

A man in a light blue shirt is seen from the side, holding a tablet computer. He is looking at the screen, which displays a technical interface with various charts and data. The background is a blurred industrial setting, likely a factory or manufacturing plant, with various pieces of machinery and equipment visible.

**SIEMENS**

Application description • 10/2013

# S7 Communication with PUT/GET

S7-1500 CPUs

---

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.

## Caution

The functions and solutions described in this article confine themselves predominantly to the realization of the automation task. Furthermore, please take into account that corresponding protective measures have to be taken in the context of Industrial Security when connecting your equipment to other parts of the plant, the enterprise network or the internet. Further information can be found in Entry ID 50203404.

<http://support.automation.siemens.com/WW/view/en/50203404>

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
<b>2</b>	<b>Configuration of the S7 Connection .....</b>	<b>4</b>
2.1	Configuration of the S7 Connection .....	4
2.2	Properties of the Configured S7 Connection .....	6
2.3	Permit Access with PUT/GET Communication from Remote Partner (PLC, HMI, OPC, ...) .....	7
2.4	Configure Clock Memory .....	8
<b>3</b>	<b>User Program of the Active S7-1500 CPU .....</b>	<b>10</b>
3.1	OB100 .....	10
3.2	OB1 .....	10
3.3	PUT_GET (FB1) .....	11
3.3.1	Write data .....	11
3.3.2	Read data .....	14
3.4	SEND_DATA (DB1) .....	17
3.5	RCVD_DATA (DB2) .....	18
<b>4</b>	<b>User Program of the Passive S7--1500 CPU .....</b>	<b>19</b>
4.1	SEND_DATA (DB2) .....	19
4.2	RCVD_DATA (DB3) .....	19

# 1 Introduction

You can use the S7 Communication, for example, for data transfer over the integrated PROFINET interface and Industrial Ethernet interface of the S7-1500 CPU.

The following communication instructions are available for S7 Communication.

- PUT for sending data
- GET for receiving data

In STEP 7 V12 SP1 Upd2 (TIA Portal) you will find the above-mentioned communication instructions in the "Instructions" task card in the "Communication > S7 Communication" palette.

This example shows how to configure an S7 connection between two S7-1500 CPUs to exchange data between the S7-1500 CPUs using the PUT and GET communication instructions.

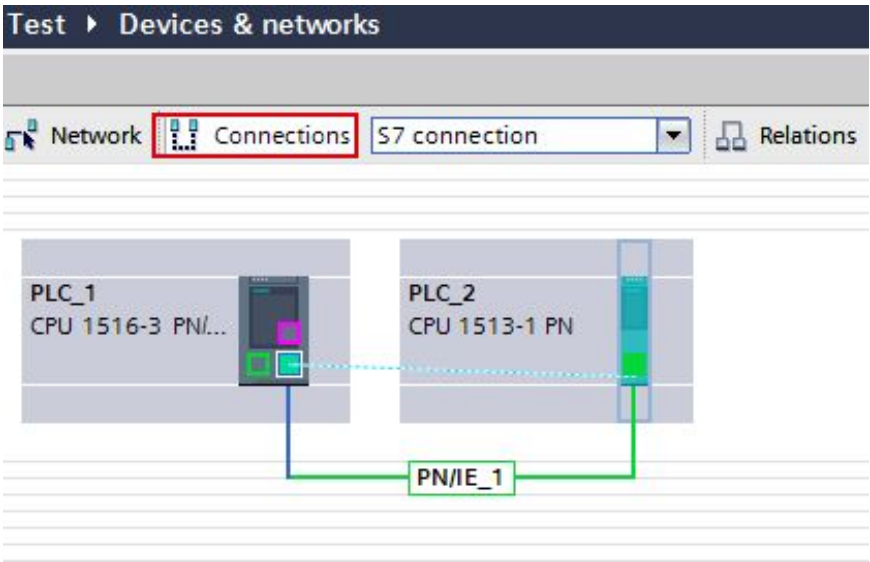
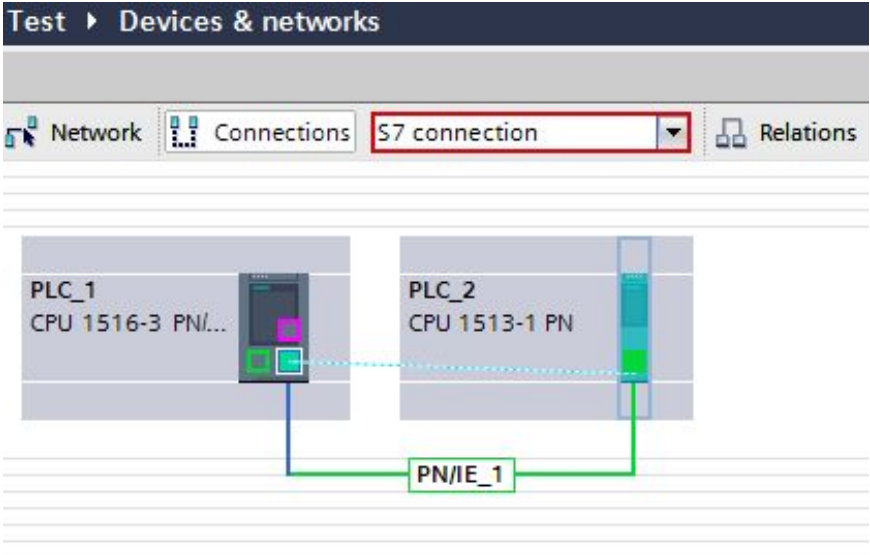
# 2 Configuration of the S7 Connection

