FETCH/WRITE service in an S7-300/400 CPU via the integrated Ethernet interface

Fetch/Write function block FB210/220

Application • October 2010

Applikationen & Tools

Answers for industry.

SIEMENS

Industry Automation and Drives Technologies Service & Support Portal

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drives Technologies. The following link takes you directly to the download page of this document.

http://support.automation.siemens.com/WW/view/en/<Item-ID>

If you have any questions concerning this document please e-mail us to the following address:

online-support.automation@siemens.com

Fetch/Write function block FB210/220
V2.2, Entrys-ID: 23592043

SIEMENS

SIMATIC
Fetch/Write function block
FB210/220

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Preface

Objective of the application

In today's automation technology old systems are upgraded step by step from SIMATIC S5 to SIMATIC S7. However, the interfaces for the data exchange on the peer (for example, existing HMI systems) are to be kept. The FETCH/WRITE service is still very frequently used for the communication via Industrial Ethernet. The communication processors (CP343-1/CP443-1) of SIMATIC S7 support the services FETCH and WRITE for data exchange on the following protocols:

- TCP
- ISO-on-TCP
- ISO transport

However, CPUs with integrated Industrial Ethernet interface are increasingly used in S7-300 and S7-400 stations to connect these to the Industrial Ethernet network and to exchange data with other nodes in the network. The FETCH/WRITE services have not been implemented here.

This application shows how any FETCH/WRITE client can access the process data via the integrated Industrial Ethernet interface of an CPU of the SIMATIC S7.

Main contents of this application

The following main points are discussed in this application:

- Implementing the FETCH and WRITE services in a SIMATIC S7 CPU with integrated Industrial Ethernet interface for the open communication via Industrial Ethernet with TCP and ISO-on-TCP protocol.
- Configuring the communication connections using the FETCH and WRITE services for the SIMATIC NET OPC server in the PC station.

Delimitation

This application does not include a description of how the FETCH and WRITE services are used with the SIMATIC S7 Industrial Ethernet CPs via S5-compatible communication connections.

Basic knowledge of this topic is required.

Structure of the document

The documentation of this application is divided into the following main parts.

Part	Description
Application Description	You are provided with a general overview of the contents. You are informed on the used components (standard hardware and software components and the specially created user software).
Principles of Operation and Program Structures	This part describes the detailed functional sequences of the involved hardware and software components, the solution structures and – where useful – the specific implementation of this application. It is only required to read this part if you want to familiarize with the interaction of the solution components to use these components e.g. as a basis for own developments.
Structure, Configuration and Operation of the Application	This part takes you step by step through structure, important configuration steps, startup and operation of the application.
Appendix	This part of the documentation includes further information, e.g. bibliographic references, glossaries, etc.

Industry Automation und Drives Technologies Service & Support Portal

This entry is from the internet service portal of Siemens AG, Industry Automation and Drives Technologies. Clicking the link below directly displays the download page of this document.

http://support.automation.siemens.com/WW/view/en/23592043

Table of Contents

Application Description

Content

You are provided with an overview of how a PC station can access the process data of an SIMATIC S7 CPU with integrated Industrial Ethernet interface using the FETCH and WRITE services. You are informed on the used components (standard hardware and software components and the function blocks that have been created especially for the FETCH and WRITE services).

The performance data displayed illustrate the performance capability of this application.

1 Automation Task

1.1 Overview

The following figure illustrates the automation task.

Figure 1-1 Automation task
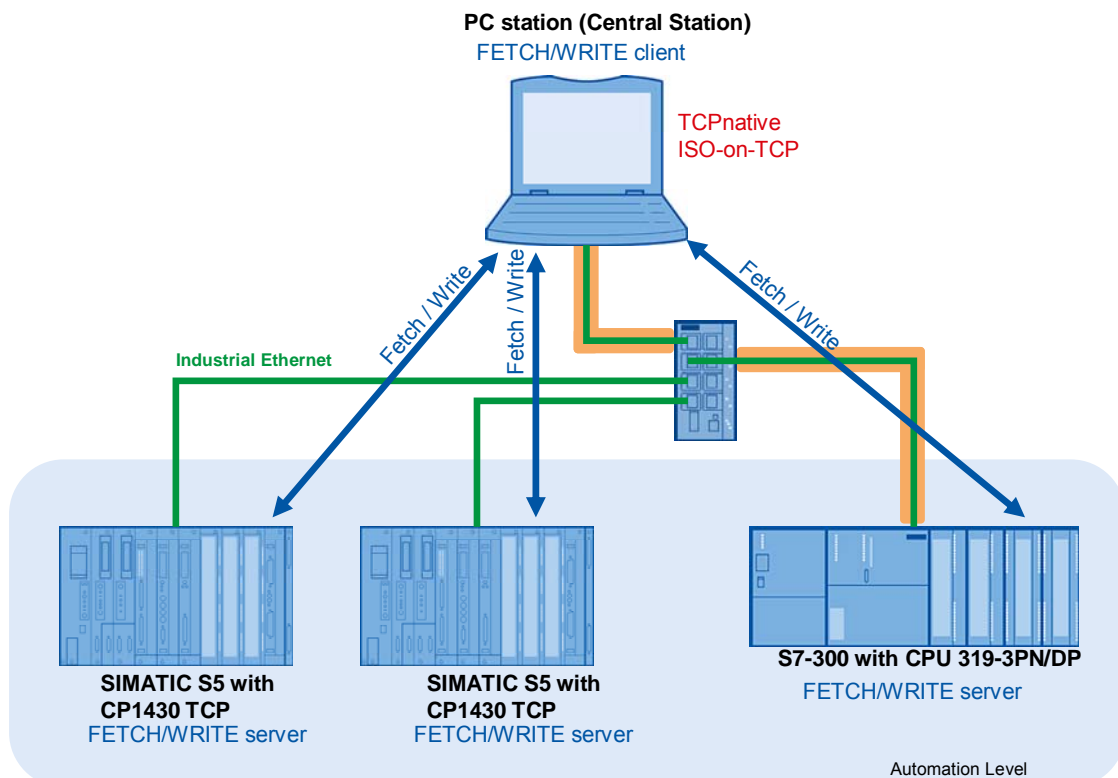
1.2 Description of the automation problem

In the automation level of an already existing plant S5 stations are used with CP1430 TCP. The communication between S5 stations and central station occurs via the ISO-on-TCP protocol using the FETCH and WRITE services. The S5 stations are only replaced by S7. A CPU with integrated Industrial Ethernet interface is used in the S7 station. Since the control station is to be kept, the FETCH/WRITE service has to be implemented in the S7 station. The control station actively establishes the communication connection to the S7 station for the bi-directional data exchange.

Note   If the communication between the S5 station and the central station occurred via ISO transport protocol, this program example can be used if the configuration of the central station is adjusted.

1.3 Requirements

Controller requirements

In this application, a CPU 319-3 PN/DP is used in the S7 station. It supports open communication via Industrial Ethernet. Consequently, the following data transmission protocols can be used:

- TCP native
- ISO-on-TCP
- UDP

The **FETCH and WRITE** services have to be implemented in the S7 program of the CPU 319-3 PN/DP. In this application they are realized via the **TCP native** and **ISO-on-TCP protocol**, since for open communication via Industrial Ethernet the data transfer via the **ISO transfer protocol is not supported**.

Here the CPU 319-3 PN/DP acts as a **FETCH/WRITE server**, i.e. it is passively involved in establishing the communication connections with the FETCH / WRITE service.

In this program example **8176 bytes = 4088 words of net data** are transmitted with the FETCH and WRITE services.

Note   The SIMATIC S7 supports the FETCH and WRITE services only passive via the TCP native and ISO-on-TCP protocol. Furthermore, the SIMATIC S7 CPUs with integrated Industrial Ethernet interface do not support the S5 compatible communication.

Requirements to the PC station

The PC station must support the communication via the TCP native or the ISO-on-TCP protocol.

The PC station acts as a FETCH/WRITE client, i.e. it actively establishes the communication connections with the FETCH and WRITE services.

In this application the SIMATIC NET OPC Scout is used as OPC client.

2 Automation Solution

2.1 Overview of the overall solution

Display

The following figure displays the most important components of the solution:

Table 2-1 Automation solution
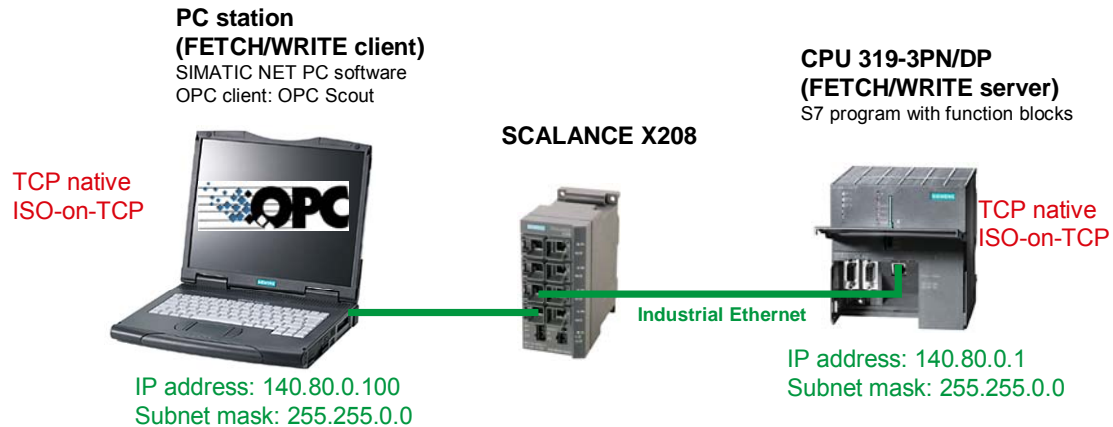
Design

The CPU 319-3 PN/DP and the PC station including the communication connections are configured and programmed with STEP 7. The CPU 319-3 PN/DP and the PC station are located in the same IP subnet. In this example, S7 station and PC station are located in Class B IP subnet 140.80.0.0 (see Table 2-1 Automation solution).

