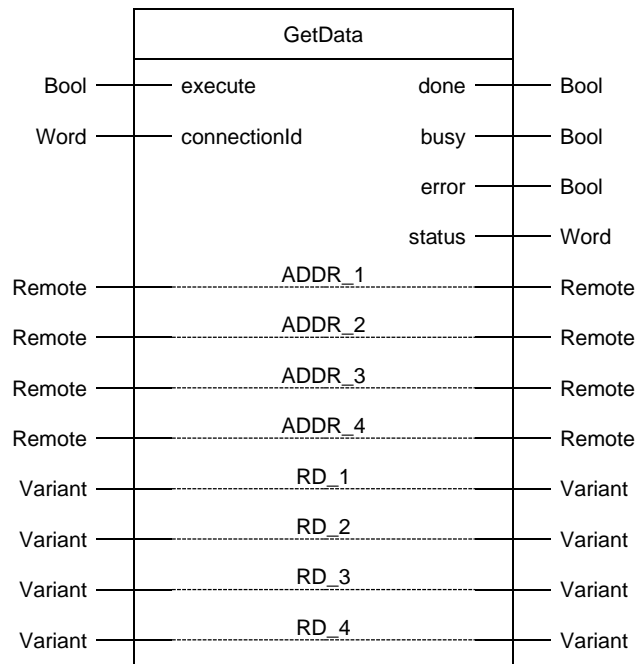
The FB13 "GetData" calls the "GET" instruction to execute the following functions:

- Receive data from the partner via the configured connection as soon as the input "execute" recognizes a positive edge. When the Read job is running, it is not possible to trigger a new Read job.
- Output the status of the FB and data transmission at the "status" output.

FB13 "GetData" is called cyclically in OB1.

The following figure shows the call of the FB13 "GetData" in OB1.

Figure 2-3



The following table shows the parameters of FB13 "GetData".

Table 2-6

Name	P type	Data type	Comment
execute	IN	Bool	Control parameter: Enables the Read job on a rising edge.
connectionId	IN	Word	Local ID: Addressing parameter for specifying the connection to the partner CPU. Note The local ID is available in the Properties of the configured S7 connection. The local ID 100 (hex) is used in this example.
done	OUT	Bool	TRUE: The Read job was executed error-free.
busy	OUT	Bool	TRUE: The read job is running.
error	OUT	Bool	Status parameters "error" and "status": <ul style="list-style-type: none"> "error" = 0: <ul style="list-style-type: none"> "status" = 0000 (hex): neither warning nor error "status" <> 0000 (hex): Warning The "status" parameter provides detailed information. "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.
status	OUT	Word	
ADDR_1	IN_OUT	Remote	Pointer to the area to be read in the partner CPU. Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> "ADDR_2" "ADDR_3" "ADDR_4" The following minimum user data size for the "GET" instruction with 1 – 4 variables is guaranteed: 160 bytes
ADDR_2	IN_OUT	Remote	
ADDR_3	IN_OUT	Remote	
ADDR_4	IN_OUT	Remote	
RD_1	IN_OUT	Variant	Pointer to the area in your own S7 CPU in which the read data is stored. Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> "RD_2" "RD_3" "RD_4" The following minimum user data size for the "GET" instruction with 1 – 4 variables is guaranteed: 160 bytes
RD_2	IN_OUT	Variant	
RD_3	IN_OUT	Variant	
RD_4	IN_OUT	Variant	

Assign the variables created in section [2.2.3](#) to the parameters of FB13 "GetData". The following table shows the assignment of the parameters of the "get" variable to the parameters of FB13 "GetData".

Table 2-7

Parameters of FB13 "GetData"	Parameters of the "get" variable	Start value	Note
execute	execute	false	Set the "execute" parameter to the value "TRUE" to start the Read job. The Read job is started on a rising edge. Reset the "execute" parameter to the value "FALSE" if the Read job is completed with "done" = true or "error" = true.
connectionId	connectionId	16#100	Enter the local ID of the connection that you defined during the connection configuration.
done	done	false	-
busy	busy	false	-
error	error	false	-
status	status	16#0	-
ADDR_1	-	P#DB4.DBX0.0 BYTE 100	Only absolute addressing is permitted.
RD_1	-	P#DB1.DBX0.0 BYTE 100	-

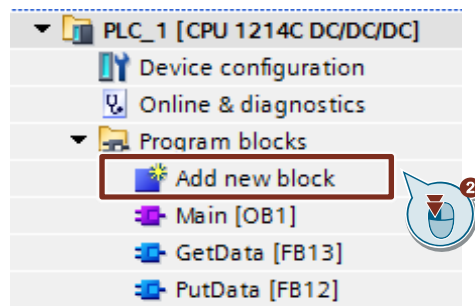
2.3 Install SIMATIC S7-1200 as Server

2.3.1 Create Send and Receive Data Areas

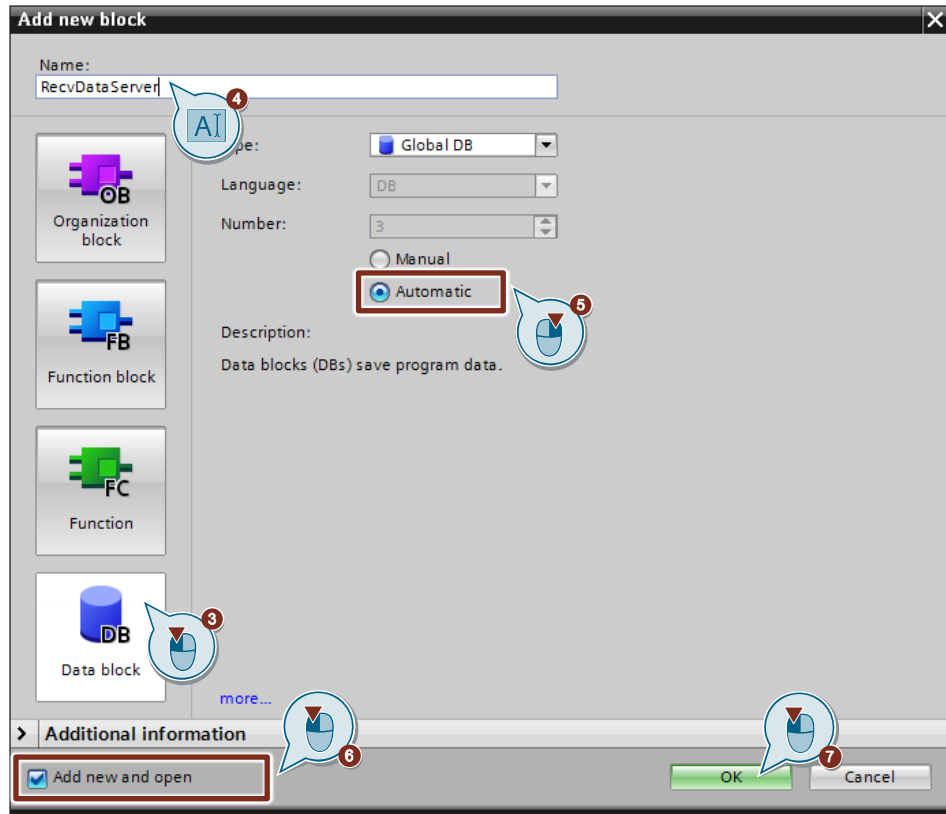
In the user program of the S7-1200 CPU you add the data blocks for saving the Send and Receive data.

Add data block for saving the Receive data

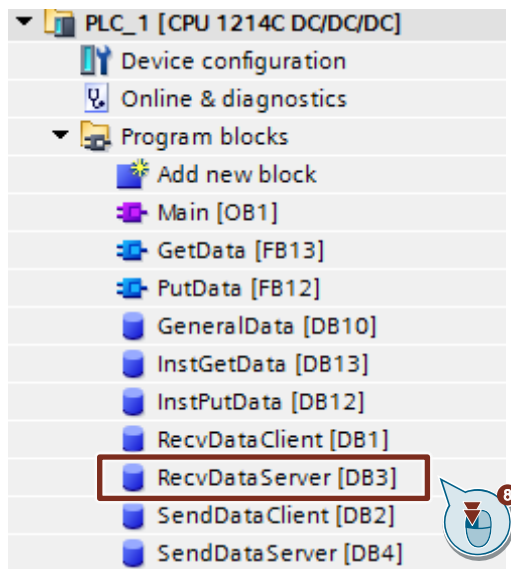
1. In the project tree you navigate to the device folder of the S7-1200 CPU. The device folder contains structured objects and actions that belong to the device.
2. In the device folder you navigate to the "Program blocks" subfolder and double-click the "Add new block" action. The "Add new block" dialog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



3. Click the "Data block (DB)" button.
4. Enter the name of the data block (DB), "RecvDataServer", 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 "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 V16.



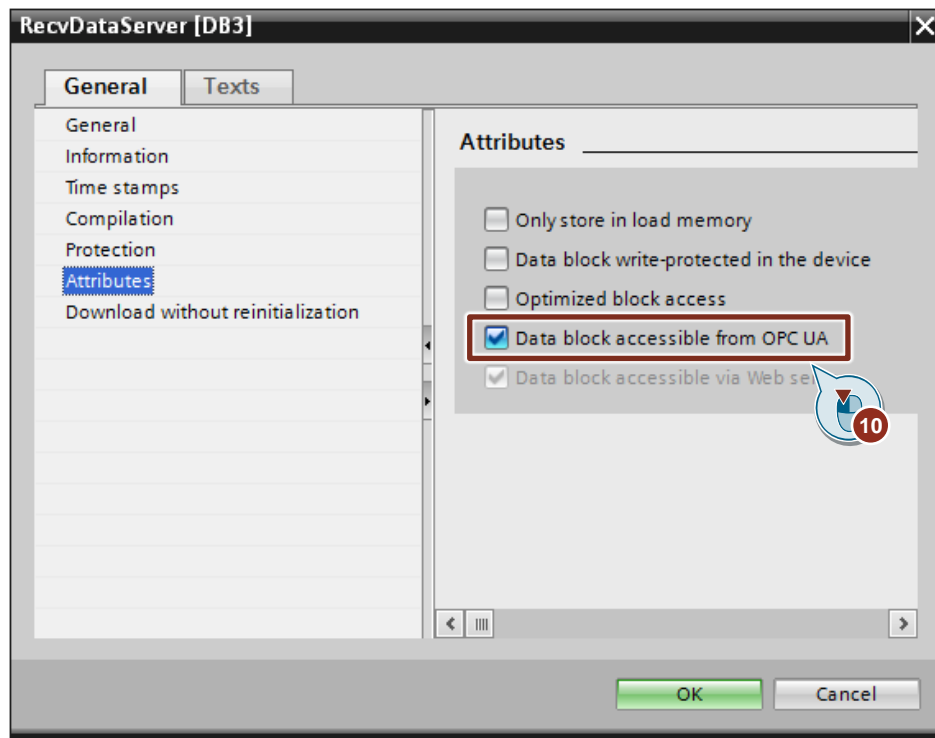
8. Alternatively, you can open the "RecvDataServer" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "RecvDataServer" DB.



- In the DB "RecvDataServer" you define the variable "data" of the data type "Array[0..199] of Byte".

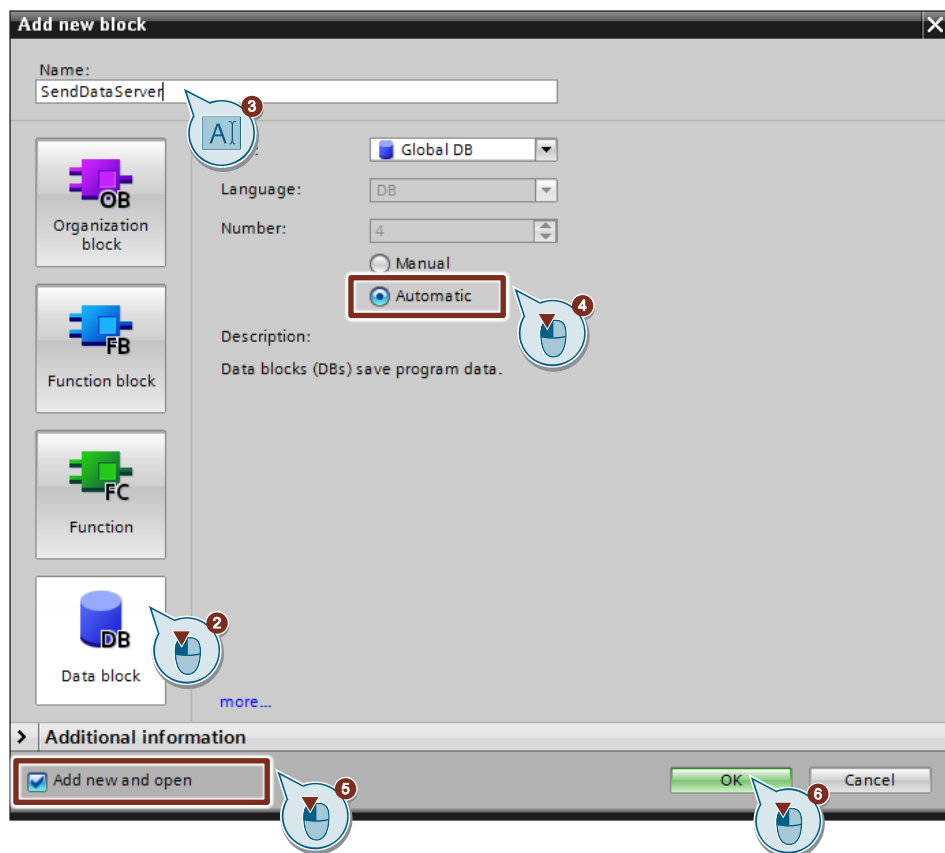
RecvDataServer				
Name	Data type	Offset	Start value	
Static				
data	Array[0..199] of Byte	0.0		
<Add new>				

- 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.

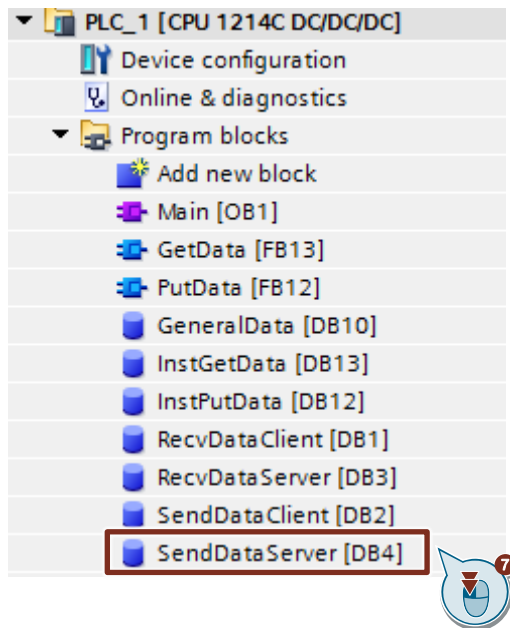


Add data block for saving the Send data

1. Add another DB for saving the Send data.
2. In the "Add new block" dialog you click the "Data block (DB)" button.
3. Enter the name of the DB, "SendDataServer", 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 "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 V16.



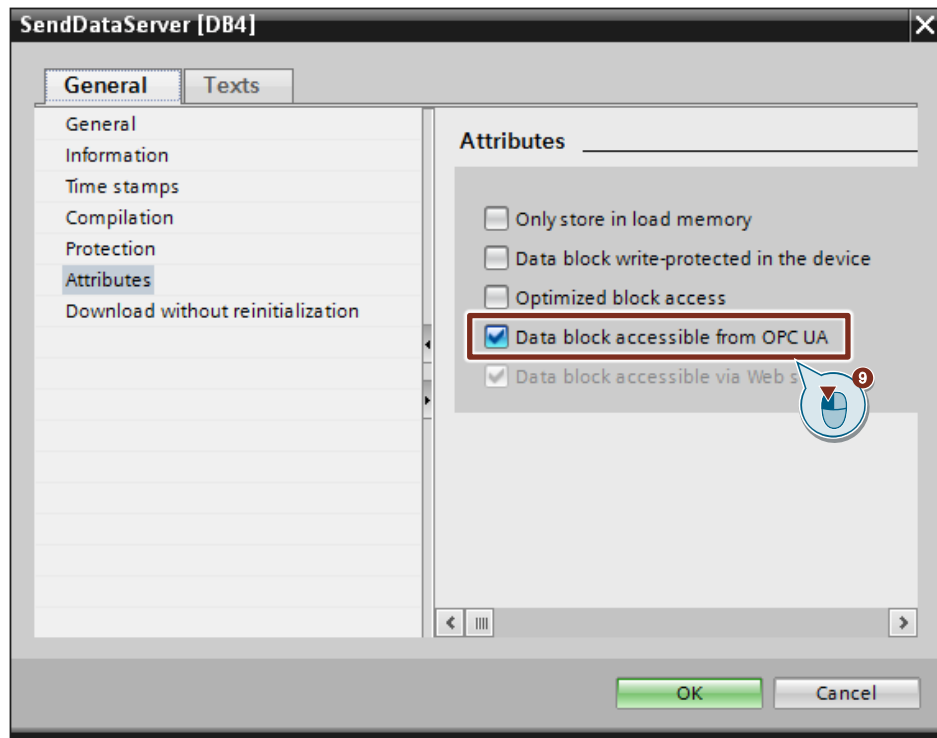
- Alternatively, you can open the "SendDataServer" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-1200 CPU and double-clicking the "SendDataServer" DB.



- In the DB "SendDataServer" you define the variable "data" of the data type "Array[0..199] of Byte".

SendDataServer				
	Name	Data type	Offset	Start value
	Static			
	data	Array[0..199] of Byte	0.0	
	<Add new>			

- 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.



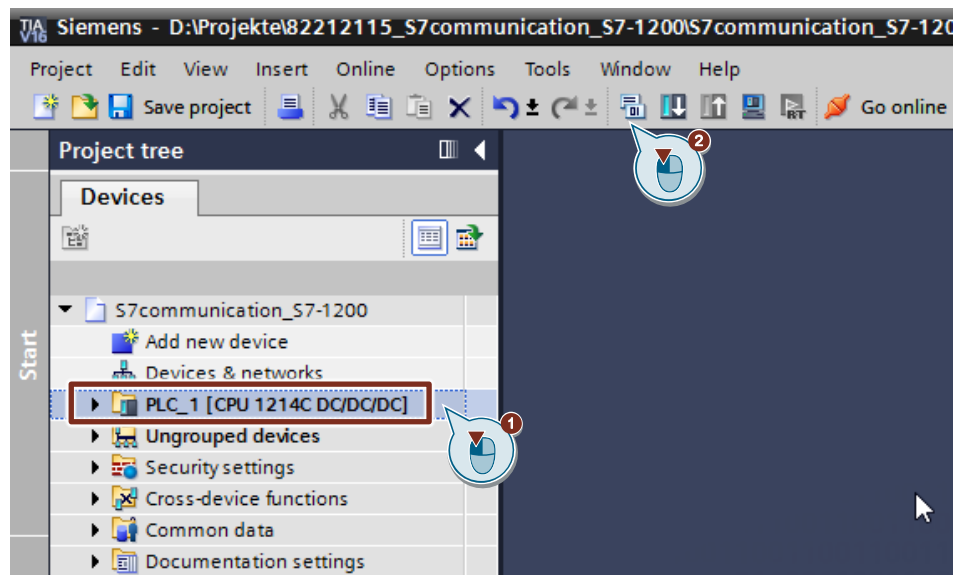
2.4 Download the Hardware Configuration and User Program

Requirements

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

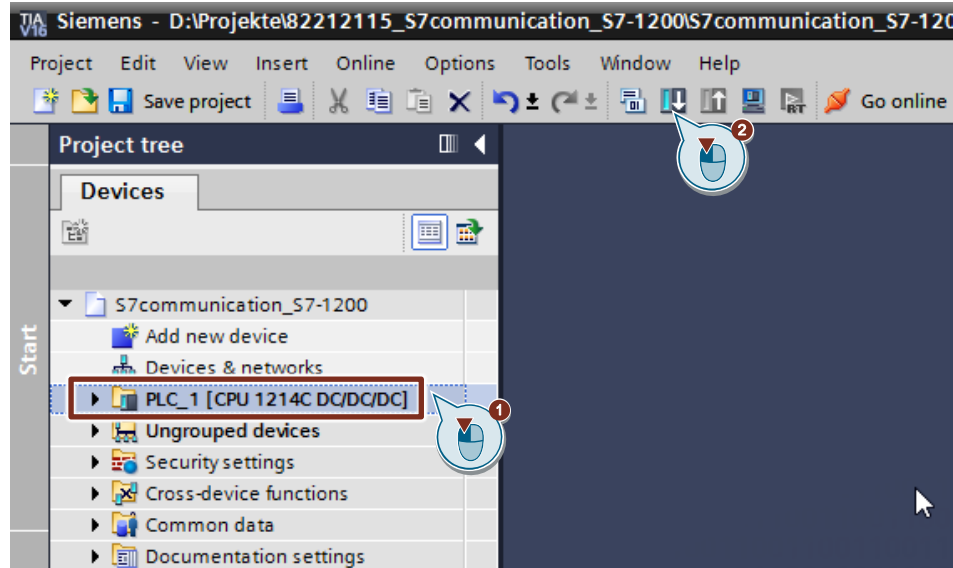
Compile

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



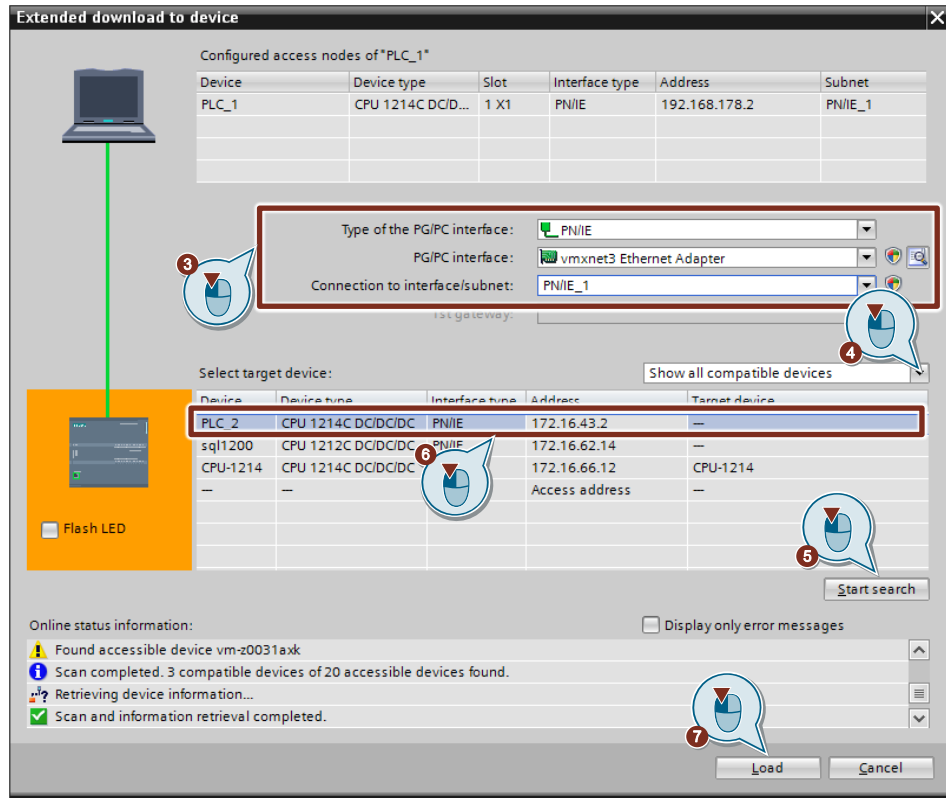
Download

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.

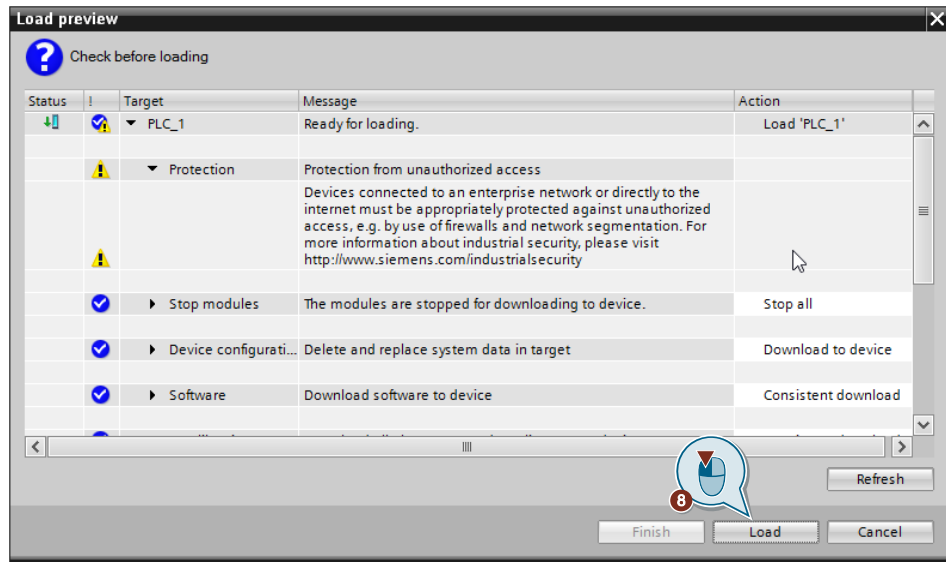


3. The "Extended download to device" dialog opens automatically only if the access path from the PG/PC to the S7-1200 CPU has to be set. Make the following settings:
 - Type of the PG/PC interface: PN/IE
 - PG/PC interface: Network card of the PG/PC
 - Connection to interface/subnet: Subnet to which 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.