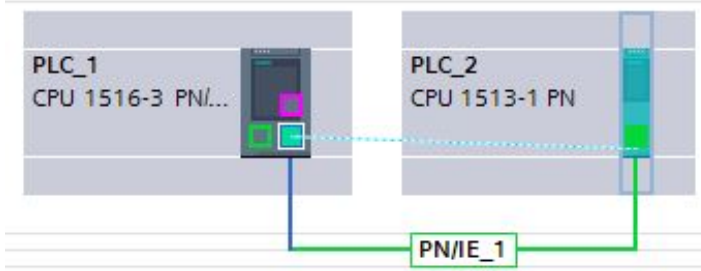
## 2.1 Configuration of the S7 Connection

In the Network view of the Hardware and network editor you create the connection partners, two S7-1500 CPUs and network them.

Proceed as follows to create an S7 connection graphically between the two S7-1500 CPUs.

Table 2-1

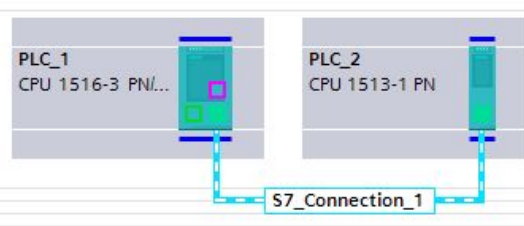
No.	Description
1.	<p>In the Network view you click the "Connections" button. This enables the Connection mode.</p> 
2.	<p>Select "S7 connection" as the connection type.</p> 

No.	Description
3.	<p>With the button held down drag the mouse cursor from the S7-1500 CPU where the S7 connection is to start to the S7-1500 CPU where the S7 connection is to finish.</p> 
4.	<p>Release the mouse button when the cursor is on the target device to create the S7 connection between the two S7-1500 CPUs.</p>

### Result

- A specified connection is created.
- The connection route is highlighted.
- The connection is entered in the connection table.

Figure 2-1



Network overview	Connections	Relations	I/O communication	VPN	
Local connection name	Local end point	Local ID (hex)	Partner ID (hex)	Partner	Connection type
S7_Verbindung_1	PLC_1	100	100	PLC_2	S7 connection
S7_Verbindung_1	PLC_2	100	100	PLC_1	S7 connection
S7_Connection_1	PLC_1	101	101	PLC_2	S7 connection
S7_Connection_1	PLC_2	101	101	PLC_1	S7 connection

## 2.2 Properties of the Configured S7 Connection

The "Properties" tab in the Inspector window shows the properties of the S7 connection to be configured. These include:

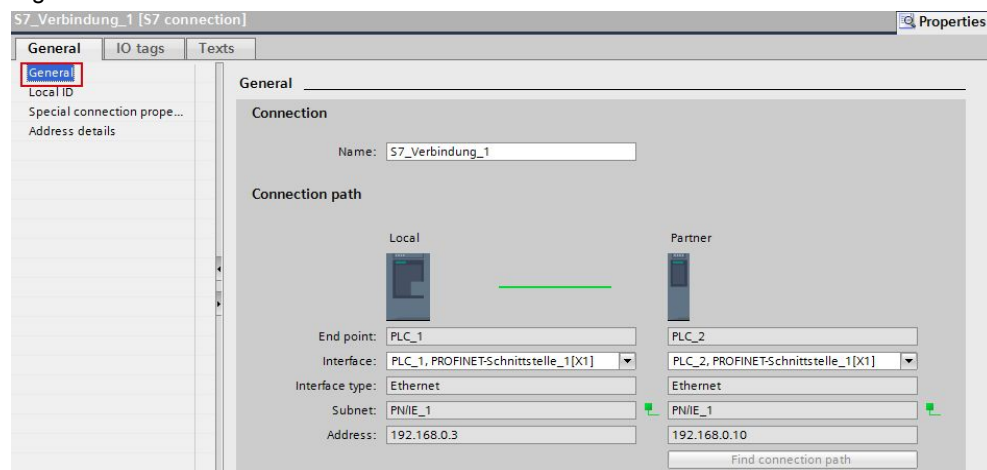
- General connection parameters
- Local ID
- Special connection parameters
- Address details

### General connection parameters

The "General" parameters group of the properties of the S7 connection shows the general connection parameters that identify the connection endpoint.

Here you can assign the connection route and specify the connection partner in full.