The SIMATIC NET PC software is installed on the PC station. The scope of delivery of the SIMATIC NET PC software includes the SIMATIC NET OPC server. It is used to access the process data of the CPU 319-3 PN/DP via the TCP or ISO-on-TCP connections using the FETCH and WRITE services.

Furthermore, the SIMATIC NET PC software includes OPC Scout as an OPC client. It is used for testing the S7 program.

2.2 Automation solution

This application shows how a PC station can access the process data of a CPU 319-3 PN/DP via the integrated Industrial Ethernet interface of this CPU using the FETCH and WRITE services.

The data between FETCH/WRITE server (CPU 319-3 PN/DP) and FETCH/WRITE client (PC station) are exchanged via four unspecified communication connections, i.e.: via two TCP respectively and two ISO-on-TCP connections. Since the FETCH and WRITE services are used, separate TCP and ISO-on-TCP connections are required for sending and receiving the data.

Via the first communication connection, the PC station reads out the data of the CPU 319-3 PN/DP (FETCH job).

Via the second communication connection, the PC station writes the data to the CPU 319-3 PN/DP (WRITE job).

The communication connections are actively established by the FETCH/WRITE client.

2.3 Required hardware and software components

Hardware components

Table 2-1 Hardware components

Component	Qty.	MLFB / order number	Hinweis
CPU 319-3 PN/DP	1	6ES7 318-2EL00-0AB0	FW V2.7
SCALANCE X208	1	6GK5208-0BA10-2AA3	
SIMATC Field PG	1		

Standard software components

Table 2-2 Software components

Component	Qty.	MLFB / order number	Note
SOFTNET S7 Lean	1	6GK1704-1LW71-3AA0	V7.1 (Edition 2008)
STEP7 V5.4 SP4	1	6ES7810-4CC08-0YA5	

Example files and projects

The following list contains all files and projects used in this example.

Table 2-3 Files and projects

Component	Note
23592043_FetchWrite_PNCPU_v21.zip	This zip file contains the STEP 7 project for configuring the CPU319-3PN/DP and PC station and the OPC Scout test project.
23592043_FetchWrite_PNCPU_v21_e.pdf	This document describes the functionality and the startup of the STEP 7 project.

2.4 Performance Data

Following table shows the size of the function blocks FB210 „FW_TCP" and FB220 „FW_IOT" as well as the size of the corresponding instance data blocks.

Table 2-4 Performance Data

Block	Symbolic name	Size in the work memory of the CPU
FB210	FW_TCP	6778 byte
FB220	FW_IOT	7346 byte
DB210	Instance data block of the FB210	8384 byte
DB220	Instance data block of the FB220	16580 byte

2.5 Alternative solutions

2.5.1 Hardware alternatives in the S7 station

Furthermore, the following hardware components can be used to connect an S7 station to any FETCH/WRITE client.

Table 2-5 Alternative hardware components

Component	Qty.	MLFB / order number	Hinweis
CPU 315-2 PN/DP	1	6ES7315-2EH14-0AB0	Firmware V3.1 and higher
CPU 315F-2 PN/DP	1	6ES7315-2FJ14-0AB0	Firmware V3.1 and higher
CPU 317-2 PN/DP	1	6ES7317-2EK13-0AB0	Firmware V2.5 and higher
CPU 317-2 PN/DP	1	6ES7317-2EK14-0AB0	
CPU 317F-2 PN/DP	1	6ES7317-2FK13-0AB0	Firmware V2.5 and higher
CPU 317F-2 PN/DP	1	6ES7317-2FK14-0AB0	
CPU 319-3 PN/DP	1	6ES7318-3EL00-0AB0	
CPU 319F-3 PN/DP	1	6ES7318-3FL00-0AB0	
CPU 319F-3 PN/DP	1	6ES7318-3FL01-0AB0	
CPU 414-2 PN	1	6ES7412-2EK06-0AB0	
CPU 414-3 PN/DP	1	6ES7414-3EM05-0AB0	
CPU 414-3 PN/DP	1	6ES7414-3EM06-0AB0	
CPU 414F-3 PN/DP	1	6ES7414-3FM06-0AB0	
CPU 416-3 PN/DP	1	6ES7416-3ER05-0AB0	
CPU 416-3 PN/DP	1	6ES7416-3ES06-0AB0	
CPU 416F-3 PN/DP	1	6ES7416-3FR05-0AB0	
CPU 416F-3 PN/DP	1	6ES7416-3FS06-0AB0	
CP443-1 Advanced	1	6GK7443-1EX40-0XE0	
CP443-1 Advanced	1	6GK7443-1EX41-0XE0	
CP443-1 Advanced	1	6GK7443-1GX20-0XE0	

2.5.2   Hardware alternatives in the PC station

In the PC station you can alternatively use a CP1613 (A2), CP1623, CP1612 or CP1512 instead of the normal standard Ethernet network card.

Note   If the PC station only supports the communication via the ISO transfer protocol or in the S5 station a communication processor (e.g. CP143, CP1430 TF or CP535) was used, which only supports the ISO transport protocol, then you must employ a communication processor in the SIMATIC S7 (CP343-1/CP443-1). It supports the ISO transport protocol including the FETCH and WRITE services.

2.5.3   Software alternatives

Table 2-6 Alternative software components

Component	Qty.	MLFB / order number	Note
SOFTNET S7 IE	1	6GK1704-1CW71-3AA0	When using the following hardware component: • Standard Ethernet network card • CP1612 • CP1615
S7-1613	1	6GK1716-1CB71-3AA0	When using the following hardware component: • CP1613 • CP1613 A2 • CP1623

Principles of Operation and Program Structures

Content

This part describes the detailed function processes of the involved hardware and software components, the solution structures and – where useful – the specific implementation of this application.

It is only required to read this part if you want to know how the individual solution components interact.

3 FETCH/WRITE communication services

3.1 Principle of the communication services FETCH and WRITE

The FETCH and WRITE services are used for the data exchange by means of the following protocols:

- TCP

- ISO-on-TCP (TCP with RFC1006) and

- ISO Transport protocol

Note
> The SIMATIC S7 CPUs with integrated Industrial Ethernet interface do not support the ISO transport protocol. Here the FETCH and WRITE services for the data exchange are only supported via TCP and ISO-on-TCP protocol.

Consequently, this communication service corresponds to "Transport Layer" 4 of the ISO/OSI reference model.

Figure 3-1 ISO/OSI reference model
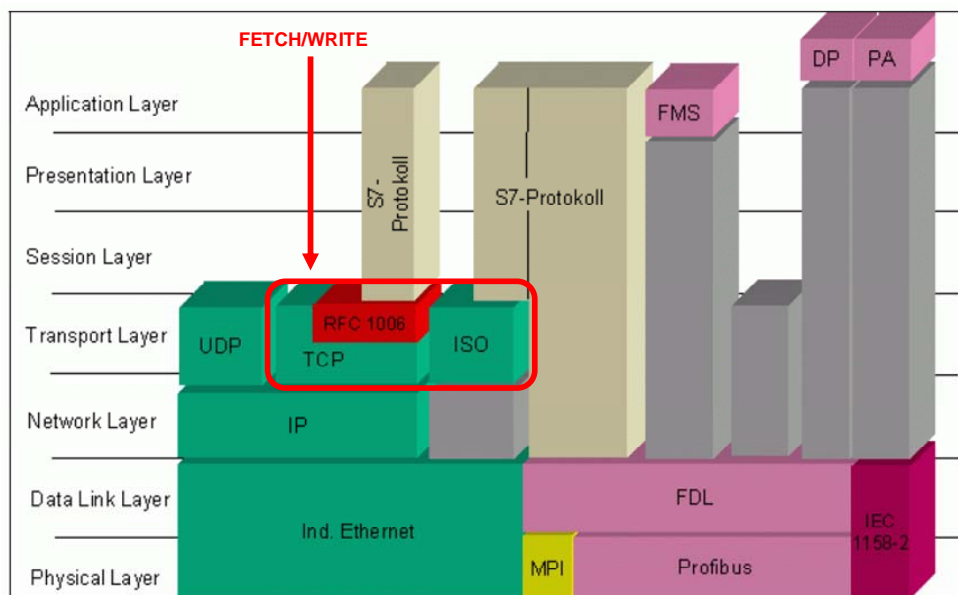
The communication of the FETCH and WRITE communication services is a server-client communication.

The FETCH/WRITE client actively establishes the communication connection.
It requests the data from the FETCH/WRITE server by means of a FETCH request.
The server responds with a positive acknowledgement of the FETCH request and
sends the requested data. Otherwise, the acknowledgement of the FETCH
response is negative.

The FETCH/WRITE client sends a WRITE request with the data required in the
FETCH/WRITE server. If the data has been successfully transmitted, the
FETCH/WRITE server responds with a positive acknowledgement. Otherwise, the
acknowledgement of the WRITE request is negative.

