

**SIEMENS**

*Ingenuity for life*



## S7 Communication with PUT/GET

S7-300 CPUs

<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](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

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
<b>2</b>	<b>Configuration</b> .....	<b>5</b>
2.1	Configuration of the Hardware .....	5
2.1.1	Open the Hardware Configuration .....	5
2.1.2	Define IP Address and Assign Subnet to the PROFINET interface of the CPU .....	6
2.1.3	Define IP Address and Assign Subnet to the PROFINET interface of the CP .....	8
2.2	Configuration of the S7 Connection .....	10
2.2.1	Add an S7 Connection .....	10
2.2.2	Define connection parameters .....	12
2.2.3	Load the S7 Connection .....	14
<b>3</b>	<b>User Program of the S7-300 CPU (Active S7 CPU)</b> .....	<b>15</b>
3.1	Overview .....	15
3.2	Create Send and Receive Data Areas .....	16
3.2.1	DB1 "RecvData" .....	16
3.2.2	DB2 "SendData" .....	16
3.3	Create Variables for Parameters of Function Blocks "PutDataSequence" and "GetDataSequence" .....	16
3.3.1	UDT1 "typePut" .....	16
3.3.2	UDT2 "typeGet" .....	17
3.3.3	UDT3 "typeDiagnostic" .....	17
3.4	FB1 "PutDataSequence" .....	18
3.4.1	Parameters of FB1 "PutDataSequence" .....	19
3.5	FB2 "GetDataSequence" .....	21
3.5.1	Parameters of FB2 "GetDataSequence" .....	22
3.6	Function .....	24
3.6.1	Overview .....	24
3.6.2	Function details .....	25
3.7	Error Handling .....	27
3.7.1	FB1 "PutDataSequence" .....	27
3.7.2	FB2 "GetDataSequence" .....	27
<b>4</b>	<b>User Program of the Passive S7 CPU</b> .....	<b>28</b>
4.1	Create Send and Receive Data Areas .....	28
4.1.1	DB1 "RecvData" .....	28
4.1.2	DB2 "SendData" .....	29
4.2	Permit Access with PUT/GET Communication from Remote Partner .....	30
4.3	Enable S7 communication to the CPU .....	30
<b>5</b>	<b>Information</b> .....	<b>31</b>
5.1	FB15 "PUT" .....	31
5.2	FB14 "GET" .....	31

# 1 Introduction

You can use the S7 Communication, for example, for data transfer via the integrated PROFINET interface and Industrial Ethernet interface of the S7-300 CPUs.

The following function blocks are available for S7 Communication:

- FB15 "PUT" for sending data
- FB14 "GET" for receiving data

If you are using the integrated PROFINET and Industrial Ethernet 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 "CPU\_300" family.

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.

This sample program shows how to call the function blocks FB15 "PUT" and FB14 "GET" sequentially in the user program of the SIMATIC S7-300 CPU in order to transfer more than 160 bytes of data via an S7 connection.

Only one job at a time can be triggered by the function blocks FB15 "GET" and FB14 "PUT" via a configured S7 connection.

It is not possible to trigger multiple jobs simultaneously by the function blocks FB15 "PUT" and FB14 "GET" via an S7 connection.

The function blocks FB15 "PUT" and FB14 "GET" can only be called sequentially, that is one after the other. A job is triggered only when the previous job is completed.

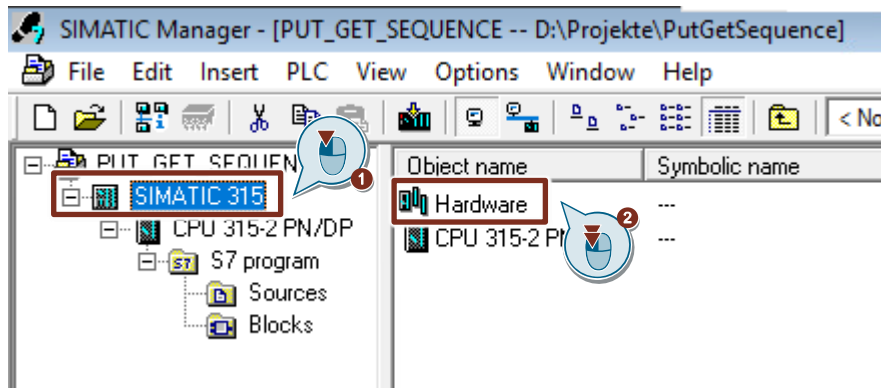
In this sample program the function blocks FB14 "GET" and FB15 "PUT" are called sequentially so that the S7-300 CPU can send and receive up to 400 bytes of data via a configured S7 connection.

## 2 Configuration

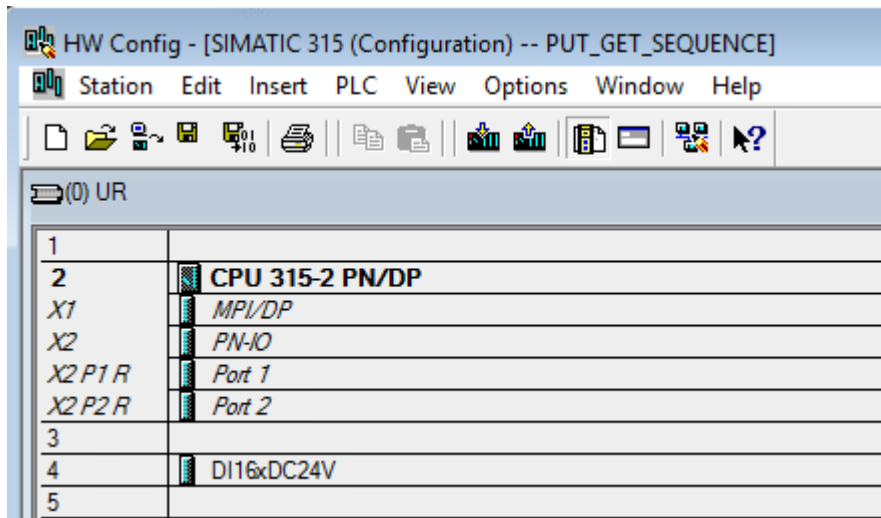
### 2.1 Configuration of the Hardware

#### 2.1.1 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.

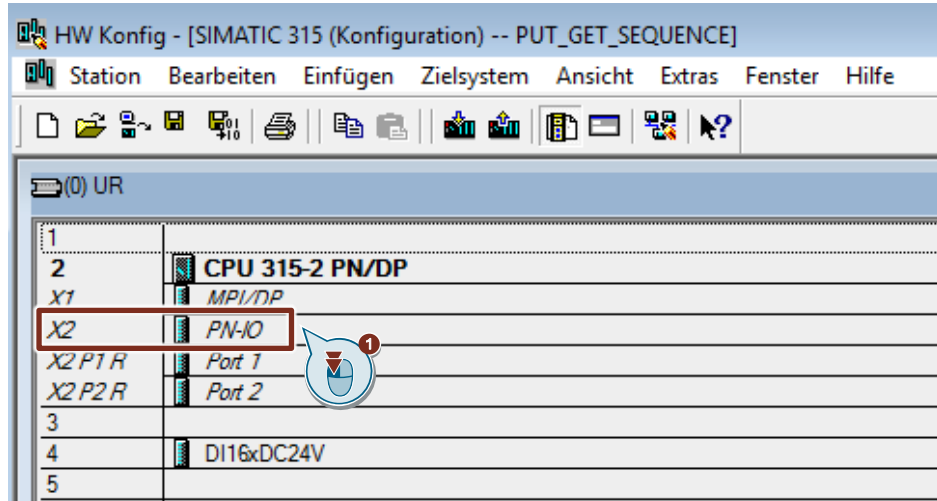


3. Drag and drop the relevant modules like Power Supply, CPU etc. from the hardware catalog into the S7-300 profile channel.