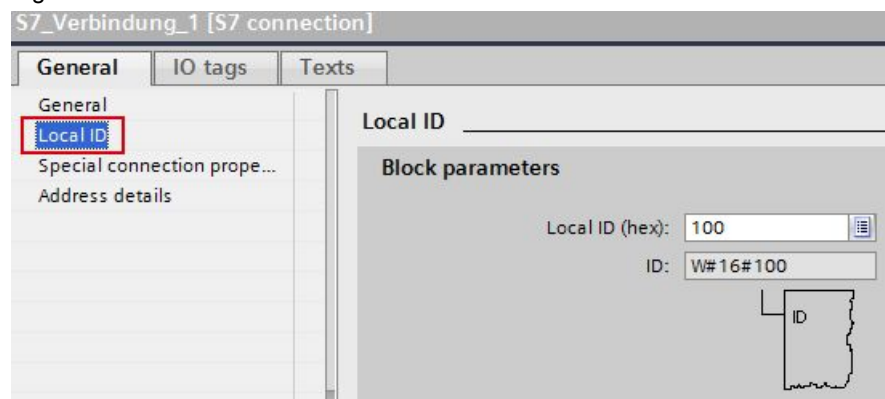
Figure 2-2



### Local ID

Here you see the local ID of the module from which the S7 connection is observed (local partner). You can change the local ID. This is necessary if you have already programmed communication function blocks and you want to use the local ID specified there for the S7 connection.

Figure 2-3



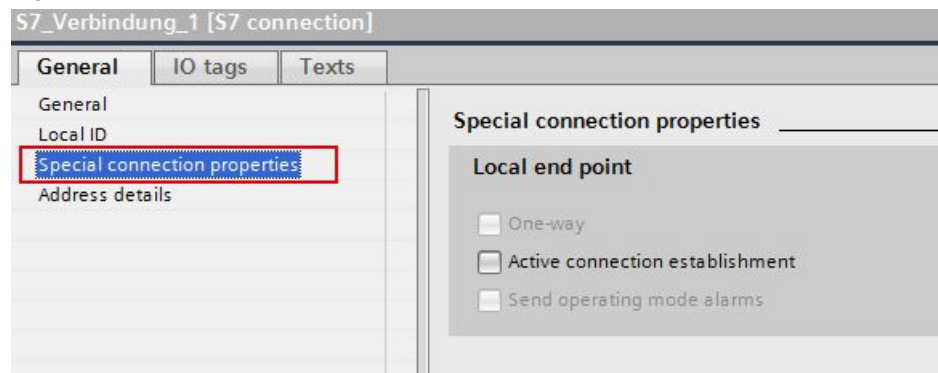


### Special connection parameters

Display of the connection parameters

- One-way  
One-way means that the connection partner is server for this connection and cannot actively send or receive.
- Active connection establishment  
In this example a two-way S7 connection is configured. This means that you can set which connection partner is to take on the active part.
- Send operating mode alarms  
In this example the local partner does not send any operating mode alarms to the connection partner.

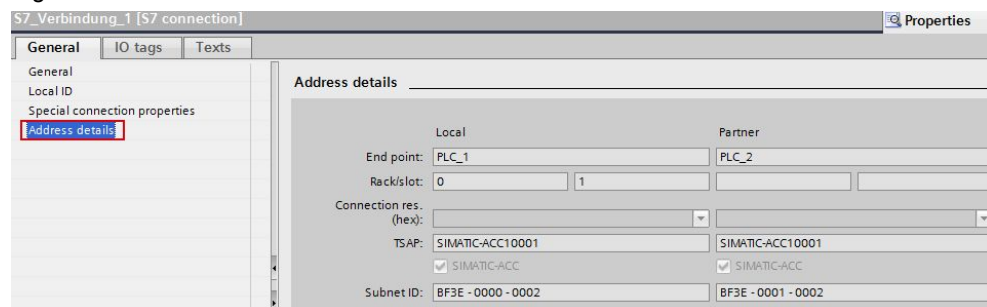
Figure 2-4



### Address details

Display of the address details of the S7 connection. In the case of an unspecified partner you can change the values for the rack and slot. All the other values are taken from the current configuration and cannot be changed.

Figure 2-5



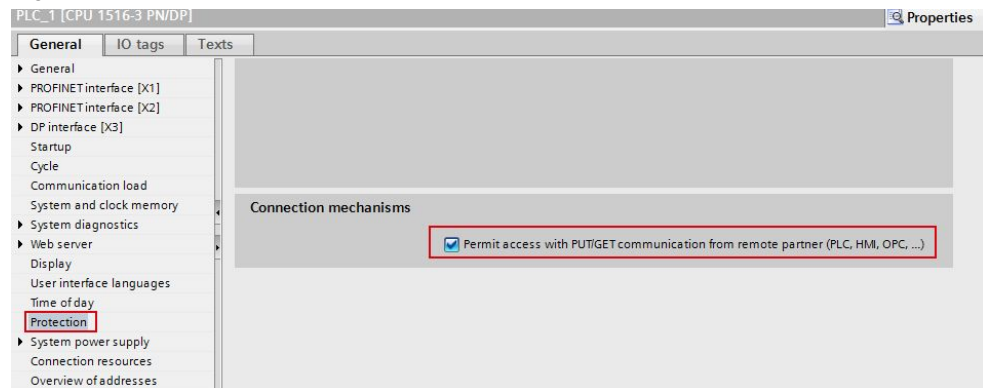
## 2.3 Permit Access with PUT/GET Communication from Remote Partner (PLC, HMI, OPC, ...)

