SIEMENS

Ingenuity for life

S7 Communication with PUT/GET

S7-1500 CPUs and S7-1200 CPUs

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

Siemens Industry Online Support

Security Information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
The customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase the customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under http://www.siemens.com/industrialsecurity.

Contents

1 Introduction

You can use the S7 Communication, for example, for data transfer via the integrated PROFINET interface and Industrial Ethernet interface of the S7-1500 CPUs and S7-1200 CPUs.

The following instructions are available for S7 Communication:

- PUT for sending data
- GET for receiving data

In STEP 7 V15.1 (TIA Portal) you will find the above-mentioned instructions in the "Instructions" task card under "Communication > S7 Communication".

The example describes how to configure an S7 connection between an S7-1500 CPU and an S7-1200 CPU to exchange data between the S7-1500 CPU and the S7-1200 CPU using the PUT and GET instructions.

Note The example can also be used for exchanging data between two S7-1500 CPUs or two S7-1200 CPUs.

2 Configuration

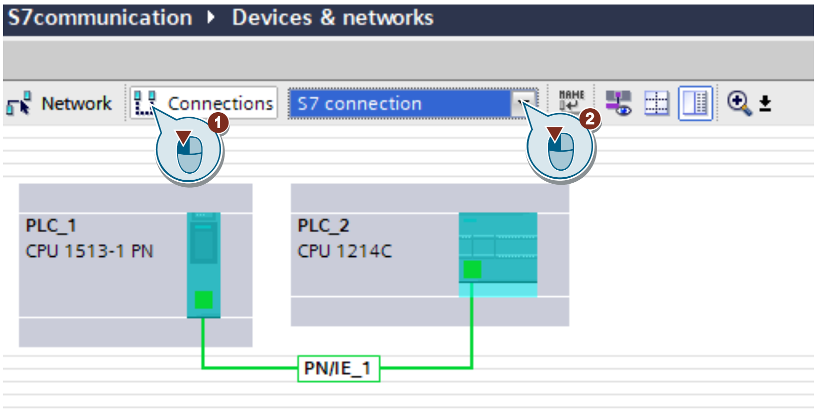
2.1 Configuration of the Hardware

1. In the network view of the "Devices & networks" editor you create the connection partners, an S7-1500 CPU and an S7-1200 CPU, for example.
2. Network the connection partners.

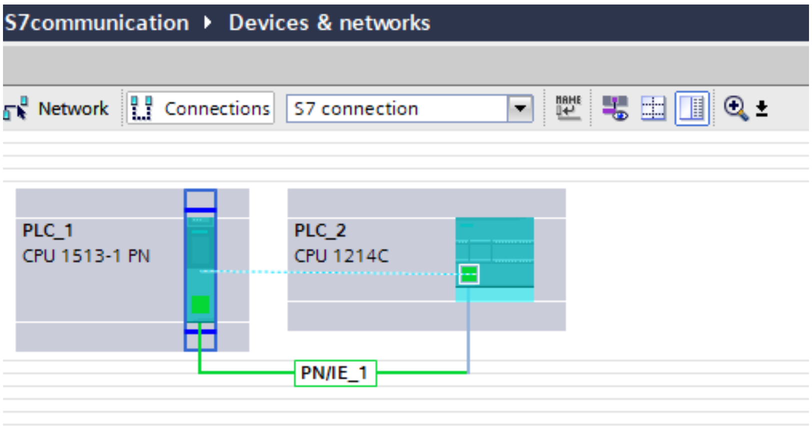
2.2 Configuration of the S7 Connection

Proceed as follows to create an S7 connection between the S7-1500 CPU and the S7-1200 CPU:

1. In the Network view you click the "Connections" icon to enable the Connection mode.
2. Select "S7 connection" as the connection type.



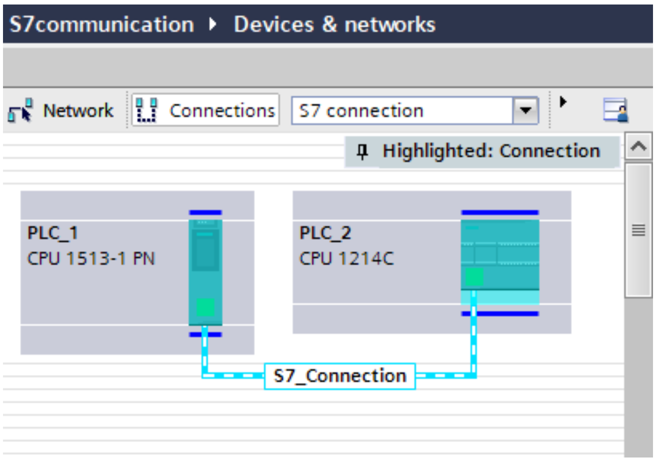
3. With the button held down drag the mouse cursor from the S7 CPU where the S7 connection is to start (active connection establishment) to the S7 CPU where the S7 connection is to finish (passive connection establishment).



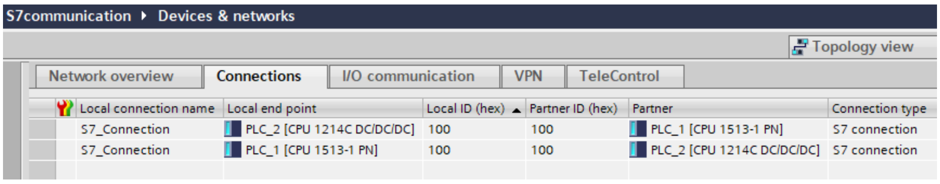
4. Release the mouse button when the cursor is on the target device to create the S7 connection between the S7-1500 CPU and the S7-1200 CPU.