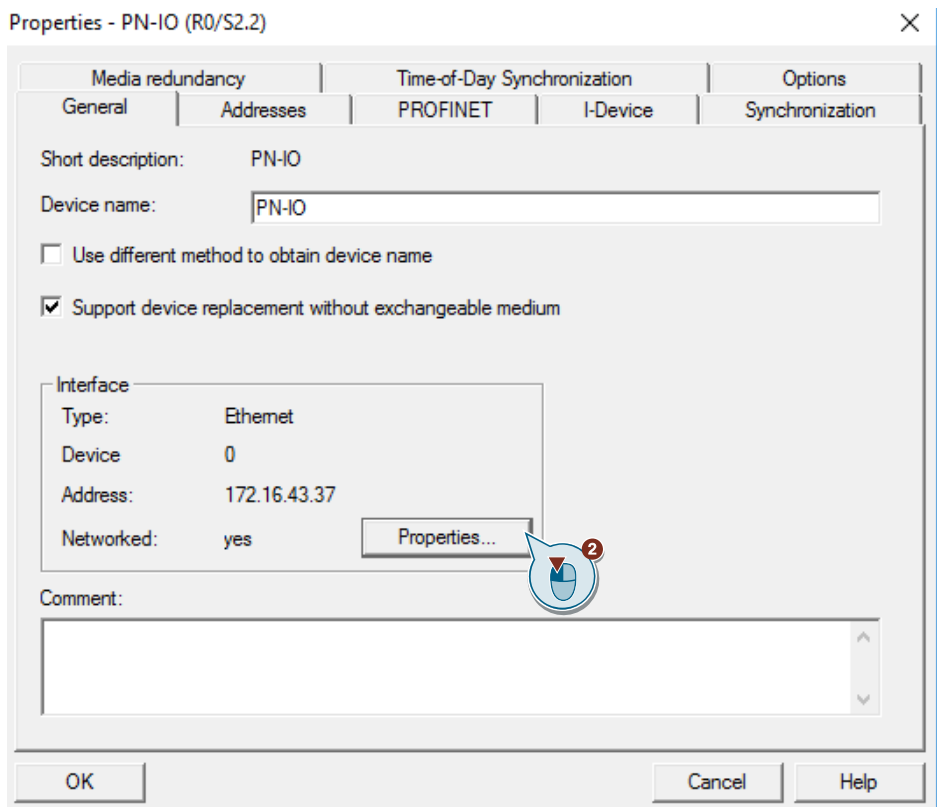


### 2.1.2 Define IP Address and Assign Subnet to the PROFINET interface of the CPU

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

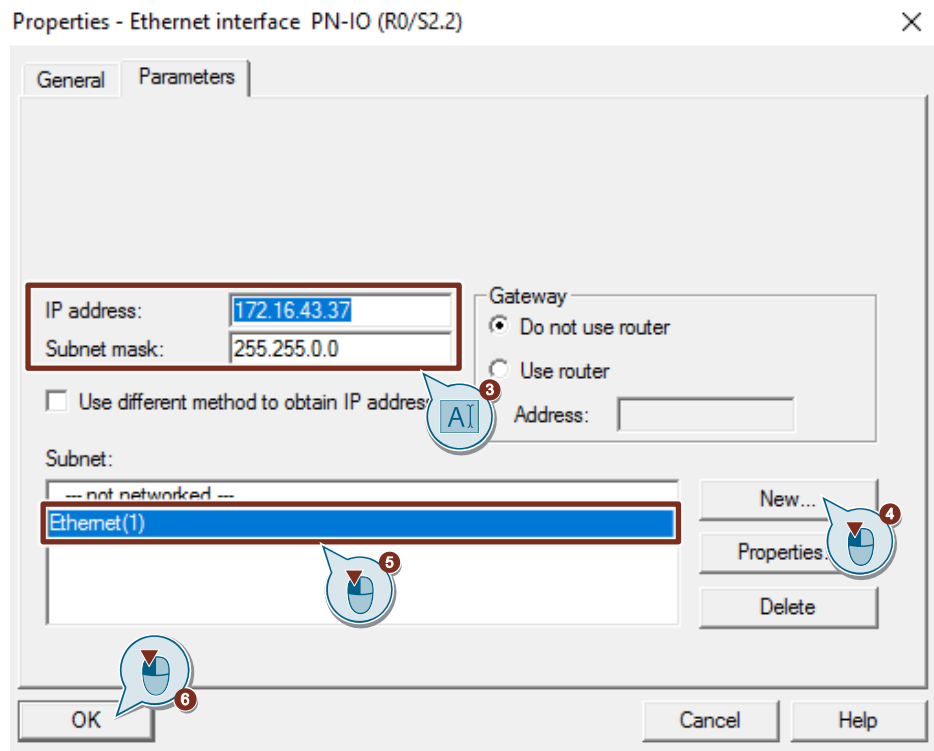


2. Click the "Properties" button to change the IP address and subnet mask and assign the subnet.



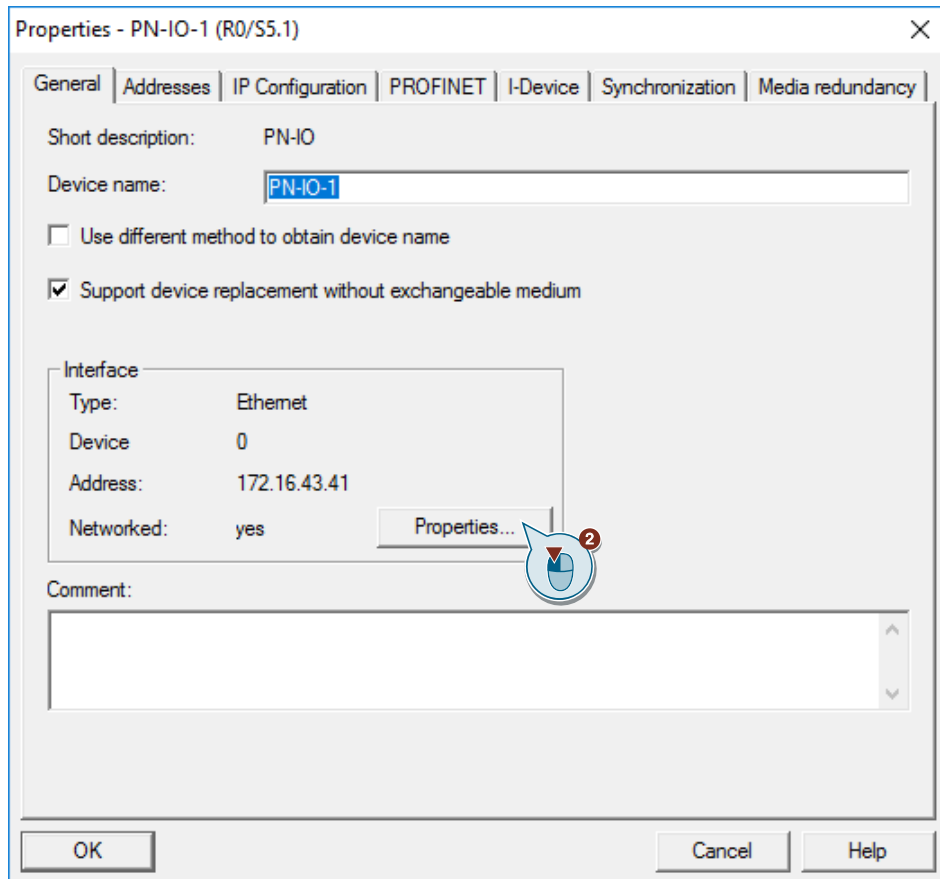
3. Enter the IP address and subnet mask, for example:
  - IP address: 172.16.43.37
  - Subnet mask: 255.255.0.0

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.