Access from the remote partner (PLC, HMI, OPC, ...) with PUT/GET communication must be permitted for both S7-1500 CPUs between which the S7 connection is configured.

Select the CPU in the device configuration of the first S7-1500 station.

In the inspector window you enable the option "Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)" under "Properties > General > Protection".

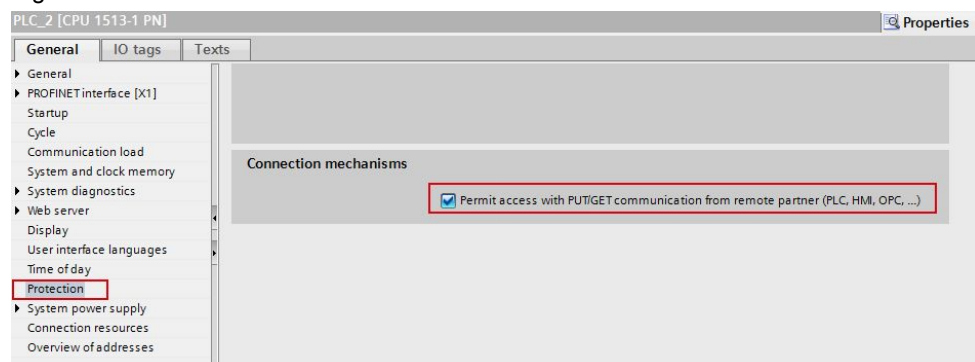
Figure 2-6



Select the CPU in the device configuration of the second S7-1500 station.

In the inspector window you enable the option "Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)" under "Properties > General > Protection".

Figure 2-7



## 2.4 Configure Clock Memory

Memory byte 0 is configured as clock memory in the active S7-1500 CPU. The send and receive job is triggered by this clock memory.

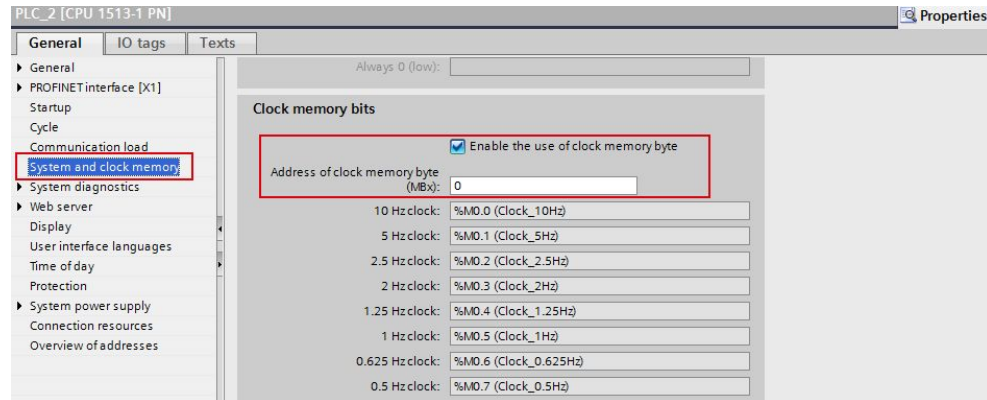
Select the CPU in the device configuration of the S7-1500 station.

In the inspector window you enable the option "Enable the use of clock memory byte" under "Properties > General > System and clock memory". The address 0 is entered for the clock memory byte in this example.



## 2 Configuration of the S7 Connection

Figure 2-8



### 3 User Program of the Active S7-1500 CPU

The user program of the active S7-1500 CPU consists of the following blocks:

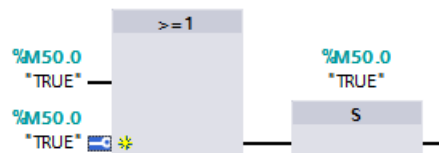
Table 3-1

Block	Symbolic name	Description
OB100	Startup	Startup OB The OB100 is executed when the CPU is restarted (warm restart).
OB1	Main	The PUT_GET (FB1) block including the associated instance data block PUT_GET_DB (DB5) is called cyclically in OB1.
FB1	PUT_GET	The communication instructions PUT and GET are called in the PUT_GET (FB1) block to transfer data via the configured S7 connection.
DB5	PUT_GET_DB	Instance data block of the PUT_GET (FB1) block
DB1	SEND_DATA	The data to be sent to the communication partner with the PUT instruction (send data) is stored in the SEND_DATA (DB1) block.
DB2	RCVD_DATA	The data received from the communication partner with the GET instruction (receive data) is stored in the RCVD_DATA (DB2) data block.

#### 3.1 OB100

When you restart (warm restart) the CPU, the M50.0 "TRUE" is set to the value "1". With M50.0 "TRUE" the job to send or receive data is enabled.