Result

- A specified connection is created.
- The connection path is displayed in the graphical area of the Network view.



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



Local connection name	Local end point	Local ID (hex)	Partner ID (hex)	Partner	Connection type
S7_Connection	PLC_2 [CPU 1214C DC/DC/DC]	100	100	PLC_1 [CPU 1513-1 PN]	S7 connection
S7_Connection	PLC_1 [CPU 1513-1 PN]	100	100	PLC_2 [CPU 1214C DC/DC/DC]	S7 connection

2.3 Properties of the S7 Connection

The "Properties" tab in the inspector window shows the properties of the configured S7 connection.

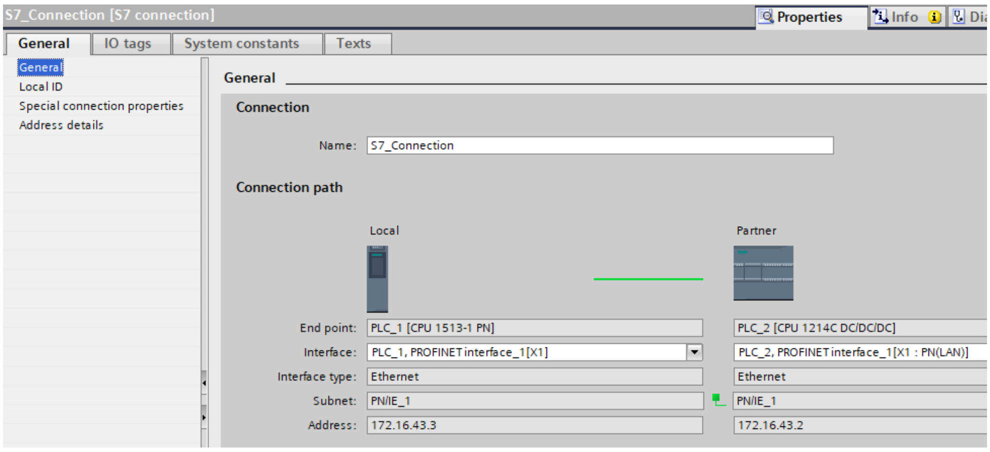
- General connection parameters
- Local ID
- Special connection parameters
- Address details

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 2-1



General

Connection

Name: S7_Connection

Connection path

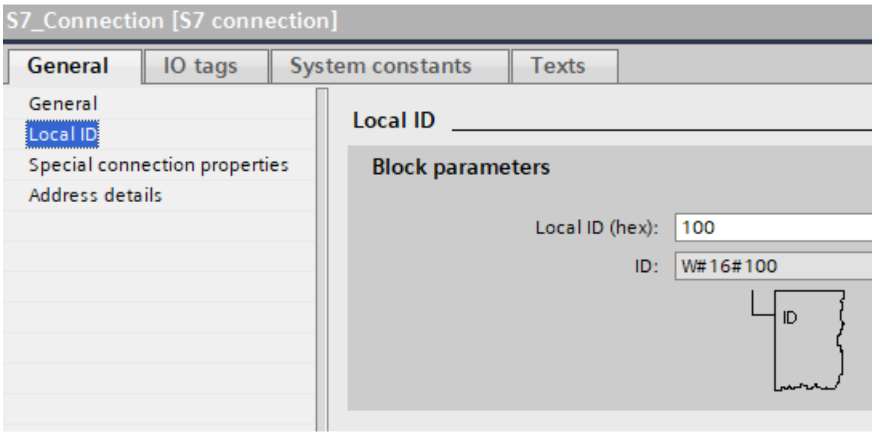
	Local	Partner
End point:	PLC_1 [CPU 1513-1 PN]	PLC_2 [CPU 1214C DC/DC/DC]
Interface:	PLC_1, PROFINET interface_1[X1]	PLC_2, PROFINET interface_1[X1 : PN(LAN)]
Interface type:	Ethernet	Ethernet
Subnet:	PN/IE_1	PN/IE_1
Address:	172.16.43.3	172.16.43.2

Local ID

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

In this example we use the local ID with the value 100 (hex).

Figure 2-2



Local ID

Block parameters

Local ID (hex): 100

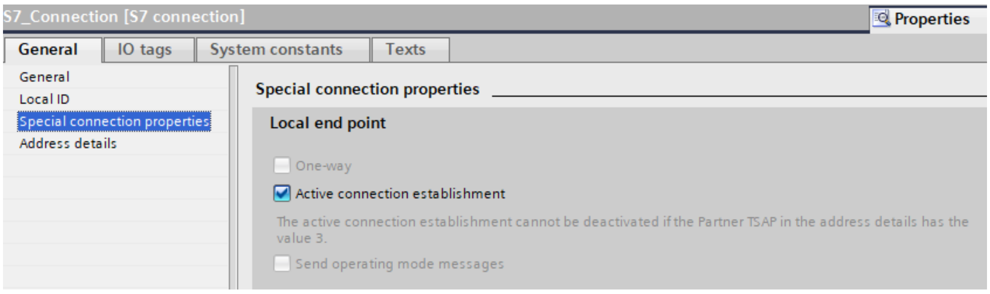
ID: W#16#100

Special connection parameters

The following connection properties are shown here:

- 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 and send and receive actively.
- Send operating mode alarms
  In this example the local partner does not send any operating mode alarms to the connection partner.