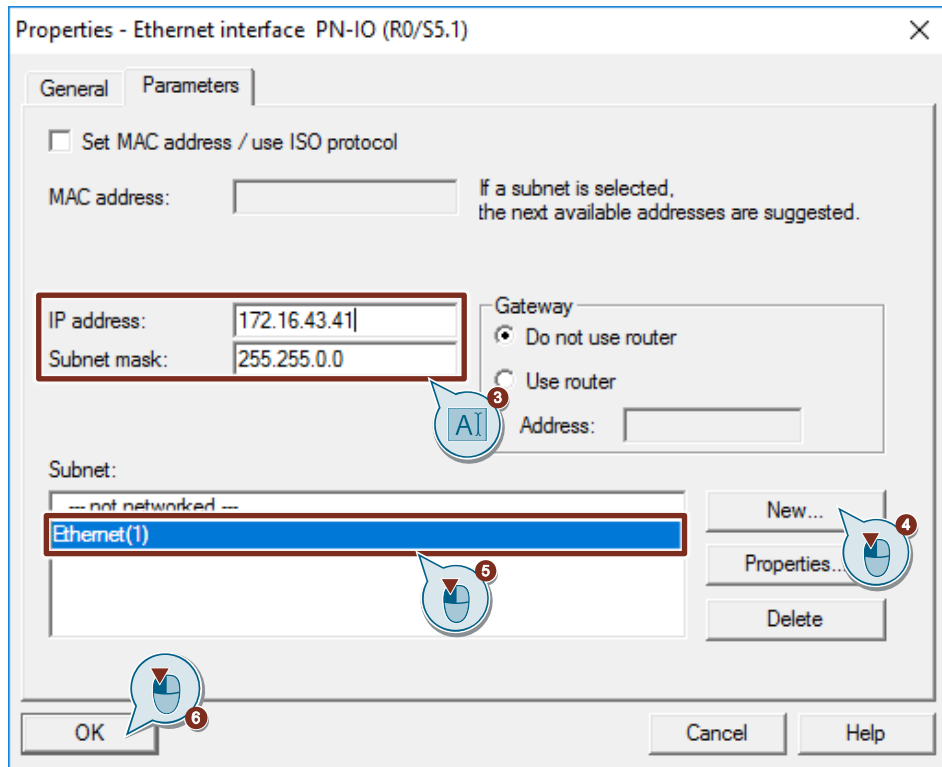
### 2.1.3 Define IP Address and Assign Subnet to the PROFINET interface of the CP

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.

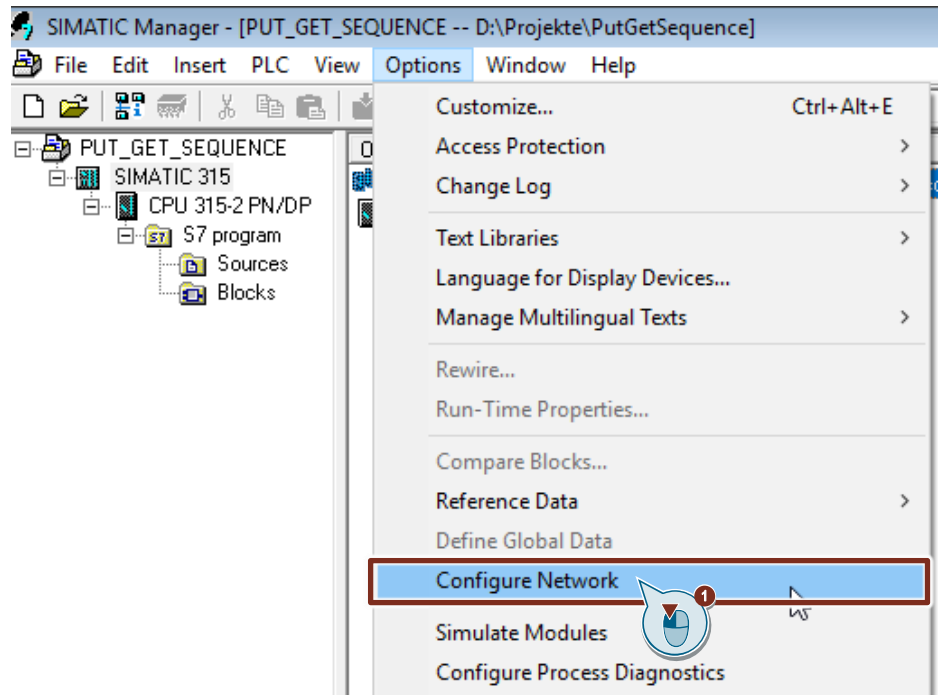




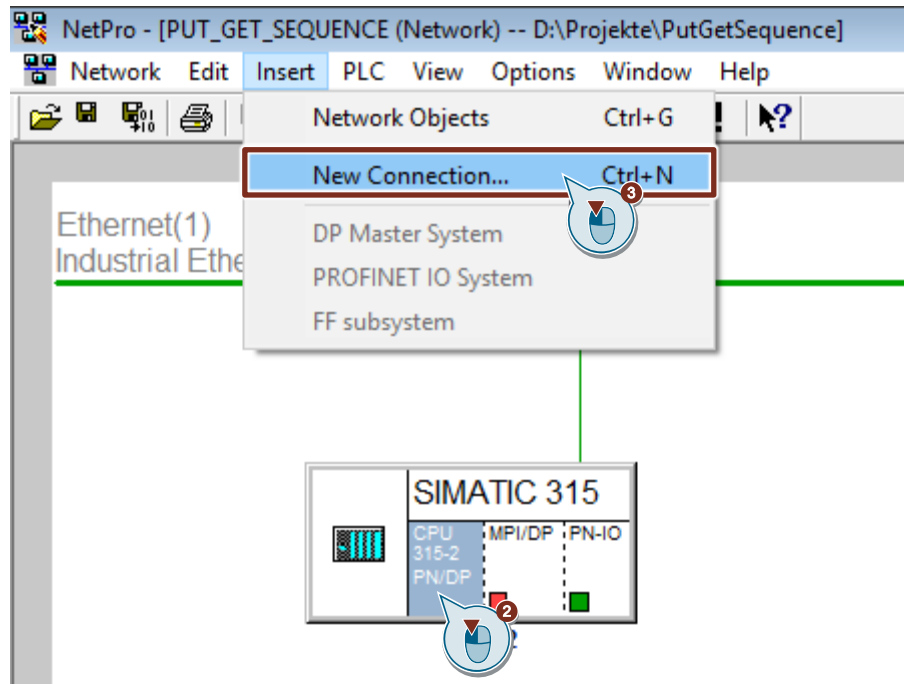
## 2.2 Configuration of the S7 Connection