Figure 3-2 FETCH/WRITE message structure
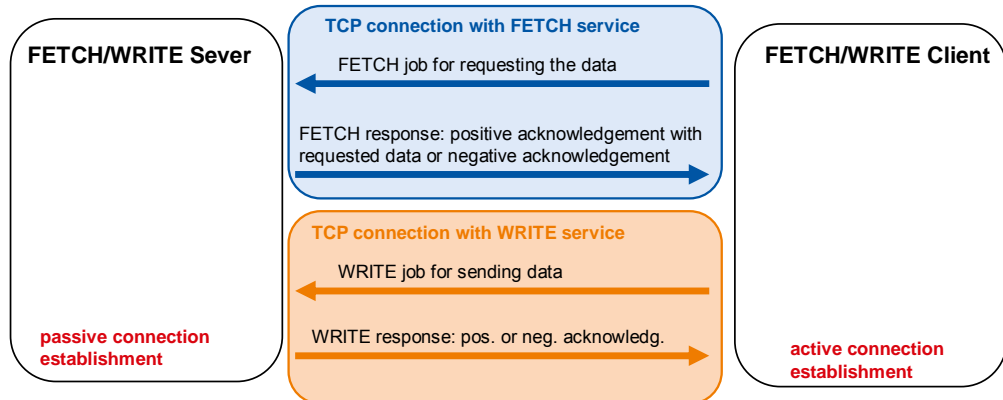
3.2 Structure FETCH/WRITE Header telegram

3.2.1 Write job

In Table 3-1 the structure of the request telegram for the WRITE job is described.

Request telegram

Table 3-1 Request telegram for WRITE job

Byte	Data type	Description	Value
0	WORD	System ID „S5"	W#16#5335
2	BYTE	Length of the Request telegram	W#16#10
3	BYTE	OP-Code ID	W#16#01
4	BYTE	Length of the OP-Code	W#16#03
5	BYTE	OP-Code	W#16#03
6	BYTE	ORG field	W#16#03
7	BYTE	Length of the ORG field	W#16#08
8	BYTE	ORG ID	Valid memory area: • W#16#01 = DB • W#16#02 = bit memory address area • W#16#03 = PII (process image of the inputs) • W#16#04 = PIQ (process image of the outputs)
9	BYTE	DB, from which the source data are taken or to which the destination data are transferred	0 bis 255 (dez.)
10	INT	Start address: Number of data word, memory byte or output / input byte in process image of the inputs / outputs (PII / PIQ) from which the data are taken or written to	see chapter 3.2.4 „Start address"
12	INT	Length of the source / destination data field	see chapter 3.2.3 "Length of the Source / Destination data field"
14	BYTE	Empty field	W#16#FF
15	BYTE	Length of the empty field	W#16#2
16…	User data		

Response telegram

In Table 3-2 the structure of the response telegram for the WRITE job is described.

Table 3-2 Response telegram for WRITE job

Byte	Data type	Description	Value
0	WORD	System ID „S5"	W#16#5335
2	BYTE	Length of the Response telegram (acknowledgement)	W#16#10
3	BYTE	OP-Code ID	W#16#01
4	BYTE	Length of the OP-Code	W#16#03
5	BYTE	OP-Code	W#16#04
6	BYTE	Acknowledgement block	W#16#0F
7	BYTE	Length of the acknowledgement block	W#16#03
8	BYTE	Error number	see 3.2.5 „Error number response telegram"
9	BYTE	Empty field	W#16#FF
10	BYTE	Length of the empty field	W#16#7
11	BYTE	Free	
12	BYTE	Free	
13	BYTE	Free	
14	BYTE	Free	
15	BYTE	Free	

3.2.2 FETCH job

In Table 3-3 the structure of the request telegram for the FETCH job is described.

Request telegram

Table 3-3 Request telegram for FETCH job

Byte	Data type	Description	Value
0	WORD	System ID „S5"	W#16#5335
2	BYTE	Length the of Request telegram	W#16#10
3	BYTE	OP-Code ID	W#16#01
4	BYTE	Length of the OP-Code	W#16#03
5	BYTE	OP-Code	W#16#05
6	BYTE	ORG field	W#16#03
7	BYTE	Length of the ORG field	W#16#08
8	BYTE	ORG ID	Valid memory area:: • W#16#01 = DB • W#16#02 = bit memory address area • W#16#03 = PII (process image of the inputs) • W#16#04 = PIQ (process image of the outputs)
9	BYTE	DB, from which the source data are taken or to which the destination data are transferred	0 bis 255 (dez.)
10	INT	Start address: Number of data word, memory byte or output / input byte in process image of the inputs / outputs (PII / PIQ) from which the data are taken or written to	see chapter 3.2.4 „Start address"
12	INT	Length of the source / destination data field	see chapter 3.2.3 "Length of the Source / Destination data field"
14	BYTE	Empty field	W#16#FF
15	BYTE	Length of the empty field	W#16#2

Response telegram

In Table 3-4 the structure of the response telegram for the FETCH job is described.

Table 3-4 Response telegram for FETCH job

Byte	Data type	Description	Value
0	WORD	System ID „S5"	W#16#5335
2	BYTE	Length of the Response telegram (acknowledgement)	W#16#10
3	BYTE	OP-Code ID	W#16#01

Byte	Data type	Description	Value
4	BYTE	Length of the OP-Code	W#16#03
5	BYTE	OP-Code	W#16#06
6	BYTE	Acknowledgement block	W#16#0F
7	BYTE	Length of the Acknowledgement block	W#16#03
8	BYTE	Error number	see 3.2.5 „Error number response telegram"
9	BYTE	Empty field	W#16#FF
10	BYTE	Length of the empty field	W#16#7
11	BYTE	Free	
12	BYTE	Free	
13	BYTE	Free	
14	BYTE	Free	
15	BYTE	Free	

3.2.3 Length of the Source / Destination data field

CPU 319-3 PN/DP

Table 3-5 max length of source / destination data field for CPU 319-3 PN/DP

ORG-ID	max length of the source / destination data field
W#16#01	= 8176 Bytes (4088 Words)
W#16#02	= 8176 Bytes (4088 Words)
W#16#03	= 4080 Bytes (2040 Words)
W#16#04	= 4080 Bytes (2040 Words)

CPU 317-2 PN/DP

Table 3-6 max length of source / destination data field for CPU 317-2 PN/DP

ORG ID	max length of the source / destination data field
W#16#01	= 8176 Bytes (4088 Words)
W#16#02	= 4080 Byte (2040 Worte)
W#16#03	= 2032 Byte (1016 Worte)
W#16#04	= 2032 Byte (1016 Worte)

CPU 315-2 PN/DP

Table 3-7 max length of source / destination data field for CPU 317-2 PN/DP

ORG ID	max length of the source / destination data field
W#16#01	= 8176 Bytes (4088 Words)
W#16#02	= 2032 Bytes (1016 Words)

ORG ID	max length of the source / destination data field
W#16#03	= 2032 Bytes (1016 Words)
W#16#04	= 2032 Bytes (1016 Words)

3.2.4 Start address

The max start address depends on:

- ORG ID, i.e. the memory area of the CPU from which the data are read and to which the data are written
- Length of the source / destination data field

ORG ID	max start address
W#16#01	= 64 KByte − Length of the source / destination data field − 16 Byte Header telegram
W#16#02	= **max** Length of the source / destination data block − Length of the source / destination data block
W#16#03	= **max** Length of the source / destination data block − Length of the source / destination data block
W#16#04	= **max** Length of the source / destination data block − Length of the source / destination data block

3.2.5 Error number response telegram

Table 3-8 error numbers in response telegram

Error number	Description
0	positive acknowledgement, i.e. valid FETCH or WRITE request telegram is received
2	**Requested block does not exist**
3	**Requested block is to small**
6	**No valid ORG ID**

4 Structure and Flow of the S7 Program of the Application

4.1 Overview of the S7 program structure

The following Figure 4-1 is a diagrammatic representation of the S7 program structure of the application.

Figure 4-1 Overview of the S7 program structure
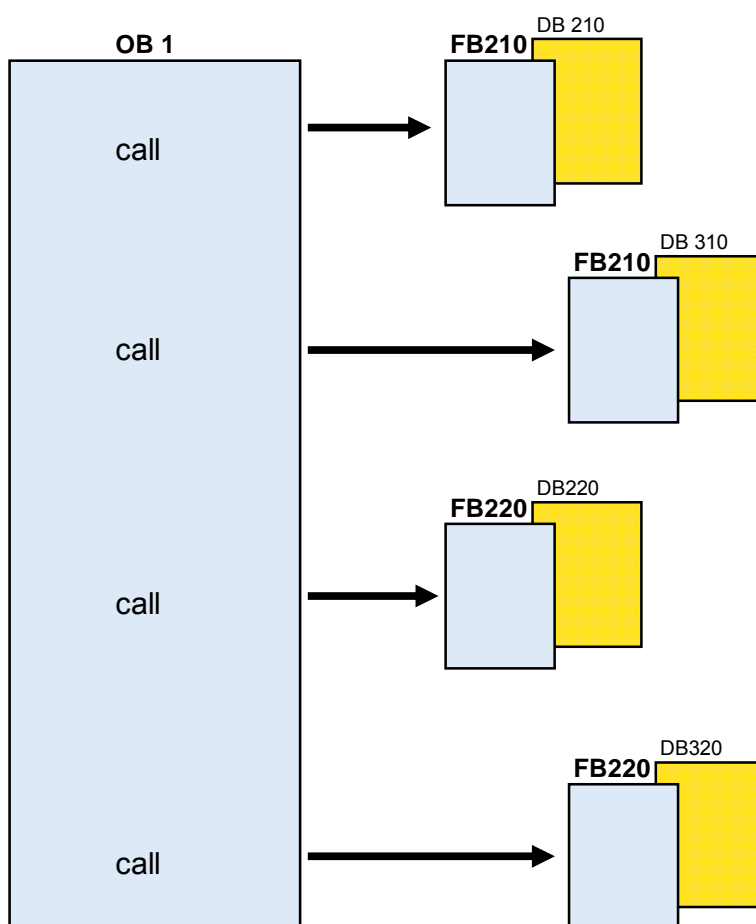
4.2 Explanations of the S7 program structure of the application

In the S7 program of the CPU 319-3 PN/DP, the FETCH and WRITE services are implemented in the following function blocks.