Figure 2-3



Special connection properties

Local end point

□ One-way

☑ Active connection establishment

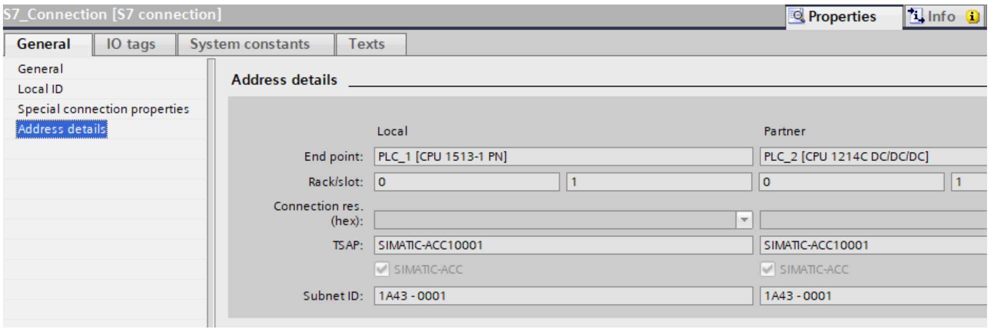
The active connection establishment cannot be deactivated if the Partner TSAP in the address details has the value 3.

□ Send operating mode messages

Address details

The address details of the S7 connection are displayed here. 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 2-4



S7_Connection [S7 connection]

General | IO tags | System constants | Texts

General
Local ID
Special connection properties
Address details

Address details

	Local		Partner	
End point:	PLC_1 [CPU 1513-1 PN]		PLC_2 [CPU 1214C DC/DC/DC]	
Rack/slot:	0	1	0	1
Connection res. (hex):				
TSAP:	SIMATIC-ACC10001		SIMATIC-ACC10001	
	☑ SIMATIC-ACC		☑ SIMATIC-ACC	
Subnet ID:	1A43 - 0001		1A43 - 0001	

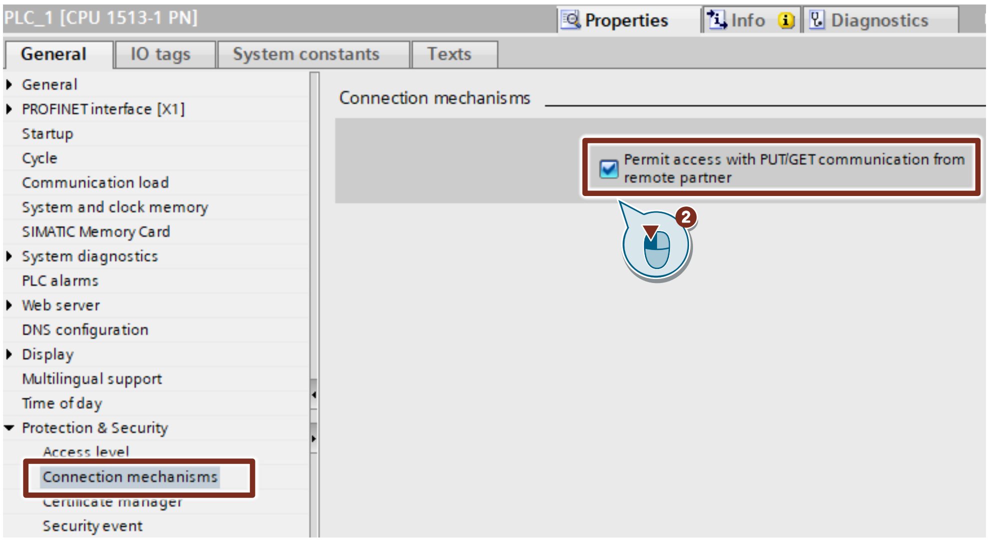
2.4 Permit Access with PUT/GET Communication from Remote Partner

Access from the remote partner via PUT/GET communication must be permitted in both S7 CPUs between which the S7 connection is configured.

S7-1500 CPU

1. In the Device view or Network view of the “Devices & networks" editor you mark the S7-1500 CPU.
   The properties of the S7-1500 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.

Figure 2-5



PLC_1 [CPU 1513-1 PN]

General | IO tags | System constants | Texts

- General
- PROFINET interface [X1]
- Startup
- Cycle
- Communication load
- System and clock memory
- SIMATIC Memory Card
- System diagnostics
- PLC alarms
- Web server
- DNS configuration
- Display
- Multilingual support
- Time of day
- Protection & Security
  - Access level
  - Connection mechanisms
  - Certificate manager
  - Security event

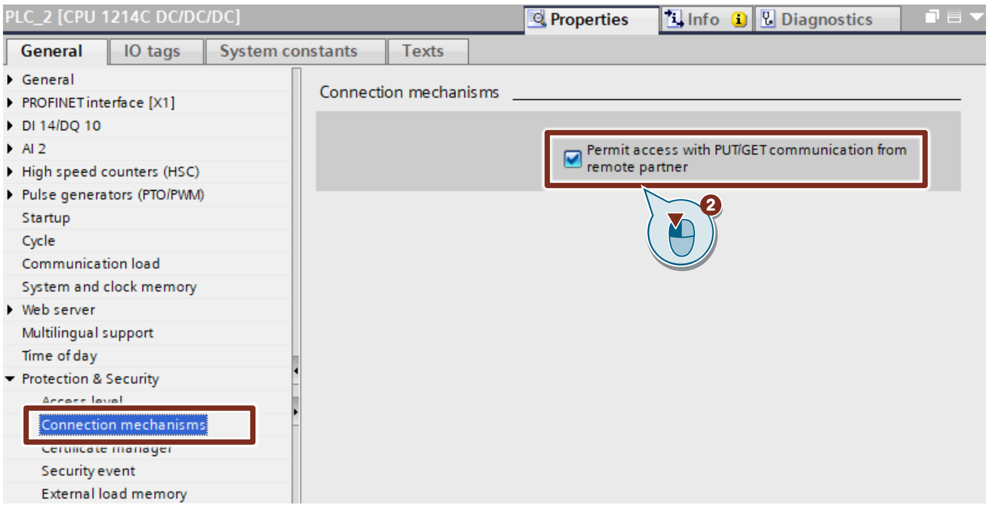
Connection mechanisms

☑ Permit access with PUT/GET communication from remote partner

S7-1200 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 "Protection & Security > Connection mechanisms" you enable the "Permit access with PUT/GET communication from remote partner" function.