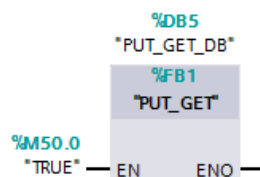
Figure 3-1



#### 3.2 OB1

The PUT\_GET (FB1) block including the associated instance data block PUT\_GET\_DB (DB5) is called cyclically in OB1.

Figure 3-2



### 3.3 PUT\_GET (FB1)

#### 3.3.1 Write data

The PUT communication instruction is called in the PUT\_GET (FB1) block. The PUT instruction is in the "Instructions" task card in the "Communication > S7 Communication" palette.

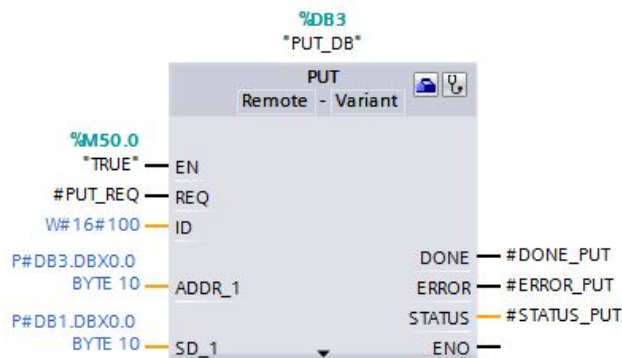
You use the PUT instruction to write data to the remote S7-1500 CPU.

#### 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".

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

Figure 3-3



#### Input parameters of the PUT instruction

The PUT instruction has the following input parameters.

Table 3-2

Input parameters	Data type	Description
REQ	BOOLEAN	Control parameter, enables data transfer on rising edge
ID	WORD	Addressing parameter for specifying the connection to the partner CPU
ADDR_1	REMOTE	Pointer to the area to be written to in the partner CPU.
SD_1	VARIANT	Pointer to the area in your own CPU that contains the data to be sent.

#### Output parameters of the PUT instruction

The PUT instruction has the following output parameters.

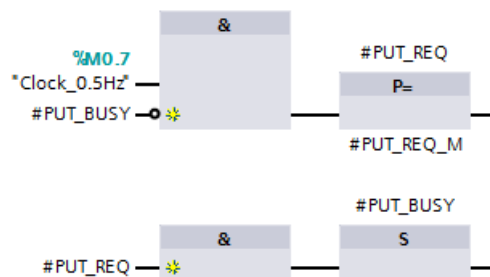
Table 3-3

Output parameters	Data type	Description
DONE	BOOLEAN	State parameter DONE <ul style="list-style-type: none"> <li>0: Job not yet started or is still being executed</li> <li>1: Job executed error-free</li> </ul>
ERROR	BOOLEAN	State parameters ERROR and STATUS, error display <ul style="list-style-type: none"> <li>ERROR = 0 STATUS has the value: <ul style="list-style-type: none"> <li>0000H: neither warning nor error</li> <li>&lt;&gt; 0000H: Warning, STATUS provides detailed information</li> </ul> </li> <li>ERROR = 1 An error has occurred, STATUS provides detailed information about the type of error</li> </ul>
STATUS	WORD	

### Enable job to write the data

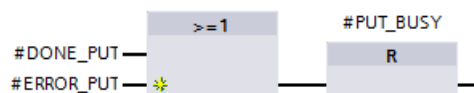
The job to write the data is enabled by a positive edge at the REQ input of the PUT instruction. Enabling of the write job is controlled by clock memory M0.7 and the "PUT\_BUSY" variable. When the write job is running, "PUT\_BUSY" is set to the value "1". This makes it impossible for a new send job to be triggered.

Figure 3-4



Only when the current write job has been completed successfully or with an error is "PUT\_BUSY" reset to the value "0" so that a new write job can be enabled.

Figure 3-5



### Setting connection parameters

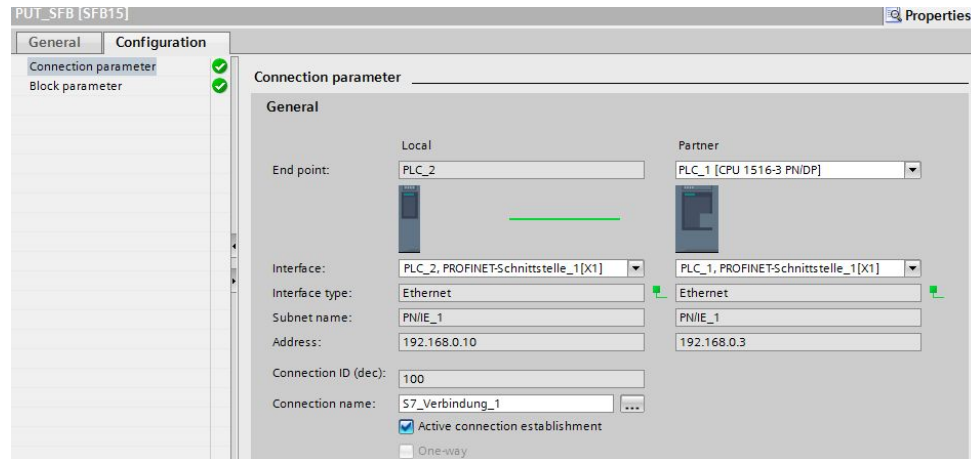
The connection parameters for the PUT instruction are set in the inspector window of the program editor. For this you select the PUT instruction called in the PUT\_GET (FB1) block. Open the "Configuration" tab in the inspector window. The "Connection Parameters" group is in the "Configuration" tab in the area navigation. This group includes the connection parameters.