### 2.2.1 Add an S7 Connection

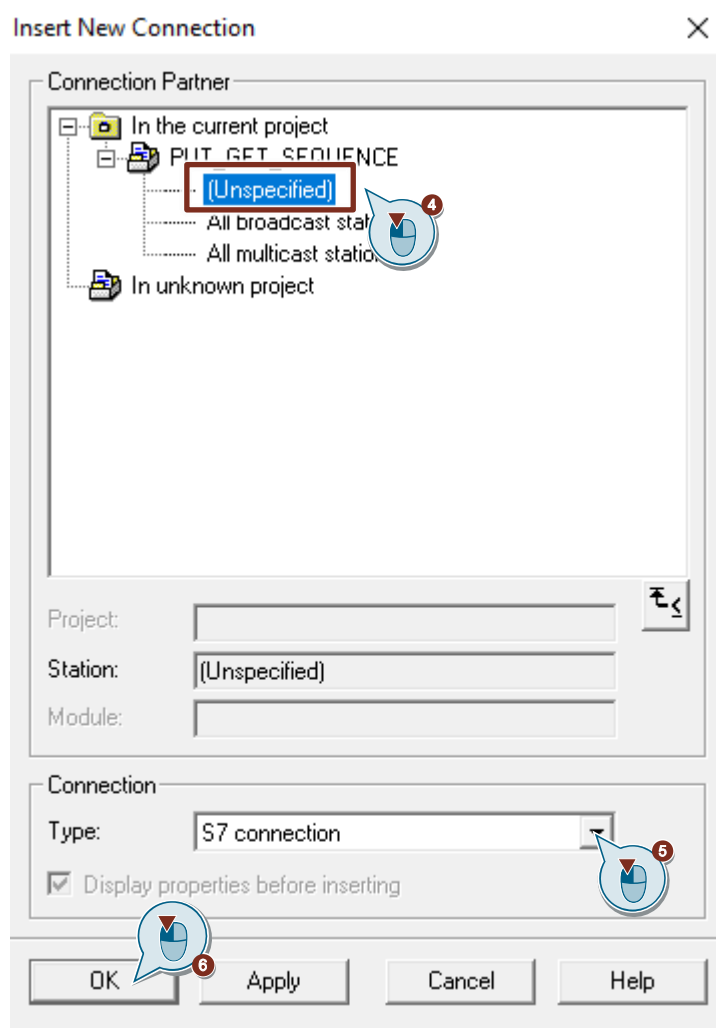
1. In the SIMATIC Manager you open the "NetPro" tool via the menu "Options > Configure Network". In "NetPro" you configure the S7 connection for the S7-300 CPU.



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



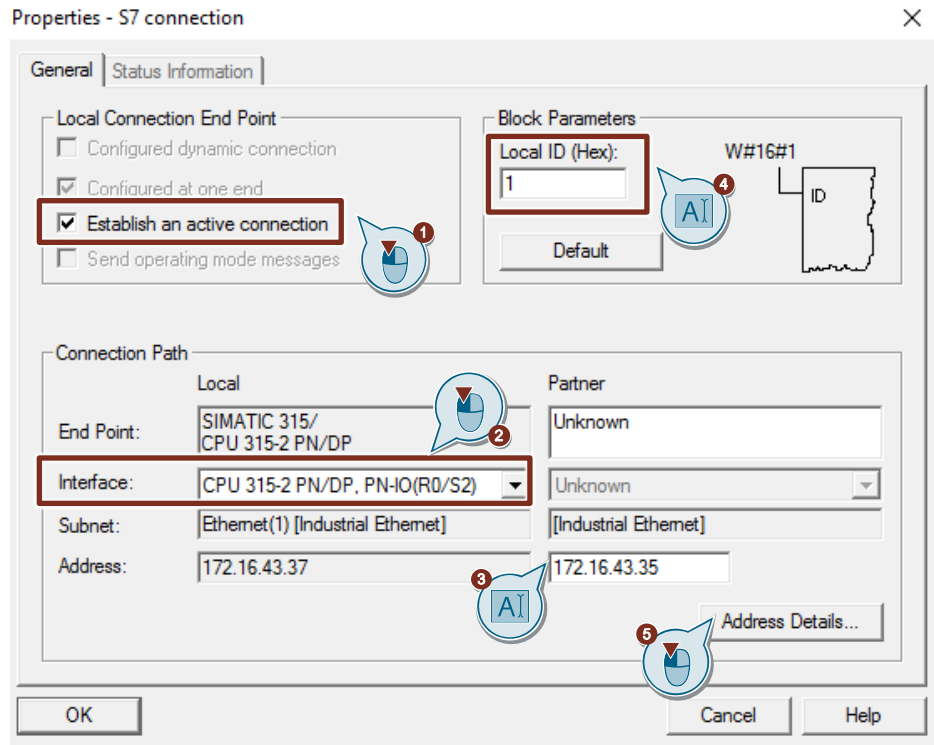
4. Select the "Unspecified" item under "Connection Partner".
5. Select "S7 connection" as the connection type.
6. Click "Apply". The Properties dialog of the S7 connection opens.



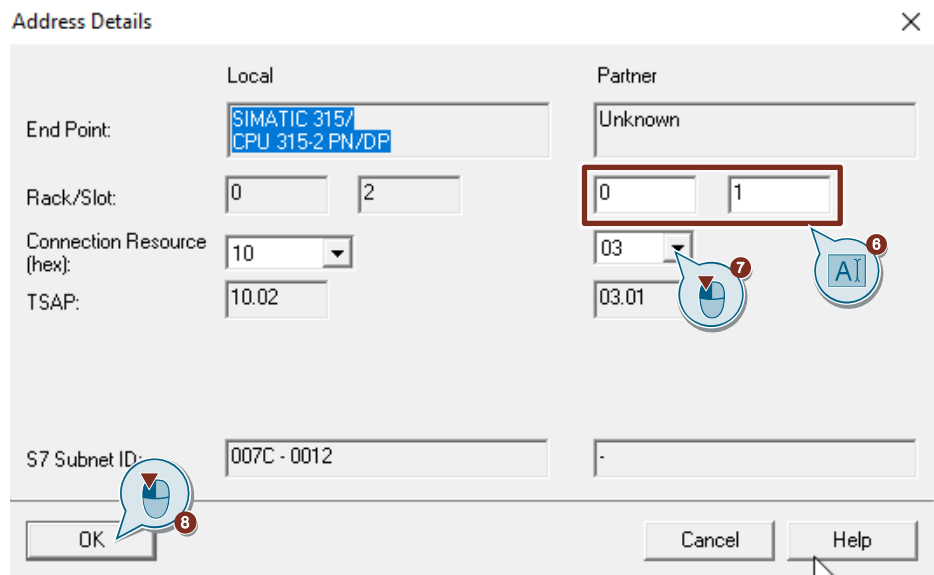
### 2.2.2 Define connection parameters

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 S7-1500 CPU: 172.16.43.35. If you use a CP as partner instead of the integrated interface of the S7-1500 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 FB14 "GET" and FB15 "PUT".
5. Click the "Address Details..." button. The "Address Details" dialog opens.



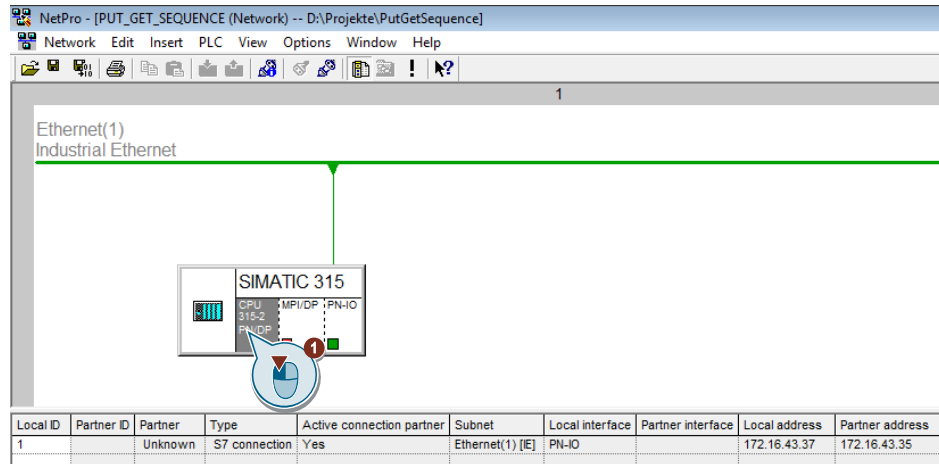
6. Enter the rack and slot of the partner CPU or CP. The S7-1500 CPU uses Rack 0 and Slot 1.
7. Select the connection resource 03(hex) for the partner CPU because the S7 connection is only configured unilaterally in the S7-300 CPU. With these settings the TSAP has the value 03.01 in the partner CPU.
8. Apply the settings with "OK".