- FB210 "FW_TCP"
- FB220 „FW_IOT"

You have to call these function blocks in OB1 cyclically.

With FB210 the FETCH/WRITE services are provided via the TCP connection. Since separate TCP connections are required for the FETCH/WRITE services, FB210 is called twice in OB1.
The first call with the DB210 instance data block provides the FETCH service via the TCP connections.
The second call with the DB310 instance data block provides the WRITE function via the TCP connection.

With FB220 the FETCH/WRITE services are provided via the ISO-on-TCP connection. Since separate ISO-on-TCP connections are required for the FETCH/WRITE services, FB220 is called twice in OB1.
The first call with the DB220 instance data block provides the FETCH service via the ISO-on-TCP connections.
The second call with the DB320 instance data block provides the WRITE service via the ISO-on-TCP connection.

Internally the FB210 "FW_TCP" and FB220 "FW_IOT" blocks call system functions for connecting and disconnecting and for sending and receiving data.

For further information on open communication via Industrial Ethernet, please refer to the "System Software for S7-300/400 System and Standard Functions" manual. Use the following link to download this manual from the internet:
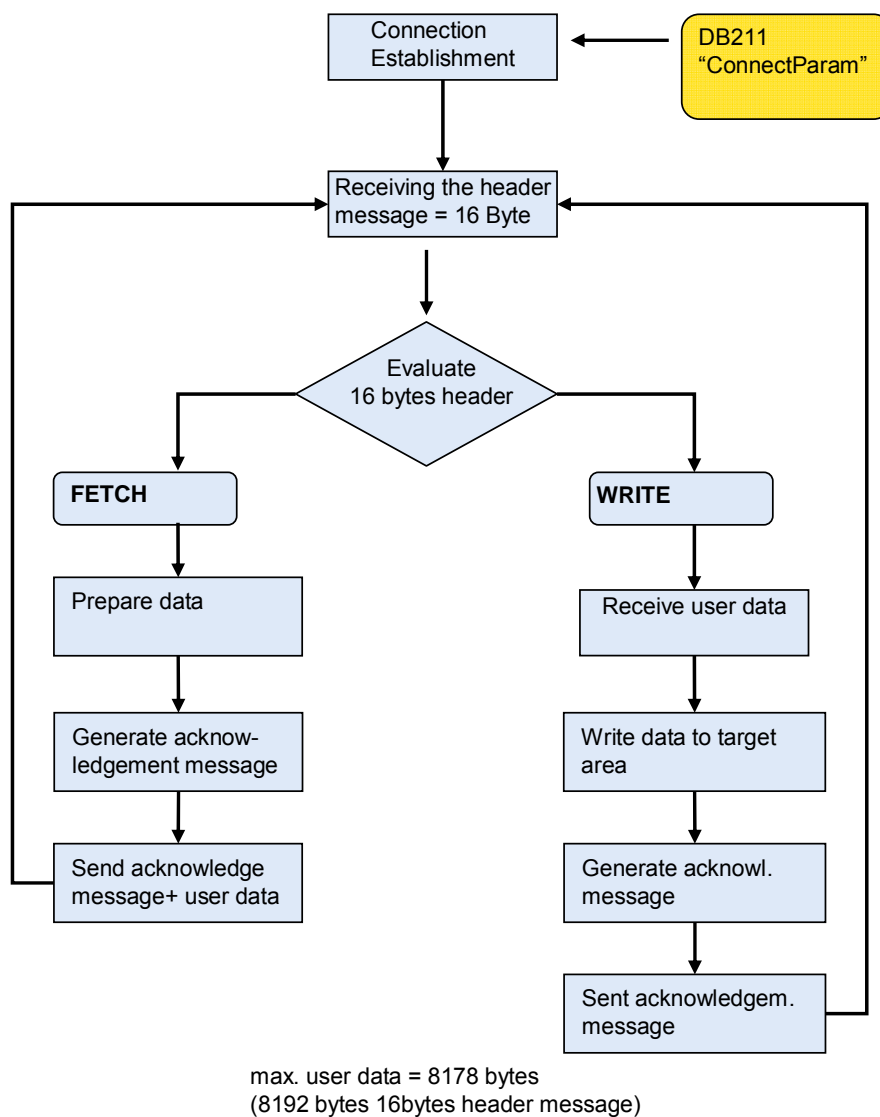
http://support.automation.siemens.com/WW/view/en/1214574

4.3 Execution of FB210 "FW_TCP"

The following Figure 4-2 is a diagrammatic representation of the execution of FB210 that is called in OB1 and that realizes the FETCH/WRITE services via TCP connections.

Overview

Figure 4-2 Execution of FB210 "FW_TCP"

max. user data = 8178 bytes
(8192 bytes 16bytes header message)

Description

After a warm, hot or cold restart of the CPU as well as changing the connection parameters the TCP connection is established automatically. The parameters for establishing the connection are filed in DB211 "ConnectParam".

With the input parameter "ENABLE" it's possible to control the connection establishment and the connection clearing manually.

When the connection parameters are changed the connection is cleared and established again automatically.

The following table shows the data structure of the connection parameters for a TCP connection. This is 64 bytes long.

Table 4-1 Data structure of the connection parameters for TCP connections

Parameter	Value	Description
id	W#16#1	Connection ID The value of the parameter must lie in the following value range: id = W#16#0001 bis W#16#0FFF
connection_type	B#16#11	Connection type For TCP connection the following applies: connection_type = B#16#11 (TCP native) or connection_type = B#16#1 (TCP native compatibility mode)
active_est	FALSE	Passive connection establishment
local_device_id	B#16#3	CPU 319-3 PN/DP
local_tsap_id_len	B#16#2	Data length of the local port in the S7-300 station
rem_subnet_id_len	B#16#0	Not relevant
rem_staddr_len	B#16#0	Unspecified connection
rem_tsap_id_len	B#16#0	Data length of the port in the communication partner
next_staddr_len	B#16#0	Used length of the next_staddr parameter
local_tsap_id	**B#16#7** **B#16#D0** B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0	Local port for the TCP connection in the S7-300 station: The value of the parameter must lie in the following value range: local_tsap_id = 1 up to 49151 (dec.)
rem_subnet_id	B#16#0 …	Not relevant

Parameter	Value	Description
rem_staddr	B#16#0 …	IP address of the communication partner: - Not relevant for unspecified connections with passive connection establishment
rem_tsap_id	B#16#0 …	Port of the communication partner: - Not relevant for unspecified connections with passive connection establishment
next_staddr	B#16#0 …	Rack and slot of the CP: - Not relevant since no CP is used
spare	W#16#0	Not relevant

The following connection parameters are checked prior to the established connection:

- connection_type
- id
- local_tsap_id

The connection_type = B#16#11 connection parameter is used in this application.

If one of these parameters wasn't defined correctly, the communication connection wouldn't establish. No data is sent or received.

After the communication connection has been established, a header message of the FETCH/WRITE services with a length of 16 bytes is received. The header message contains the information whether a FETCH or WRITE job is pending.

When there is a FETCH job, then the data requested by the communication partner is prepared in instance data block DB210 and the acknowledgement message is generated. The acknowledgement telegram is sent including the prepared data.

If there is a WRITE job, then the user data is received after evaluating the 16 bytes header message and written to the target area. Subsequently, the acknowledgement message is generated in the instance data block DB310 and sent.

Input and output parameters

FB210 has the following input and output parameters:

Figure 4-3 FB210 "FW_TCP"
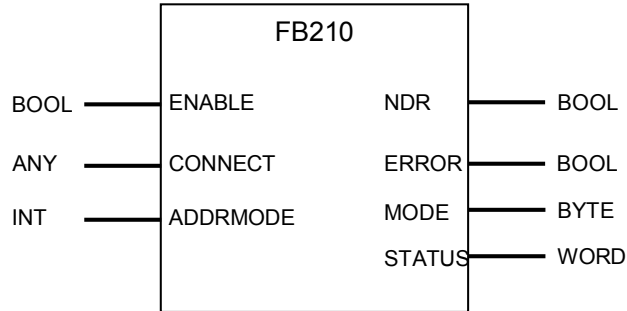
```
                    ┌─────────────────────────┐
                    │          FB210           │
                    │                          │
 BOOL ──────────────┤ ENABLE          NDR      ├────────────── BOOL
                    │                          │
 ANY  ──────────────┤ CONNECT         ERROR    ├────────────── BOOL
                    │                          │
 INT  ──────────────┤ ADDRMODE        MODE     ├────────────── BYTE
                    │                          │
                    │                 STATUS   ├────────────── WORD
                    │                          │
                    └─────────────────────────┘
```