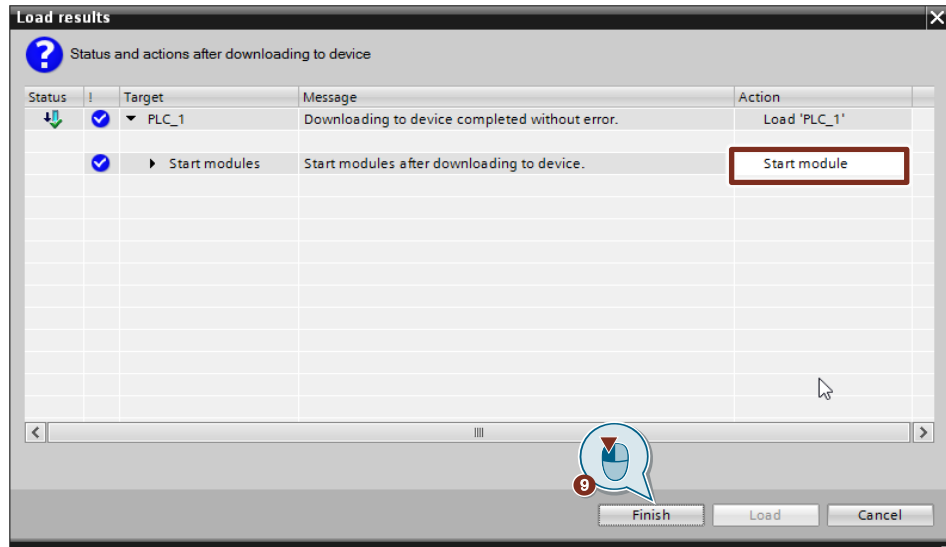
2 Configuration of SIMATIC S7-1200



- 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.



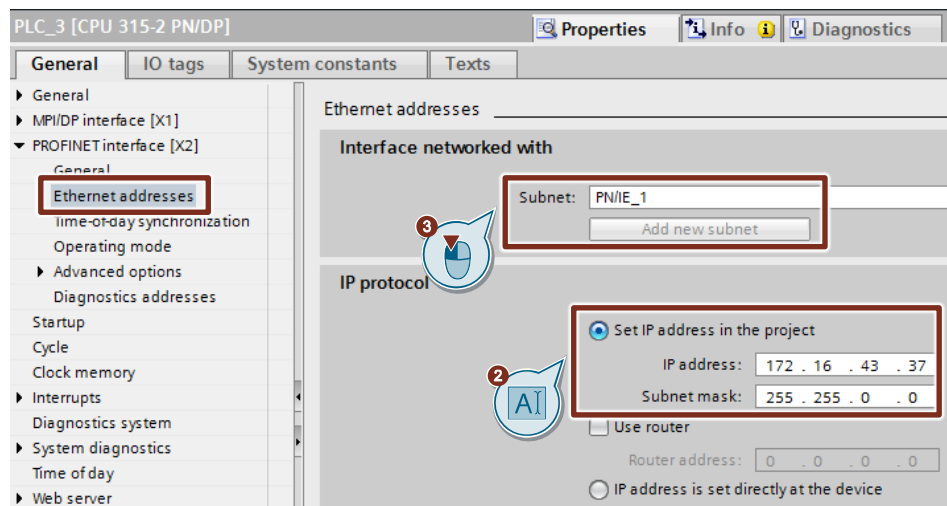
3 Configuration of SIMATIC S7-300 in STEP 7 V16

1. Create a new project or open an existing project in STEP 7 V16 (TIA Portal V16).
2. Add the S7-300 CPU and create the hardware configuration.
3. Install the S7-300 CPU as client or server.
 - Client: Configure a new S7 connection and create the user program. In the user program you call the "PUT" instruction to write data to the S7-1200 CPU. In the user program you call the "GET" instruction to read data from the S7-1200 CPU. Add the data blocks for saving the Send and Receive data.
 - Server: Create the user program. In the user program you add data blocks in which the sent and received data is stored.

3.1 Create the Hardware Configuration

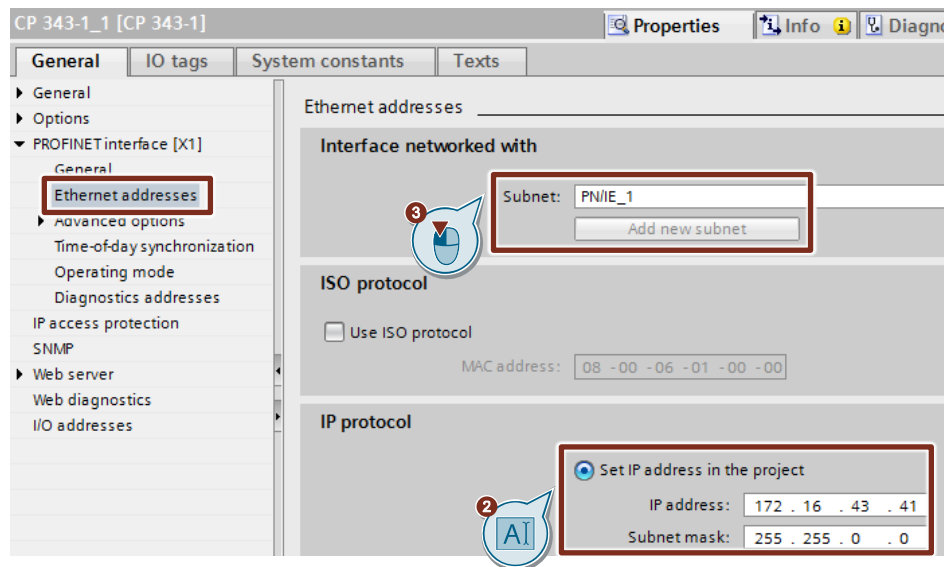
3.1.1 Set the IP address of the CPU

1. In the Device view or Network view of the "Devices & networks" editor you mark the S7-300 CPU. The properties of the S7-300 CPU are displayed in the inspector window.
2. In the "General" tab under "PROFINET interface [X2] > Ethernet addresses" you enter the IP address and the subnet mask of the S7-300 CPU.
 - IP address: 172.16.43.37
 - Subnet mask: 255.255.0.0
3. Assign a subnet to the Ethernet interface of the S7-300 CPU. You have two options:
 - Click the "Add new subnet" button to add a new subnet.
 - Select an existing subnet.



3.1.2 Set the IP address of the CP

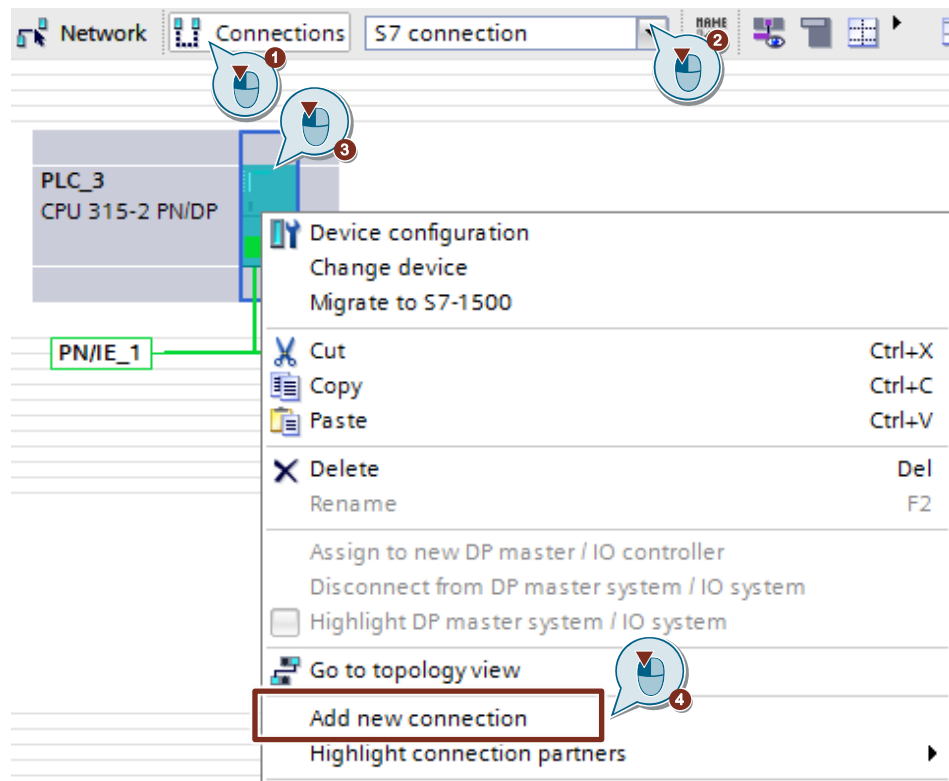
1. In the Device view or Network view of the "Devices & networks" editor you mark the CP in the S7-300.
The properties of the S7-300 CPU are displayed in the inspector window.
2. In the "General" tab under "PROFINET interface [X1] > Ethernet addresses" you enter the IP address and the subnet mask of the CP.
 - IP address: 172.16.43.41
 - Subnet mask: 255.255.0.0
3. Assign a subnet to the Ethernet interface of the CP. You have two options:
 - Click the "Add new subnet" button to add a new subnet.
 - Select an existing subnet.



3.2 Install SIMATIC S7-300 as Client

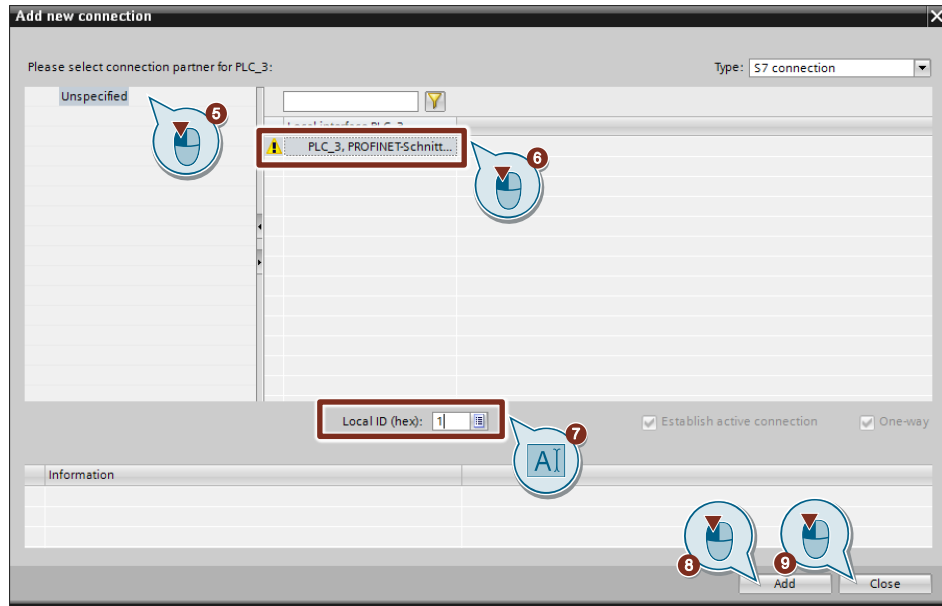
3.2.1 Configuration of the S7 Connection

1. In the Network view you click the "Connections" icon to enable the Connection mode.
2. Select "S7 connection" as the connection type in the adjacent drop-down list box.
In the Network view, all devices that are available for an S7 connection are highlighted in color.
3. Right-click the S7-1200 CPU. The pop-up menu opens.
4. Select the "Add new connection" item.
The "Add new connection" dialog opens.



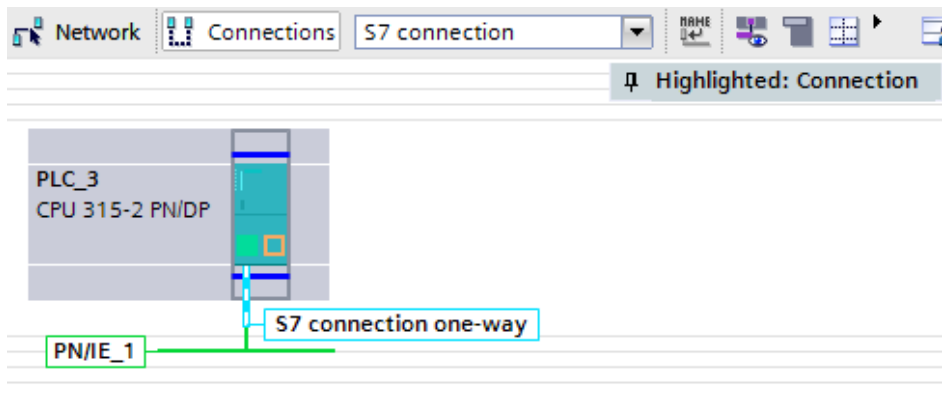
5. Specify the following connection partner: "Unspecified".
6. Select the local interface. In this example, the integrated interface of the S7-300 CPU is used. If the S7 connection is to be established via a CP, select the CP.
7. Enter the local ID for the S7 connection, 1 (hex), for example.
8. Click the "Add" button to add the unspecified S7 connection.
9. Click the "Close" button to end the dialog.

3 Configuration of SIMATIC S7-300 in STEP 7 V16



Result

- An unspecified S7 connection is created.
- The connection path is displayed highlighted in the graphical area of the Network view.

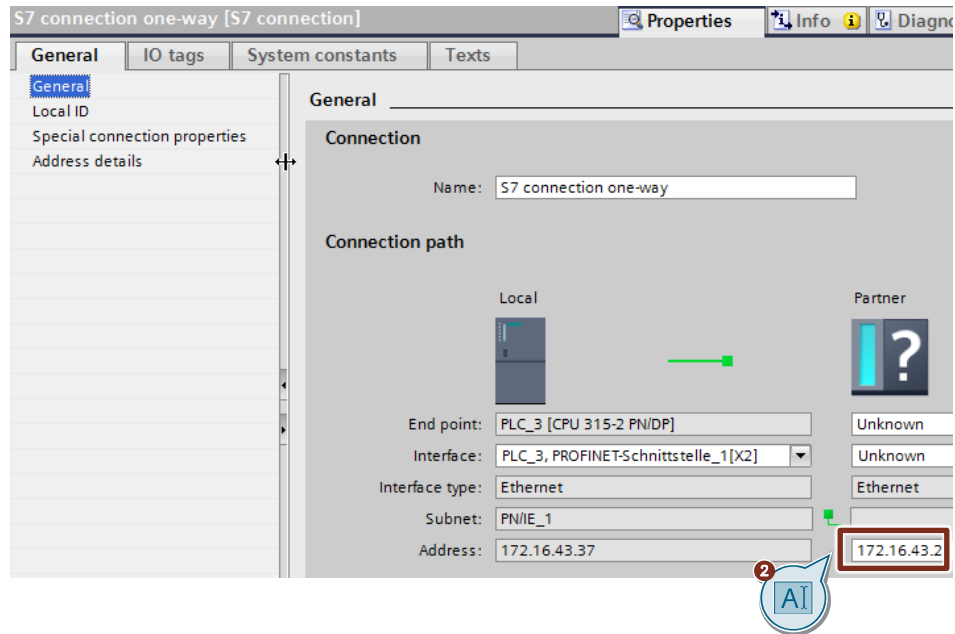


- The S7 connection is entered in the "Connections" table in the table area of the Network view.

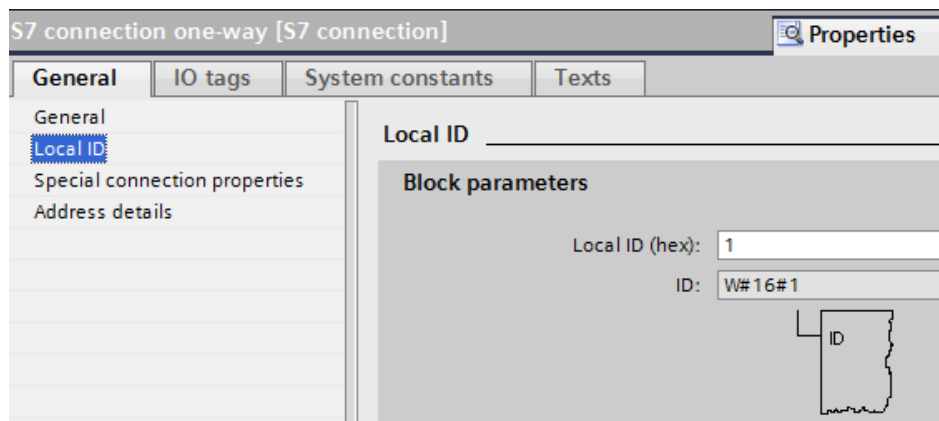
Network overview	Connections	I/O communication	VPN	TeleControl	
	Local connection name	Local end point	Local ID (hex)	Partner ID (hex)	Partner
	S7 connection one-way	PLC_3 [CPU 315-2 PN/DP]	1		Unknown
					S7 connection

Setting the connection parameters

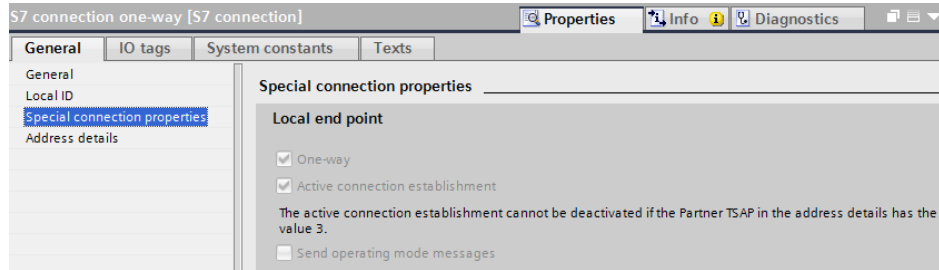
1. In the "Connections" table, mark the unspecified S7 connections.
The Properties of the S7 connection are displayed in the inspector window.
2. Under "General" the "General" tab displays the general connection parameters that identify the local endpoint. Under Partner you enter the IP address of the S7-1200 CPU, for example: 172.16.43.2. If you use a CP as partner instead of the integrated interface of the S7-1200 CPU, enter the IP address of the CP.



3. In the "General" tab under "Local ID" you see the local ID of the module from which the connection is viewed (local partner). You can change the local ID. You specify the local ID later at the "ID" input parameter of the "PUT" or "GET" instruction.
In this example we use the local ID with the value 1 (hex).

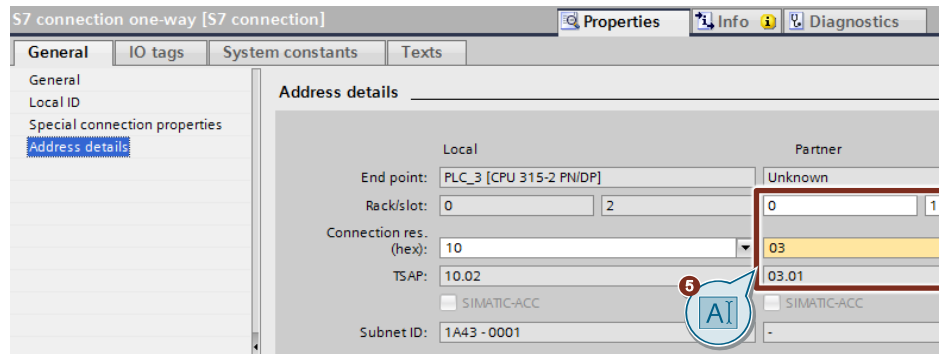


- In the "General" under "Special connection properties" you see that the S7 connection is configured unilaterally. Unilaterally means that the connection partner is server for this connection and cannot actively send or receive. The S7 CPU that is used as the local endpoint for the S7 connection, actively establishes the S7 connection as a client and actively sends and receives.



- In the "General" tab under "Address details" you enter the partner TSAP. Since the S7 connection is configured unilaterally to an unspecified partner, the partner TSAP is composed as follows:
03 <slot of CPU or CP>, for example:

- S7-1200 CPU in slot 1: 03.01

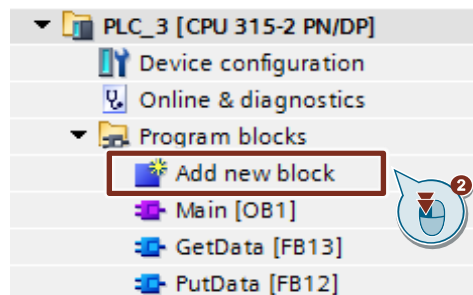


3.2.2 Create Send and Receive Data Areas

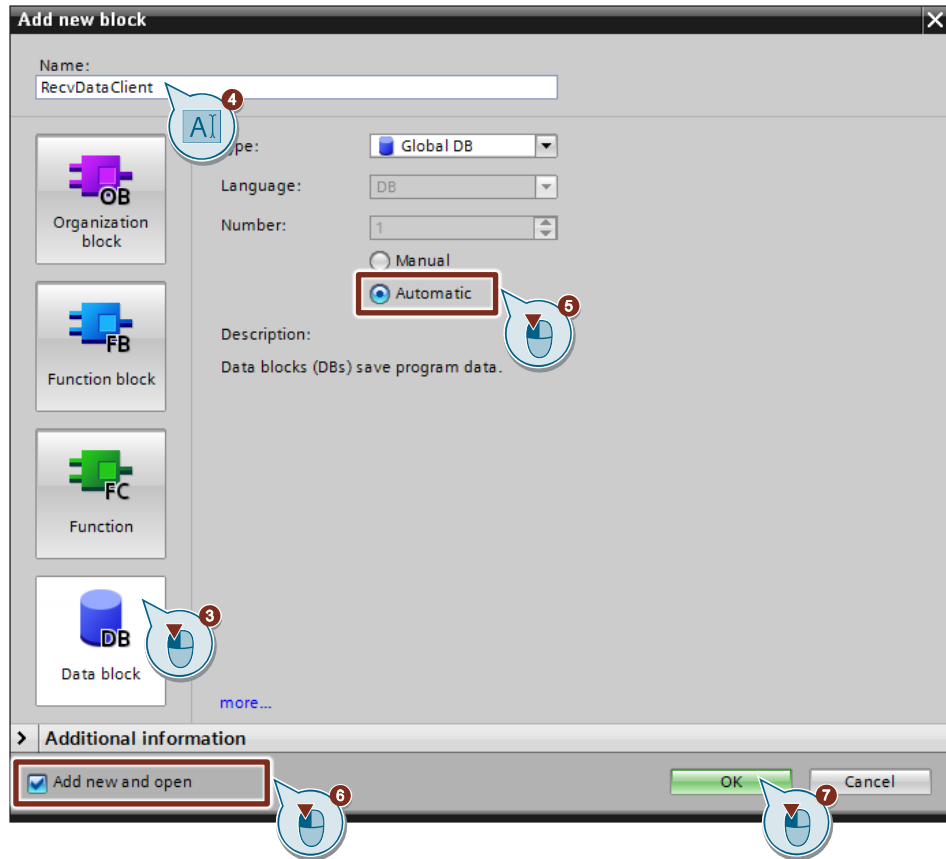
Add data block for saving the Receive data

The data received from the S7-1200 CPU is stored in data block DB1 "RecvDataClient" of the S7-300 CPU.

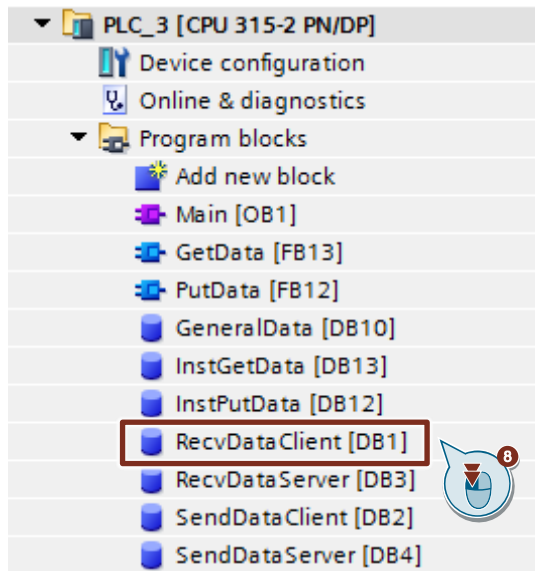
1. In the project tree you navigate to the device folder of the S7-300 CPU. The device folder contains structured objects and actions that belong to the device.
2. In the device folder you navigate to the "Program blocks" subfolder and double-click the "Add new block" action. The "Add new block" dialog opens.



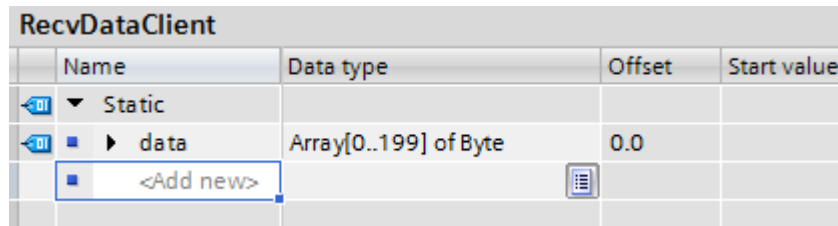
3. Click the "Data block (DB)" button.
4. Enter the name of the data block (DB), "RecvDataClient", for example.
5. Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
6. Enable the "Add new and open" function.
7. Click the "OK" button to apply the settings and close the dialog.
 - In the project tree the "RecvDataClient" DB is added to the "Program blocks" folder of the S7-300 CPU.
 - If the "Add new and open" function is enabled, the "RecvDataClient" DB opens in the working area of STEP 7 V16.



8. Alternatively, you can open the "RecvDataClient" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-300 CPU and double-clicking the "RecvDataClient" DB.