Figure 2-6



Connection mechanisms

☑ Permit access with PUT/GET communication from remote partner

3 User Program of the Active S7 CPU

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

Table 3-1

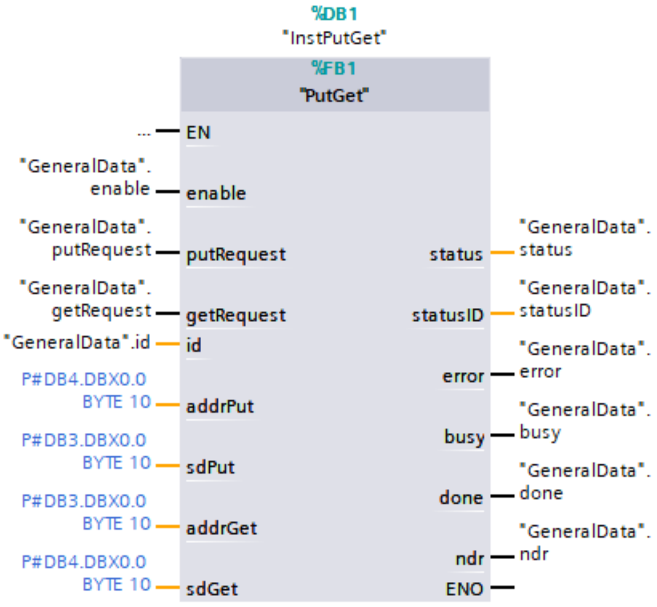
Block	Symbolic name	Description
OB1	Main	The FB1 "PutGet" including the associated instance data block DB1 "InstPutGet" is called cyclically in OB1.
FB1	PutGet	The instructions PUT and GET are called in the FB1 "PutGet" to transfer data via the configured S7 connection.
DB1	InstPutGet	Instance data block of FB1 "PutGet".
DB2	GeneralData	Includes the variables for the parameters of the FB1 "PutGet".
DB3	SendData	The data to be sent to the communication partner with the "PUT" instruction (SendData) is stored in the data block DB3 "SendData".
DB4	RecvData	The data received from the communication partner with the GET instruction (RecvData) is stored in the data block DB4 "RecvData".

3.1 OB1

The FB1 "PutGet" including the associated instance data block DB1 "InstPutGet" is called cyclically in OB1.

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

Figure 3-1


%DB1
"InstPutGet"
%FB1
"PutGet"
... EN
"GeneralData".enable enable
"GeneralData".putRequest putRequest
"GeneralData".getRequest getRequest
"GeneralData".id id
P#DB4.DBX0.0 BYTE 10 addrPut
P#DB3.DBX0.0 BYTE 10 sdPut
P#DB3.DBX0.0 BYTE 10 addrGet
P#DB4.DBX0.0 BYTE 10 sdGet
status "GeneralData".status
statusID "GeneralData".statusID
error "GeneralData".error
busy "GeneralData".busy
done "GeneralData".done
ndr "GeneralData".ndr
ENO

3.2 FB1 "PutGet"

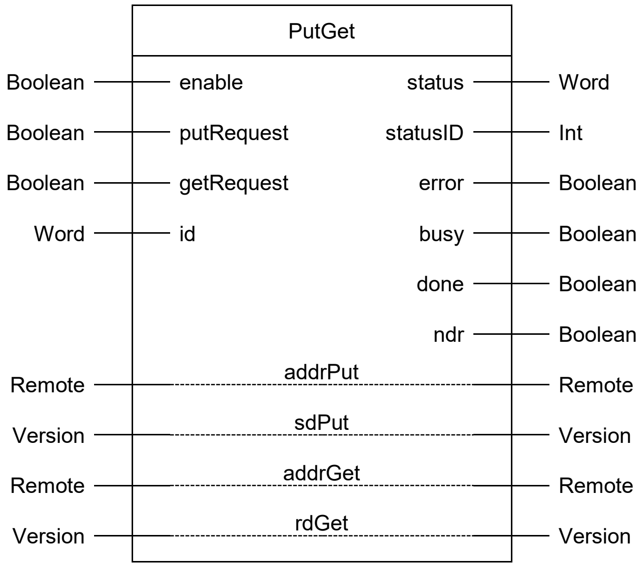
Note The FB1 "PutGet" can be used in S7-1200 CPUs and S7-1500 CPUs.

The FB1 "PutGet" encapsulates the "PUT" and "GET" instruction in a user-friendly shell to execute the following functions:

- Send data to the partner as soon as the input "putRequest" recognizes a positive edge. When the Send job is running, it is not possible to trigger a new Send job.
- Receive data from a partner and store it in the defined receive area as soon as the input "getRequest" recognizes a positive edge. When the Receive job is running, it is not possible to trigger a new Receive job.
- Output the status of the transmission at the "status" output.