Table 4-2 Input parameters

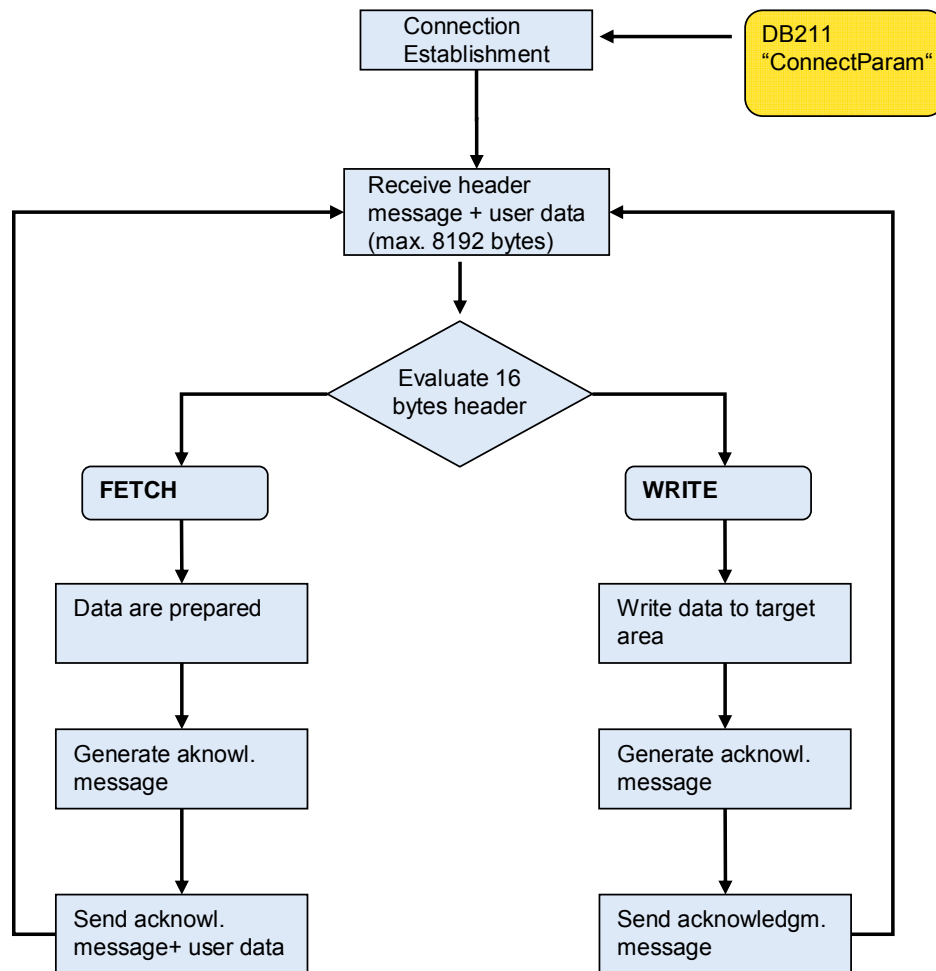| Input parameters	Description
ENABLE	This parameter is used to initialize the connection establishment and connection clearing respectively. • 0 = connection clearing • 1 = connection establishment
CONNECT	ANY pointer to the data area of DB211 that includes the parameters for establishing the TCP connection. This data area must be at least 64 bytes.
ADDMODE	This parameter is used to define how the data of FETCH and WRITE job respectively has to be addressed (S7 or S5 addressing mode): • 0 = S7 addressing mode (the start address of data is interpreted as byte address) • 1 = S5 addressing mode (the start address of data is interpreted as word address)

Table 4-3 Output parameters

Output parameters	Description
NDR (new data record)	With this parameter it is indicated that the data of the WRITE job have been adopted successfully and the acknowledgement telegram was generated and sent.
ERROR	This parameter is set when: • an error has occurred during connection or disconnection • an error has occurred during sending or receiving the data • an invalid FETCH/WRITE Header telegram was received
MODE	This parameter indicates whether a FETCH or WRITE job is executed. • 0 = no active job • 1 = WRITE job • 2 = FETCH job
STATUS	Internal communication status bits are output in the STATUS parameter (see Table 8-1 Structure of the STATUS parameter).

4.4 Execution of FB220 "FW_IOT"

The following figure is a diagrammatic representation of the execution of FB220 that is called in OB1 and that realizes the FETCH/WRITE services via connections.

Overview

Figure 4-4 Execution of FB220 "FW_IOT"

Description

After a warm, hot or cold restart of the CPU as well as changing the connection parameters the ISO-on-TCP connection is established automatically. The parameters for establishing the connection are filed in DB211 "ConnectParam".

With the input parameter "ENABLE" it's possible to control the connection establishment and the connection clearing manually.

When the connection parameters are changed the connection is cleared and established again automatically.

The following table shows the data structure of the connection parameters for an ISO-on-TCP connection. It is 64 bytes long.

Table 4-4 Data structure of the connection parameters for ISO-on-TCP connections

Parameter	Value	Description
Id	W#16#1	Connection ID The value of the parameter must lie in the following value range: id = W#16#0001 up to W#16#0FFF
connection_type	B#16#12	Connection type For ISO-on-TCP connection the following applies: connection_type = B#16#12
active_est	FALSE	Passive connection
local_device_id	B#16#3	CPU 319-3 PN/DP
local_tsap_id_len	B#16#7	Data length of the local TSAP in the S7-300 station The value of the parameter must lie in the following value range: local_tsap_id_len = 2 up to 16
rem_subnet_id_len	B#16#0	Not relevant
rem_staddr_len	B#16#0	Unspecified connection
rem_tsap_id_len	B#16#0	Data length of the TSAP in the communication partner
next_staddr_len	B#16#0	Used length of the next_staddr parameter
local_tsap_id	**B#16#E0** **B#16#02** **B#16#54** **B#16#43** **B#16#50** **B#16#2D** **B#16#31** B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0 B#16#0	Local TSAP for the ISO-on-TCP connection in the S7-300 station: The first byte of the parameter must have the following value: local_tsap_id[1]=W#16#E0 or local_tsap_id[1] is an ASCII character
rem_subnet_id	B#16#0 …	Not relevant
rem_staddr	B#16#0 …	IP address of the communication partner: - Not relevant for unspecified connections with passive connection establishment
rem_tsap_id	B#16#0 …	TSAP of the communication partner: - Not relevant for unspecified connections with passive connection establishment
next_staddr	B#16#0 …	Rack and slot of the CP: - Not relevant since no CP is used
spare	W#16#0	Not relevant

The following connection parameters are checked prior to the established connection:

- connection_type

- id

- local_tsap_id

- local_tsap_id_len

If one of these parameters wasn't defined correctly, the communication connection wouldn't establish. No data is sent or received.

After the communication connection has been established, a header message of the FETCH/WRITE services with a length of 16 bytes is received. The header message contains the information whether a FETCH or WRITE job is pending.

When there is a FETCH job, then the data requested by the communication partner is prepared in instance data block DB220 and the acknowledgement message is generated. The acknowledgement telegram is sent including the prepared data.

During a WRITE job the user data are received together with the 16 Bytes Header telegram and written to the target area. Subsequently, the acknowledgement message is generated in the instance data block DB320 and sent.

Input and output parameters

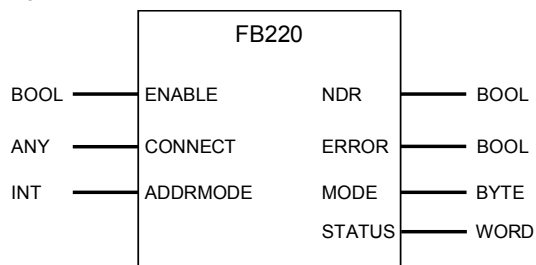FB220 has the following input and output parameters:

Figure 4-5 FB220 "FW_IOT"

Table 4-5 Input parameters

Input parameters	Description
ENABLE	This parameter is used to initialize the connection establishment and connection clearing respectively. • 0 = connection clearing • 1 = connection establishment
CONNECT	ANY pointer to the data area of DB211 that includes the parameters for establishing the ISO-on-TCP connection. This data area must be at least 64 bytes.
ADDRMODE	This parameter is used to define how the data of FETCH and WRITE job respectively has to be addressed (S7 or S5 addressing mode): • 0 = S7 addressing mode (the start address of data is interpreted as byte address) • 1 = S5 addressing mode (the start address of data is interpreted as word address)

Table 4-6 Output parameters

Output parameters	Description
NDR (new data record)	With this parameter it is indicated that the data of the WRITE job have been adopted successfully and the acknowledgement telegram was generated and sent.
ERROR	This parameter is set when: • an error has occurred during connection or disconnection • an error has occurred during sending or receiving the data • an invalid FETCH/WRITE Header telegram was received
MODE	This parameter indicates whether a FETCH or WRITE job is executed. • 0 = no active job • 1 = WRITE job

Output parameters	Description
	• 2 = FETCH job
STATUS	Internal communication status bits are output in the STATUS parameter (see Table 8-1 Structure of the STATUS parameter).

5 Explanation of the Configuration

5.1 Programming the communication connections in the controller

The parameters of the communication connections are stored in the CPU memory as a data block.

The connection parameters can be generated with the "Open Communication Wizard" tool and be stored in the data block of the STEP 7 project.

In this application the connection parameters are saved in the DB211 "ConnectParam".

Use the following link to download the "Open Communication Wizard" from the internet:

http://support.automation.siemens.com/WW/view/en/25209116

5.1.1 Creating the connection parameters for the TCP connections

When generating these connection parameters with the Open Communication Wizard, please follow the instructions listed below:

Installing the Open Communication Wizard

Install the Open Communication Wizard on the PC on which you configure the S7 station. Please note that at least Microsoft .NET Framework V2.0 is required for the installation of the Open Communication Wizard.