9. In the DB "RecvDataClient" you define the variable "data" of the data type Array[0..199] of Byte.



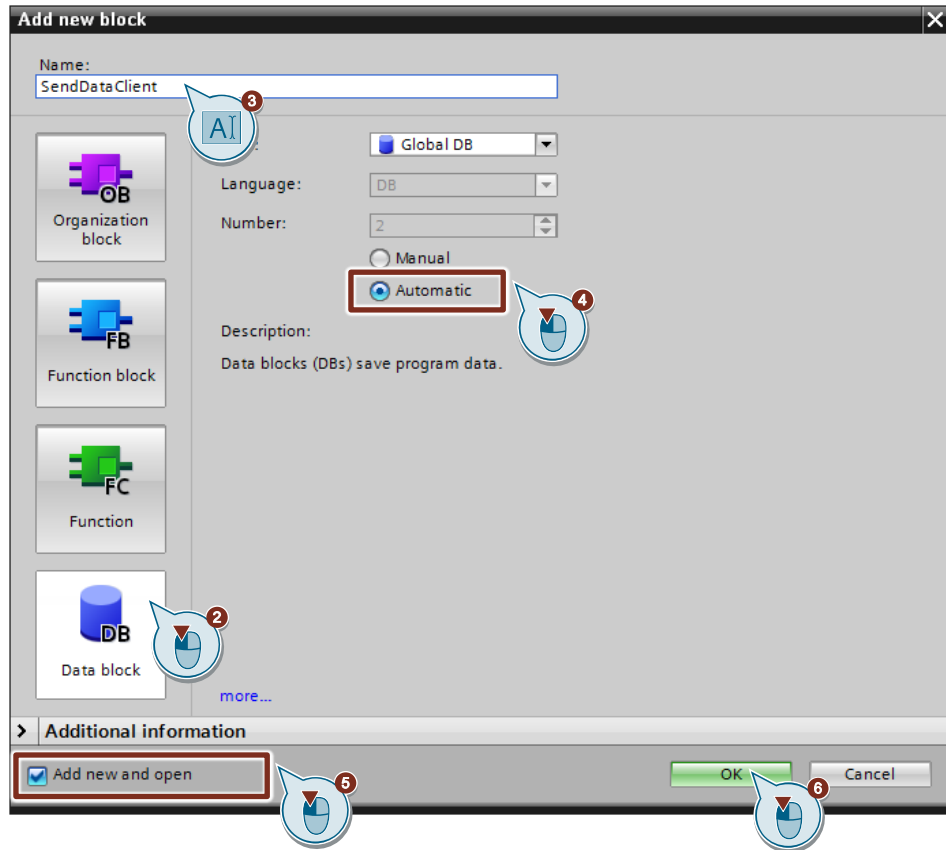
The screenshot shows the 'RecvDataClient' data block configuration in SIMATIC Manager. It features a table with four columns: Name, Data type, Offset, and Start value. Under the 'Static' section, a variable named 'data' is defined with the data type 'Array[0..199] of Byte', an offset of '0.0', and a start value of '0.0'. Below the table, there is an '<Add new>' button.

Name	Data type	Offset	Start value
Static			
data	Array[0..199] of Byte	0.0	
<Add new>			

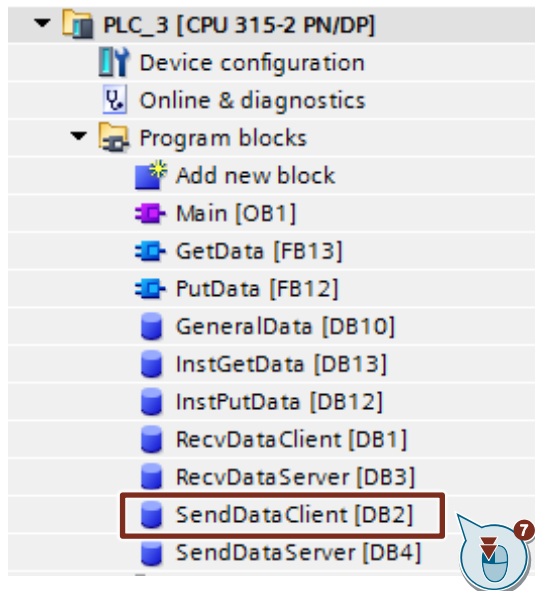
Add data block for saving the Send data

The data that is transferred to the S7-1200 CPU is stored in data block DB2 "SendDataClient" of the S7-300 CPU.

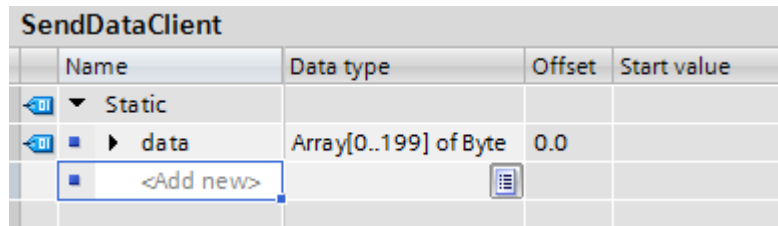
1. Add another DB for saving the Send data.
2. In the "Add new block" dialog you click the "Data block (DB)" button.
3. Enter the name of the DB, "SendDataClient", for example.
4. Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
5. Enable the "Add new and open" function.
6. Click the "OK" button to apply the settings and close the dialog.
 - In the project tree the "SendDataClient" DB is added to the "Program blocks" folder of the S7-1200 CPU.
 - If the "Add new and open" function is enabled, the "SendDataClient" DB opens in the working area of STEP 7 V16.



- Alternatively, you can open the "SendDataClient" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-300 CPU and double-clicking the "SendDataClient" DB.



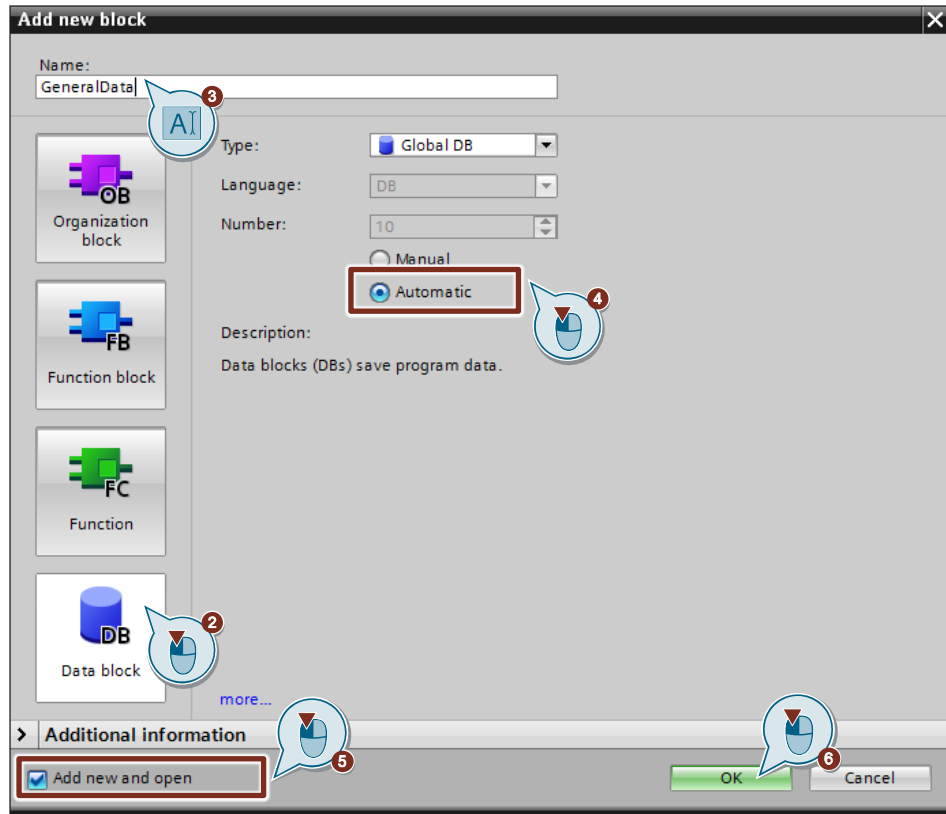
8. In the DB "SendDataClient" you define the variable "data" of the data type "Array[0..199] of Byte".



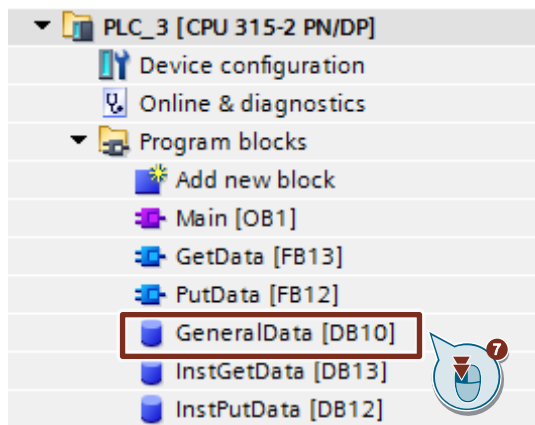
SendDataClient			
Name	Data type	Offset	Start value
Static			
data	Array[0..199] of Byte	0.0	
<Add new>			

3.2.3 Create Variables for Parameters of Function Blocks "PutData" and "GetData"

1. Add another DB to create variables for the parameters of function blocks "PutData" and "GetData".
2. In the "Add new block" dialog you click the "Data block (DB)" button.
3. Enter the name of the DB, "GeneralData", 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 "GeneralData" DB is added to the "Program blocks" folder of the S7-300 CPU.
 - If the "Add new and open" function is enabled, the "GeneralData" DB opens in the working area of STEP 7 V16.



- Alternatively, you can open the "GeneralData" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-300 CPU and double-clicking the "GeneralData" DB.



- Create the following variables to assign the input and output parameters of the FBs "PutData" and "GetData". PLC data types are used as data type.

Variable	PLC data type	Description
put	typePut	Variables for parameterizing FB12 "PutData".
get	typeGet	Variables for parameterizing FB13 "GetData".
diagnostic	typeDiagnostic	Variables to store the status of FBs "PutData" and "GetData" in case of error.

PLC Data Type "typePut"

The following table shows the structure of "typePut" data type.

Table 3-1

Parameter	Data type	Start value	Description
execute	Bool	false	Control parameter for FB12 "PutData".
connectionId	Word	16#0	Addressing parameter for specifying the connection to the communication partner.
done	Bool	false	Status parameters
busy	Bool	false	Status parameters
error	Bool	false	Status parameters
status	Word	16#0	Status parameters

PLC Data Type "typeGet"

The following table shows the structure of the PLC data type "typeGet".

Table 3-2

Parameter	Data type	Start value	Description
execute	Bool	false	Control parameter for FB13 "GetData".
connectionId	Word	16#0	Addressing parameter for specifying the connection to the communication partner.
done	Bool	false	Status parameters
busy	Bool	false	Status parameters
error	Bool	false	Status parameters
status	Word	16#0	Status parameters

PLC Data Type "typeDiagnostic"

The following table shows the structure of the "typeDiagnostic" data type.

Table 3-3

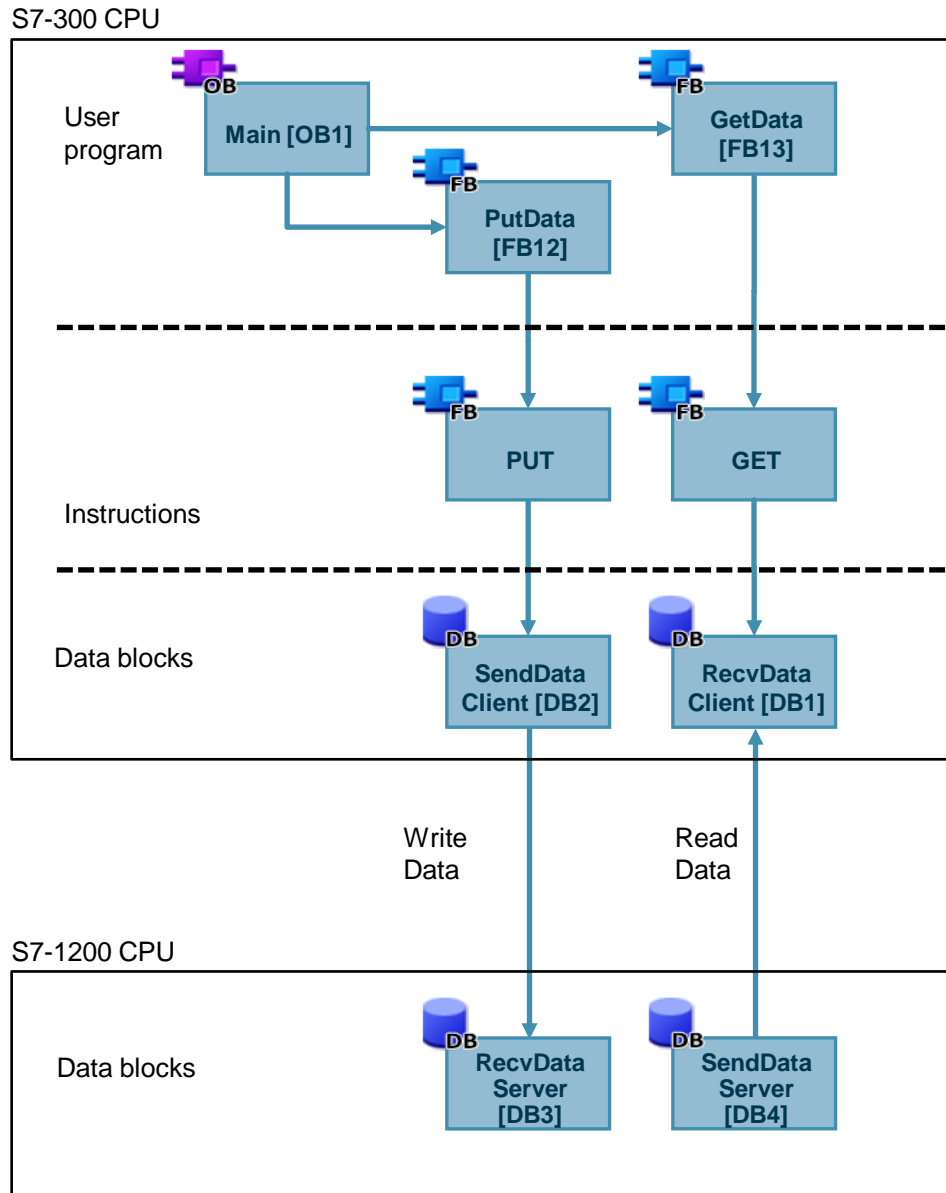
Parameter	Data type	Start value	Description
statusPut	Word	16#0	Parameter to store the status of FB12 "PutData".
statusGet	Word	16#0	Parameter to store the status of FB13 "GetData".

3.2.4 Create a User Program

Overview

The following figure shows an overview of the user program in the S7-300 CPU and S7-1200 CPU.

Figure 3-1



FB12 "PutData"

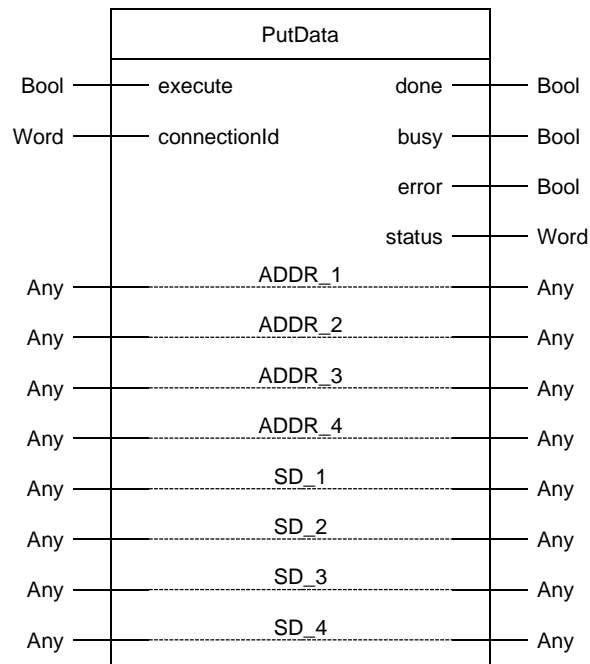
The FB12 "PutData" calls the "PUT" instruction to execute the following functions:

- Send data to the partner via the configured connection as soon as the input "execute" recognizes a positive edge. When the Write job is running, it is not possible to trigger a new Write job.
- Output the status of the FB and data transmission at the "status" output.

FB12 "PutData" is called cyclically in OB1.

The following figure shows the call of the FB12 "PutData" in OB1.

Figure 3-2



The following table shows the parameters of FB12 "PutData".

Table 3-4

Name	P type	Data type	Comment
execute	IN	Bool	Control parameter: Enables the Write job on a rising edge.
connectionId	IN	Word	Local ID: Addressing parameter for specifying the connection to the partner CPU. Note The local ID is available in the Properties of the configured S7 connection. The local ID 1 (hex) is used in this example.
done	OUT	Bool	TRUE: The Write job was executed error-free.
busy	OUT	Bool	TRUE: The Write job is running.
error	OUT	Bool	Status parameters "error" and "status": <ul style="list-style-type: none"> • "error" = 0: <ul style="list-style-type: none"> - "status" = 0000 (hex): neither warning nor error - "status" <> 0000 (hex): Warning The "status" parameter provides detailed information. • "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.
status	OUT	Word	
ADDR_1	IN_OUT	Any	Pointer to the area to be written to in the partner CPU. Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> • "ADDR_2" • "ADDR_3" • "ADDR_4" The following minimum user data size for the "PUT" instruction with 1 – 4 variables is guaranteed: 160 bytes
ADDR_2	IN_OUT	Any	
ADDR_3	IN_OUT	Any	
ADDR_4	IN_OUT	Any	
SD_1	IN_OUT	Any	Pointer to the area in your own S7 CPU that contains the data to be sent. Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> • "SD_2" • "SD_3" • "SD_4" The following minimum user data size for the "PUT" instruction with 1 – 4 variables is guaranteed: 160 bytes
SD_2	IN_OUT	Any	
SD_3	IN_OUT	Any	
SD_4	IN_OUT	Any	

Assign the variables created in section [3.2.3](#) to the parameters of FB12 "PutData".
 The following table shows the assignment of the parameters of the "put" variable to the parameters of FB12 "PutData".

Table 3-5

Parameters of FB12 "PutData"	Parameters of the "put" variable	Start value	Note
execute	execute	false	Set the "execute" parameter to the value "TRUE" to start the Write job. The Write job is started on a rising edge. Reset the "execute" parameter to the value "FALSE" if the Write job is completed with "done" = true or "error" = true.
connectionId	connectionId	16#1	Enter the local ID of the connection that you defined during the connection configuration.
done	done	false	-
busy	busy	false	-
error	error	false	-
status	status	16#0	-
ADDR_1	-	P#DB1.DBX0.0 BYTE 100.	Only absolute addressing is permitted.
SD_1	-	P#DB2.DBX0.0 BYTE 100	-

FB13 "GetData"

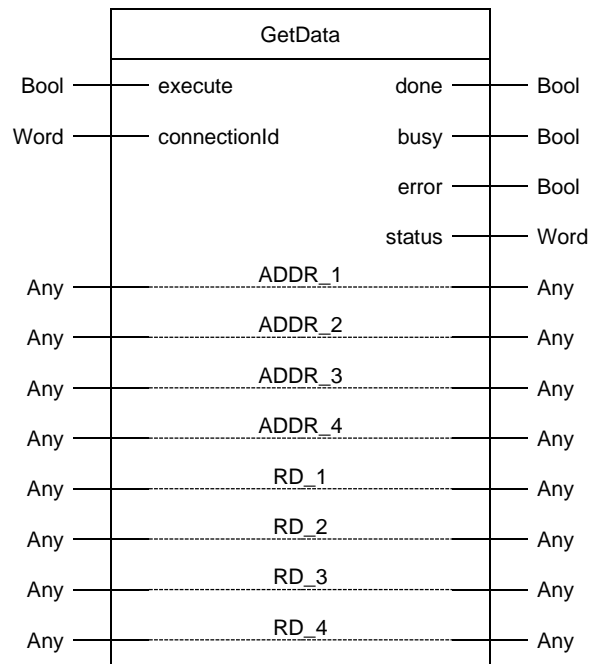
The FB13 "GetData" calls the "GET" instruction to execute the following functions:

- Receive data from the partner via the configured connection as soon as the input "execute" recognizes a positive edge. When the Read job is running, it is not possible to trigger a new Read job.
- Output the status of the FB and data transmission at the "status" output.

FB13 "GetData" is called cyclically in OB1.

The following figure shows the call of the FB13 "GetData" in OB1.

Figure 3-3



The following table shows the parameters of FB13 "GetData".

Table 3-6

Name	P type	Data type	Comment
execute	IN	Bool	Control parameter: Enables the Read job on a rising edge.
connectionId	IN	Word	Local ID: Addressing parameter for specifying the connection to the partner CPU. Note The local ID is available in the Properties of the configured S7 connection. The local ID 1 (hex) is used in this example.
done	OUT	Bool	TRUE: The Read job was executed error-free.
busy	OUT	Bool	TRUE: The read job is running.
error	OUT	Bool	Status parameters "error" and "status": <ul style="list-style-type: none"> "error" = 0: <ul style="list-style-type: none"> "status" = 0000 (hex): neither warning nor error "status" <> 0000 (hex): Warning The "status" parameter provides detailed information. "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.
status	OUT	Word	
ADDR_1	IN_OUT	Any	Pointer to the area to be read in the partner CPU. Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> "ADDR_2" "ADDR_3" "ADDR_4" The following minimum user data size for the "GET" instruction with 1 – 4 variables is guaranteed: 160 bytes
ADDR_2	IN_OUT	Any	
ADDR_3	IN_OUT	Any	
ADDR_4	IN_OUT	Any	
RD_1	IN_OUT	Any	Pointer to the area in your own S7 CPU in which the read data is stored. Note The following parameters are hidden and are preset with the value "NULL": <ul style="list-style-type: none"> "RD_2" "RD_3" "RD_4" The following minimum user data size for the "GET" instruction with 1 – 4 variables is guaranteed: 160 bytes
RD_2	IN_OUT	Any	
RD_3	IN_OUT	Any	
RD_4	IN_OUT	Any	

Assign the variables created in section [3.2.3](#) to the parameters of FB13 "GetData".
 The following table shows the assignment of the parameters of the "get" variable to the parameters of FB13 "GetData".

Table 3-7

Parameters of FB13 "GetData"	Parameters of the "get" variable	Start value	Note
execute	execute	false	Set the "execute" parameter to the value "TRUE" to start the Read job. The Read job is started on a rising edge. Reset the "execute" parameter to the value "FALSE" if the Read job is completed with "done" = true or "error" = true.
connectionId	connectionId	16#100	Enter the local ID of the connection that you defined during the connection configuration.
done	done	false	-
busy	busy	false	-
error	error	false	-
status	status	16#0	-
ADDR_1	-	P#DB4.DBX0.0 BYTE 100	Only absolute addressing is permitted.
RD_1	-	P#DB1.DBX0.0 BYTE 100	-

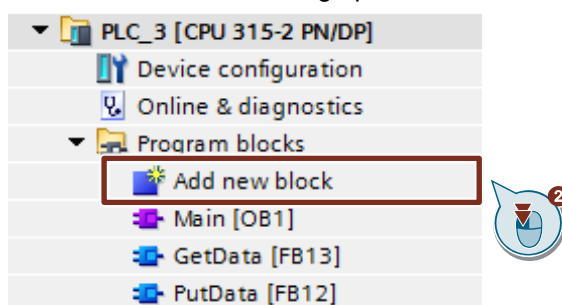
3.3 Install SIMATIC S7-300 as Server

3.3.1 Create Send and Receive Data Areas

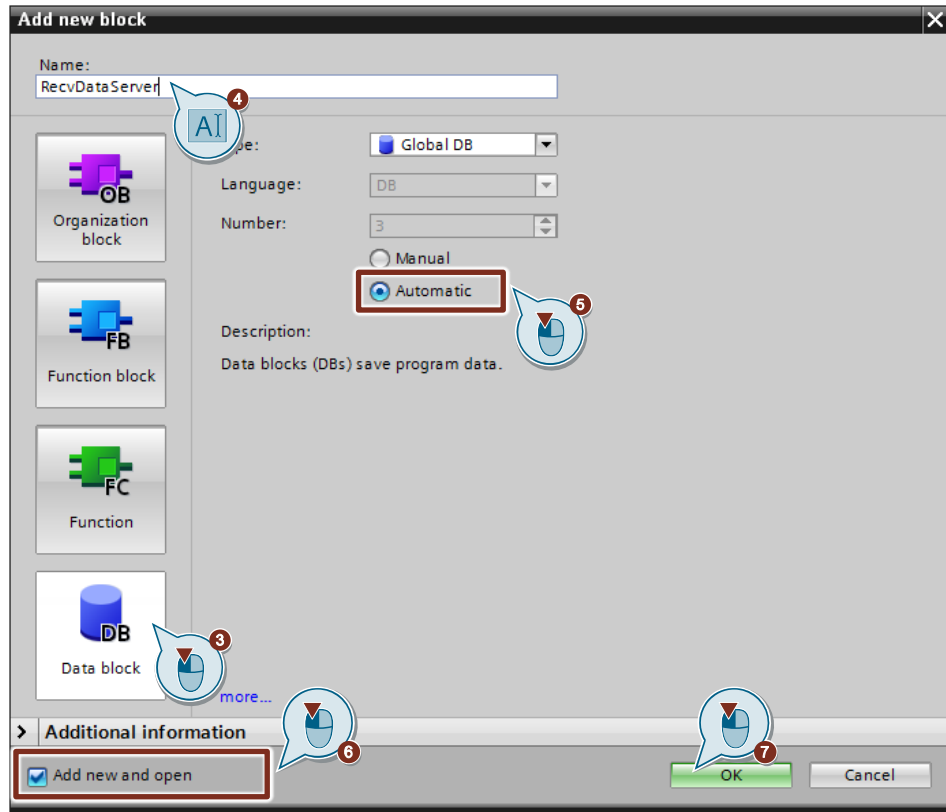
In the user program of the S7-300 CPU you add the data blocks for saving the Send and Receive data.