3.2.1 Block interface

Figure 3-2



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

Table 3-2

Name	P type	Data type	Comment
enable	IN	Boolean	Enable signal for data exchange
putRequest	IN	Boolean	Control parameter: Enables the job to send data with the "PUT" instruction on a rising edge.
getRequest	IN	Boolean	Control parameter: Enables the job to receive data with the "GET" instruction on a rising edge.
id	IN	Word	Local ID: Addressing parameter for specifying the connection to the partner CPU. **Note** You can set the local ID in the Properties of the S7 connection (see chapter 2.3). In this example we use the local ID with the value 100 (hex).
status	OUT	Word	Status display of the "PUT" and "GET" instructions.
statusID	OUT	Int	Parameter shows which instruction is producing the error (see Table 3-4).
error	OUT	Boolean	Error display
busy	OUT	Boolean	FB is being processed.
done	OUT	Boolean	Status display 0: Job to send data not yet started or is still being executed. 1: Job to send data was executed error-free.
ndr	OUT	Boolean	Status display 0: Job to send receive data not yet started or is still being executed. 1: Job to receive data was executed error-free.
addrPut	IN_OUT	Remote	Pointer to the area to be written to in the partner CPU.
sdPut	IN_OUT	Version	Pointer to the area to be written to in your own CPU.
addrGet	IN_OUT	Remote	Pointer to the area to be read in the partner CPU.
rdGet	IN_OUT	Version	Pointer to the area in your own CPU in which the read data is stored.

Receive data area in the partner CPU

At the input parameter "addrPut" of the FB1 "PutGet" 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 starting at address 0 are stored in DB3 of the partner CPU.

- P#DB4.DBX0.0 BYTE 10

Send data area in the local CPU

At the input parameter "sdPut" of the FB1 "PutGet" 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#DB1.DBX0.0 BYTE 10.

Send data area in the partner CPU

At the input parameter "addrGet" of the FB1 "PutGet" you specify the memory area of the partner CPU from which the data is to be read.

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

- P#DB1.DBX0.0 BYTE 10.

Receive data area in the local CPU

At the input parameter "rdGet" of the FB1 "PutGet" you specify the memory area of the local CPU to which the data is to be written.

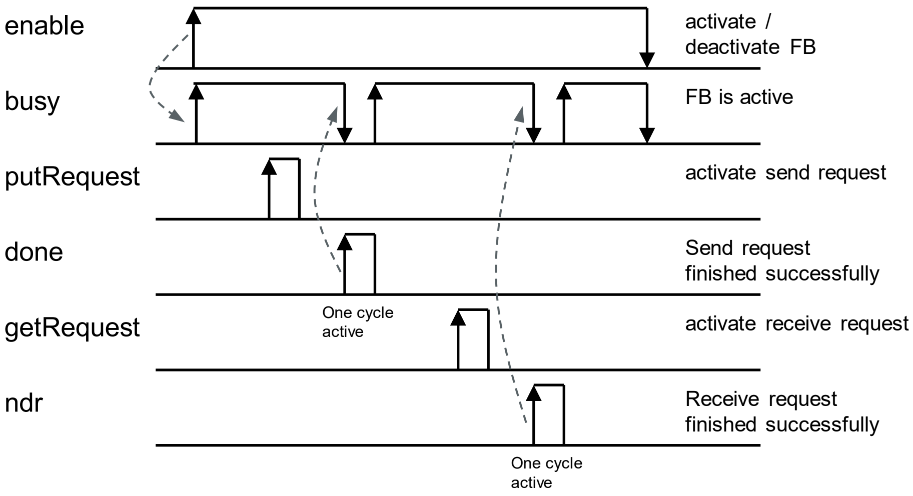
In this example 10 bytes of data are stored starting at address 0 in DB4 of the local CPU.

- P#DB4.DBX0.0 BYTE 10

3.2.2 Flow Chart

The following flow chart shows how the main output parameters are set depending on the input parameters.

Figure 3-3



Enable processing of the FB1 "PutGet"

Processing of the FB1 "PutGet" is started via a positive edge at the input parameter "enable". When processing of the FB1 "PutGet" is started the output parameter busy = true is set.
As long as processing of the FB1 "PutGet" is not started it is not possible to enable any Send or Receive job.

Enable Send job

The job to send the data is enabled by a positive edge at the input parameter "putRequest" of the FB1 "PutGet". When the Send job is running, it is not possible to trigger a new Send job.

If a Send job with the "PUT" instruction terminates successfully and the data is transferred to the Send data area in the partner CPU, the output parameters "done" and "busy" are set to the following values for one cycle:

- done = true
- busy = false

Enable Receive job

The job to receive the data is enabled by a positive edge at the input parameter "getRequest" of the FB1 "PutGet". When the Receive job is running, it is not possible to trigger a new Receive job.

If a Receive job with the "GET" instruction terminates successfully and the data is transferred to the Receive data area in the local CPU, the output parameters "ndr" and "busy" are set to the following values for one cycle:

- ndr = true
- busy = false

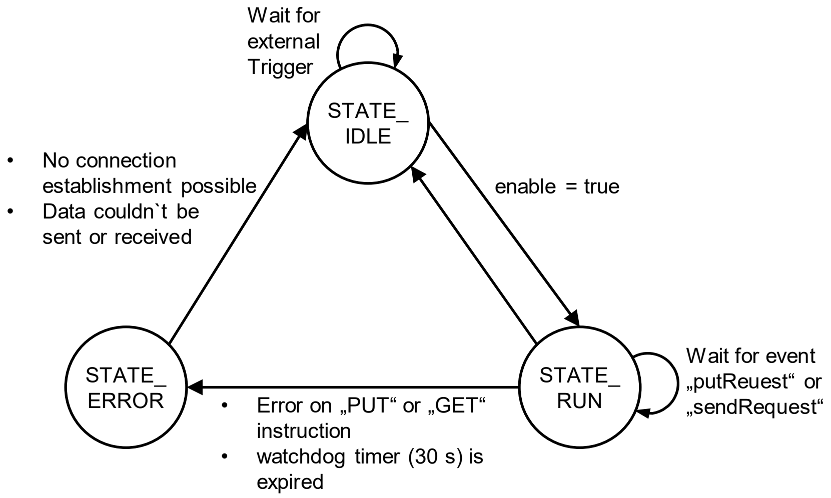
3.2.3 Function

Implementation as state machine

The FB1 "PutGet" is implemented as a state machine. The design model of a state machine is particularly suitable for modeling more complex asynchronous processes, such as communication between partners, which can extend over several cycles.