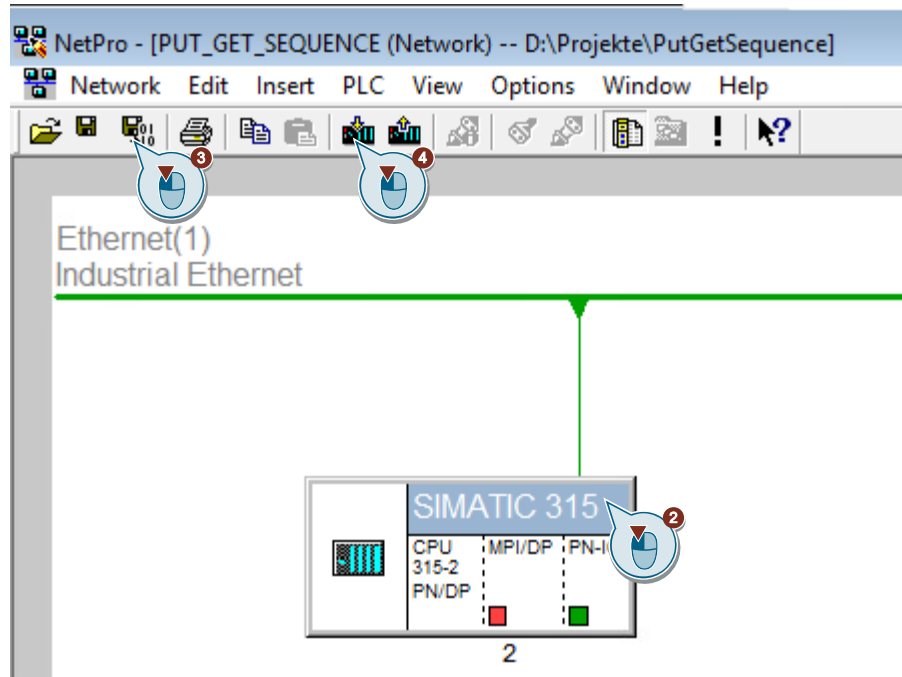
9. Likewise apply the settings in the Properties dialog of the S7 connection with "OK".
10. In the "Insert New Connection" dialog you click the "OK" button to close the dialog. The S7 connection is inserted in "NetPro".

### 2.2.3 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.



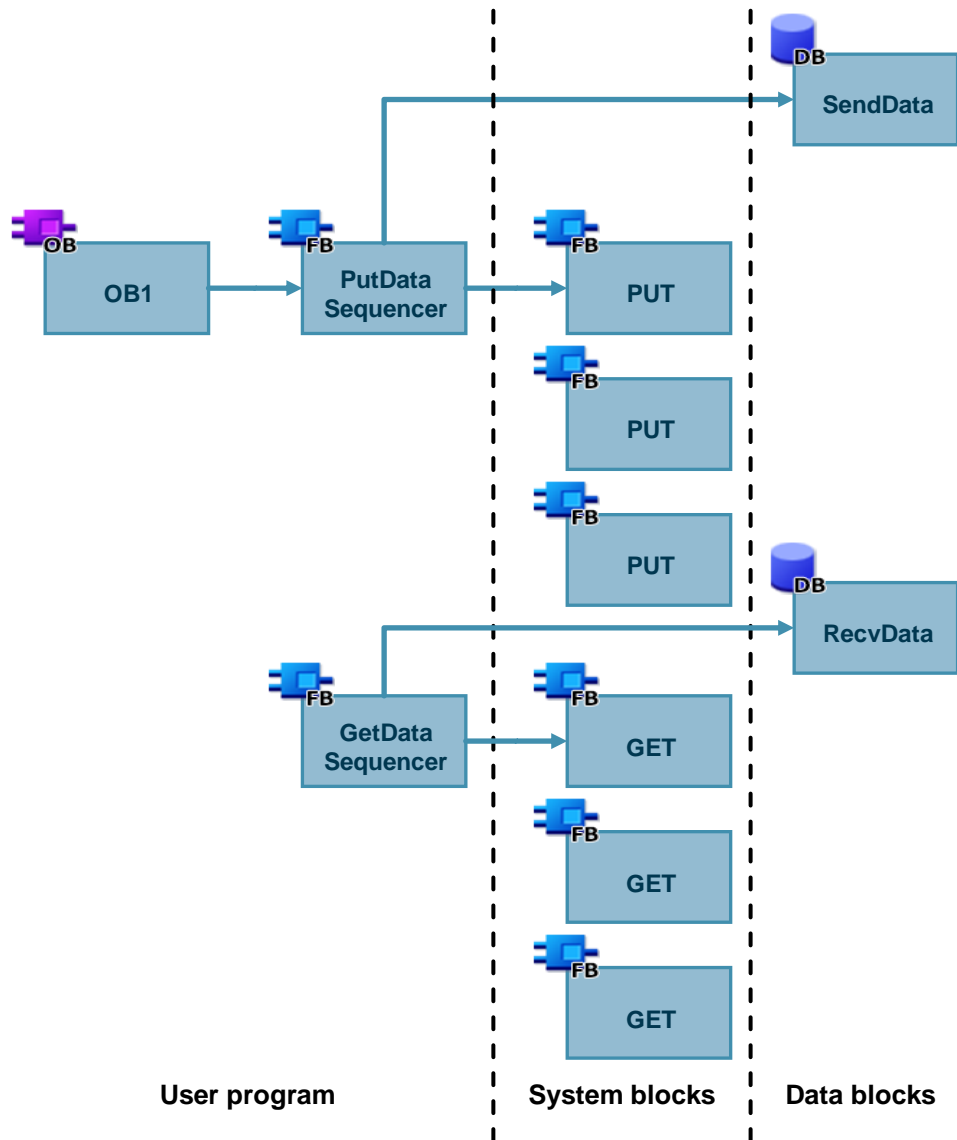
2. In "NetPro" you mark the SIMATIC S7-300 station.
3. In the toolbar you click the "Save and Compile" button to save and compile the connection configuration.
4. In the toolbar you click the "Load marked station(s)" button to load the connection configuration into the S7 CPU.



# 3 User Program of the S7-300 CPU (Active S7 CPU)

## 3.1 Overview

The following figure shows an overview of the user program of the active S7 CPU.  
Figure 3-1



## 3.2 Create Send and Receive Data Areas

### 3.2.1 DB1 "RecvData"

The data read from the partner CPU is stored in the data block DB1 "RecvData" of the active S7 CPU.

### 3.2.2 DB2 "SendData"

The data that is transferred to the partner CPU is stored in the data block DB2 "SendData" of the active S7 CPU.

## 3.3 Create Variables for Parameters of Function Blocks "PutDataSequence" and "GetDataSequence"

Create the following variables in DB10 "GeneralData" to assign the input and output parameters of the FBs "PutDataSequence" and "GetDataSequence". UDTs are used as data types.

Table 3-1

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

### 3.3.1 UDT1 "typePut"

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

Table 3-2

Parameter	Data type	Start value	Description
execute	Boolean	false	Control parameter for FB1 "PutDataSequence"
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



### 3.3.2 UDT2 "typeGet"

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

Table 3-3

Parameter	Data type	Start value	Description
execute	Boolean	false	Control parameter for FB2 "GetDataSequence"
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

### 3.3.3 UDT3 "typeDiagnostic"

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

Table 3-4

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

### 3.4 FB1 "PutDataSequence"

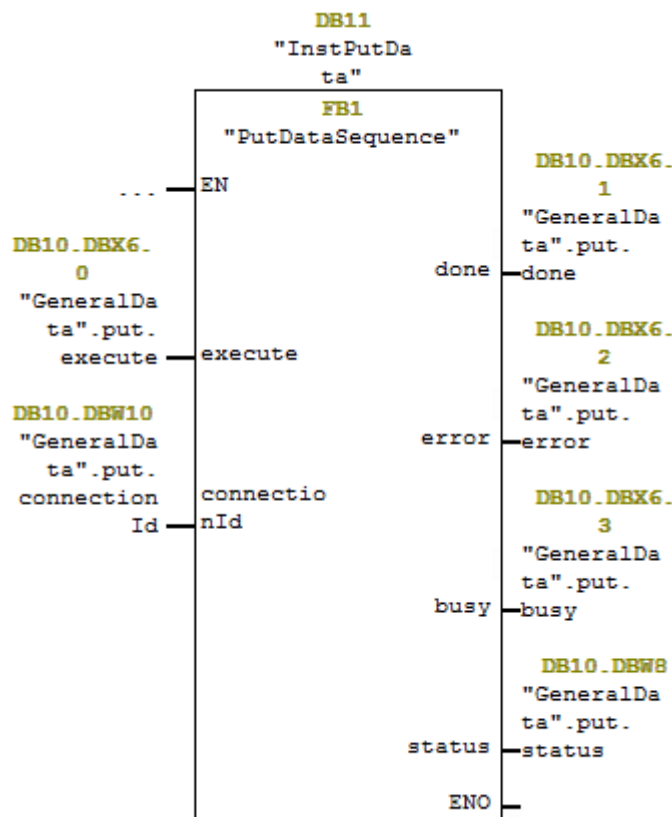
The FB1 "PutDataSequence" calls the "PUT" instruction to execute the following functions:

- 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.