Add data block for saving the Receive data

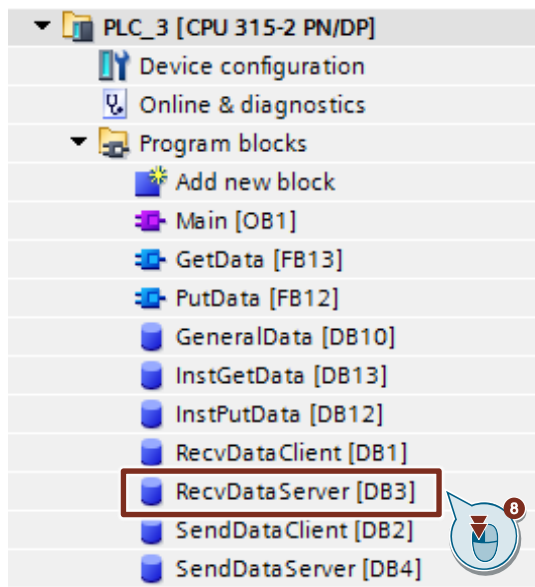
1. In the project tree you navigate to the device folder of the S7-300 CPU. The device folder contains structured objects and actions that belong to the device.
2. In the device folder you navigate to the "Program blocks" subfolder and double-click the "Add new block" action. The "Add new block" dialog opens.



3. Click the "Data block (DB)" button.
4. Enter the name of the data block (DB), "RecvDataServer", 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 "RecvDataServer" DB is added to the "Program blocks" folder of the S7-300 CPU.
 - If the "Add new and open" function is enabled, the "RecvDataServer" DB opens in the working area of STEP 7 V16.



8. Alternatively, you can open the "RecvDataServer" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-300 CPU and double-clicking the "RecvDataServer" DB.

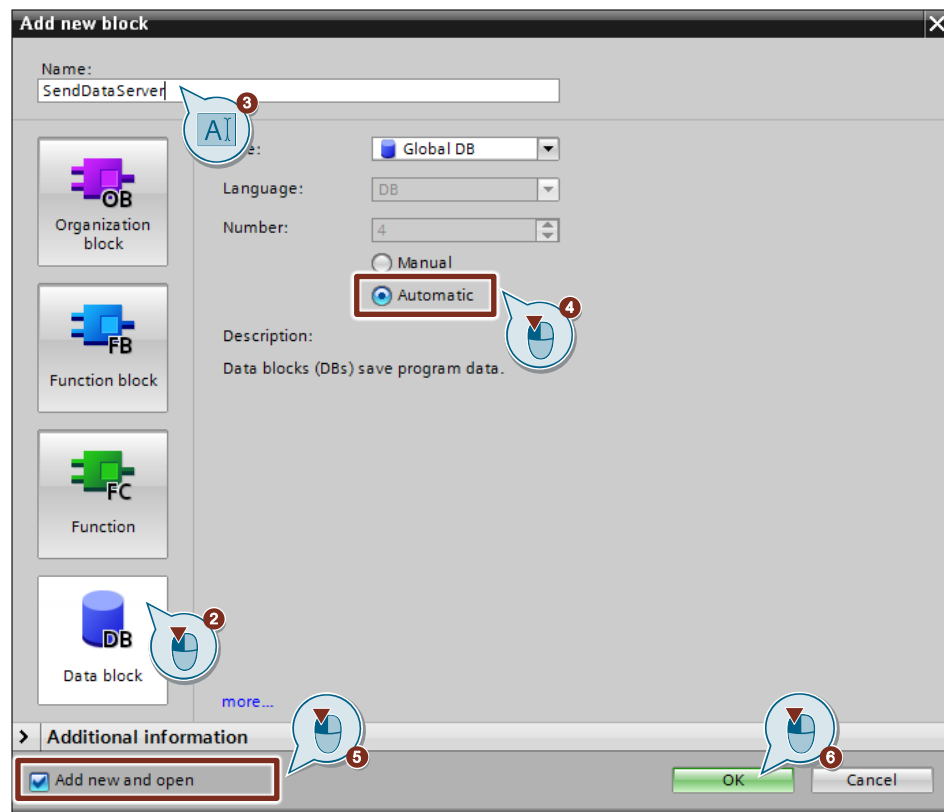


- In the DB "RecvDataServer" you define the variable "data" of the data type "Array[0..199] of Byte".

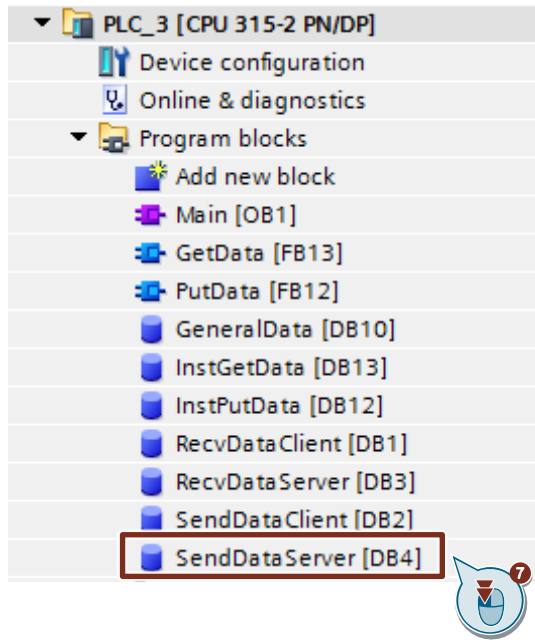
RecvDataServer			
Name	Data type	Offset	Start value
Static			
data	Array[0..199] of Byte	0.0	
<Add new>			

Add data block for saving the Send data

- Add another DB for saving the Send data.
- In the "Add new block" dialog you click the "Data block (DB)" button.
- Enter the name of the DB, "SendDataServer", for example.
- Enable the "Automatic" option to have the number of the DB assigned automatically by STEP 7.
- Enable the "Add new and open" function.
- 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 S7-300 CPU.
 - If the "Add new and open" function is enabled, the "SendDataServer" DB opens in the working area of STEP 7 V16.



- Alternatively, you can open the "SendDataServer" DB in the working area of STEP 7 V16 by going in the project tree to the "Program blocks" folder of the S7-300 CPU and double-clicking the "SendDataServer" DB.



- In the DB "SendDataServer" you define the variable "data" of the data type "Array[0..199] of Byte".

SendDataServer				
	Name	Data type	Offset	Start value
	Static			
	data	Array[0..199] of Byte	0.0	
	<Add new>			

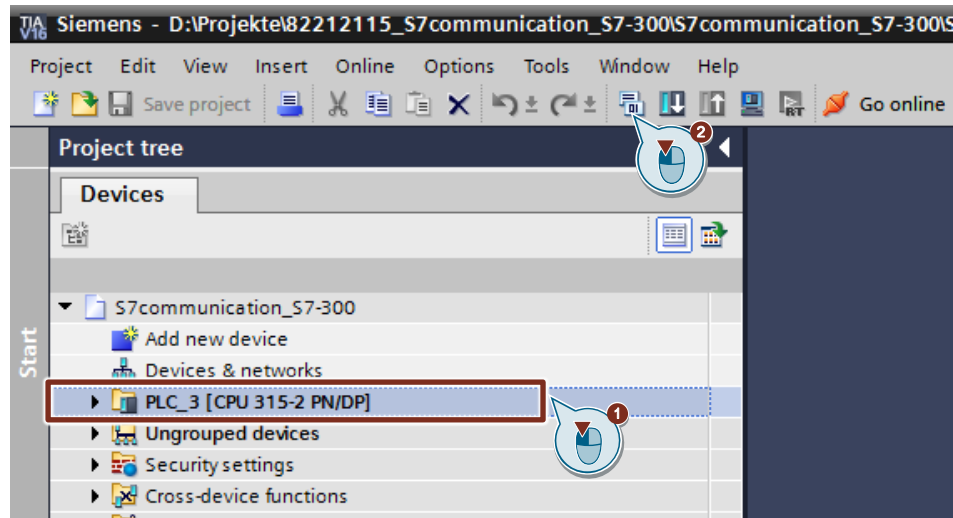
3.4 Download the Hardware Configuration and User Program

Requirements

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

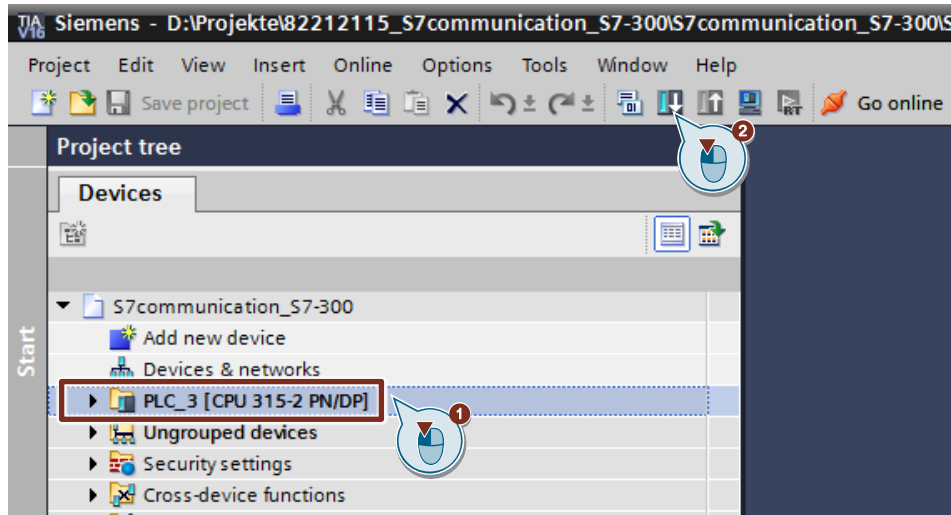
Compile

1. In the project tree you mark the device folder of the S7-300 CPU.
2. Click the "Compile" button in the toolbar. The hardware configuration and the software of the S7-300 are compiled.



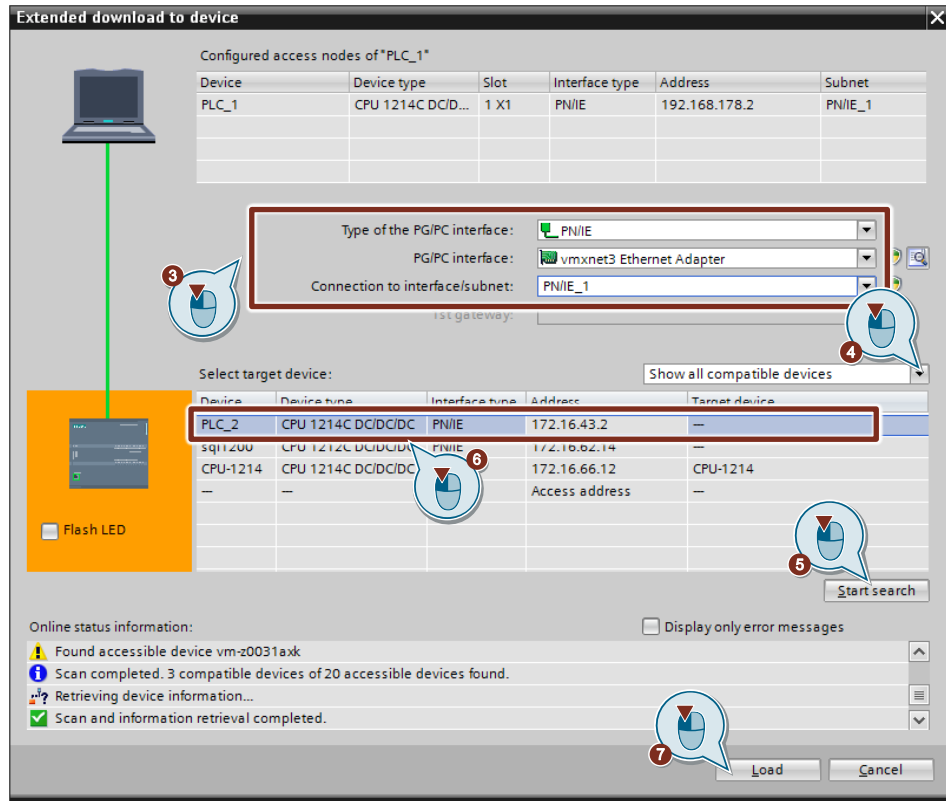
Download

1. In the project tree you mark the device folder of the S7-300 CPU.
2. Click the "Download to device" button in the toolbar. The "Extended download to device" or "Load preview" dialog opens automatically.

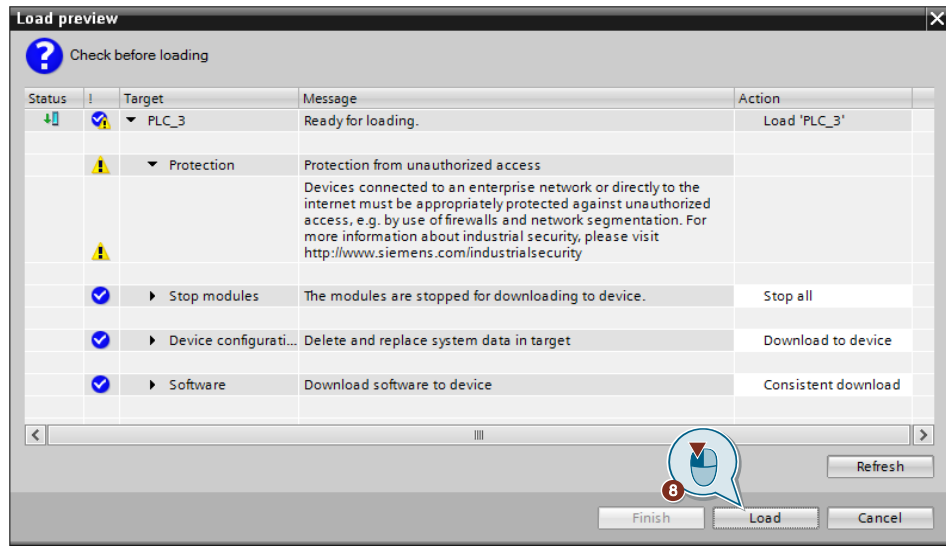


3. The "Extended download to device" dialog opens automatically only if the access path from the PG/PC to the S7-300 CPU has to be set. Make the following settings:
 - Type of the PG/PC interface: PN/IE
 - PG/PC interface: Network card of the PG/PC
 - Connection to interface/subnet: Subnet to which the S7-300 CPU is connected
4. In the drop-down list box, you select "Show all compatible devices".
5. Click "Start Search".
6. The S7-300 CPU is displayed in the "Select target device:" list. Mark the S7-300 CPU.
7. Click the "Load" button.

3 Configuration of SIMATIC S7-300 in STEP 7 V16

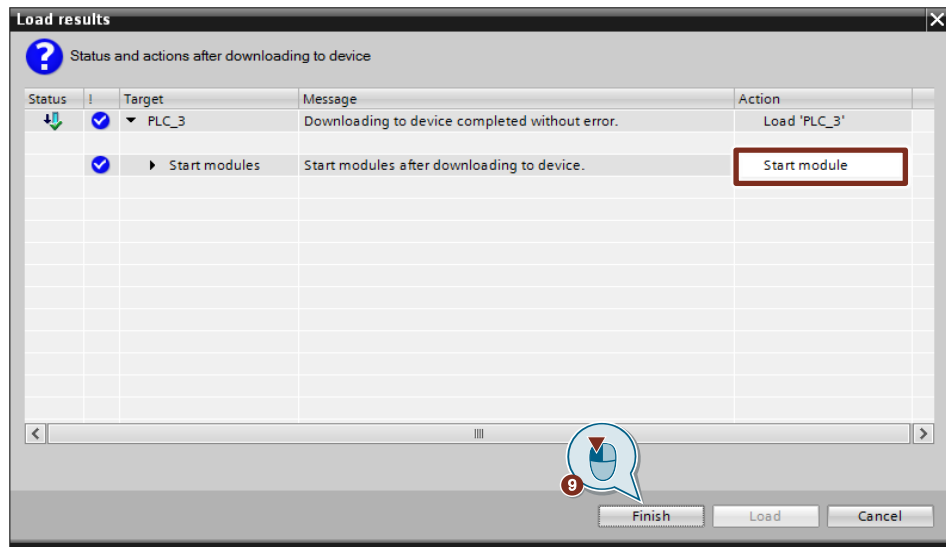


- In the "Load preview" dialog you click the "Load" button to start the loading procedure.



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

3 Configuration of SIMATIC S7-300 in STEP 7 V16



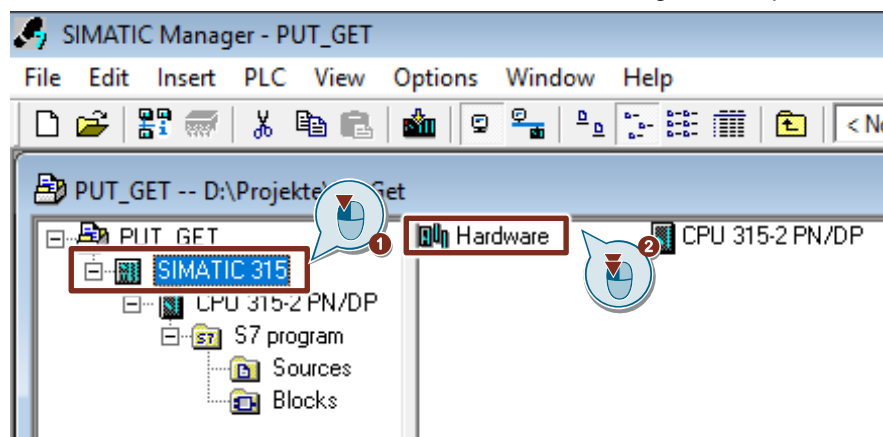
4 Configuration of SIMATIC S7-300 in STEP 7 V5.6

1. Create a new project or open an existing project in STEP 7 V5.6.
2. Add an S7-300 station and create the hardware configuration.
3. Install the S7-300 CPU as client or server.
 - Client: Configure a new S7 connection and create the user program. In the user program you call the "PUT" instruction to write data to the S7-300 CPU. In the user program you call the "GET" instruction to read data from the S7-300 CPU. Add the data blocks for saving the Send and Receive data.
 - Server: Create the user program. In the user program you add data blocks in which the sent and received data is stored.

4.1 Create the Hardware Configuration

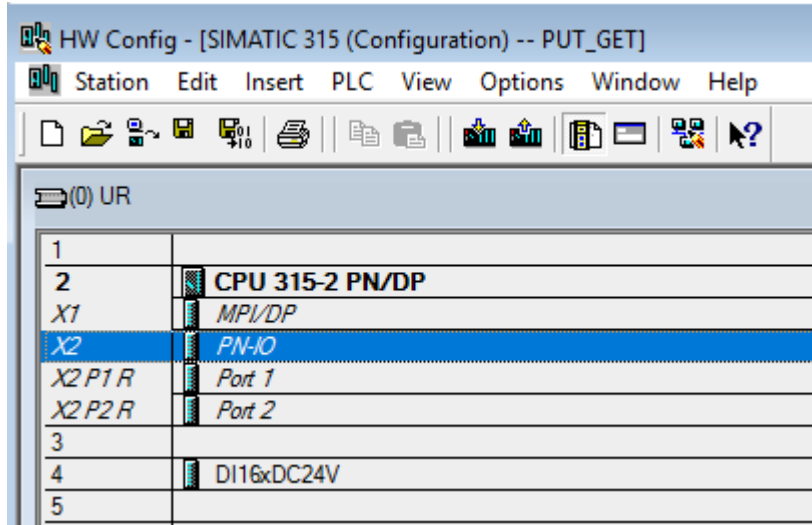
4.1.1 Open the Hardware Configuration

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



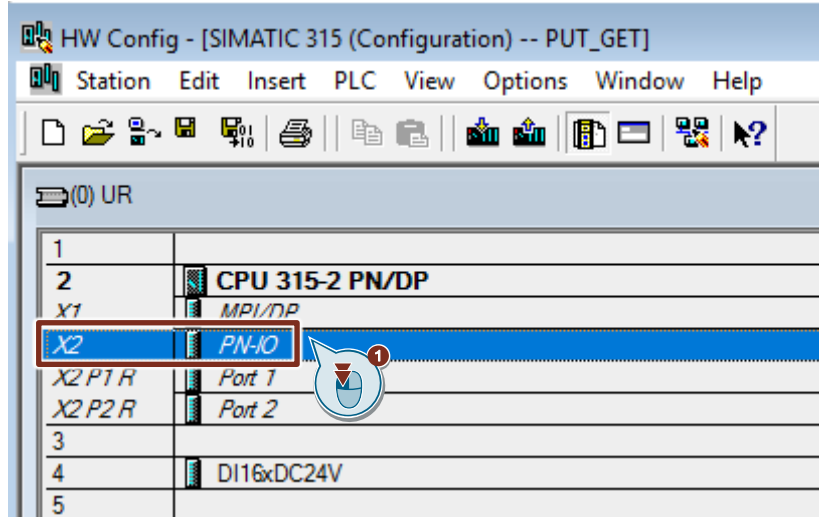
4 Configuration of SIMATIC S7-300 in STEP 7 V5.6

3. Drag and drop the relevant modules like Power Supply, CPU, CP etc. from the hardware catalog into the S7-300 profile channel.

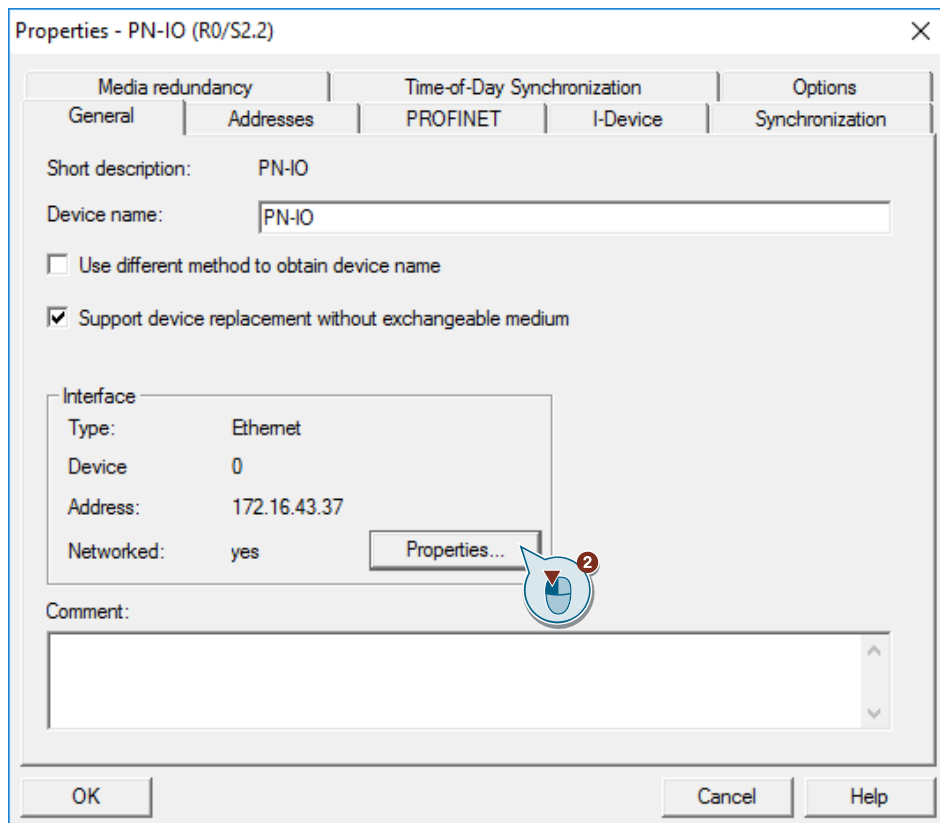


4.1.2 Define IP Address and Assign Subnet to the PROFINET interface of the CPU

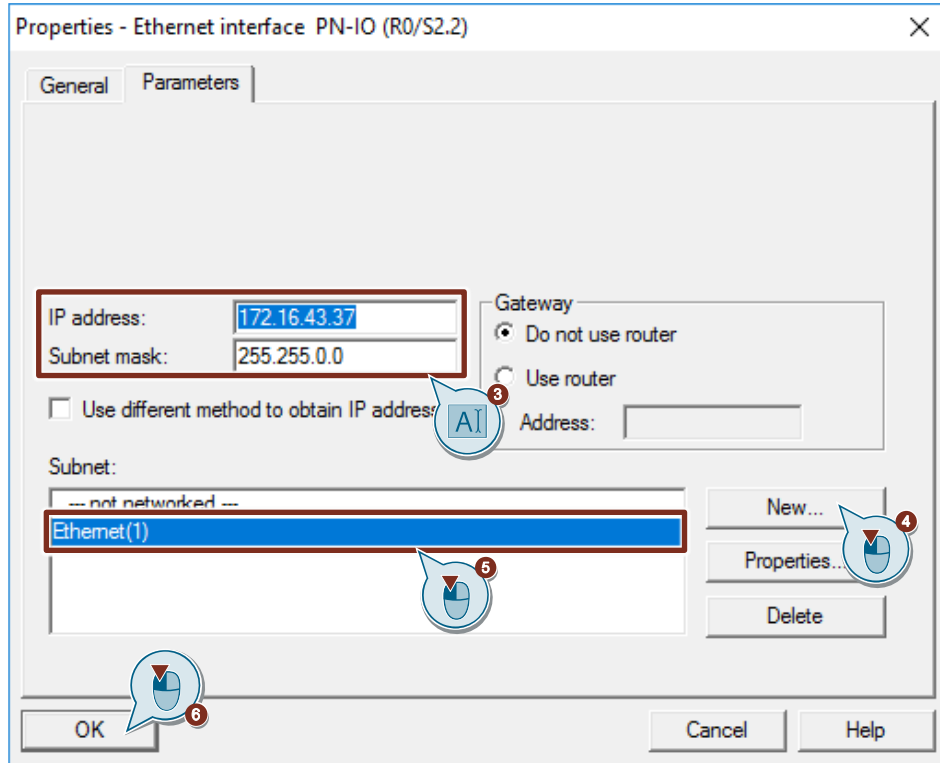
1. Double-click the PROFINET interface of the S7-300 CPU. The Properties dialog of the PROFINET interface opens.