A certain state is run through cyclically until a transition condition is fulfilled and the machine switches to the next subsequent state. This not only improves the clarity compared to conventional link control, but also makes it easier to find any errors in the program logic more quickly.

Figure 3-4


Wait for external Trigger
STATE_ IDLE
• No connection establishment possible
• Data couldn`t be sent or received
enable = true
STATE_ ERROR
STATE_ RUN
Wait for event „putReuest“ or „sendRequest“
• Error on „PUT“ or „GET“ instruction
• watchdog timer (30 s) is expired

Description of the states

The following table gives an overview of the states and possible transitions.

Table 3-3

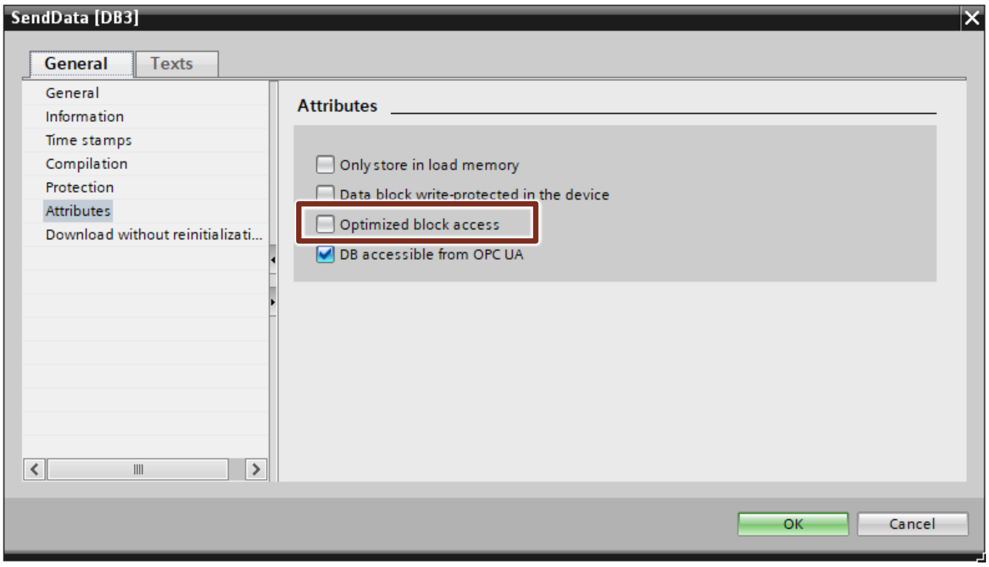
State	Description	Transition condition
STATE_IDLE	In the idle state "STATE_IDLE" the FB has the following properties: • Status variables are reset. • No Send or Receive job is active.	The idle state "STATE_IDLE" is exited if the FB is processed via the parameter enable = true.
STATE_RUN	In the state "STATE_RUN" the FB executes the following actions: • It waits for the triggering of the Send or Receive job to send and receive data via the configured S7 connection. • It monitors whether the "PUT" instruction has executed successfully (DONE = true) or with an error (ERROR = true). As long as the Send job is running no new Send job can be started. • It monitors whether the "GET" instruction has executed successfully (NDR = true) or with an error (ERROR = true). As long as the Receive job is running no new Receive job can be started.	The state "STATE_RUN" is exited when one of the following requirements is fulfilled: • If the Send or Receive job is not yet completed after the watchdog timer (30s) has expired, the FB changes to the state "STATE_ERROR”. • If an error occurs while sending the data with the "PUT" instruction, the FB changes to the state "STATE_ERROR". • If an error occurs while receiving the data with the "GET" instruction, the FB changes to the state "STATE_ERROR".
STATE_ERROR	In the state "STATE_ERROR" the output parameters are supplied with the error information.	The FB changes to the state "STATE_IDLE" without transition condition.

3.3 DB3 "SendData"

The data that is transferred to the partner CPU is stored in the data block DB3 "SendData" of the local CPU.

In the Properties of the DB3 "SendData", under "Attributes" you disable the "Optimized block access" function.

Figure 3-5



Attributes

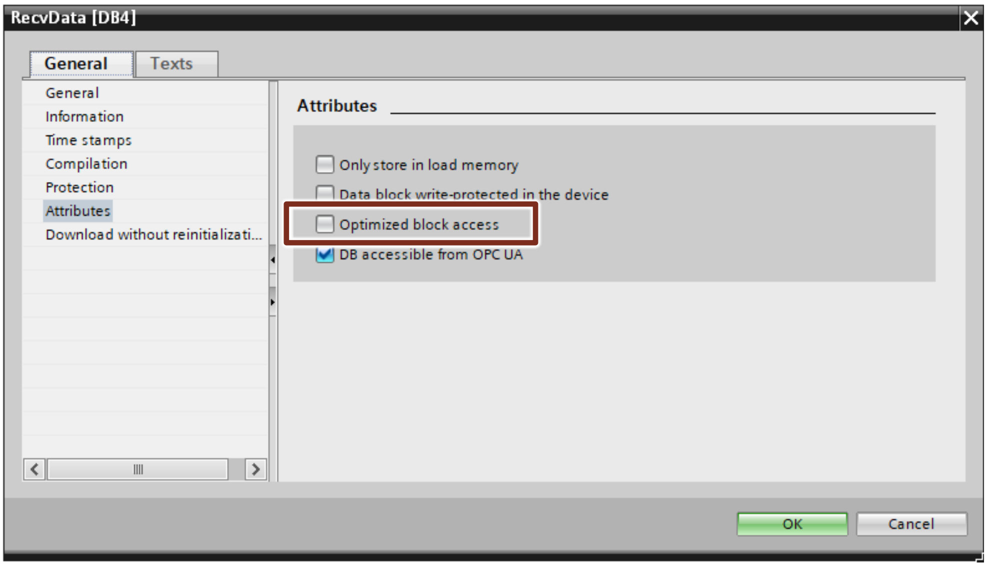
☐ Only store in load memory
☐ Data block write-protected in the device
☐ Optimized block access
☑ DB accessible from OPC UA