FB1 "PutDataSequence" is called cyclically in OB1.

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

Figure 3-2



### 3.4.1 Parameters of FB1 "PutDataSequence"

The following table shows the parameters of FB1 "PutDataSequence".

Table 3-5

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 "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": <ul style="list-style-type: none"> <li>• "error" = 0:                             <ul style="list-style-type: none"> <li>- "status" = 0000 (hex): neither warning nor error</li> <li>- "status" &lt;&gt; 0000 (hex): Warning, The "status" parameter provides detailed information.</li> </ul> </li> <li>• "error" = 1: An error has occurred. The "status" parameter provides detailed information about the type of error.</li> </ul>
status	OUT	Word	

Assign the variables created in section 3.3 to the parameters of FB1 "PutDataSequence".

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

Table 3-6

Parameters of FB1 "PutDataSequence"	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	-

Open FB1 "PutDataSequence". FB1 "PutDataSequence" contains 3 calls of FB15 "PUT". Define the Receive and Send data areas.

#### Receive data area

At input "ADDR\_1" of FB15 "PUT" you refer to the area in the partner CPU (S7-1500 CPU, for example) to which you want to write.

- First call of FB15 "PUT": P#DB1.DBX0.0 BYTE 160
- Second call of FB15 "PUT": P#DB1.DBX160.0 BYTE 160
- Third call of FB15 "PUT": P#DB1.DBX320.0 BYTE 80

#### 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.

- First call of FB15 "PUT": P#DB2.DBX0.0 BYTE 160
- Second call of FB15 "PUT": P#DB2.DBX160.0 BYTE 160
- Third call of FB15 "PUT": P#DB2.DBX320.0 BYTE 80

#### Note

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

### 3.5 FB2 "GetDataSequence"

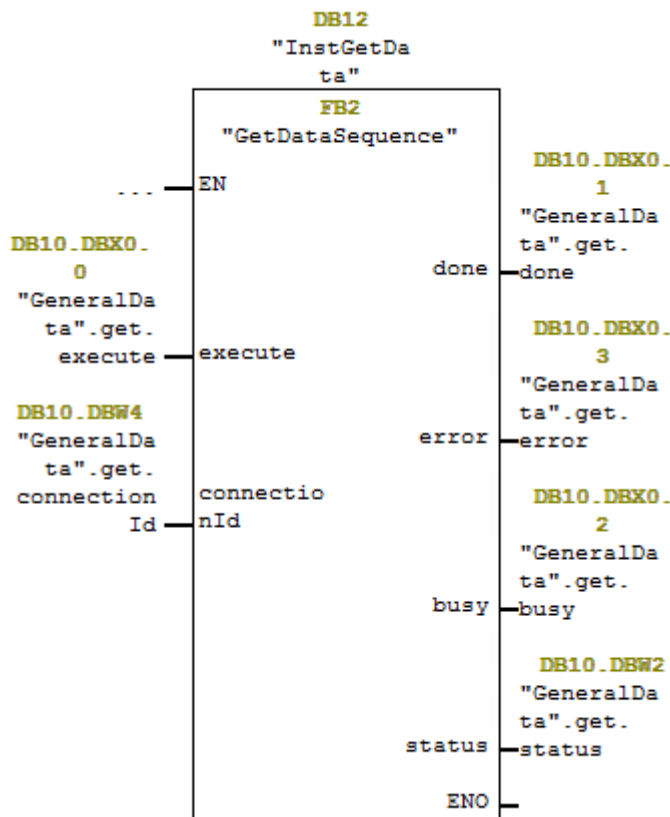
The FB2 "GetDataSequence" calls the "GET" instruction to execute the following functions:

- 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 "GetDataSequence" is called cyclically in OB1.

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

Figure 3-3



### 3.5.1 Parameters of FB2 "GetDataSequence"

The following table shows the parameters of FB2 "GetDataSequence".

Table 3-7

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": <ul style="list-style-type: none"> <li>• "error" = 0:                             <ul style="list-style-type: none"> <li>- "status" = 0000 (hex): neither warning nor error</li> <li>- "status" &lt;&gt; 0000 (hex): Warning, The "status" parameter provides detailed information.</li> </ul> </li> <li>• "error" = 1:                             <ul style="list-style-type: none"> <li>An error has occurred. The "status" parameter provides detailed information about the type of error.</li> </ul> </li> </ul>
status	OUT	Word	

Assign the variables created in section [3.3](#) to the parameters of FB2 "GetDataSequence".

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

Table 3-8

Parameters of FB2 "GetDataSequence"	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	-

Open FB2 "GetDataSequence". FB2 "GetDataSequence" contains 3 calls of FB14 "GET". Define the Receive and Send data areas.

#### Send data area

At input "ADDR\_1" of FB14 "GET" you refer to the area in the partner CPU (S7-1500 CPU, for example) which you want to read.

- First call of FB14 "GET": P#DB2.DBX0.0 BYTE 160
- Second call of FB14 "GET": P#DB2.DBX160.0 BYTE 160
- Third call of FB14 "GET": P#DB2.DBX320.0 BYTE 80

#### 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.

- First call of FB14 "GET": P#DB1.DBX0.0 BYTE 160
- Second call of FB14 "GET": P#DB1.DBX160.0 BYTE 160
- Third call of FB14 "GET": P#DB1.DBX320.0 BYTE 80