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

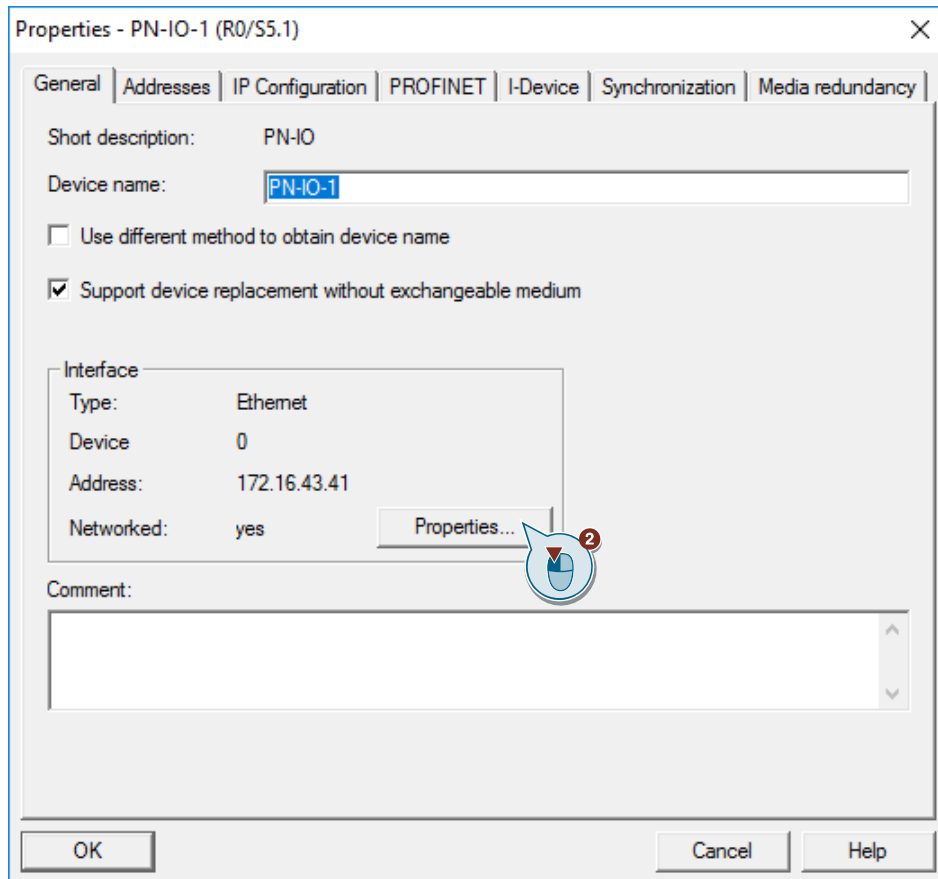


3. Enter the IP address and subnet mask, for example:
 - IP address: 172.16.43.37
 - Subnet mask: 255.255.0.0
4. Click the "New..." button to create a new subnet.
5. Select the new subnet or another existing subnet.
6. Click the "OK" button to apply the IP address and subnet mask and assign the selected subnet to the PROFINET interface of the S7-300 CPU.

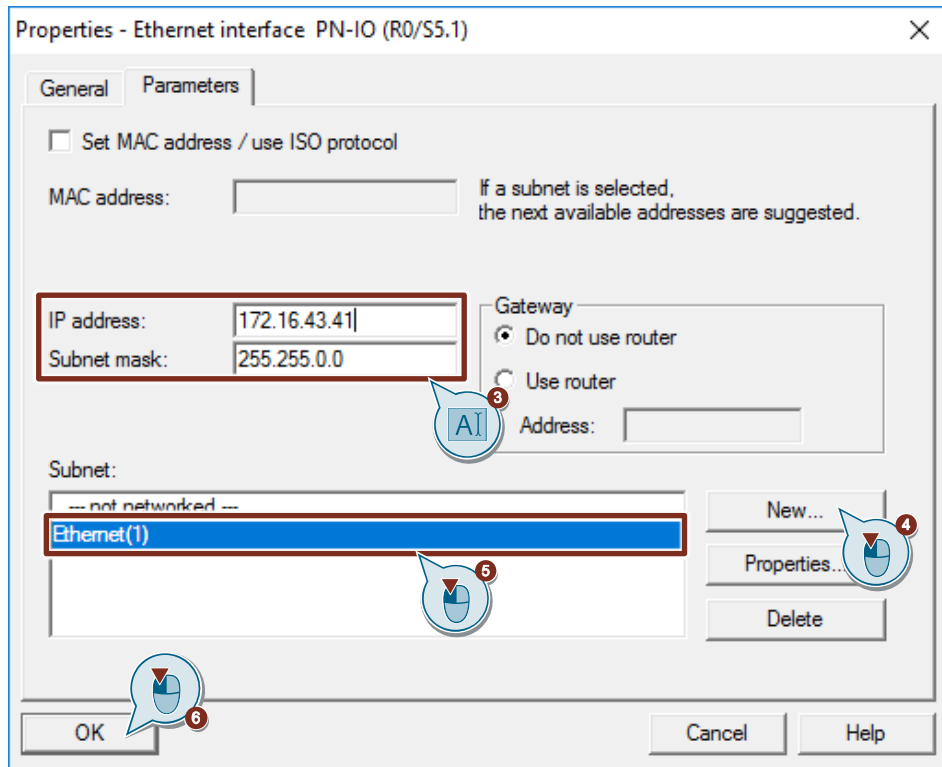


4.1.3 Define IP Address and Assign Subnet to the PROFINET interface of the CP

1. Double-click the PROFINET interface of the CP. 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: 172.16.43.41
 - 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 CP.



4.2 Configure the SIMATIC S7-300 as Client

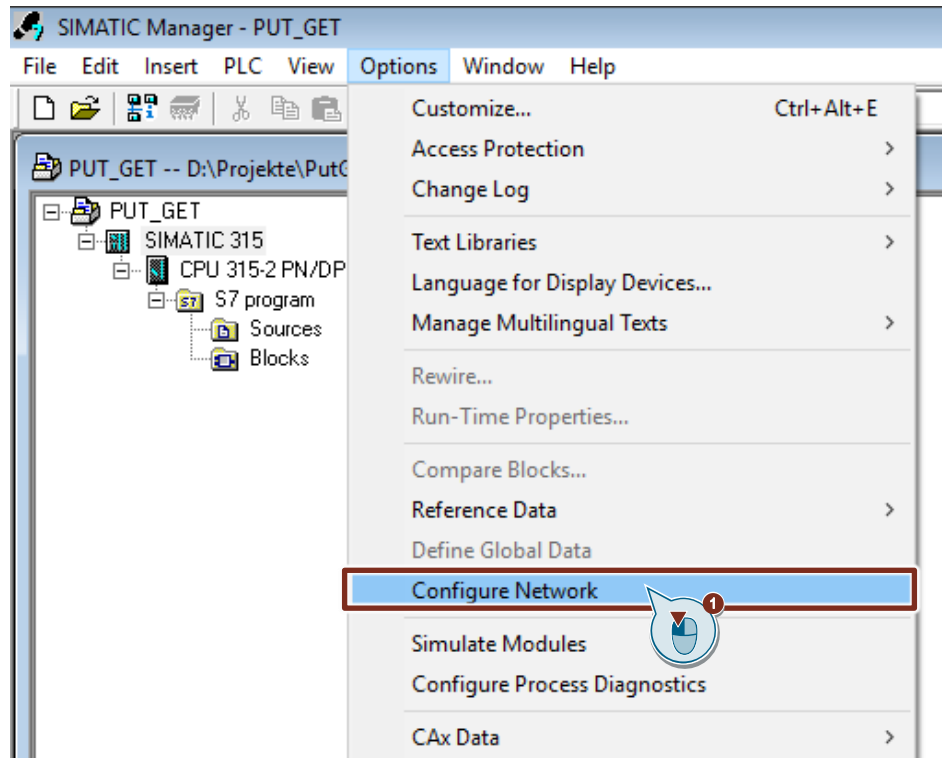
The S7-300 station actively sets up the S7 connection. In this case you must configure an S7 connection for the S7-300 CPU and call the following function blocks (FBs) in the user program:

- 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".

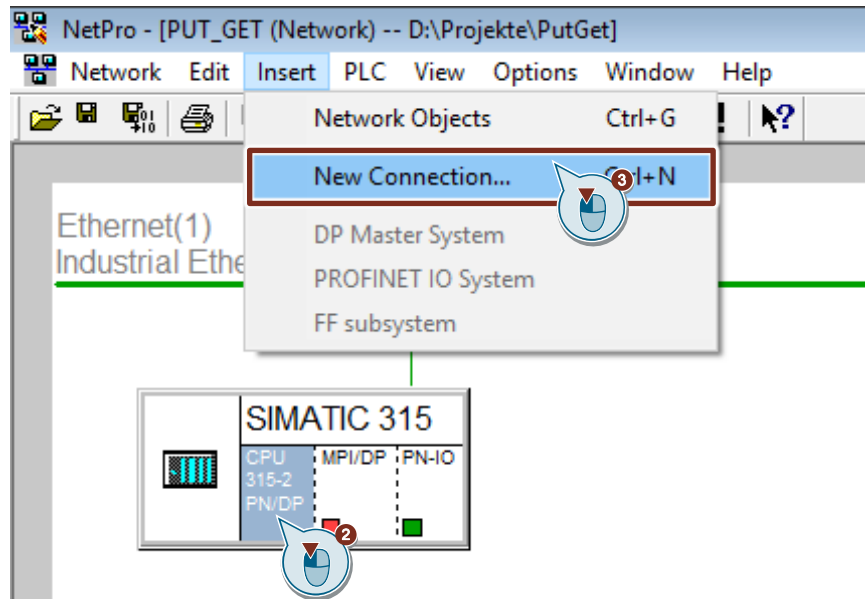
4.2.1 Configuration of the S7 Connection

Add an S7 Connection

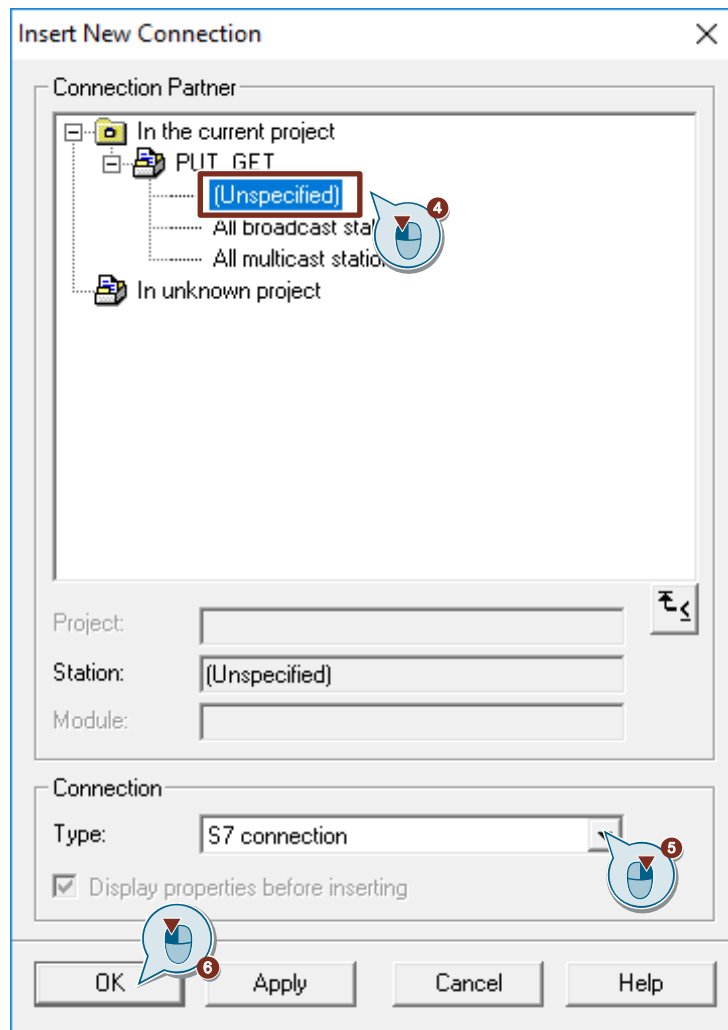
1. In the SIMATIC Manager you open the "NetPro" tool via the menu "Options > Configure Network". In "NetPro" you configure the S7 connection for the S7-300 CPU.



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



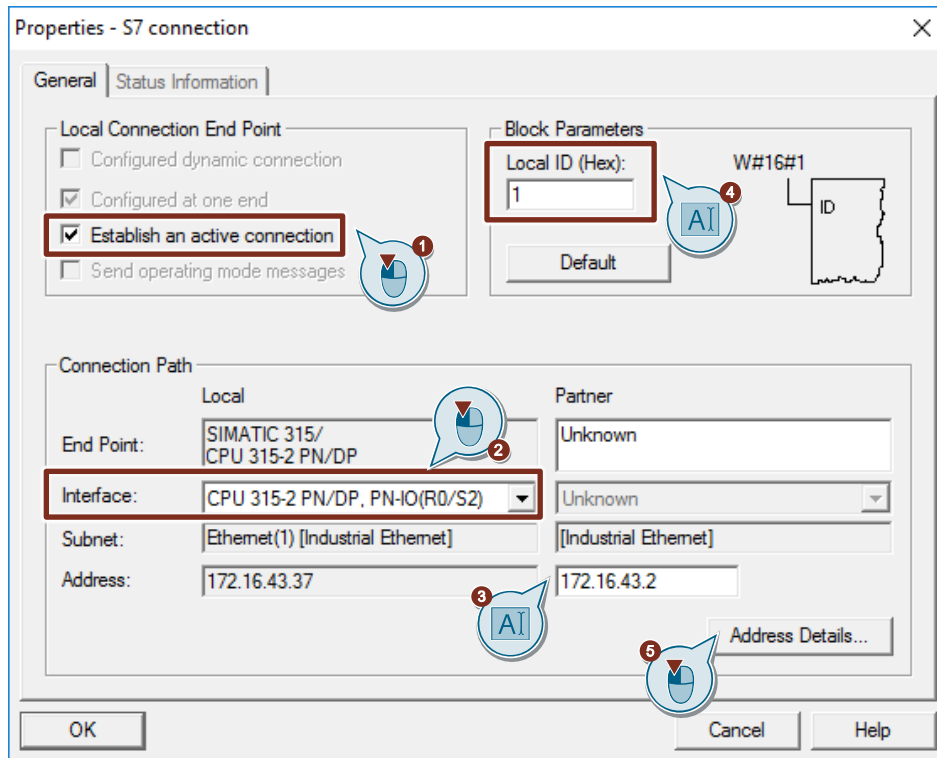
4. Select the "Unspecified" item under "Connection Partner".
5. Select "S7 connection" as the connection type.
6. Click "Apply". The Properties dialog of the S7 connection opens.



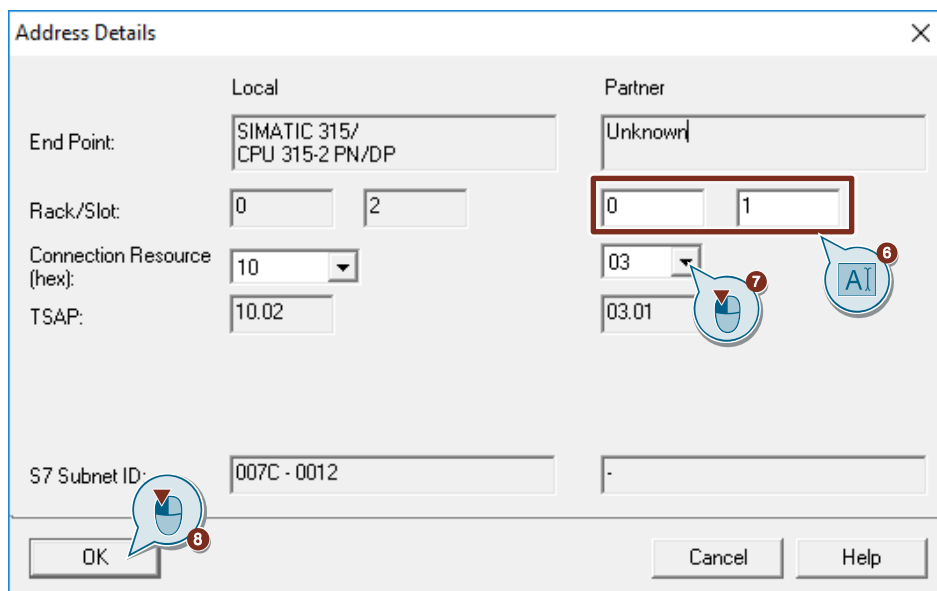
Define connection parameters

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

1. Enable the "Establish an active connection" function.
2. Select the local interface. In this example, the integrated interface of the S7-300 CPU is used. If the S7 connection is to be established via a CP, select the CP.
3. 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.
If you use a CP as partner instead of the integrated interface of the S7-1200 CPU, enter the IP address of the CP.
4. You specify the local ID of the connection in the user program at the input parameter "ID" of the function blocks FB/SFB14 "GET" and FB/SFB15 "PUT".
5. Click the "Address Details..." button. The "Address Details" dialog opens.



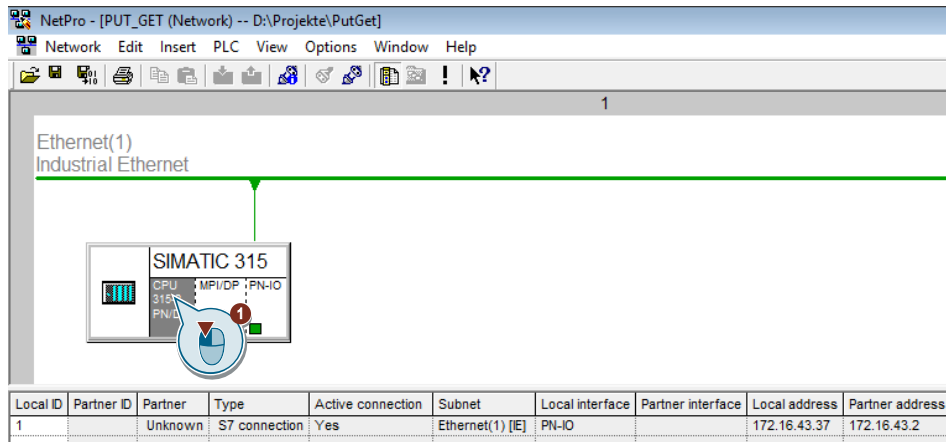
6. Enter the rack and slot of the partner CPU or CP. The S7-1200 CPU uses Rack 0 and Slot 1.
7. Select the connection resource 03(hex) for the partner CPU because the S7 connection is only configured unilaterally in the S7-300 CPU. With these settings the TSAP has the value 03.01 in the partner CPU.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



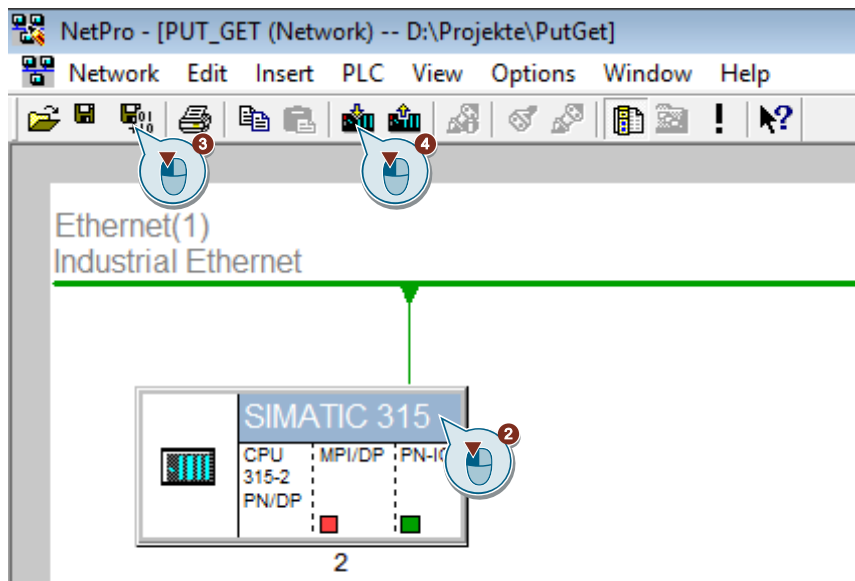
9. Likewise apply the settings in the Properties dialog of the S7 connection with "OK".
10. In the "Insert New Connection" dialog you click the "OK" button to close the dialog. The S7 connection is inserted in "NetPro".

Load the S7 Connection

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

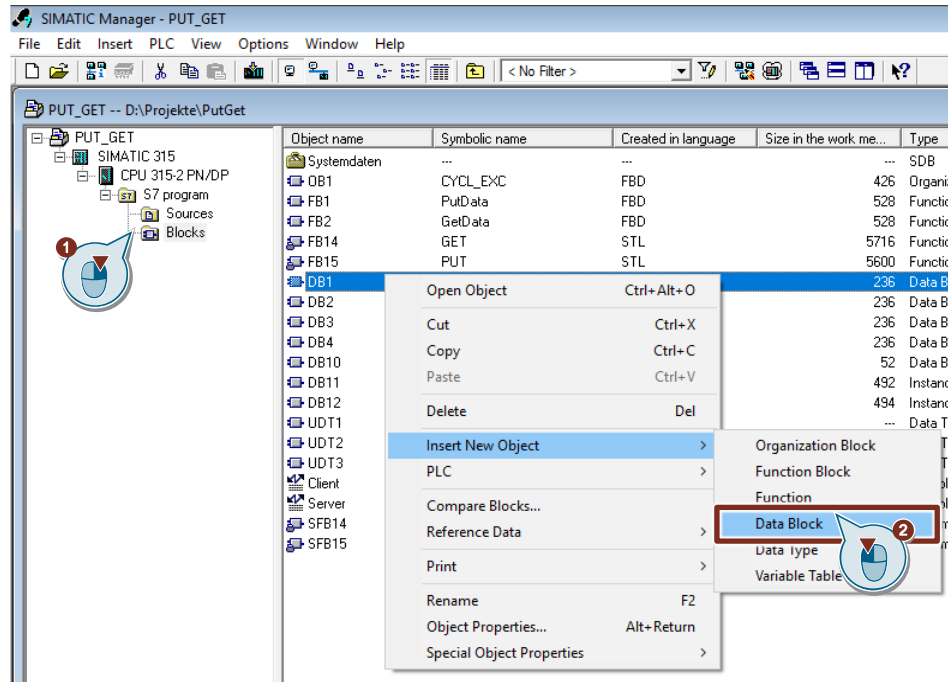


4.2.2 Create Send and Receive Data Areas

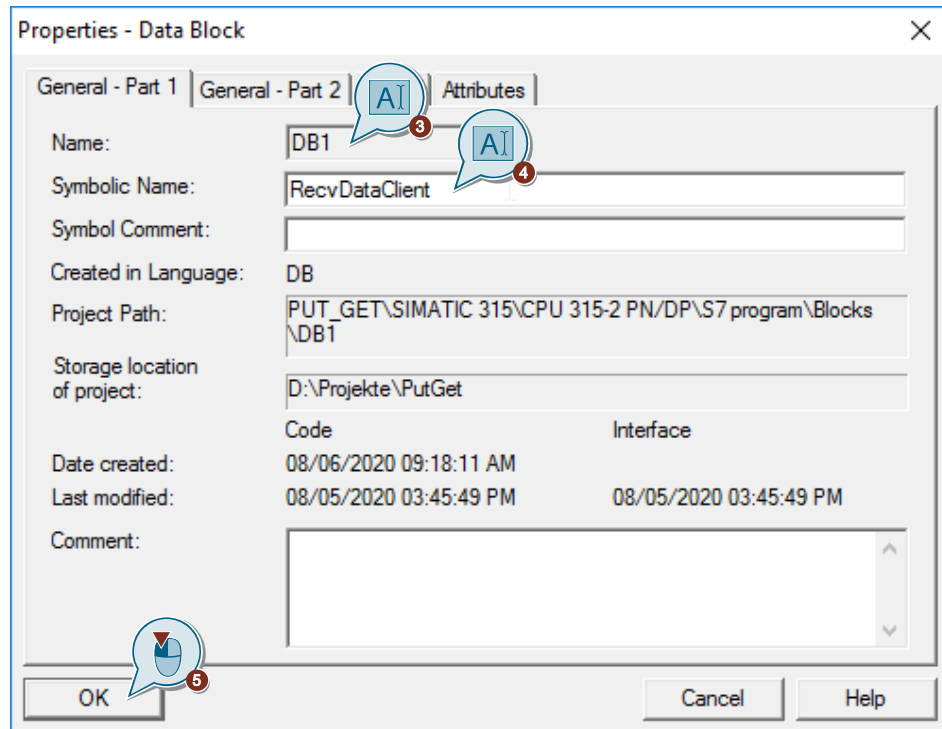
Add data block for saving the Receive data

The data received from the S7-1200 CPU is stored in data block DB1 "RecvDataClient" of the S7-300 CPU.