3.4 DB4 "RecvData"

The data received from the partner CPU is stored in the data block DB4 "RecvData" of the local CPU.

In the Properties of the DB4 "RecvData", under "Attributes" you disable the "Optimized block access" function.

Figure 3-6



Attributes

☐ Only store in load memory
☐ Data block write-protected in the device
☐ Optimized block access
☑ DB accessible from OPC UA

3.5 Error Evaluation

The following table shows the values and meanings of the output parameters "status" and "statusID" of the FB1 "PutGet".

Table 3-4

statusID value (dec)	Meaning	status value (dec)	Comment
1	Internal error in the function block	16#8101	Function block could not be enabled
1	Internal error in the function block	16#8102	The watchdog timer (30s) has expired because the Send or Receive job has not been completed.
1	Internal error in the function block	16#8103	The Send or Receive job could not be started because enable = 0 or a job is already running.
2	PUT error	–	The status of "PUT" is output at the "status" output of the function block.
3	GET error	–	The status of "GET" is output at the "status" output of the function block.

Note If an error occurs, the values of the output parameters are set for only one cycle. This is why it is recommended to save the values of the output parameters "status" and "statusID" if error = true.

4 User Program of the Passive S7 CPU

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

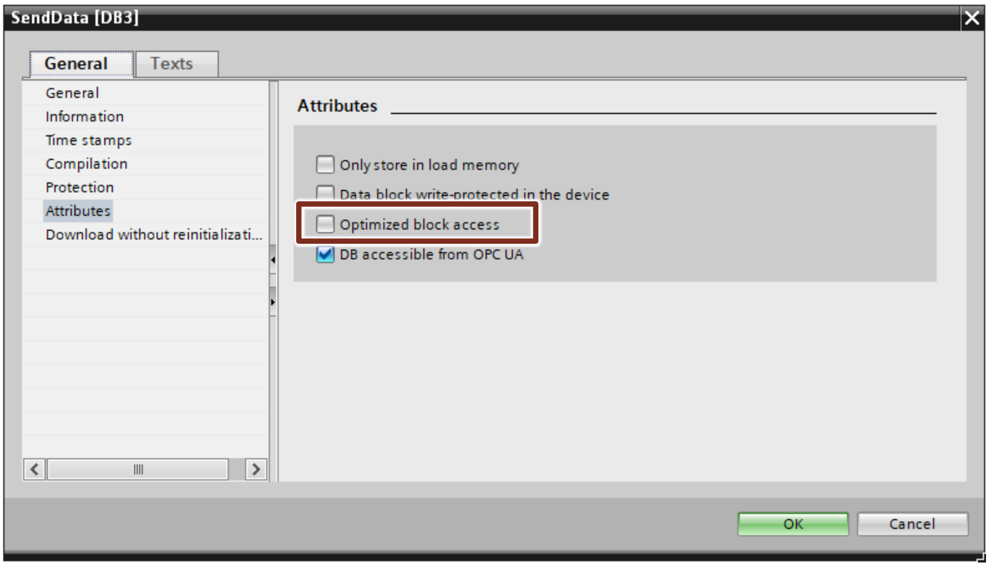
All you need are data blocks in which the sent and received data is stored.

4.1 DB3 "SendData"

The data that is transferred to the partner CPU is stored in the data block DB3 "SendData" of the local CPU.

In the Properties of the DB3 "SendData", under "Attributes" you disable the "Optimized block access" function.