Define the connection endpoints of the S7 connection via which the data will be sent and received. In this example you select as partner the S7-1500 CPU that participates passively in establishing the configured S7 connection.

The interface type, interface name and address of the communication partners are entered automatically once you have defined the connection endpoint.

Select the name of the S7 connection via which the data is to be transferred. The ID of the selected S7 connection will be entered automatically at the ID parameter of the PUT instruction.

Figure 3-6



#### Receive data area in the partner CPU

At the ADDR\_1 input parameter of the PUT instruction you specify the memory area of the partner CPU to which the data is to be written.

Only absolute addressing is permitted. In this example 10 bytes of data are stored in DB3 of the partner CPU starting at address 0:

- P#DB1.DBX0.0 BYTE 10.

#### Send data area in the local CPU

At the SD\_1 input data area of the PUT instruction you specify the memory area of the local CPU from which the data is to be read.

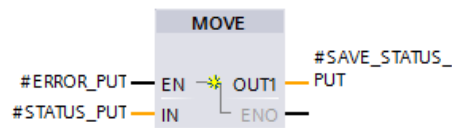
In this example 10 bytes of data are stored in DB3 of the local CPU starting at address 0:

- P#DB3.DBX0.0 BYTE 10.

#### Error evaluation

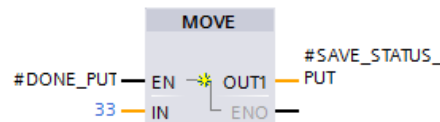
If an error occurs during processing of the PUT instruction, the STATUS of the PUT instruction is stored in the static variable "SAVE\_STATUS\_PUT".

Figure 3-7



If the PUT instruction is executed and completed successfully, the "SAVE\_STATUS\_PUT" variable is set to the value 33 (dec) = 21 (hex).

Figure 3-8



### 3.3.2 Read data

The GET communication instruction is called in the PUT\_GET (FB1) block. The GET instruction is in the "Instructions" task card in the "Communication > S7 Communication" palette.

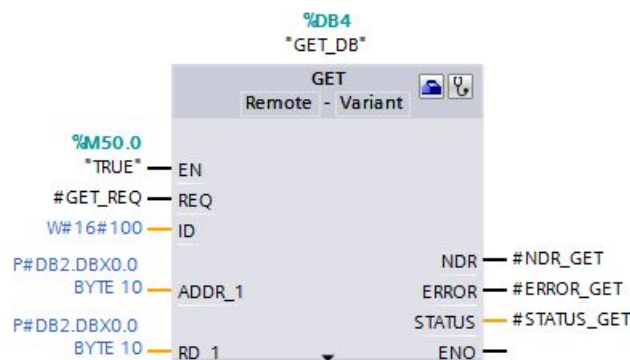
You use the GET instruction to read data from the remote S7-1500 CPU.

#### 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".

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

Figure 3-9





### Input parameters of the GET instruction

The GET instruction has the following input parameters.

Table 3-4

Input parameters	Data type	Description
REQ	BOOLEAN	Control parameter, enables data transfer on a rising edge
ID	WORD	Addressing parameter for specifying the connection to the partner CPU
ADDR_1	REMOTE	Pointer to the area to be read in the partner CPU.
RD_1	VARIANT	Pointer to the area in your own CPU in which the read data is stored.

### Output parameters of the GET instruction

The GET instruction has the following output parameters.

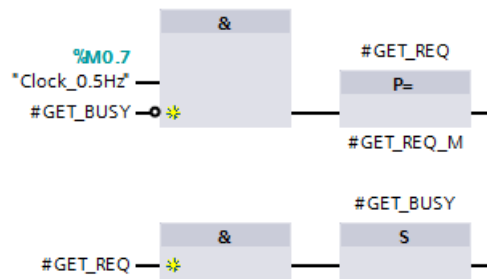
Table 3-5

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

### Enable job to read the data

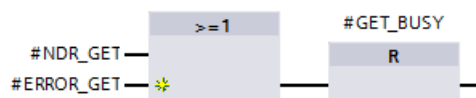
The job to read the data is enabled by a positive edge at the REQ input of the GET instruction. Enabling of the read job is controlled by clock memory M0.7 and the "GET\_BUSY" variable. When the read job is running, "GET\_BUSY" is set to the value "1". Triggering a new read job is not then possible.

Figure 3-10



Only when the current read job has been completed successfully or with an error is "GET\_BUSY" reset to the value "0" so that a new read job can be enabled.