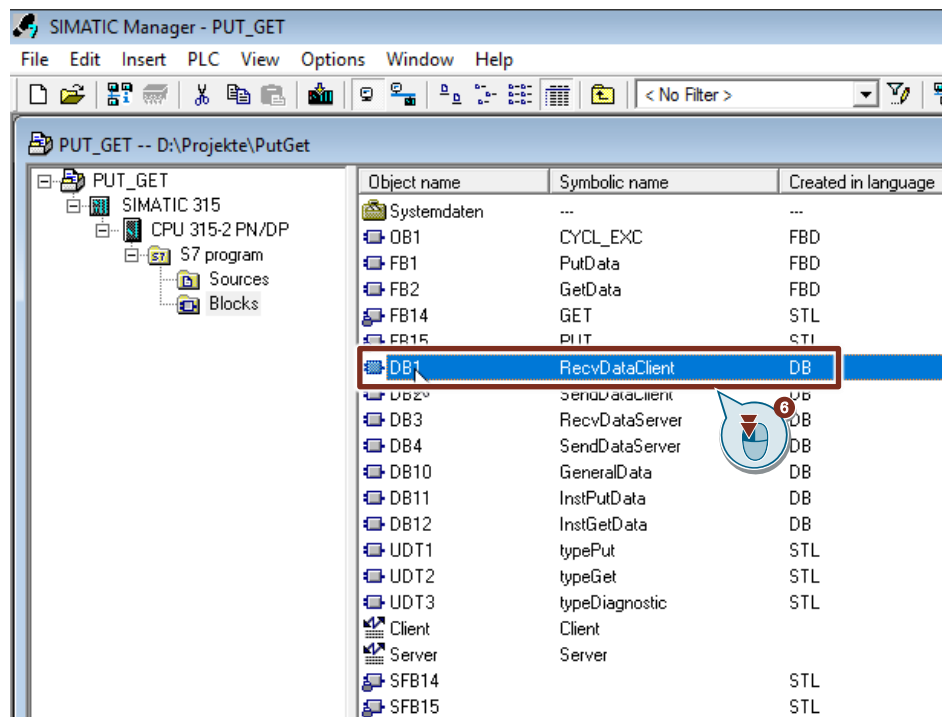
1. In your STEP 7 project you right-click the "Blocks" folder. The pop-up menu opens.
2. You select the menu "Insert New Object > Data Block". The Properties dialog of the DB opens.



3. You enter the name and type of block that is to be created, DB1, for example.
4. Enter a symbolic name for the block, "RecvDataClient", for example.
5. Apply the settings with "OK".
Block DB1 "RecvDataClient" is added to the "Blocks" folder.



6. Double-click DB1 "RecvDataClient" in the "Blocks" folder. The DB opens.



7. In the DB "RecvDataClient" you define the variable "data" of the data type "Array[0..199] of Byte".

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	data	ARRAY[0..199]	B#16#0
+1.0		BYTE	
=200.0		END_STRUCT	

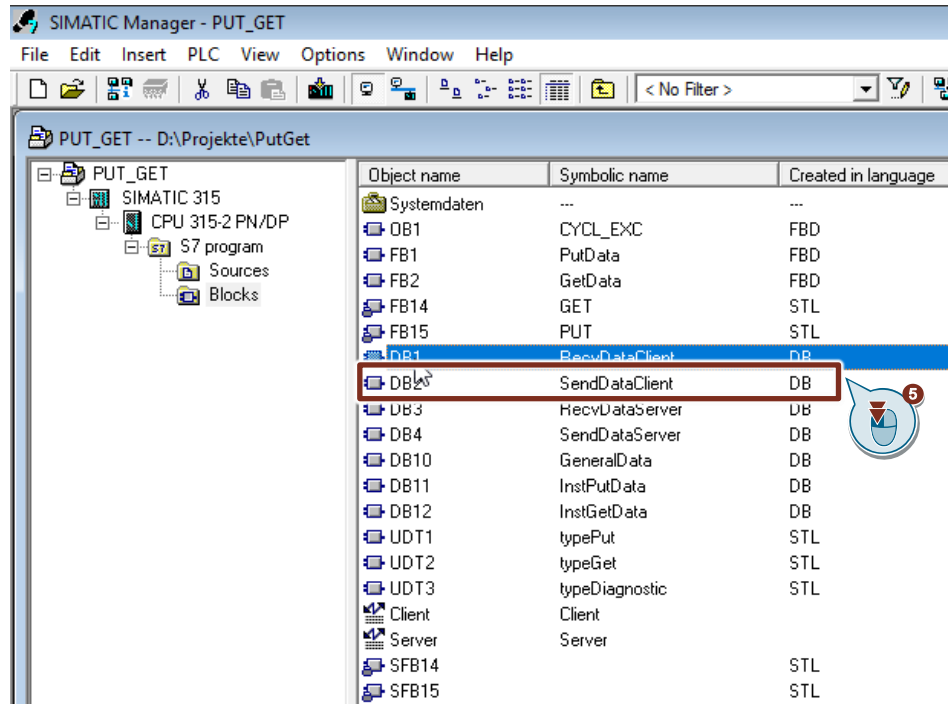
Add data block for saving the Send data

The data that is transferred to the S7-1200 CPU is stored in data block DB2 "SendDataClient" of the S7-300 CPU.

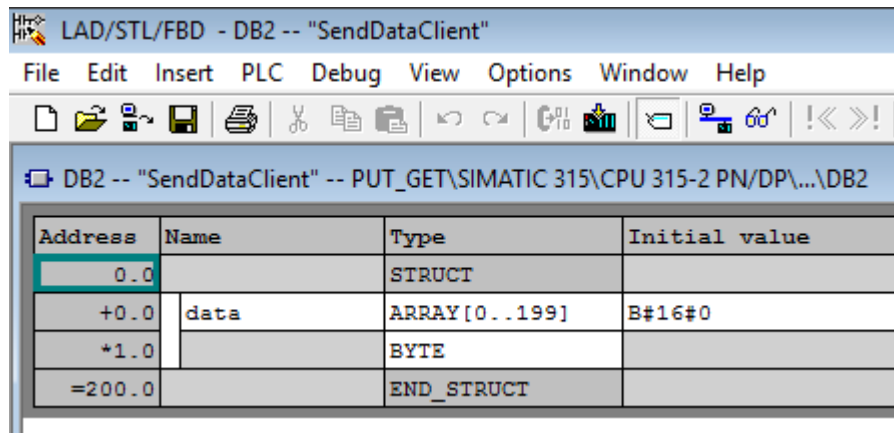
1. Add another DB for saving the Send data.
2. You enter the name and type of block that is to be created, DB2, for example.
3. Enter a symbolic name for the block, "SendDataClient", for example.
4. Apply the settings with "OK".

Block DB2 "SendDataClient" is added to the "Blocks" folder.

5. Double-click DB2 "SendDataClient" in the "Blocks" folder.
The DB opens.

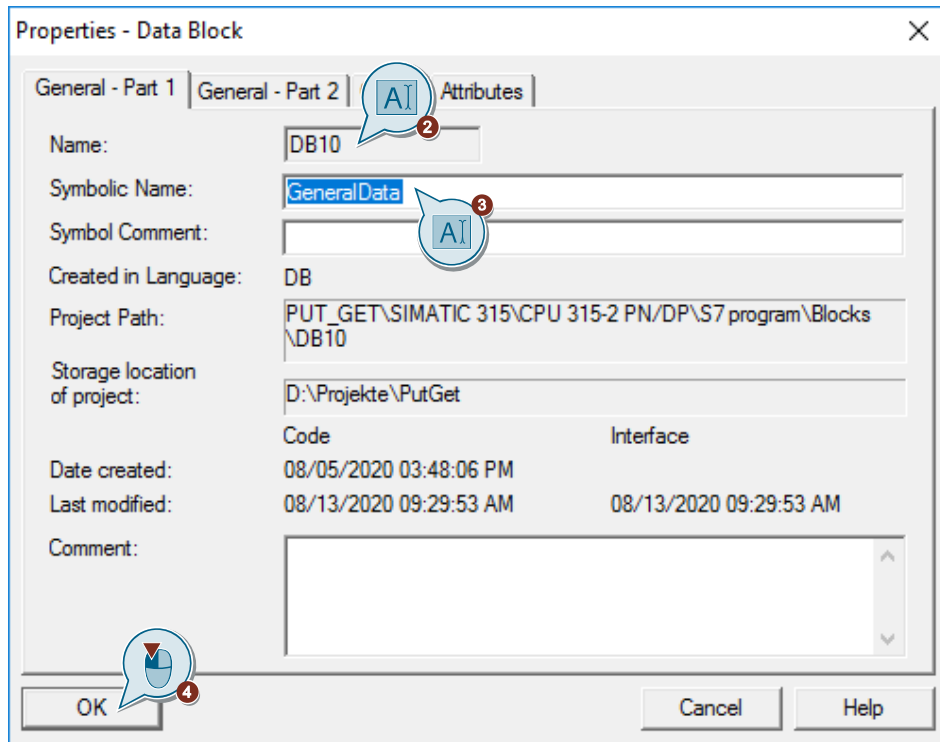


6. In the DB2 "SendDataClient" you define the variable "data" of the data type "Array[0..99] of Byte".



4.2.3 Create Variables for Parameters of Function Blocks "PutData" and "GetData"

1. Add another DB to create variables for the parameters of function blocks "PutData" and "GetData".
2. You enter the name and type of block that is to be created, DB10, for example.
3. Enter a symbolic name for the block, "GeneralData", for example.
4. Apply the settings with "OK".
Block DB10 "GeneralData" is added to the "Blocks" folder.



5. Create the following variables to assign the input and output parameters of the FBs "PutData" and "GetData". UDTs are used as data types.

Variable	UDT	Description
put	UDT1 "typePut"	Variables for parameterizing FB1 "PutData".
get	UDT2 "typeGet"	Variables for parameterizing FB2 "GetData".
diagnostic	UDT3 "typeDiagnostic"	Variables to store the status of FBs "PutData" and "GetData" in case of error.

UDT1 "typePut"

The following table shows the structure of UDT1 "typePut".

Table 4-1

Parameter	Data type	Start value	Description
execute	Boolean	false	Control parameter for FB1 "PutData".
connectionId	Word	16#0	Addressing parameter for specifying the connection to the communication partner.
done	Bool	false	Status parameters
busy	Bool	false	Status parameters
error	Bool	false	Status parameters
status	Word	16#0	Status parameters

UDT2 "typeGet"

The following table shows the structure of UDT2 "typeGet".

Table 4-2

Parameter	Data type	Start value	Description
execute	Bool	false	Control parameter for FB2 "GetData".
connectionId	Word	16#0	Addressing parameter for specifying the connection to the communication partner.
done	Bool	false	Status parameters
busy	Bool	false	Status parameters
error	Bool	false	Status parameters
status	Word	16#0	Status parameters

UDT3 "typeDiagnostic"

The following table shows the structure of UDT3 "typeDiagnostic".

Table 4-3

Parameter	Data type	Start value	Description
statusPut	Word	16#0	Parameter to store the status of FB1 "PutData".
statusGet	Word	16#0	Parameter to store the status of FB2 "GetData".

4.2.4 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.

If you are using the integrated PROFINET interface of the S7-300 CPU for data communication via S7 connections, then utilize function blocks FB14 "GET" and FB15 "PUT" from the Standard Library under "Communication Blocks -> Blocks" with the family "CPU_300".

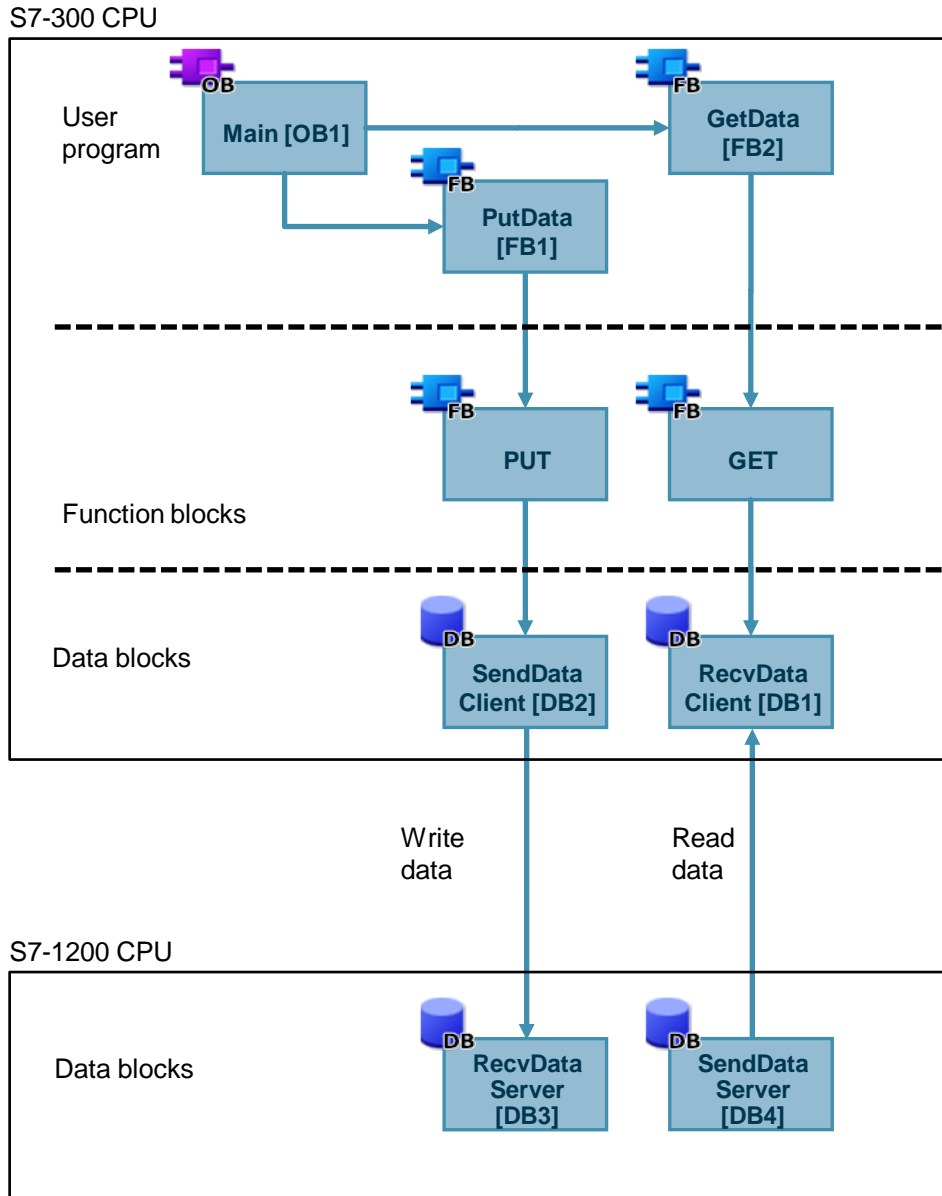
If you are using an Industrial Ethernet CP or PROFIBUS CP for data communication via S7 connections in the S7-300 station, then utilize the function blocks FB14 "GET" and FB15 "PUT" from the "SIMATIC_NET_CP" library under "CP 300 > Blocks".

Copy the FBs from the Standard Library or from the "SIMATIC_NET_CP" 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 CPU and S7-1200 CPU.
Figure 4-1



FB1 "PutData"

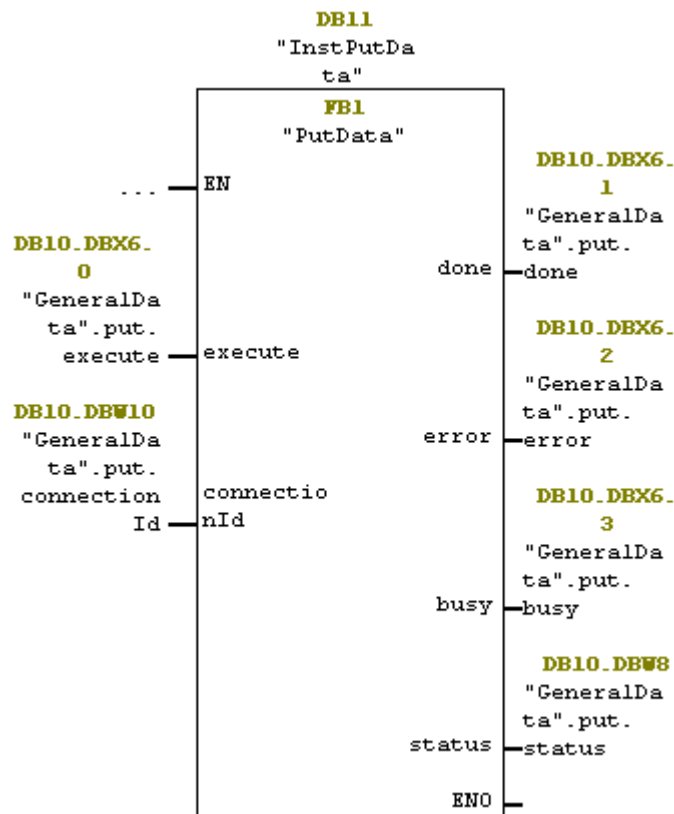
The FB1 "PutData" calls the FB15 "PUT" to execute the following functions:

- Send data to the partner via the configured connection as soon as the input "execute" recognizes a positive edge. When the Write job is running, it is not possible to trigger a new Write job.
- Output the status of the FB and data transmission at the "status" output.

FB1 "PutData" is called cyclically in OB1.

The following figure shows the call of the FB1 "PutData" in OB1.

Figure 4-2



FB1 "PutData" has the following parameters.

Table 4-4

Name	P type	Data type	Comment
execute	IN	Bool	Control parameter: Enables the Write job on a rising edge.
connectionId	IN	Word	Reference to the local connection description (preset by the connection configuration in in "NetPro"). In this example the value w#16#1 is preset by the configuration connection.
done	OUT	Bool	TRUE: The Write job was executed error-free.
busy	OUT	Bool	TRUE: The Write job is running.
error	OUT	Bool	Status parameters "error" and "status":
status	OUT	Word	<ul style="list-style-type: none"> • "error" = 0: <ul style="list-style-type: none"> - "status" = 0000 (hex): neither warning nor error - "status" <> 0000 (hex): Warning The "status" parameter provides detailed information. • "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.

Assign the variables created in section [4.2.3](#) to the parameters of FB1 "PutData".

The following table shows the assignment of the parameters of the "put" variable to the parameters of FB1 "PutData".

Table 4-5

Parameters of FB1 "PutData"	Parameters of the "put" variable	Start value	Note
execute	execute	false	Set the "execute" parameter to the value "TRUE" to start the Write job. The Write job is started on a rising edge. Reset the "execute" parameter to the value "FALSE" if the Write job is completed with "done" = true or "error" = true.
connectionId	connectionId	16#1	Enter the local ID of the connection that you defined during the connection configuration.
done	done	false	-
busy	busy	false	-
error	error	false	-
status	status	16#0	-

Open FB1 "PutData" to define the Receive and Send data areas in FB15 "PUT".

Receive data area

At input "ADDR_1" of FB15 "PUT" you refer to the area in the partner CPU (S7-1200 CPU, for example) to which you want to write.

- P#DB1.DBX0.0 BYTE 100.

Note The following minimum user data size for the "PUT" instruction is guaranteed:
160 bytes

Send data area

At input "SD_1" of FB15 "PUT" you refer to the area in your own S7 CPU (S7-300 CPU) that contains the data to be sent.

- P#DB2.DBX0.0 BYTE 100

Note The following minimum user data size for the "PUT" instruction is guaranteed:
160 bytes

FB 2 "GetData"

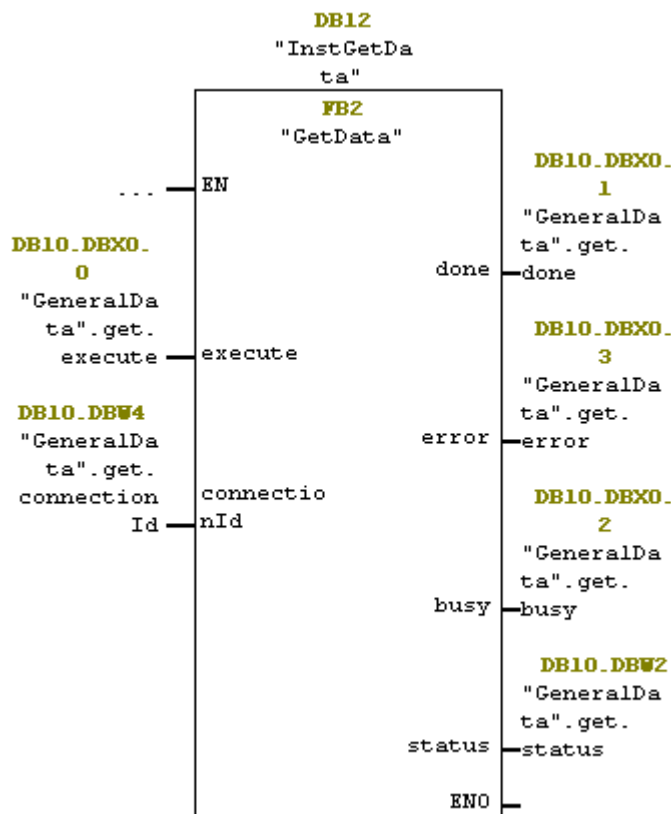
The FB2 "GetData" calls the "GET" instruction to execute the following functions:

- Receive data from the partner via the configured connection as soon as the input "execute" recognizes a positive edge. When the Read job is running, it is not possible to trigger a new Read job.
- Output the status of the FB and data transmission at the "status" output.

FB2 "GetData" is called cyclically in OB1.

The following figure shows the call of the FB2 "GetData" in OB1.

Figure 4-3



FB2 "PutData" has the following parameters.

Table 4-6

Name	P type	Data type	Comment
execute	IN	Bool	Control parameter: Enables the Read job on a rising edge.
connectionId	IN	Word	Reference to the local connection description (preset by the connection configuration in in "NetPro"). In this example the value w#16#1 is preset by the configuration connection.
done	OUT	Bool	TRUE: The Read job was executed error-free.
busy	OUT	Bool	TRUE: The read job is running.
error	OUT	Bool	Status parameters "error" and "status":
status	OUT	Word	<ul style="list-style-type: none"> • "error" = 0: <ul style="list-style-type: none"> - "status" = 0000 (hex): neither warning nor error - "status" <> 0000 (hex): Warning The "status" parameter provides detailed information. • "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.

Assign the variables created in section [4.2.3](#) to the parameters of FB2 "GetData".

The following table shows the assignment of the parameters of the "get" variable to the parameters of FB2 "GetData".

Table 4-7

Parameters of FB2 "GetData"	Parameters of the "get" variable	Start value	Note
execute	execute	false	Set the "execute" parameter to the value "TRUE" to start the Read job. The Read job is started on a rising edge. Reset the "execute" parameter to the value "FALSE" if the Read job is completed with "done" = true or "error" = true.
connectionId	connectionId	16#1	Enter the local ID of the connection that you defined during the connection configuration.
done	done	false	-
busy	busy	false	-
error	error	false	-
status	status	16#0	-

Open FB2 "GetData" to define the Receive and Send data areas in FB14 "GET".

Send data area

At input "ADDR_1" of FB14 "GET" you refer to the area in the partner CPU (S7-1200 CPU, for example) which you want to read.

- P#DB4.DBX0.0 BYTE 100

Note The following minimum user data size for the "GET" instruction is guaranteed:
160 bytes

Receive data area

At input "RD_1" of FB14 "GET" you refer to the area in your own S7 CPU (S7-300 CPU) that contains the data to be read.