#### Note

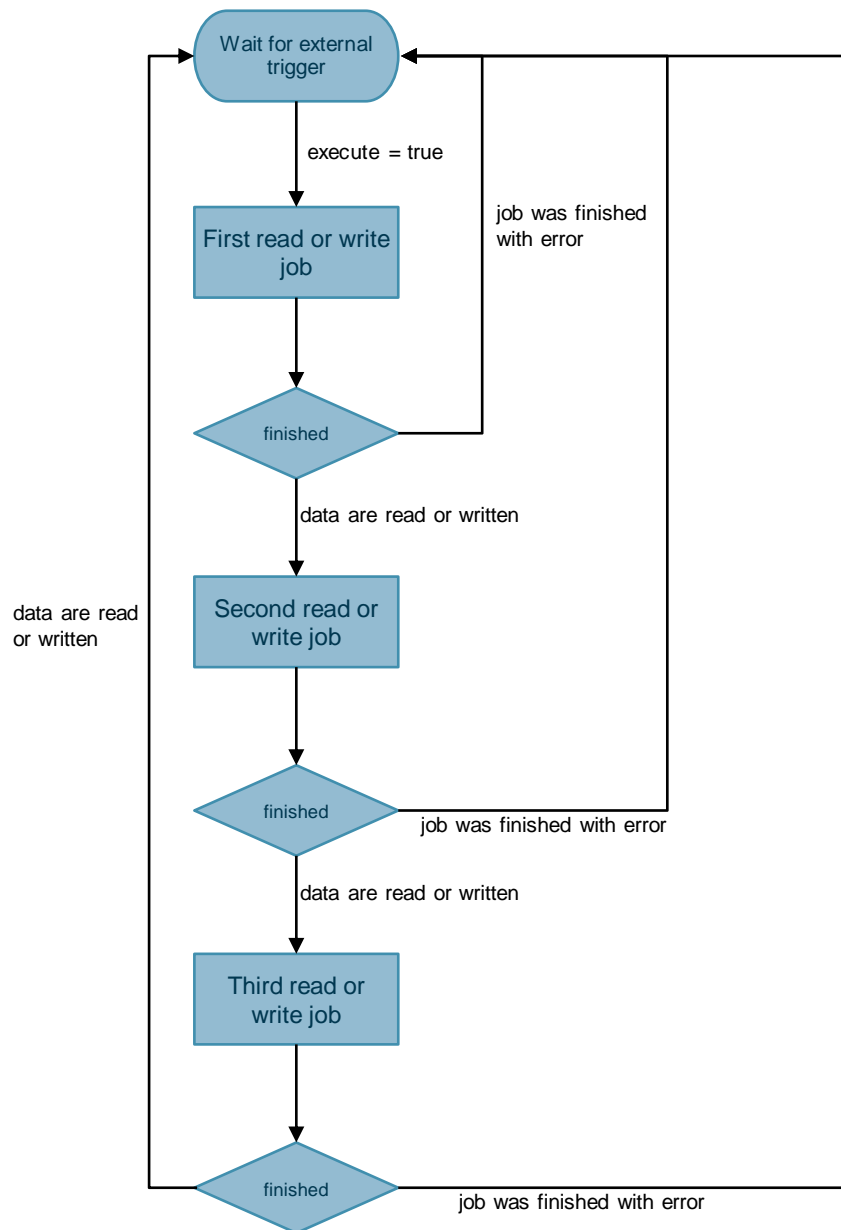
The following minimum user data size for the "GET" instruction is guaranteed: 160 bytes

### 3.6 Function

#### 3.6.1 Overview

The following figure shows the call of the FBs "PutDataSequence" and "GetDataSequence".

Figure 3-4





### 3.6.2 Function details

#### FB1 "PutDataSequence"

The first "PUT" write job is started when a positive edge is detected at the "execute" input of FB "PutDataSequence".

If the first Write job is completed successfully with DONE = true, the second write job is started.

If the first Write job is completed with an error (ERROR = true), the following actions are performed:

- The "error" output of FB1 "PutDataSequence" is set to the value "true" for one cycle.
- The status display of FB15 "PUT" is output at the "status" output of FB1 "PutDataSequence".

If the second Write job is completed successfully with DONE = true, the third write job is started.

If the second Write job is completed with an error (ERROR = true), the following actions are performed:

- The "error" output of FB1 "PutDataSequence" is set to the value "true" for one cycle.
- The status display of FB15 "PUT" is output at the "status" output of FB1 "PutDataSequence".

If the third Write job is completed successfully with DONE = true, the "done" output of FB1 "PutDataSequence" is set to the value "true" for one cycle. This indicates that all the data has been sent successfully.

If the third Write job is completed with an error (ERROR = true), the following actions are performed:

- The "error" output of FB1 "PutDataSequence" is set to the value "true" for one cycle.
- The status display of FB15 "PUT" is output at the "status" output of FB1 "PutDataSequence".

### FB2 "GetDataSequence"

The first "GET" read job is started when a positive edge is detected at the "execute" input of FB "GetDataSequence".

If the first Read job is completed successfully with NDR = true, the second read job is started.

If the first Read job is completed with an error (ERROR = true), the following actions are performed:

- The "error" output of FB2 "GetDataSequence" is set to the value "true" for one cycle.
- The status display of FB14 "GET" is output at the "status" output of FB2 "GetDataSequence".

If the second Read job is completed successfully with NDR = true, the third read job is started.

If the second Read job is completed with an error (ERROR = true), the following actions are performed:

- The "error" output of FB2 "GetDataSequence" is set to the value "true" for one cycle.
- The status display of FB14 "GET" is output at the "status" output of FB2 "GetDataSequence".

If the third Read job is completed successfully with NDR = true, the "done" output of FB2 "GetDataSequence" is set to the value "true" for one cycle. This indicates that all the data has been received successfully.

If the third Read job is completed with an error (ERROR = true), the following actions are performed:

- The "error" output of FB2 "GetDataSequence" is set to the value "true" for one cycle.
- The status display of FB14 "GET" is output at the "status" output of FB2 "GetDataSequence".

## 3.7 Error Handling

### 3.7.1 FB1 "PutDataSequence"

If an error occurs in FB1 "PutDataSequence", the cause of the error is written to the "status" output parameter.

Table 3-9

error	status	Description	Remedy
0	16#7000	FB1 "PutDataSequence" is not enabled.	Status information Enable FB1 "PutDataSequence" by setting the "execute" input to "true".
0	16#7001	FB1 "PutDataSequence" is initialized.	Status information
0	16#7002	Write job is running.	Status information
1	<>16#0000	Status display of FB15 "PUT"	Detailed information is available in the STEP 7 Online Help or in the following manual: <a href="#">SIMATIC System Software for S7-300/400 System and Standard Functions - Volume 1/2</a>

### 3.7.2 FB2 "GetDataSequence"

If an error occurs in FB2 "GetDataSequence", the cause of the error is written to the "status" output parameter.

Table 3-10

error	status	Description	Remedy
0	16#7000	FB2 "GetDataSequence" is not enabled.	Status information Enable FB2 "GetDataSequence" by setting the "execute" input to "true".
0	16#7001	FB2 "GetDataSequence" is initialized.	Status information
0	16#7002	Read job is running.	Status information
1	<>16#0000	Status display of FB14 "GET"	Detailed information is available in the STEP 7 Online Help or in the following manual: <a href="#">SIMATIC System Software for S7-300/400 System and Standard Functions - Volume 1/2</a>

## 4 User Program of the Passive S7 CPU

No instructions for data transfer are called in the user program of the passive S7 CPU.

All you need are data blocks in which the sent and received data is stored.

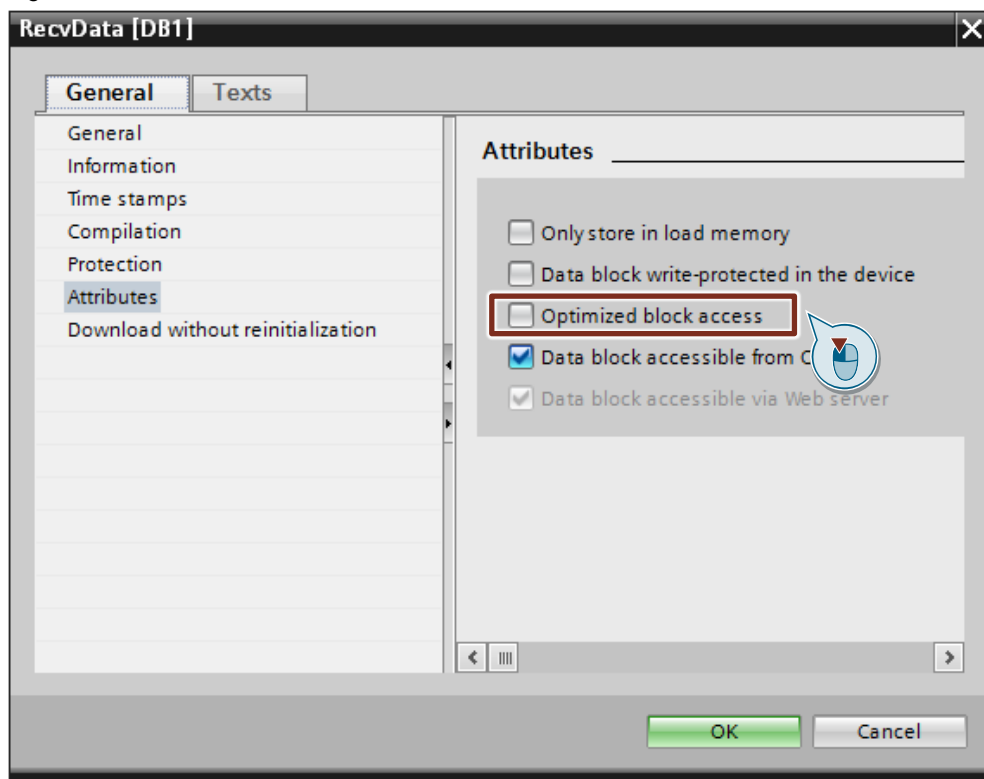
### 4.1 Create Send and Receive Data Areas

#### 4.1.1 DB1 "RecvData"

The data written by the partner CPU is stored in the data block DB1 "RecvData" of the passive S7 CPU.