Starting the Open Communication Wizard

Start the Open Communication Wizard in the Windows START menu by selecting SIMATIC → Open Communication Wizard. The following dialog box opens (see Figure 5-1 starting the Open Communication Wizard). Click the button "Next" to continue with setting the connection parameters.

Figure 5-1 starting the Open Communication Wizard
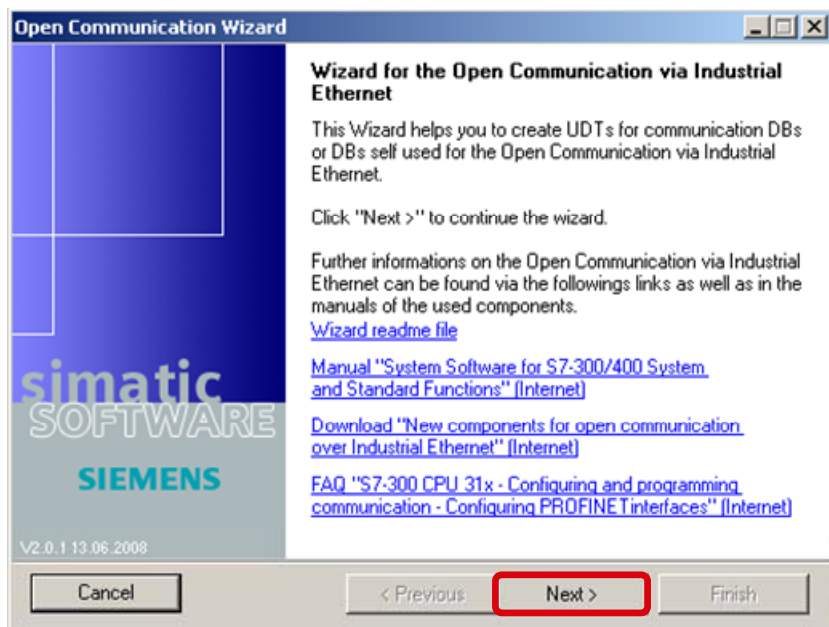
In the following dialog you select the STEP 7 project and the block folder in which the connection parameters are saved. Subsequently you click the button "Next" to go to the next dialog.

Figure 5-2 select STEP 7 project and block folder
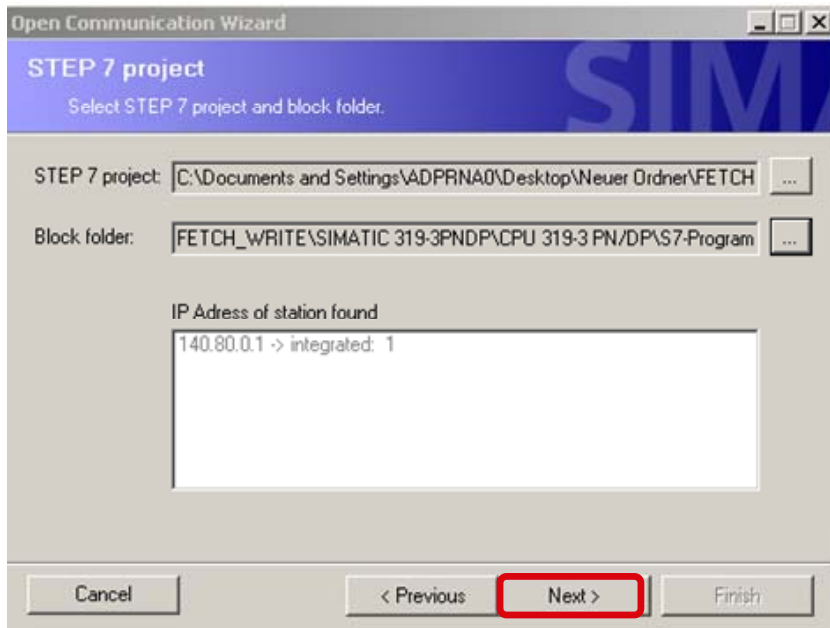
Select the "New" function to generate a new connection. Subsequently you click the button "Next" to go to the next dialog, where you select the connection type.
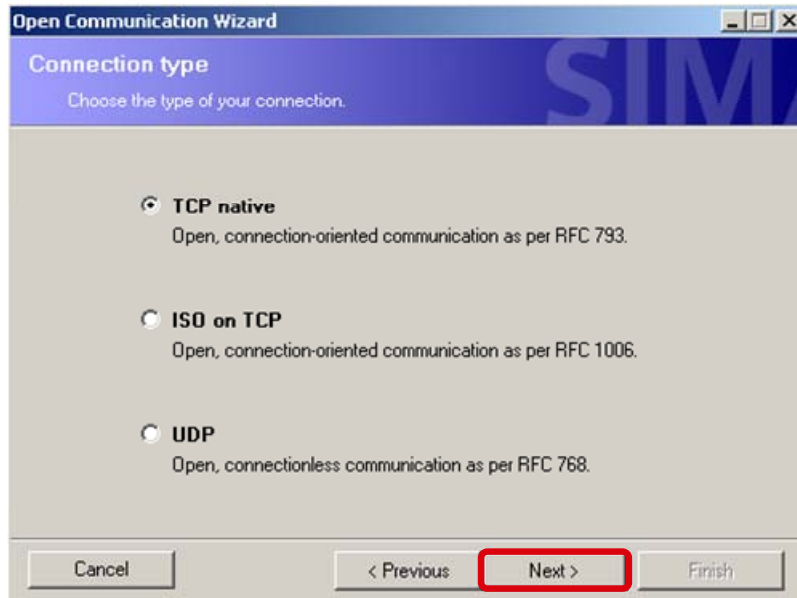
Figure 5-3 generate a new connection
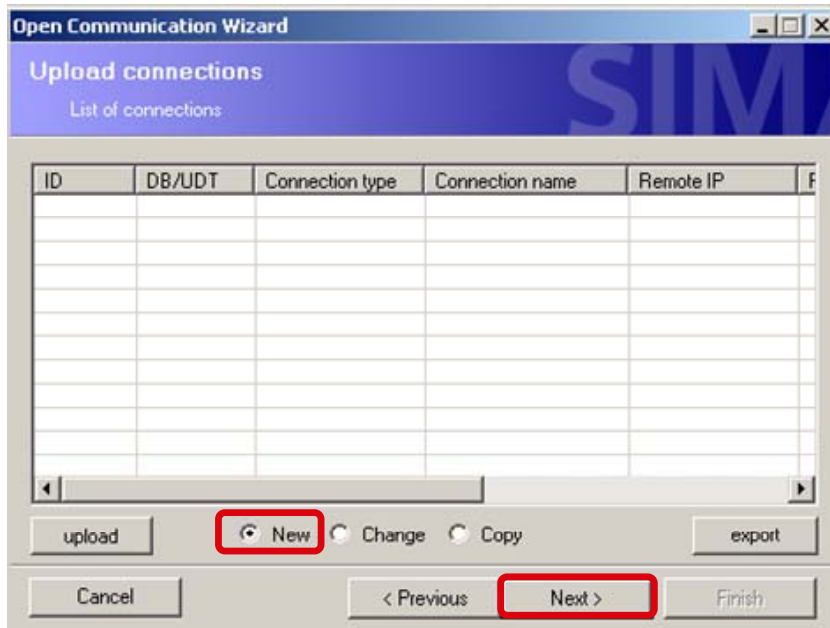
Selecting the connection type

To configure the TCP connection, select the "TCP native" connection type.

Figure 5-4 selecting the connection type

Configuring the communication partners

Now the communication partner is configured. In this application, only the S7-300 station (FETCH/WRITE server) is configured since the communication partner is a PC station (FETCH/WRITE client).

If you use two S7 controllers as communication partners, the parameters, which you have to configure here, will change.

Click the button "Next" to go to the next dialog, where you define the properties of the communication partners.

Figure 5-5 configuring the communication partners
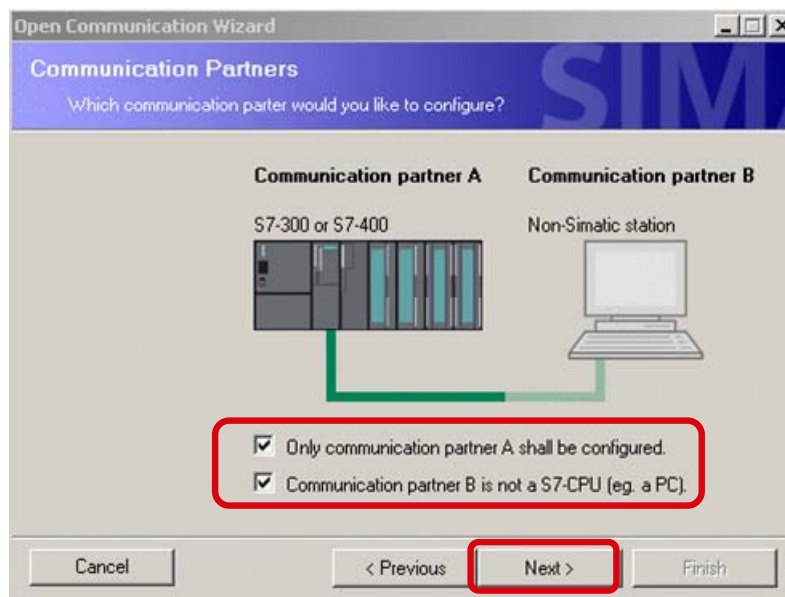
Defining the properties of the communication partners

In this application, the following properties are defined for the TCP connection in the CPU 319-3 PN/DP:

Table 5-1 Properties of the communication partners

Property	TCP connection with the FETCH service	TCP connection with the WRITE service
Connection ID	W#16#0003	W#16#0004
Connection establishment	Passive	Passive
Connection	Unspecified	Unspecified
Interface	CPU 319-3 PN/DP	CPU 319-3 PN/DP

Define the TCP connections for the FETCH and WRITE services with different connection ID and different connection name.