- P#DB1.DBX0.0 BYTE 100

Note The following minimum user data size for the "GET" instruction is guaranteed:
160 bytes

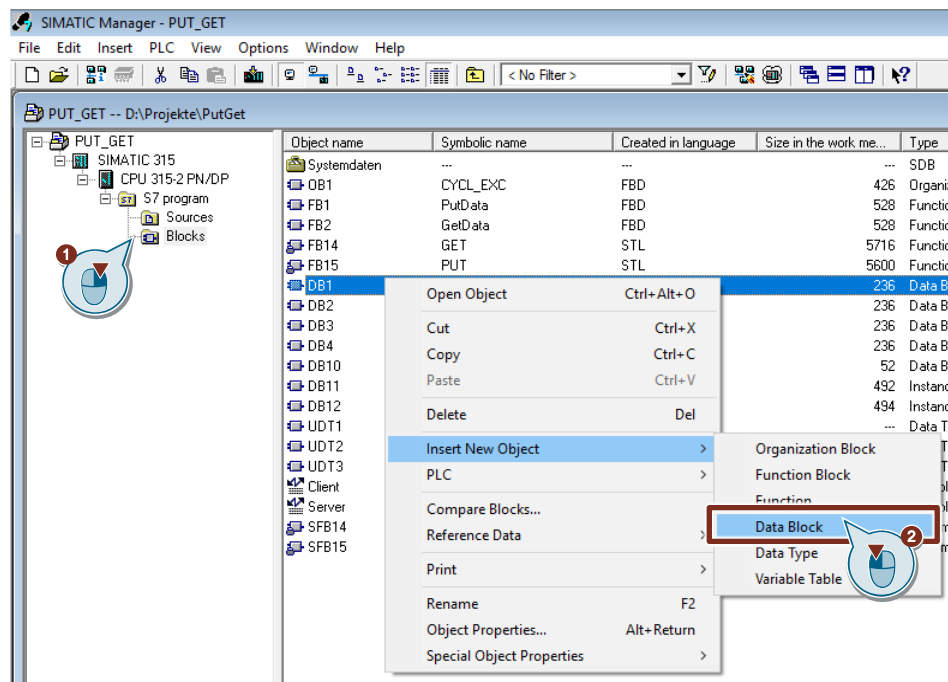
4.3 Configure the SIMATIC S7-300 as Server

4.3.1 Create a User Program

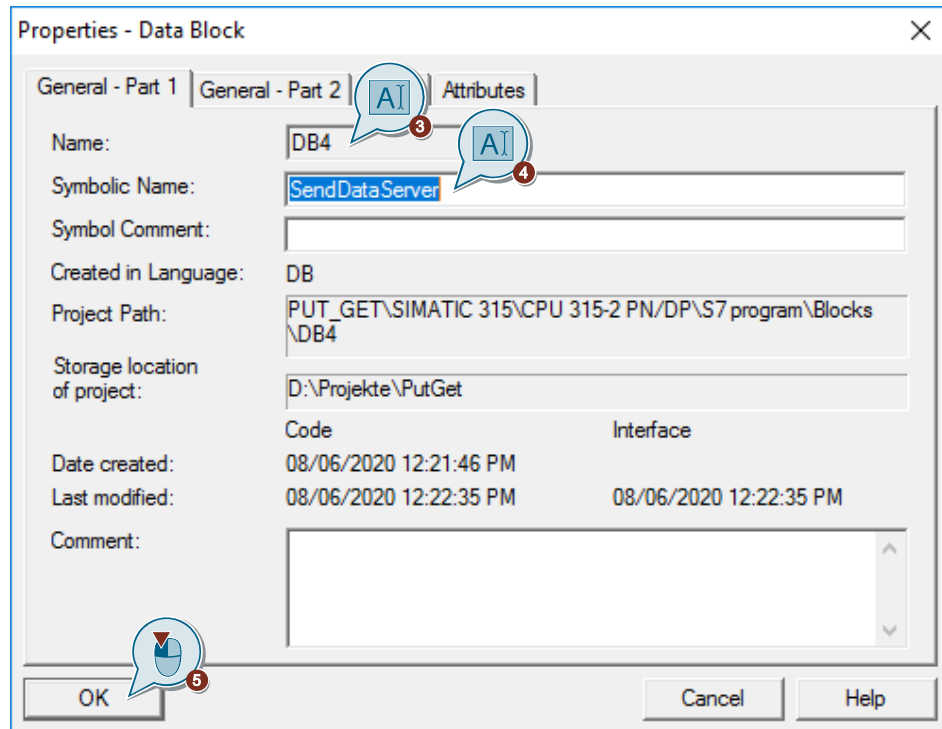
In the user program of the S7-300 CPU you add the data blocks for saving the Send and Receive data.

Add data block for saving the Send data

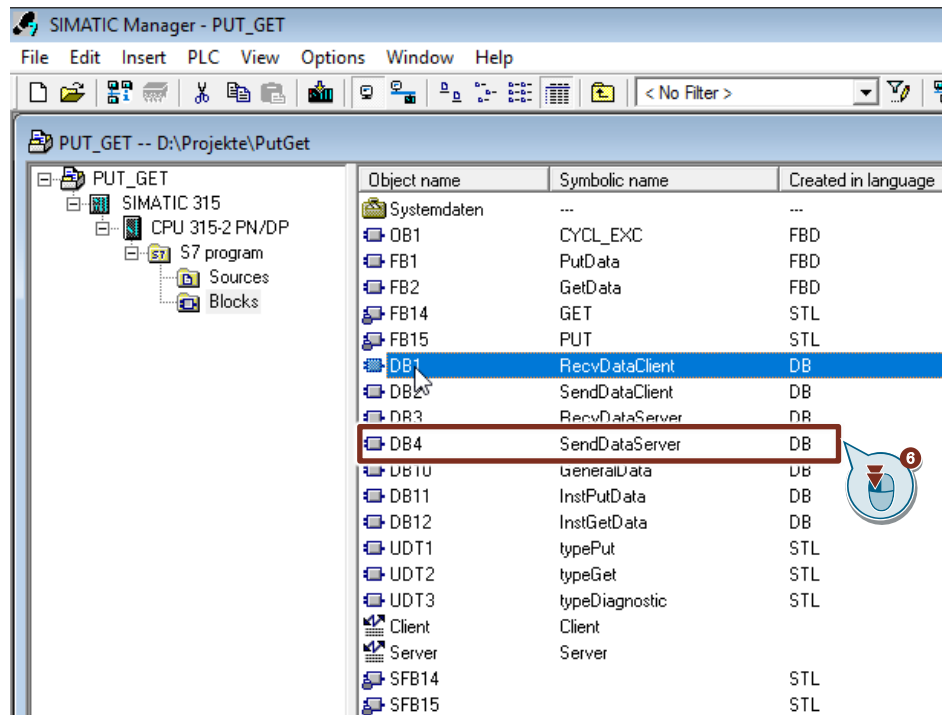
1. In your STEP 7 project you right-click the "Blocks" folder. The pop-up menu opens.
2. You select the menu "Insert New Object > Data Block". The Properties dialog of the DB opens.



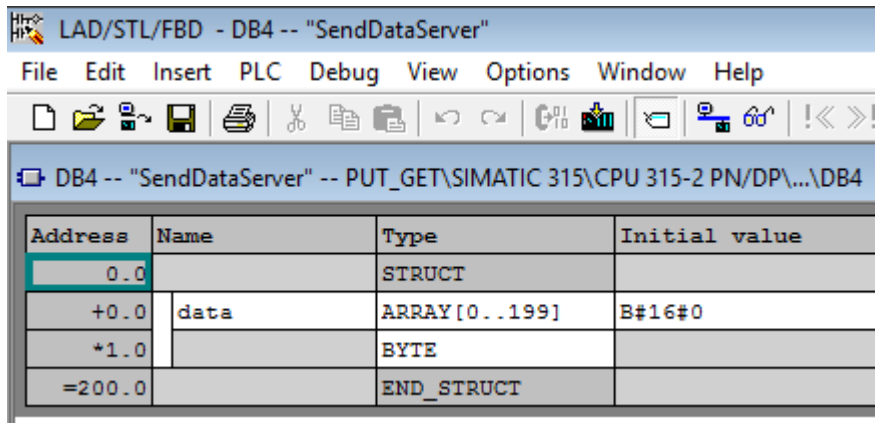
3. You enter the name and type of block that is to be created, DB4, for example.
4. Enter a symbolic name for the block, "SendDataServer", for example.
5. Apply the settings with "OK".
Block DB4 "SendDataServer" is added to the "Blocks" folder.



6. Double-click DB4 "SendDataServer" in the "Blocks" folder. The DB opens.

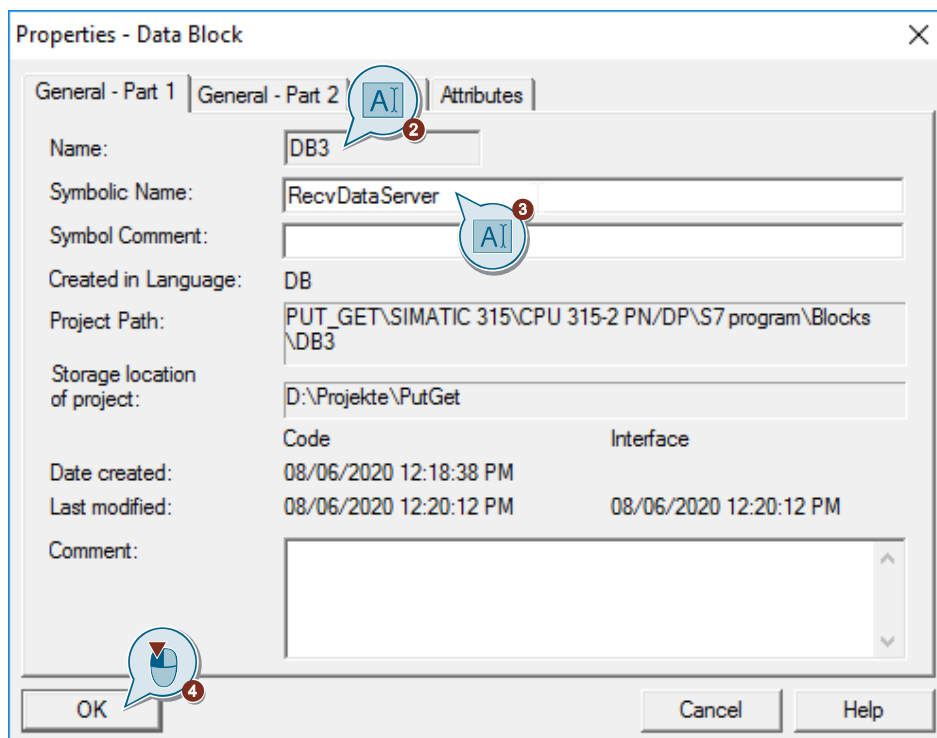


- In the DB4 "SendDataServer" you define the variable "data" of the data type "Array[0..199] of Byte".



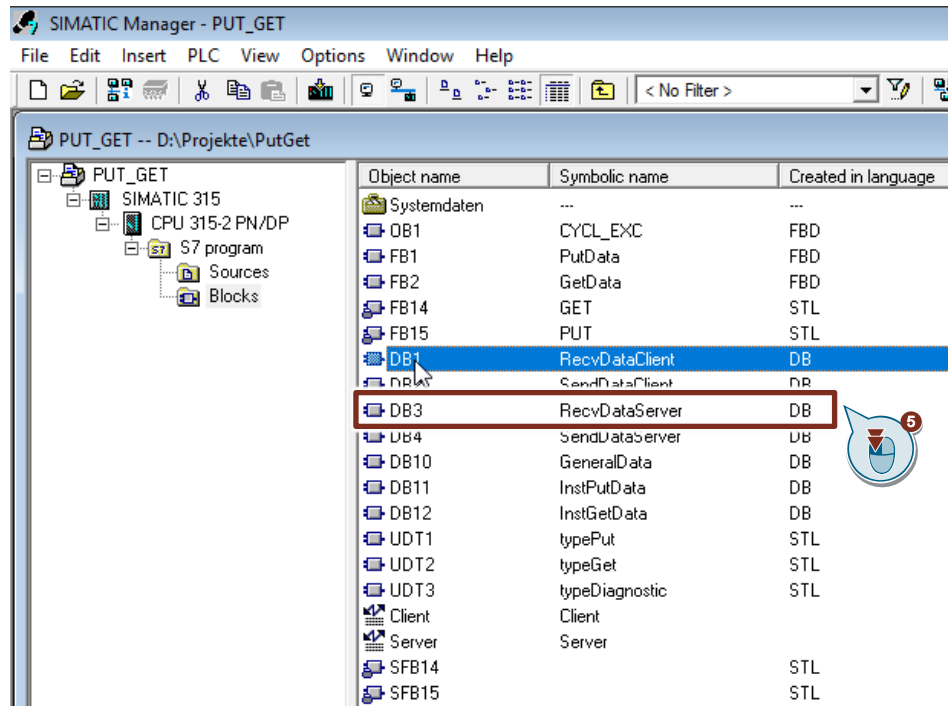
Add data block for saving the Receive data

- Add another DB for saving the Receive data.
- You enter the name and type of block that is to be created, DB3, for example.
- Enter a symbolic name for the block, "RecvDataServer", for example.
- Apply the settings with "OK".
Block DB3 "RecvDataServer" is added to the "Blocks" folder.

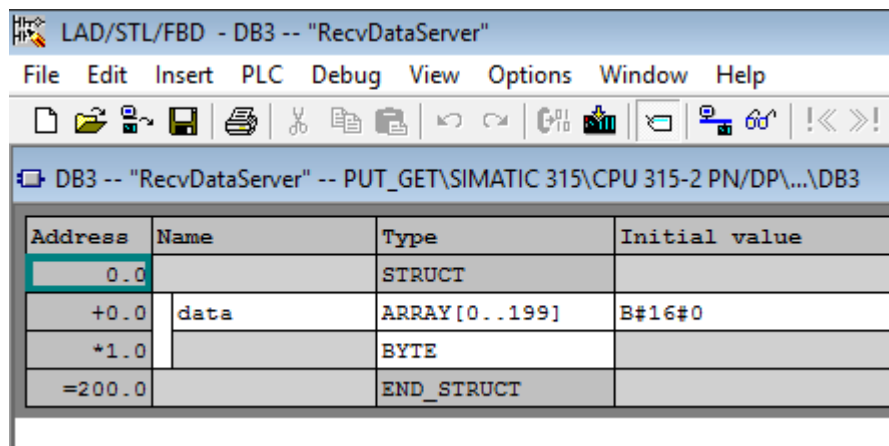


- Double-click DB3 "RecvDataServer" in the "Blocks" folder.
The DB opens.

4 Configuration of SIMATIC S7-300 in STEP 7 V5.6



- In the DB3 "RecvDataServer" you define the variable "data" of the data type "Array[0..99] of Byte".



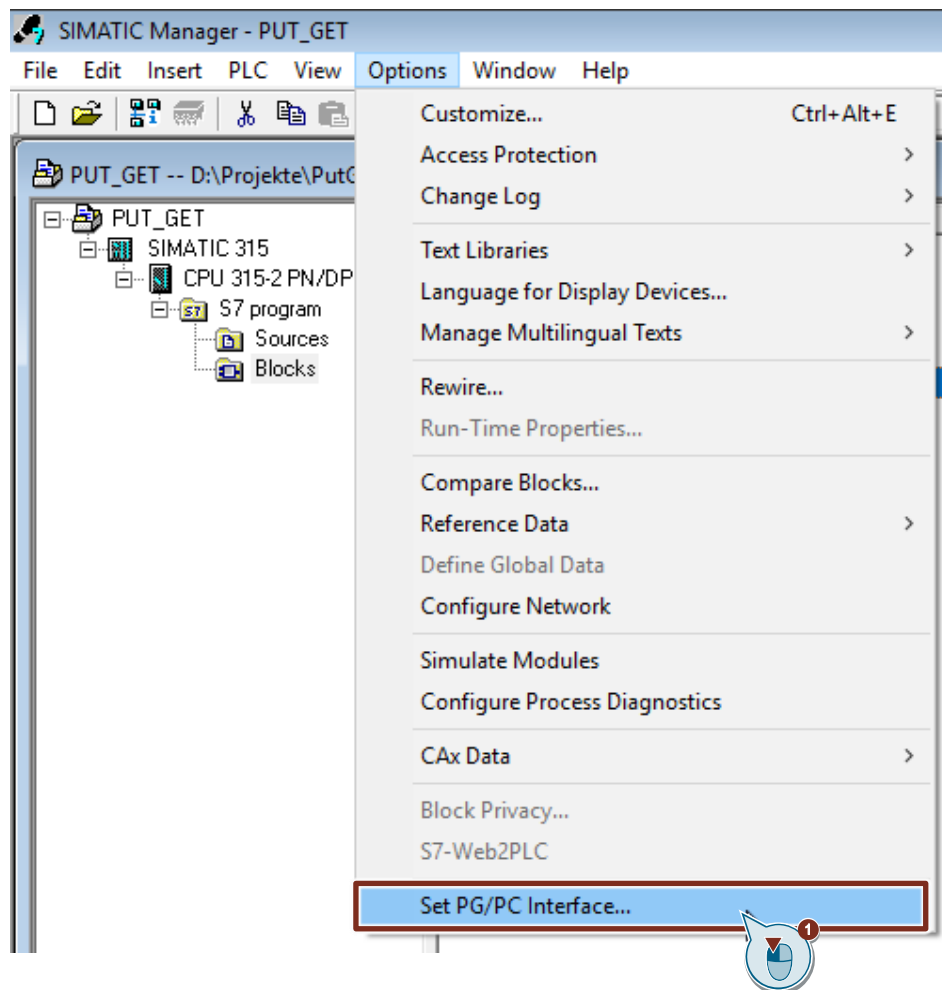
4.4 Download the Hardware Configuration and User Program

Requirements

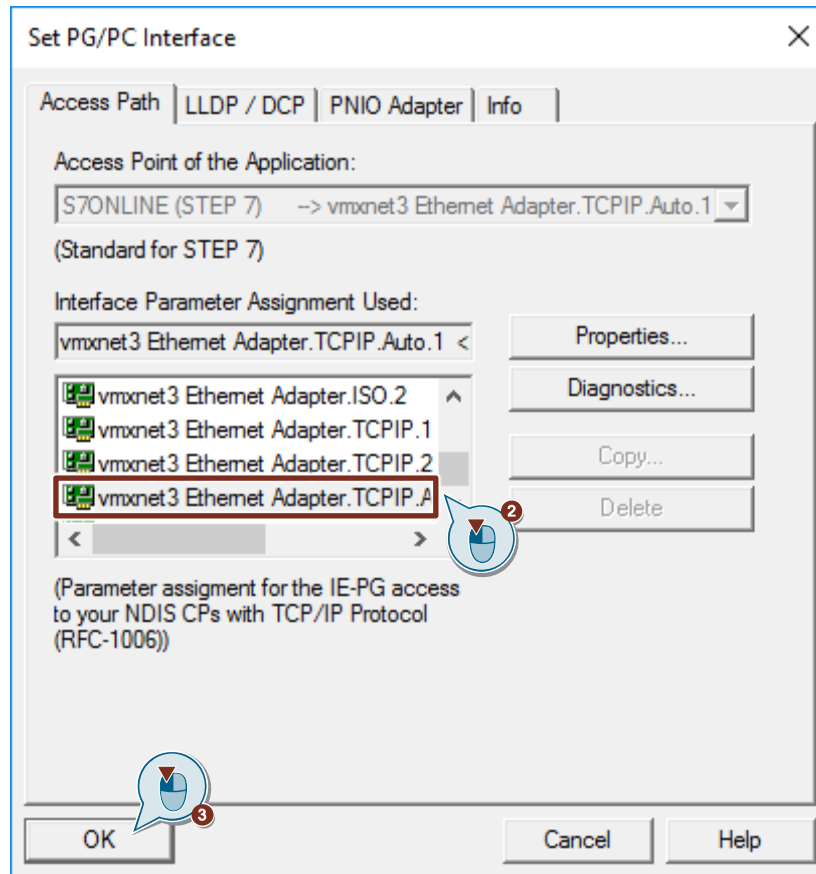
You have already assigned the configured IP address and subnet mask to the S7-300 CPU.

Instructions

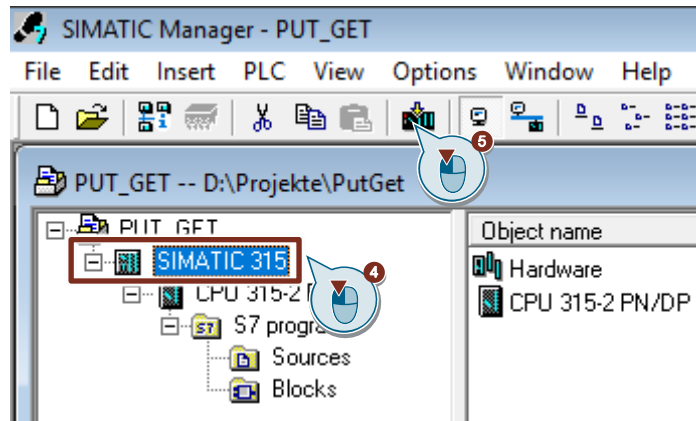
1. In the SIMATIC Manager, you select the menu "Options > Set PG/PC Interface...". The "Set PG/PC Interface" dialog opens.



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.



5 Error Handling

5.1 FB "PutData"

If an error occurs in FB "PutData", the cause of the error is written to the "status" output parameter.

Table 5-1

error	status	Description	Remedy
0	16#7000	FB "PutData" is not active.	Status information Enable FB "PutData" by setting the "execute" input to "true".
0	16#7001	FB "PutData" is initialized.	Status information
0	16#7002	Write job is running.	Status information
1	<>16#0000	Status display of the "PUT" instruction.	Detailed information is available in the STEP 7 Online Help or in the following manuals: <ul style="list-style-type: none"> • "SIMATIC STEP 7 Basic/Professional V16 and SIMATIC WinCC V16". • SIMATIC System Software for S7-300/400 System and Standard Functions - Volume 1/2

5.2 FB "GetData"

If an error occurs in FB "GetData", the cause of the error is written to the "status" output parameter.

Table 5-2

error	status	Description	Remedy
0	16#7000	FB "GetData" is not active.	Status information Enable FB "GetData" by setting the "execute" input to "true".
0	16#7001	FB "GetData" is initialized.	Status information
0	16#7002	Read job is running.	Status information
1	<>16#0000	Status display of the "GET" instruction.	Detailed information is available in the STEP 7 Online Help or in the following manuals: <ul style="list-style-type: none"> • "SIMATIC STEP 7 Basic/Professional V16 and SIMATIC WinCC V16". • SIMATIC System Software for S7-300/400 System and Standard Functions - Volume 1/2

6 Operating and Monitoring

Operating and monitoring is done via a variable table in STEP 7 V5.6 and STEP 7 V16.

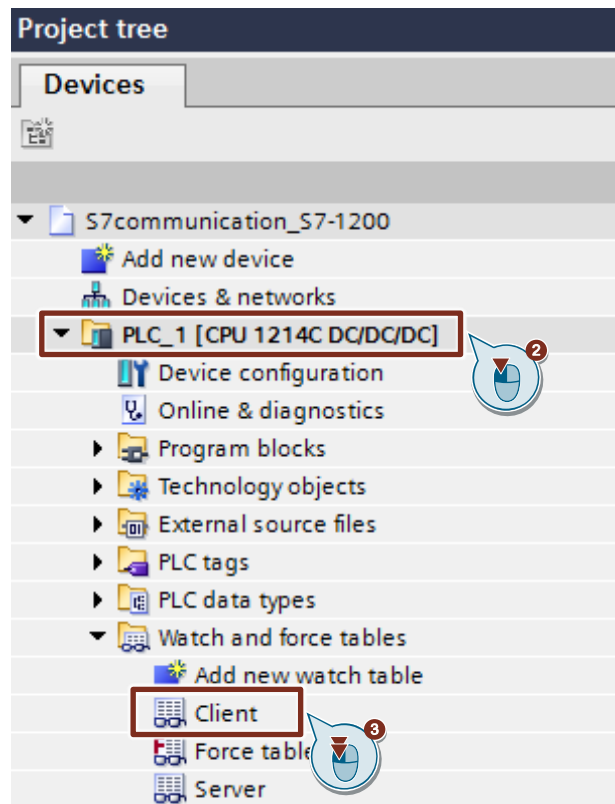
- "Client":
 - The variable table is for starting the Read job and for monitoring the data that the S7 CPU as client reads from the server.
 - The variable table is for starting the Write job and for monitoring the data that the S7 CPU writes to the server.
- "Server":
 - The variable table is for monitoring the data that the S7 CPU as server receives from the client.
 - The variable table is for controlling the data that the client reads from the S7 CPU.

6.1 SIMATIC S7-1200 Writes and Reads Data as Client

6.1.1 Write Data to the S7-300

1. In STEP 7 V16 you open the device folder of the S7-1200 CPU.
2. In the project tree you open the device folder of the S7-1200 CPU.
3. In the "Watch and force tables" folder you double-click the variable table "Client".

The variable table opens in the working area of STEP 7 V16.



4. 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).
5. In the "Modify value" column you enter the values that are to be put in the Send data area (DB2) of the S7-1200 CPU and written to the S7300 CPU.
6. Set the "GeneralData" variable to the value "True" to start the Write job.

Note

The variable "GeneralData".put.execute is reset to the value "False" in the user program as soon as the Write job is completed with "done" = true or "error" = true.

7. In the toolbar you click the "Modify all enabled values once and immediately" button.

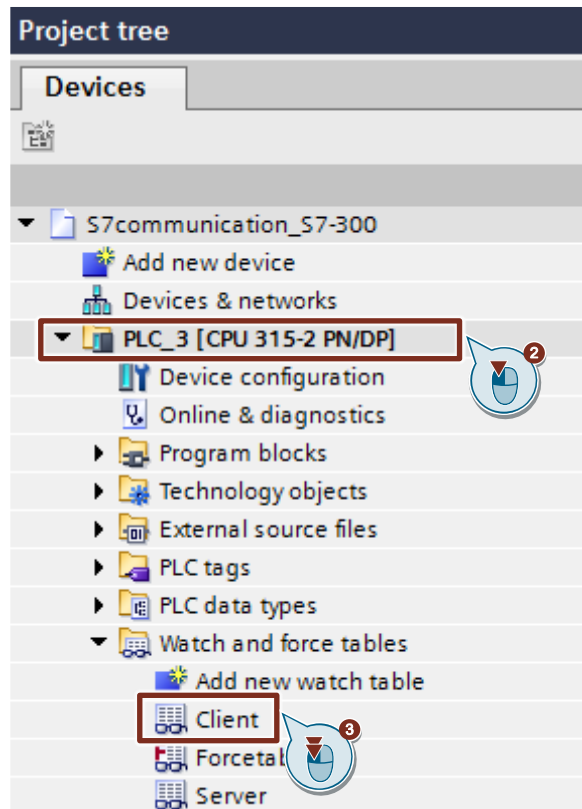
	Name	Address	Display format	Monitor value	Modify value
1	// Send Data				
2	*GeneralData*.put.execute		Bool	<input type="checkbox"/> FALSE	TRUE
3	*SendDataClient*.data[0]	%DB2.DBB0	DEC+/-	15	15
4	*SendDataClient*.data[1]	%DB2.DBB1	DEC+/-	16	16
5	*SendDataClient*.data[2]	%DB2.DBB2	DEC+/-	17	17
6	*SendDataClient*.data[3]	%DB2.DBB3	DEC+/-	18	18
7	*SendDataClient*.data[4]	%DB2.DBB4	DEC+/-	19	19
8	*SendDataClient*.data[5]	%DB2.DBB5	DEC+/-	20	20
9	*SendDataClient*.data[6]	%DB2.DBB6	DEC+/-	21	21
10	*SendDataClient*.data[7]	%DB2.DBB7	DEC+/-	22	22
11	*SendDataClient*.data[8]	%DB2.DBB8	DEC+/-	23	23
12	*SendDataClient*.data[9]	%DB2.DBB9	DEC+/-	24	24
13	*SendDataClient*.data[99]	%DB2.DBB99	DEC+/-	99	99
14	*GeneralData*.diagnostic.statusPut		Hex	16#0000	16#0000
15	// Receive Data				
16	*GeneralData*.get.execute		Bool	<input type="checkbox"/> FALSE	TRUE
17	*RecvDataClient*.data[0]	%DB1.DBB0	DEC+/-	17	
18	*RecvDataClient*.data[1]	%DB1.DBB1	DEC+/-	18	
19	*RecvDataClient*.data[2]	%DB1.DBB2	DEC+/-	19	
20	*RecvDataClient*.data[3]	%DB1.DBB3	DEC+/-	20	
21	*RecvDataClient*.data[4]	%DB1.DBB4	DEC+/-	21	
22	*RecvDataClient*.data[5]	%DB1.DBB5	DEC+/-	22	
23	*RecvDataClient*.data[6]	%DB1.DBB6	DEC+/-	23	
24	*RecvDataClient*.data[7]	%DB1.DBB7	DEC+/-	24	
25	*RecvDataClient*.data[8]	%DB1.DBB8	DEC+/-	25	
26	*RecvDataClient*.data[9]	%DB1.DBB9	DEC+/-	32	
27	*RecvDataClient*.data[99]	%DB1.DBB99	DEC+/-	33	
28	*GeneralData*.diagnostic.statusGet		Hex	16#0000	
29		<Add new>			

Result

The modified values are put in the Send data area (DB1) of the S7-1200 CPU and written to the S7-300 CPU.