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

Figure 4-1

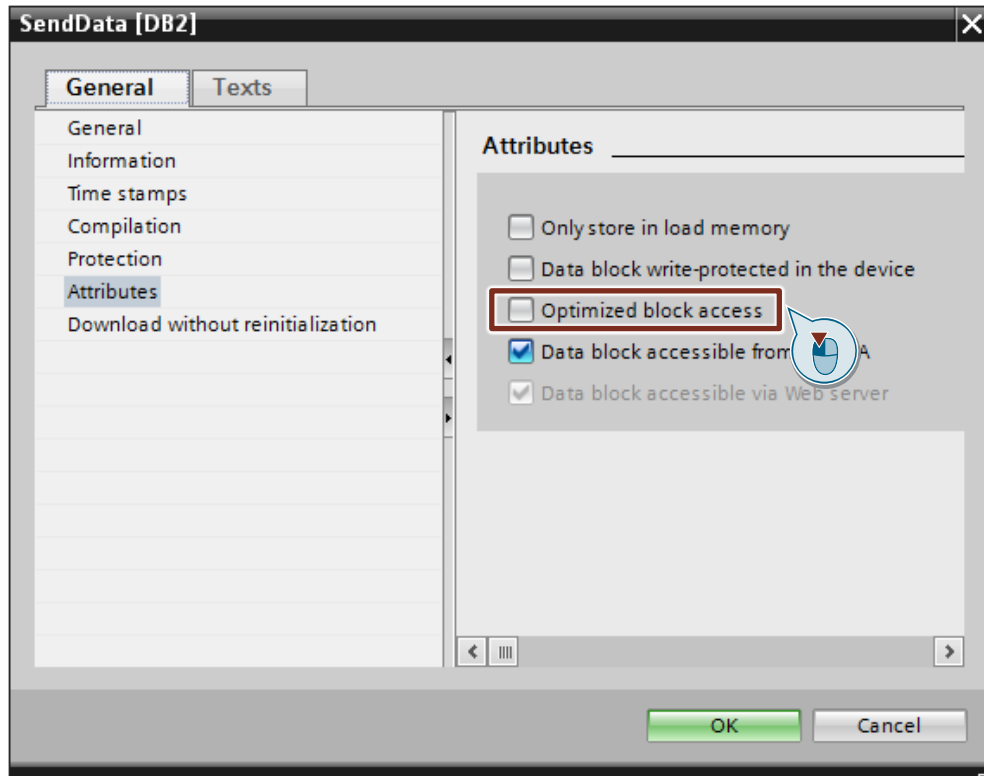


### 4.1.2 DB2 "SendData"

The data that is read by the partner CPU is stored in the data block DB2 "SendData" of the passive S7 CPU.

In the Properties of the DB2 "SendData", under "Attributes" you disable the "Optimized block access" function.

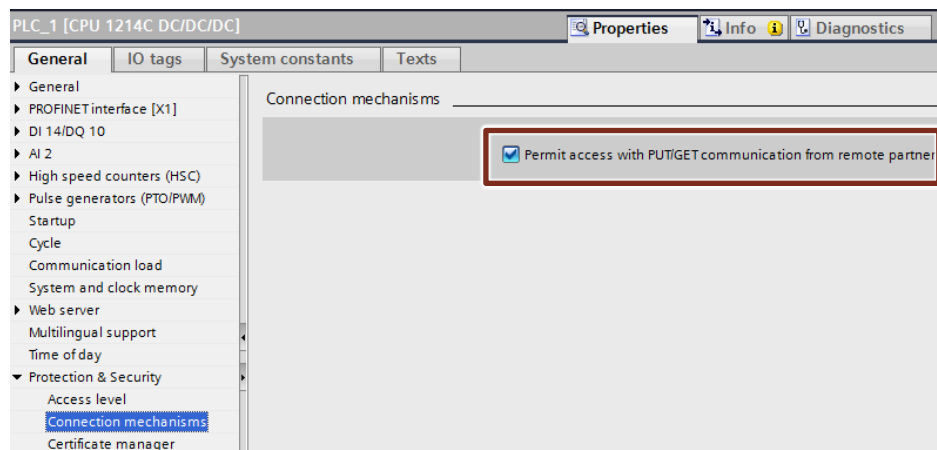
Figure 4-2



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

In the S7-1200 CPU and in the S7-1500 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 or S7-1500 CPU using the "PUT" and "GET" instructions.

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

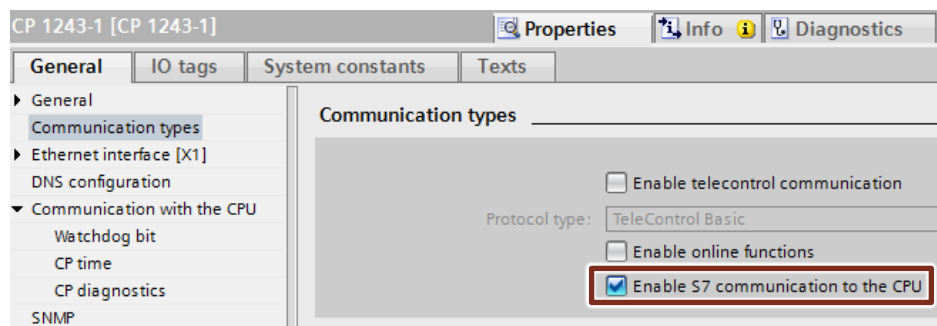


## 4.3 Enable S7 communication to the CPU

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.



## 5 Information

### 5.1 FB15 "PUT"

FB15 "PUT" is called in FB1 "PutDataSequence". This is located in the Standard Library under "Communication Blocks". Copy the FB from the Standard Library and insert it in your STEP 7 project.

You use FB15 "PUT" to write data to the partner CPU.

**Note**

This is only possible if the function "Permit access with PUT/GET communication from remote partner" has been enabled for the partner CPU in the Properties of the S7 CPU under "Protection & Security > Connection mechanisms".

With FB15 "PUT" you cannot access blocks that have been created with the "optimized" type of access.

Detailed information about FB15 "PUT" is available in the manual [SIMATIC System Software for S7-300/400 System and Standard Functions - Volume 1/2](#).

### 5.2 FB14 "GET"

FB14 "GET" is called in FB2 "GetDataSequence". This is located in the Standard Library under "Communication Blocks". Copy the FB from the Standard Library and insert it in your STEP 7 project.

You use FB14 "GET" to read data from the partner CPU.

**Note**

This is only possible if the function "Permit access with PUT/GET communication from remote partner" has been enabled for the partner CPU in the Properties of the CPU under "Protection & Security > Connection mechanisms".

With FB14 "GET" you cannot access blocks that have been created with the "optimized" type of access.

Detailed information about FB14 "GET" is available in the manual [SIMATIC System Software for S7-300/400 System and Standard Functions - Volume 1/2](#)