Figure 3-11



#### Setting connection parameters

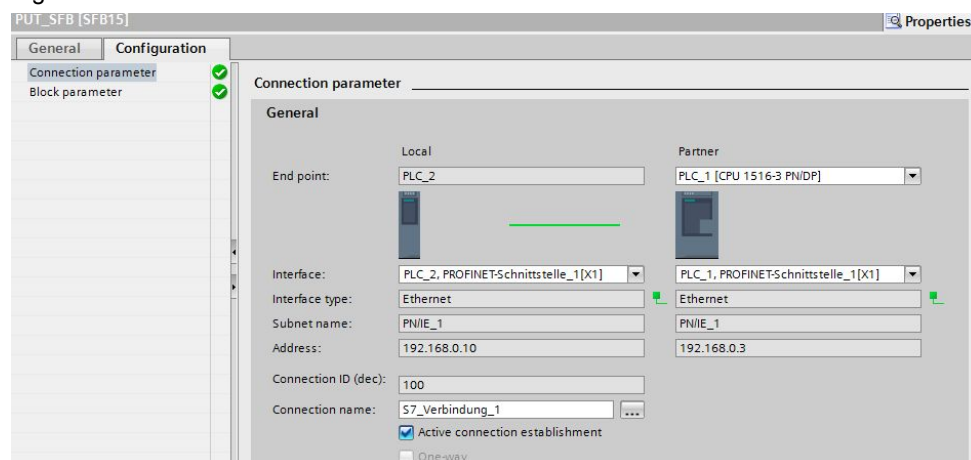
The connection parameters for the GET instruction are set in the inspector window of the program editor. For this you select the GET instruction called in the PUT\_GET (FB1) block. Open the "Configuration" tab in the inspector window. The "Connection Parameters" group is in the "Configuration" tab in the area navigation. This group includes the connection parameters.

Define the connection endpoints of the S7 connection via which the data will be sent and received. In this example you select as partner the S7-1500 CPU that participates passively in establishing the configured S7 connection.

The interface type, interface name and address of the communication partners are entered automatically once you have defined the connection endpoint.

Select the name of the S7 connection via which the data is to be transferred. The ID of the selected S7 connection will be entered automatically at the ID parameter of the GET instruction.

Figure 3-12



**Send data area in the partner CPU**

At the ADDR\_1 input data area of the GET instruction you specify the memory area of the local CPU from which the data is to be read.

Only absolute addressing is permitted. In this example 10 bytes of data are stored in DB2 of the partner CPU starting at address 0:

- P#DB2.DBX0.0 BYTE 10

**Receive data area in the local CPU**

At the RD\_1 input parameter of the GET instruction you specify the memory area of the local CPU to which the data is to be written.

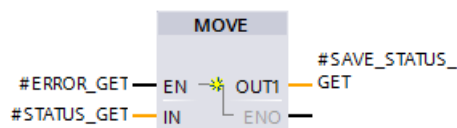
In this example 10 bytes of data are stored in DB2 of the local CPU starting at address 0.

- P#DB2.DBX0.0 BYTE 10

**Error evaluation**

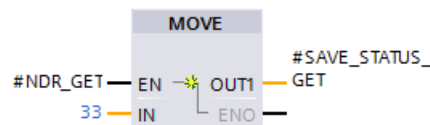
If an error occurs during processing of the GET instruction, the STATUS of the GET instruction is stored in the static variable "SAVE\_STATUS\_GET".

Figure 3-13



If the GET instruction is executed and completed successfully, the "SAVE\_STATUS\_GET" variable is set to the value 33 (dec) = 21 (hex).

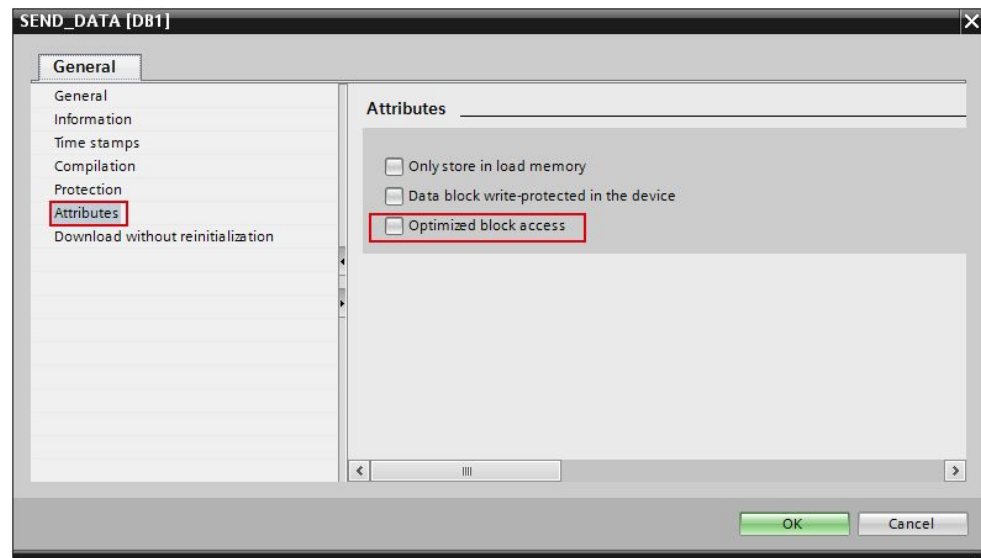
Figure 3-14

**3.4 SEND\_DATA (DB1)**

The data that is transferred to the partner CPU is stored in the SEND\_DATA (DB1) data block of the local CPU.