The TCP connections are unspecified since only an unspecified configuration of TCP connections is possible in NetPro for the data exchange between PC station and CPU with integrated Industrial Ethernet interface (see chapter 5.2 "Configuring the communication connections in the PC station").

Select the interface according to the hardware you are using. A CPU 319-3 PN/DP is used in this application. The integrated Industrial Ethernet interface of the CPU 319-3 PN/DP is thus defined in the Open Communication Wizard. Click the button "Next" to go to the next dialog, where you define the local port for the TCP connection in the CPU.

Figure 5-6 defining the properties of the communication partners
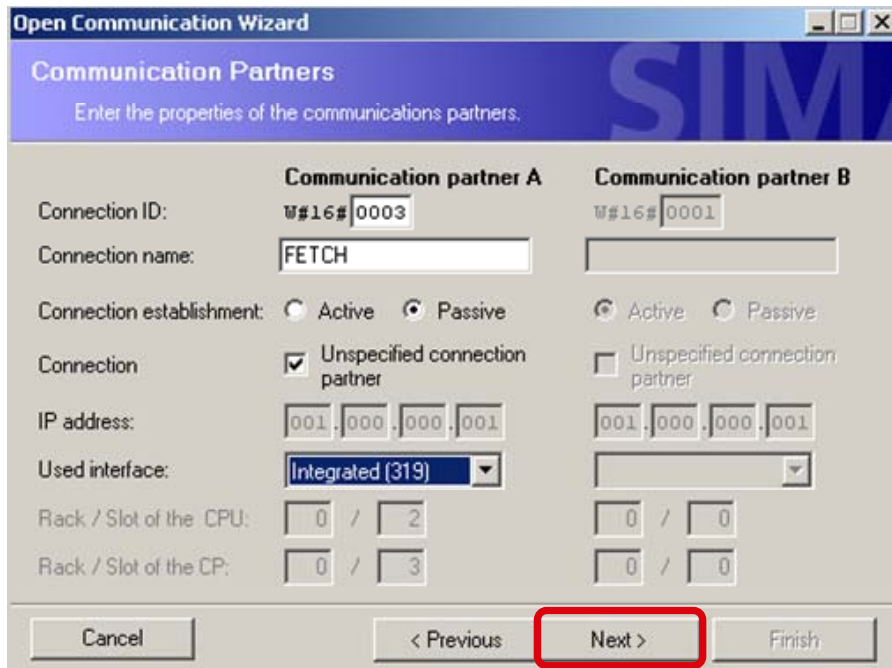
Defining the ports of the TCP connections

In this application, the following ports are defined in the S7-300 station for the TCP connections.

Table 5-2 Ports of the TCP connections

Communication connection	Port
TCP connection for FETCH job	2000 (dec) = 7D0 (hex)
TCP connection for WRITE job	2001 (dec) = 7D1 (hex)

Figure 5-7 ports of the TCP connections
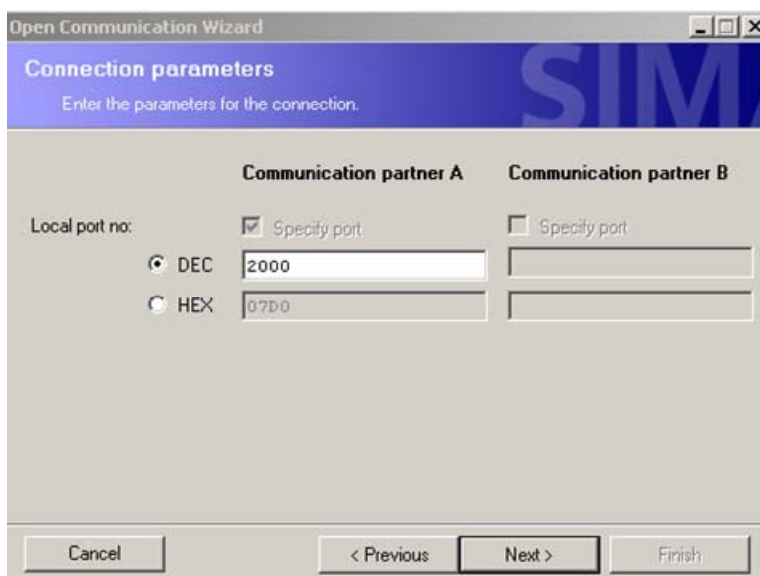
| Note	With this parameterization of the S7-300 station, any station supporting the TCP native communication service can access the data in the controller via the configured port.

Defining the name of the data structure

In the following dialog of the Open Communication Wizard you specify the data block in which you wish to save the connection parameters. Furthermore the already selected STEP 7 project and the already selected block folder is displayed by generating the data block with the connection parameters. If you have not yet selected a STEP 7 project and block folder (see Figure 5-2 select STEP 7 project and block folder), then you can do this in the following dialog.

Figure 5-8 generate data block with connection parameters

According to the configuration of the connection parameters with the Open Communication Wizard the specified data block is generated with the correct connection parameters (see Table 4-1 Data structure of the connection parameters for TCP connections).

If the given data block already exists in the STEP 7 project, the connection parameters are attached to the already existing data in the data block. In this case you acknowledge the following message with "OK".

Figure 5-9 Data block already exists
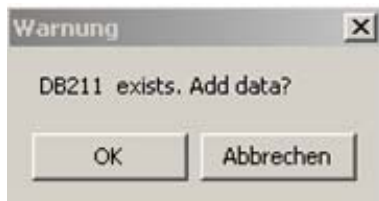
5.1.2 Creating the connection parameters for the ISO-on-TCP connections

When generating these connection parameters with the Open Communication Wizard, please follow the instructions listed below:

Installing the Open Communication Wizard

Install the Open Communication Wizard on the PC on which you configure the S7 station. Please note that at least Microsoft .NET Framework V2.0 is required for the installation of the Open Communication Wizard.

**Starting the Open Communication Wizard**

Start the Open Communication Wizard in the Windows START menu by selecting SIMATIC → Open Communication Wizard. The following dialog box opens (see Figure 5-10 starting the Open Communication Wizard). Click the button "Next" to continue with setting the connection parameters.

Figure 5-10 starting the Open Communication Wizard
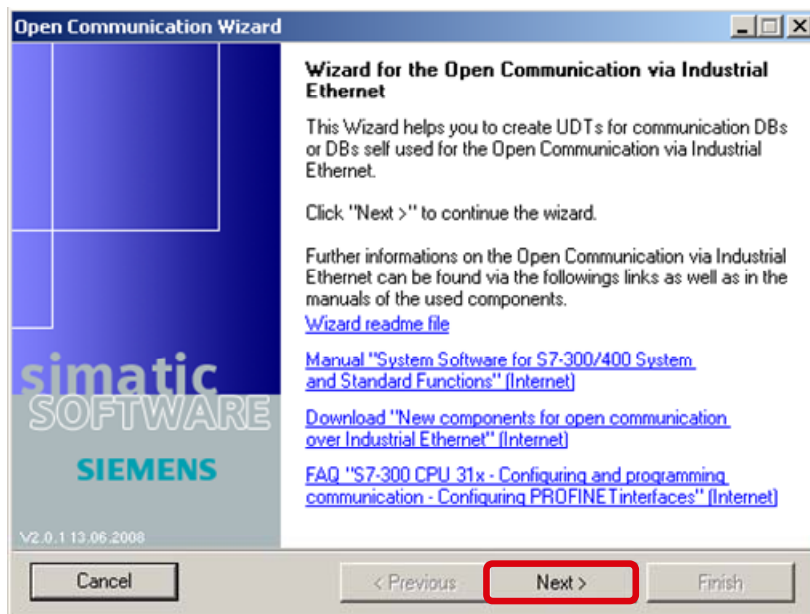
In the following dialog you select the STEP 7 project and the block folder in which the connection parameters are saved. Subsequently you click the button "Next" to go to the next dialog.

Figure 5-11 select STEP 7 project and block folder
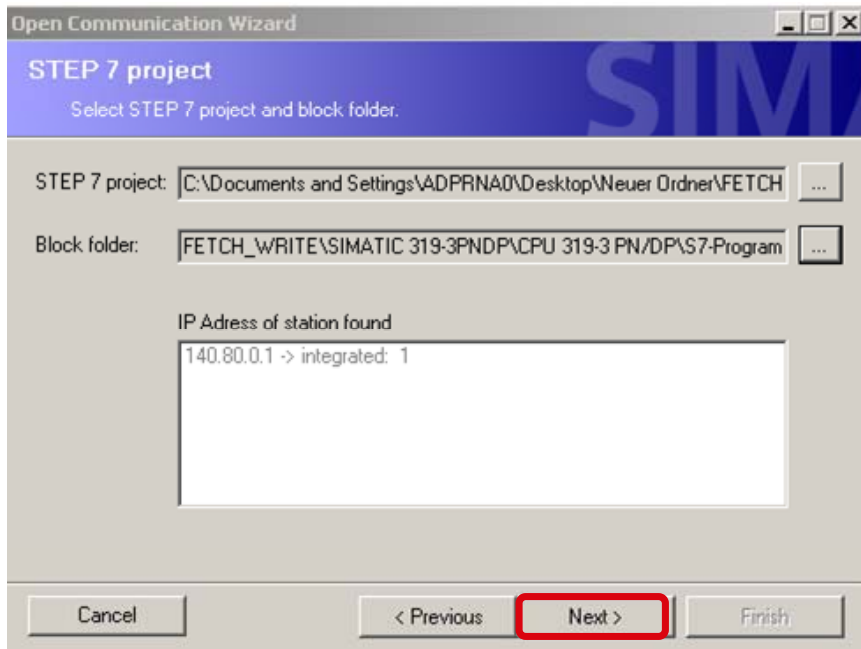
Select the "New" function to generate a new connection. Subsequently you click the button "Next" to go to the next dialog, where you select the connection type.