6.1.2 Read Data from the S7-300

1. In STEP 7 V16 you open the S7-1200 CPU project.
2. In the project tree you open the device folder of the S7-1200 CPU.
3. In the "Watch and force tables" folder you double-click the variable table "Client".
The variable table opens in the working area of STEP 7 V16.



4. Click the "Monitor all" button in the toolbar. The "Status value" column displays the current values stored in the Send data area (DB2) and Receive data area (DB1).
5. Set the "GeneralData" variable to the value "True" to start the Read job.
Note
The variable "GeneralData".get.execute is reset to the value "False" in the user program as soon as the Read job is completed with "done" = true or "error" = true.
6. In the toolbar you click the "Modify all enabled values once and immediately" button.

S7communication_S7-1200 PLC_1 [CPU 1214C DC/DC/DC] Watch and force tables Client

	Name	Address	Display format	Monitor value	Modify value
1	// Send Data				
2	"GeneralData".put.execute		Bool	<input type="checkbox"/> FALSE	TRUE
3	"SendDataClient".data[0]	%DB2.DBB0	DEC+/-	15	15
4	"SendDataClient".data[1]	%DB2.DBB1	DEC+/-	16	16
5	"SendDataClient".data[2]	%DB2.DBB2	DEC+/-	17	17
6	"SendDataClient".data[3]	%DB2.DBB3	DEC+/-	18	18
7	"SendDataClient".data[4]	%DB2.DBB4	DEC+/-	19	19
8	"SendDataClient".data[5]	%DB2.DBB5	DEC+/-	20	20
9	"SendDataClient".data[6]	%DB2.DBB6	DEC+/-	21	21
10	"SendDataClient".data[7]	%DB2.DBB7	DEC+/-	22	22
11	"SendDataClient".data[8]	%DB2.DBB8	DEC+/-	23	23
12	"SendDataClient".data[9]	%DB2.DBB9	DEC+/-	24	24
13	"SendDataClient".data[99]	%DB2.DBB99	DEC+/-	99	99
14	"GeneralData".diagnostic.statusPut		Hex	16#0000	16#0000
15	// Receive Data				
16	"GeneralData".get.execute		Bool	<input type="checkbox"/> FALSE	TRUE
17	"RecvDataClient".data[0]	%DB1.DBB0	DEC+/-	17	
18	"RecvDataClient".data[1]	%DB1.DBB1	DEC+/-	18	
19	"RecvDataClient".data[2]	%DB1.DBB2	DEC+/-	19	
20	"RecvDataClient".data[3]	%DB1.DBB3	DEC+/-	20	
21	"RecvDataClient".data[4]	%DB1.DBB4	DEC+/-	21	
22	"RecvDataClient".data[5]	%DB1.DBB5	DEC+/-	22	
23	"RecvDataClient".data[6]	%DB1.DBB6	DEC+/-	23	
24	"RecvDataClient".data[7]	%DB1.DBB7	DEC+/-	24	
25	"RecvDataClient".data[8]	%DB1.DBB8	DEC+/-	25	
26	"RecvDataClient".data[9]	%DB1.DBB9	DEC+/-	32	
27	"RecvDataClient".data[99]	%DB1.DBB99	DEC+/-	33	
28	"GeneralData".diagnostic.statusGet		Hex	16#0000	
29		<Add new>			

© Siemens AG 2020. All rights reserved

Result

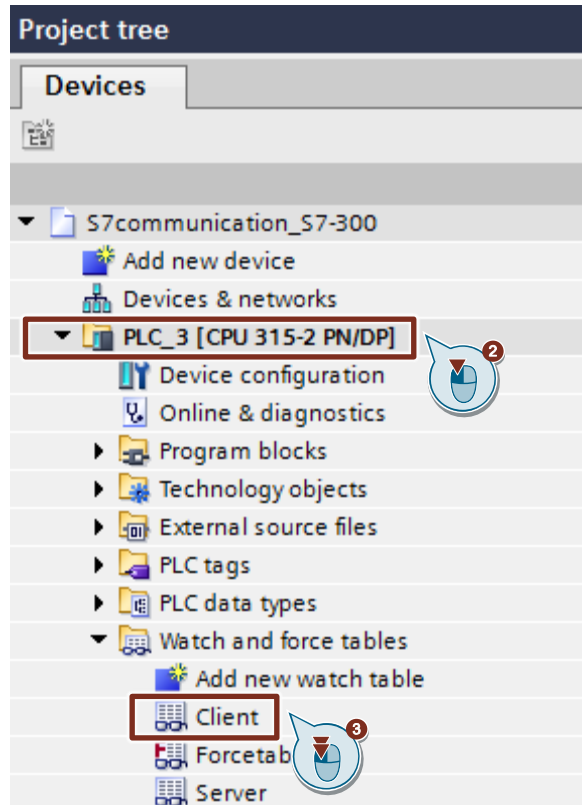
The "Monitor value" column displays the data read from the S7-300 and stored in the Receive data area (DB1).

6.2 SIMATIC S7-300 Writes and Reads Data as Client (STEP 7 V16)

6.2.1 Write Data to the S7-1200

1. In STEP 7 V16 you open the S7-300 CPU project.
2. In the project tree you open the device folder of the S7-300 CPU.
3. In the "Watch and force tables" folder you double-click the variable table "Client".

The variable table opens in the working area of STEP 7 V16.



4. 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).
5. In the "Modify value" column you enter the values that are to be put in the Send data area (DB2) of the S7-300 CPU and written to the S7-1200 CPU.
6. Set the "GeneralData" variable to the value "True" to start the Write job.
Note
The variable "GeneralData".put.execute is reset to the value "False" in the user program as soon as the Write job is completed with "done" = true or "error" = true.
7. In the toolbar you click the "Modify all enabled values once and immediately" button.

S7communication_S7-300...C_3 [CPU 315-2 PN/DP] ▶ Watch and force tables ▶ Client

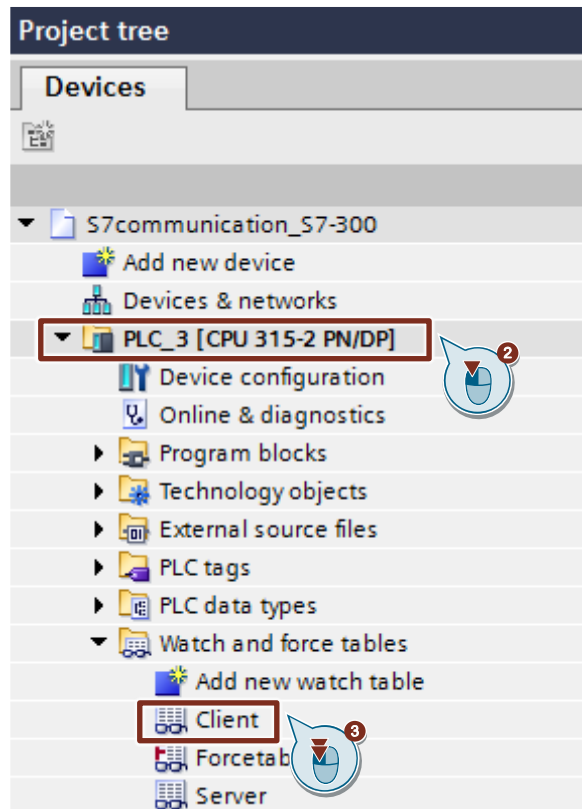
	Name	Address	Display format	Monitor value	Modify value
1	// send data				
2	"GeneralData".put.execute	%DB10.DBX0.0	Bool	<input type="checkbox"/> FALSE	TRUE
3	"SendDataClient".data[0]	%DB2.DBB0	DEC+/-	17	17
4	"SendDataClient".data[1]	%DB2.DBB1	DEC+/-	18	18
5	"SendDataClient".data[2]	%DB2.DBB2	DEC+/-	19	19
6	"SendDataClient".data[3]	%DB2.DBB3	DEC+/-	20	20
7	"SendDataClient".data[4]	%DB2.DBB4	DEC+/-	21	21
8	"SendDataClient".data[5]	%DB2.DBB5	DEC+/-	33	33
9	"SendDataClient".data[6]	%DB2.DBB6	DEC+/-	34	34
10	"SendDataClient".data[7]	%DB2.DBB7	DEC+/-	35	35
11	"SendDataClient".data[8]	%DB2.DBB8	DEC+/-	36	36
12	"SendDataClient".data[9]	%DB2.DBB9	DEC+/-	37	37
13	"SendDataClient".data[99]	%DB2.DBB99	DEC+/-	49	49
14	"GeneralData".diagnostic.statusPut	%DB10.DBW16	Hex	16#0000	
15					
16	// receive data				
17	"GeneralData".get.execute	%DB10.DBX8.0	Bool	<input type="checkbox"/> FALSE	TRUE
18	"RecvDataClient".data[0]	%DB1.DBB0	DEC+/-	26	
19	"RecvDataClient".data[1]	%DB1.DBB1	DEC+/-	9	
20	"RecvDataClient".data[2]	%DB1.DBB2	DEC+/-	19	
21	"RecvDataClient".data[3]	%DB1.DBB3	DEC+/-	81	
22	"RecvDataClient".data[4]	%DB1.DBB4	DEC+/-	24	
23	"RecvDataClient".data[5]	%DB1.DBB5	DEC+/-	10	
24	"RecvDataClient".data[6]	%DB1.DBB6	DEC+/-	20	
25	"RecvDataClient".data[7]	%DB1.DBB7	DEC+/-	18	
26	"RecvDataClient".data[8]	%DB1.DBB8	DEC+/-	3	
27	"RecvDataClient".data[9]	%DB1.DBB9	DEC+/-	11	
28	"RecvDataClient".data[99]	%DB1.DBB99	DEC+/-	60	
29	"GeneralData".diagnostic.statusGet	%DB10.DBW18	Hex	16#0000	
30		<Add new>			

Result

The modified values are put in the Send data area (DB1) of the S7-300 CPU and written to the S7-1200 CPU.

6.2.2 Read Data from the S7-1200

1. In STEP 7 V16 you open the S7-300 CPU project.
2. In the project tree you open the device folder of the S7-300 CPU.
3. In the "Watch and force tables" folder you double-click the variable table "Client".
The variable table opens in the working area of STEP 7 V16.



4. Click the "Monitor all" button in the toolbar. The "Status value" column displays the current values stored in the Send data area (DB2) and Receive data area (DB1).
5. Set the "GeneralData" variable to the value "True" to start the Read job.
Note
The variable "GeneralData".get.execute is reset to the value "False" in the user program as soon as the Read job is completed with "done" = true or "error" = true.
6. In the toolbar you click the "Modify all enabled values once and immediately" button.

S7communication_S7-300...C_3 [CPU 315-2 PN/DP] ▶ Watch and force tables ▶ Client

	Name	Address	Display format	Monitor value	Modify value
1	// send data				
2	"GeneralData".put.execute	%DB10.DBX0.0	Bool	<input type="checkbox"/> FALSE	TRUE
3	"SendDataClient".data[0]	%DB2.DBB0	DEC+/-	17	17
4	"SendDataClient".data[1]	%DB2.DBB1	DEC+/-	18	18
5	"SendDataClient".data[2]	%DB2.DBB2	DEC+/-	19	19
6	"SendDataClient".data[3]	%DB2.DBB3	DEC+/-	20	20
7	"SendDataClient".data[4]	%DB2.DBB4	DEC+/-	21	21
8	"SendDataClient".data[5]	%DB2.DBB5	DEC+/-	33	33
9	"SendDataClient".data[6]	%DB2.DBB6	DEC+/-	34	34
10	"SendDataClient".data[7]	%DB2.DBB7	DEC+/-	35	35
11	"SendDataClient".data[8]	%DB2.DBB8	DEC+/-	36	36
12	"SendDataClient".data[9]	%DB2.DBB9	DEC+/-	37	37
13	"SendDataClient".data[99]	%DB2.DBB99	DEC+/-	49	49
14	"GeneralData".diagnostic.statusPut	%DB10.DBW16	Hex	16#0000	
15					
16	// receive data				
17	"GeneralData".get.execute	%DB10.DBX8.0	Bool	<input type="checkbox"/> FALSE	TRUE
18	"RecvDataClient".data[0]	%DB1.DBB0	DEC+/-	26	
19	"RecvDataClient".data[1]	%DB1.DBB1	DEC+/-	9	
20	"RecvDataClient".data[2]	%DB1.DBB2	DEC+/-	19	
21	"RecvDataClient".data[3]	%DB1.DBB3	DEC+/-	81	
22	"RecvDataClient".data[4]	%DB1.DBB4	DEC+/-	24	
23	"RecvDataClient".data[5]	%DB1.DBB5	DEC+/-	10	
24	"RecvDataClient".data[6]	%DB1.DBB6	DEC+/-	20	
25	"RecvDataClient".data[7]	%DB1.DBB7	DEC+/-	18	
26	"RecvDataClient".data[8]	%DB1.DBB8	DEC+/-	3	
27	"RecvDataClient".data[9]	%DB1.DBB9	DEC+/-	11	
28	"RecvDataClient".data[99]	%DB1.DBB99	DEC+/-	60	
29	"GeneralData".diagnostic.statusGet	%DB10.DBW18	Hex	16#0000	
30		<Add new>			

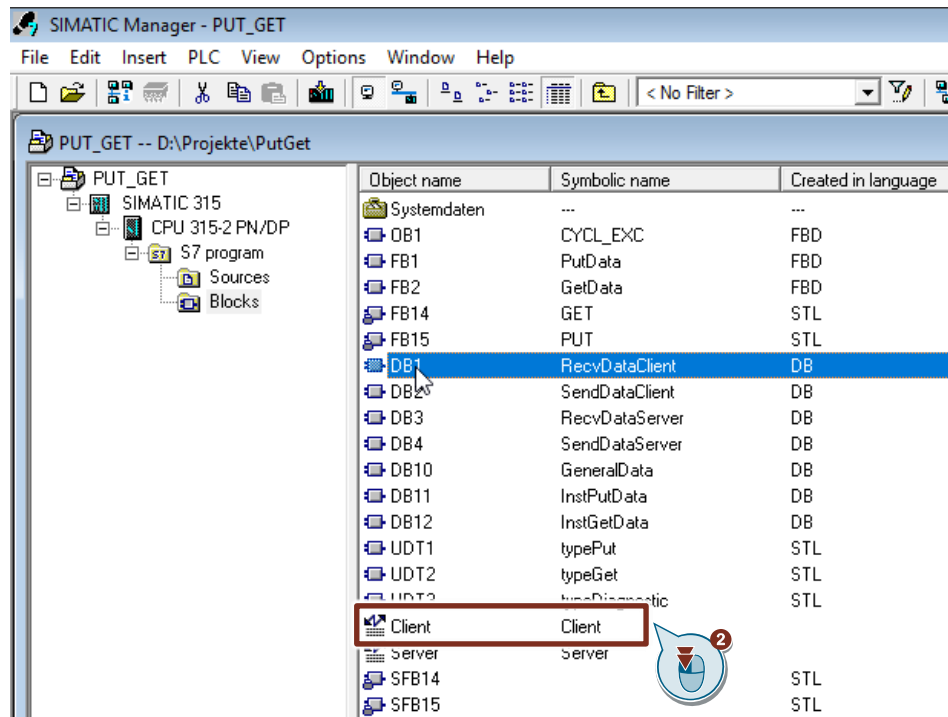
Result

The "Monitor value" column displays the data read from the S7-300 and stored in the Receive data area (DB1).

6.3 SIMATIC S7-300 Writes and Reads Data as Client (STEP 7 V5.6)

6.3.1 Write Data to the S7-1200 CPU

1. In STEP 7 V5.6 you open the S7-300 CPU project.
2. In the "Blocks" folder you double-click the variable table "Client". The variable table opens.



3. Click the "Monitor variable" button in the toolbar. The "Status value" column displays the current values stored in the Send data area (DB2) and Receive data area (DB1).
4. In the "Modify value" column you enter the values that are to be put in the Send data area (DB2) of the S7-300 CPU and written to the S7-1200 CPU.
5. Set the "GeneralData" variable to the value "True" to start the Write job.

Note

The variable "GeneralData".put.execute is reset to the value "False" in the user program as soon as the Write job is completed with "done" = true or "error" = true.

6. Click the "Activate modify values" button in the toolbar.

	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	DB10.DBX 6.0	"GeneraData".put.execute	BOOL	false	true
2	//Send data				
3	DB2.DBB 0	"SendDataClient".data[0]	DEZ	26	26
4	DB2.DBB 1	"SendDataClient".data[1]	DEZ	9	9
5	DB2.DBB 2	"SendDataClient".data[2]	DEZ	19	19
6	DB2.DBB 3	"SendDataClient".data[3]	DEZ	81	81
7	DB2.DBB 4	"SendDataClient".data[4]	DEZ	16	16
8	DB2.DBB 5	"SendDataClient".data[5]	DEZ	2	2
9	DB2.DBB 6	"SendDataClient".data[6]	DEZ	20	20
10	DB2.DBB 7	"SendDataClient".data[7]	DEZ	13	13
11	DB2.DBB 8	"SendDataClient".data[8]	DEZ	3	3
12	DB2.DBB 9	"SendDataClient".data[9]	DEZ	10	10
13	DB2.DBB 99	"SendDataClient".data[99]	DEZ	1	1
14	DB10.DBW 12	"GeneraData".diagnostic.statusPut	HEX	W#16#0000	
15					
16	DB10.DBX 0.0	"GeneraData".get.execute	BOOL	false	true
17	//Receive Data				
18	DB1.DBB 0	"RecvDataClient".data[0]	DEZ	11	
19	DB1.DBB 1	"RecvDataClient".data[1]	DEZ	12	
20	DB1.DBB 2	"RecvDataClient".data[2]	DEZ	13	
21	DB1.DBB 3	"RecvDataClient".data[3]	DEZ	14	
22	DB1.DBB 4	"RecvDataClient".data[4]	DEZ	15	
23	DB1.DBB 5	"RecvDataClient".data[5]	DEZ	16	
24	DB1.DBB 6	"RecvDataClient".data[6]	DEZ	17	
25	DB1.DBB 7	"RecvDataClient".data[7]	DEZ	18	
26	DB1.DBB 8	"RecvDataClient".data[8]	DEZ	19	
27	DB1.DBB 9	"RecvDataClient".data[9]	DEZ	20	
28	DB1.DBB 99	"RecvDataClient".data[99]	DEZ	21	
29	DB10.DBW 14	"GeneraData".diagnostic.statusGet	HEX	W#16#0000	
30					

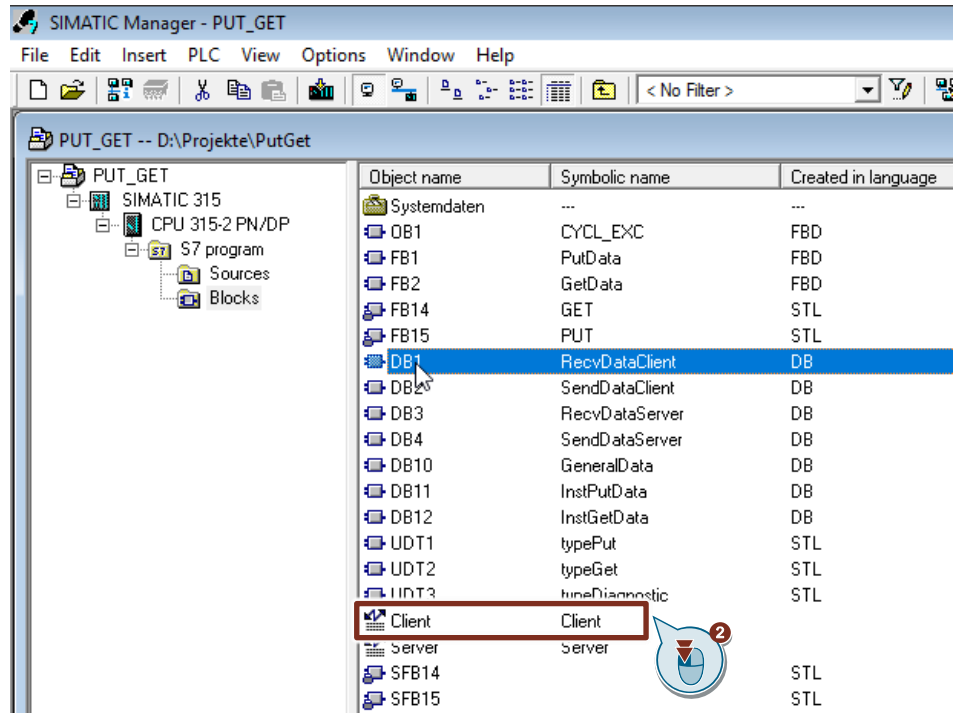
© Siemens AG 2020. All rights reserved

Result

The modified values are put in the Send data area (DB1) of the S7-300 CPU and written to the S7-1200 CPU.

6.3.2 Read Data from the S7-1200 CPU

1. In STEP 7 V5.6 you open the S7-300 CPU project.
2. In the "Blocks" folder you double-click the variable table "Client". The variable table opens.



3. Click the "Monitor variable" button in the toolbar. The "Status value" column displays the current values stored in the Send data area (DB2) and Receive data area (DB1).
4. Set the "GeneralData" variable to the value "True" to start the Read job.

Note

The variable "GeneralData".get.execute is reset to the value "False" in the user program as soon as the Read job is completed with "done" = true or "error" = true.

- Click the "Activate modify values" button in the toolbar.

	Operand	Symbol	Anzeigeformat	Status	Steuerwert
1	DB10.DBX 6.0	"GeneralData".put.execute	BOOL	false	true
2	//Send data				
3	DB2.DBB 0	"SendDataClient".data[0]	DEZ	26	26
4	DB2.DBB 1	"SendDataClient".data[1]	DEZ	9	9
5	DB2.DBB 2	"SendDataClient".data[2]	DEZ	19	19
6	DB2.DBB 3	"SendDataClient".data[3]	DEZ	81	81
7	DB2.DBB 4	"SendDataClient".data[4]	DEZ	16	16
8	DB2.DBB 5	"SendDataClient".data[5]	DEZ	2	2
9	DB2.DBB 6	"SendDataClient".data[6]	DEZ	20	20
10	DB2.DBB 7	"SendDataClient".data[7]	DEZ	13	13
11	DB2.DBB 8	"SendDataClient".data[8]	DEZ	3	3
12	DB2.DBB 9	"SendDataClient".data[9]	DEZ	10	10
13	DB2.DBB 99	"SendDataClient".data[99]	DEZ	1	1
14	DB10.DBW 12	"GeneralData".diagnostic.statusPut	HEX	W#16#0000	
15					
16	DB10.DBX 0.0	"GeneralData".get.execute	BOOL	false	true
17	//Receive Data				
18	DB1.DBB 0	"RecvDataClient".data[0]	DEZ	11	
19	DB1.DBB 1	"RecvDataClient".data[1]	DEZ	12	
20	DB1.DBB 2	"RecvDataClient".data[2]	DEZ	13	
21	DB1.DBB 3	"RecvDataClient".data[3]	DEZ	14	
22	DB1.DBB 4	"RecvDataClient".data[4]	DEZ	15	
23	DB1.DBB 5	"RecvDataClient".data[5]	DEZ	16	
24	DB1.DBB 6	"RecvDataClient".data[6]	DEZ	17	
25	DB1.DBB 7	"RecvDataClient".data[7]	DEZ	18	
26	DB1.DBB 8	"RecvDataClient".data[8]	DEZ	19	
27	DB1.DBB 9	"RecvDataClient".data[9]	DEZ	20	
28	DB1.DBB 99	"RecvDataClient".data[99]	DEZ	21	
29	DB10.DBW 14	"GeneralData".diagnostic.statusGet	HEX	W#16#0000	
30					

Result

The "Status value" column displays the data read from the S7-1200 and stored in the Receive data area (DB1).