In the Properties of the SEND\_DATA (DB1) data block you disable the "Optimized block access" function under "Attributes".

Figure 3-15

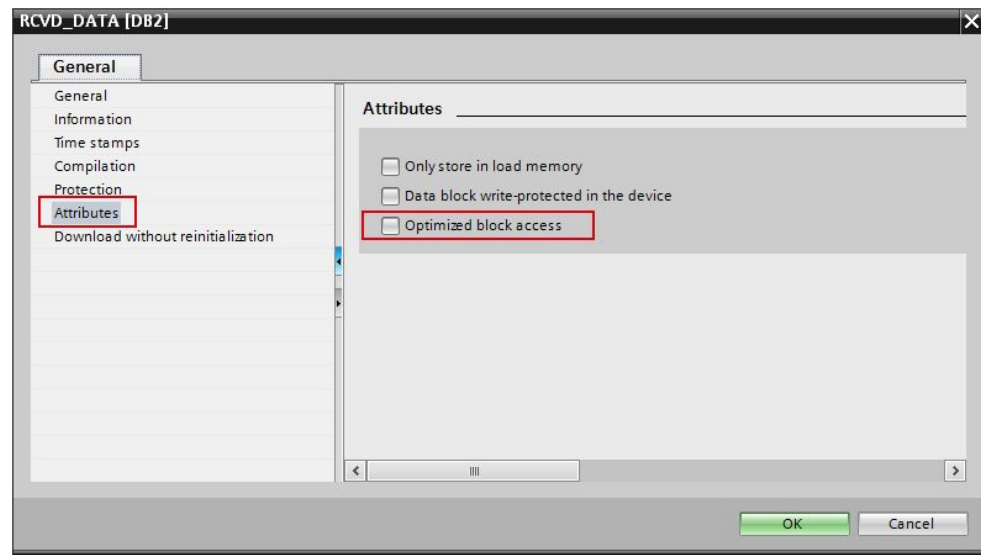


## 3.5 RCVD\_DATA (DB2)

The data received from the partner CPU is stored in the RCVD\_DATA (DB2) data block of the local CPU.

In the Properties of the RCVD\_DATA (DB2) data block you disable the "Optimized block access" function under "Attributes".

Figure 3-16



## 4 User Program of the Passive S7--1500 CPU

No communication instructions for data transfer are called in the user program of the passive S7-1500 CPU.

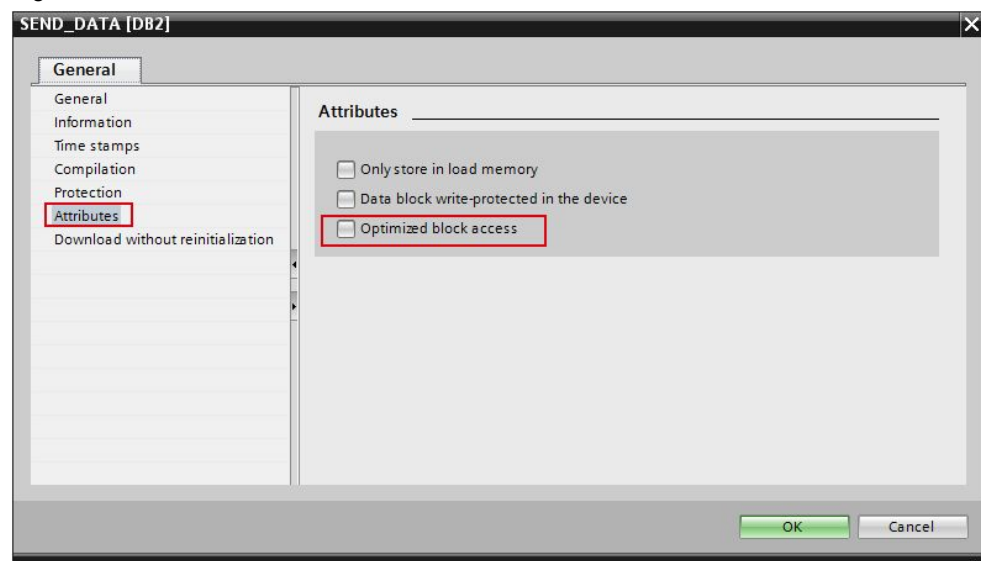
All you need are data blocks in which the sent and received data is stored.

### 4.1 SEND\_DATA (DB2)

The data read by the active S7-1500 CPU from the passive S7-1500 CPU is stored in the SEND\_DATA (DB2) data block.

In the Properties of the SEND\_DATA (DB2) data block you disable the "Optimized block access" function under "Attributes".

Figure 4-1



### 4.2 RCVD\_DATA (DB3)

The data written by the active S7-1500 CPU to the passive S7-1500 CPU is stored in the RCVD\_DATA (DB3) data block.

In the Properties of the RCVD\_DATA (DB2) data block you disable the "Optimized block access" function under "Attributes".

Figure 4-2