Figure 4-1



Attributes

☐ Only store in load memory

☐ Data block write-protected in the device

☐ Optimized block access

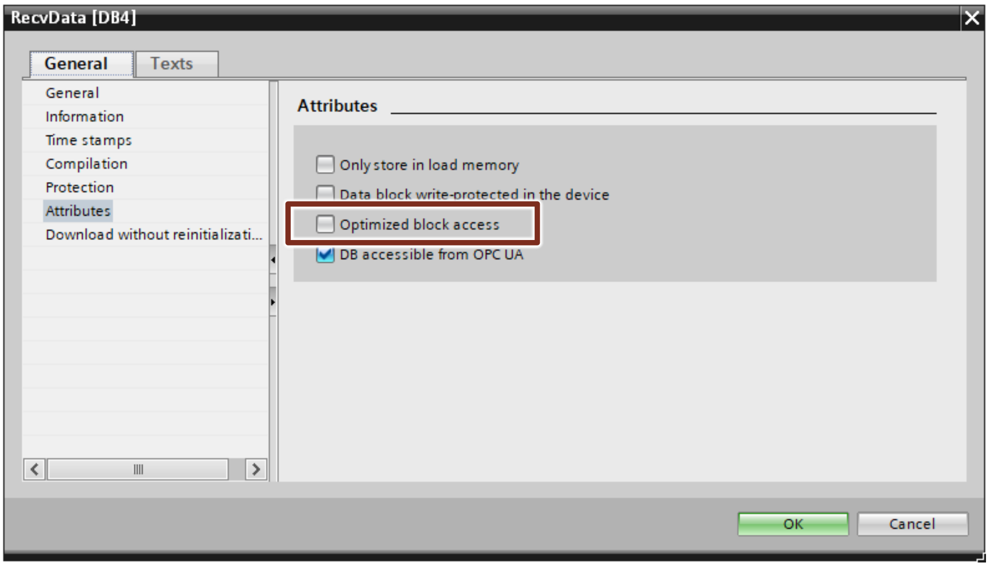
☑ DB accessible from OPC UA

4.2 DB4 "RecvData"

The data received from the partner CPU is stored in the data block DB4 "RecvData" of the local CPU.

In the Properties of the DB4 "RecvData", under "Attributes" you disable the "Optimized block access" function.

Figure 4-2



RecvData [DB4]

Attributes

□ Only store in load memory

□ Data block write-protected in the device

□ Optimized block access

☑ DB accessible from OPC UA

5 Information

5.1 "PUT" Instruction

The "PUT" instruction is called in the FB1 "PutGet". This is to be found in the "Instructions" task card under "Communication".

You use the "PUT" instruction 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 CPU under "Protection & Security > Connection mechanisms".

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

Detailed information about the "PUT" instruction is given in the manual entitled "SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1".

5.2 "GET" Instruction

The "GET" instruction is called in the FB1 "PutGet". This is to be found in the "Instructions" task card under "Communication > S7 Communication".

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

Detailed information about the "PUT" instruction is given in the manual entitled "SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1".

5.3 Configuration of the "PUT" and "GET" Instructions

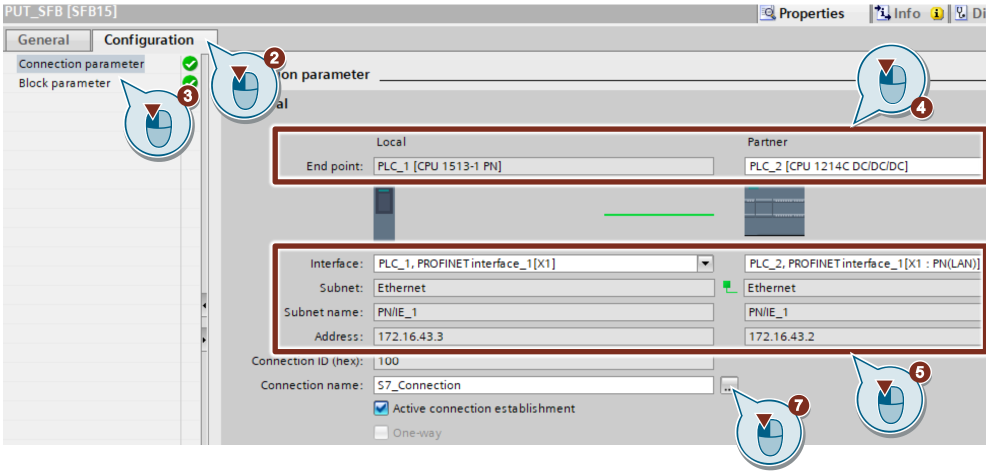
Note If you use the FB1 "PutGet", you specify the connection ID of the configured S7 connection at the input parameter "id". The connection ID is used internally in the FB on the instructions "PUT" and "GET".

You have the option of having the connection ID entered automatically by TIA Portal at the input parameter "ID" of the "PUT" and "GET" instructions.

You configure the "PUT" and "GET" instructions in the inspector window of the program editor. Proceed as instructed below:

1. Mark the call of the "PUT" or "GET" instruction.
2. Open the "Configuration" tab in the inspector window.
3. In the area navigation of the "Configuration" tab you select the "Connection Parameters" group. This group includes the connection parameters.
4. Define the connection endpoints of the S7 connection via which the data will be sent and received.
5. The following parameters of the communication partners are entered automatically once you have defined the connection endpoint.
   - Interface
   - Subnet
   - Subnet name
   - Address
6. Select the name of the S7 connection via which the data is to be transferred.

Figure 5-1



	Local	Partner
End point:	PLC_1 [CPU 1513-1 PN]	PLC_2 [CPU 1214C DC/DC/DC]
Interface:	PLC_1, PROFINET interface_1[X1]	PLC_2, PROFINET interface_1[X1 : PN(LAN)]
Subnet:	Ethernet	Ethernet
Subnet name:	PN/IE_1	PN/IE_1
Address:	172.16.43.3	172.16.43.2

Connection ID (hex): 100

Connection name: S7_Connection

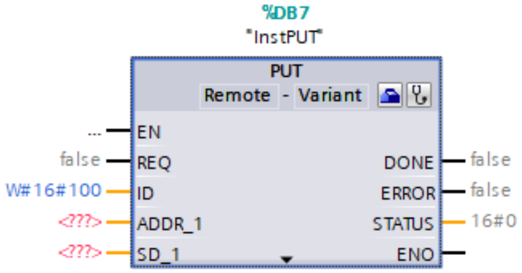
☑ Active connection establishment

□ One-way

Result

The ID of the selected S7 connection will be entered automatically at the input parameter "ID" of the "PUT" or "GET" instruction.

Figure 5-2


%DB7
"InstPUT"
PUT
Remote - Variant
... EN
false REQ
W#16#100 ID
<???> ADDR_1
<???> SD_1
DONE false
ERROR false
STATUS 16#0
ENO