Figure 5-12 generate a new connection
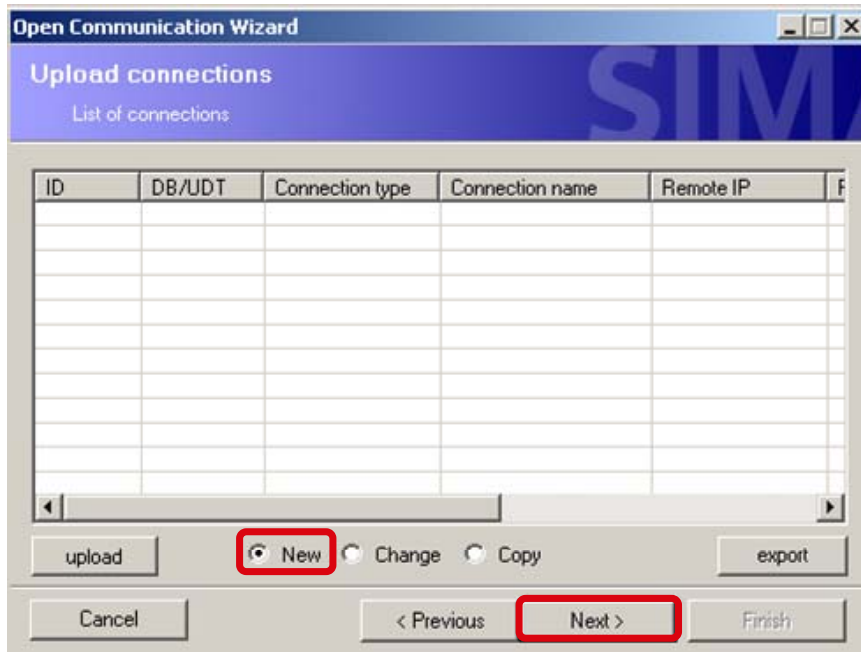
Selecting the connection type

To configure the ISOonTCP connection, select the "ISO on TCP" connection type.

Figure 5-13 selecting the connection type
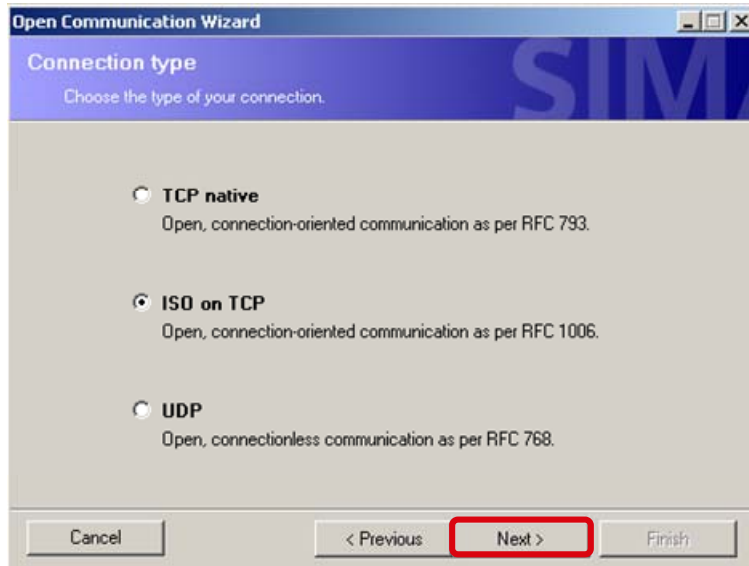
Configuring the communication partners

Now the communication partner is configured. In this application, only the S7-300 station (FETCH/WRITE server) is configured since the communication partner is a PC station (FETCH/WRITE client).

If you use two S7 controllers as communication partners, the parameters, which you have to configure here, will change.

Click the button "Next" to go to the next dialog, where you define the properties of the communication partners.

Figure 5-14 configuring the communication partners
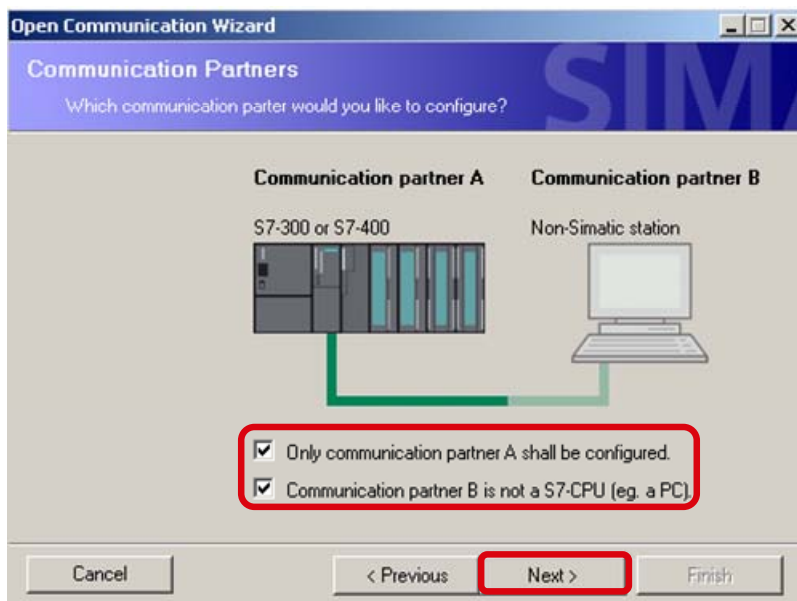
Defining the properties of the communication partners

In this application, the following properties are defined for the ISO-on-TCP connection in the CPU 319-3 PN/DP:

Table 5-3 Properties of the communication partners

Property	ISO-on-TCP connection for FETCH job	ISO-on-TCP connection for WRITE job
Connection ID	W#16#0001	W#16#0002
Connection establishment	Passive	Passive
Connection	Unspecified	Unspecified
Interface	CPU 319-3 PN/DP	CPU 319-3 PN/DP

Define the ISO-on-TCP connections for the FETCH and WRITE services with different connection ID and different connection name.

The ISO-on-TCP connections are unspecified since only an unspecified configuration of TCP connections is possible in NetPro for the data exchange between PC station and CPU with integrated Industrial Ethernet interface (see chapter 5.2 "Configuring the communication connections in the PC station").

Select the interface according to the hardware you are using. A CPU 319-3 PN/DP is used in this application. The integrated Industrial Ethernet interface of the CPU 319-3 PN/DP is thus defined in the Open Communication Wizard. Click the button "Next" to go to the next dialog, where you define the local TSAP of the ISO-on-TCP connection in the CPU.

Figure 5-15 defining the properties of the communication partners

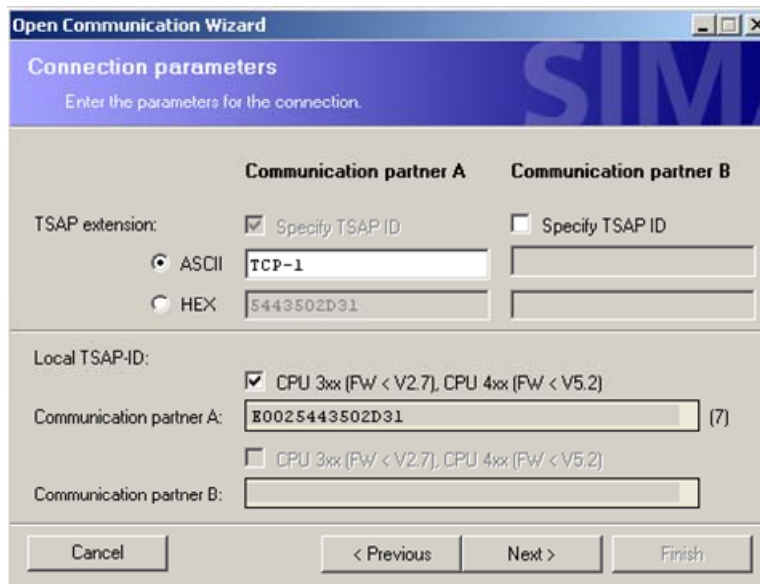**Defining the TSAPs of the ISO-on-TCP connection**

In this application, the following TSAPs are defined in the S7-300 station for the ISO-on-TCP connections.

Table 5-4 TSAPs of the communication connections

Communication connection	TSAP
ISO-on-TCP connection for FETCH job	TCP-1 (ASC) = **E0.02**.54.43.50.2D.31 (hex)
ISO-on-TCP connection for WRITE job	TCP-2 (ASC) = **E0.02**.54.43.50.2D.32 (hex)

The checkbox "CPU 3xx (FW<V2.7), CP4xx (FW<V5.2)" has been activated as a standard. Only deactivate this checkbox if you are using a S7-300 CPU with FW ≥ V2.7 or an S7-400 with FW ≥ V5.2, since the application of ASCII-TSAPs is supported from these firmware versions on.

Figure 5-16 TSAPs of the ISO-on-TCP connections

Note With this parameterization of the S7-300 station, any station supporting the ISO-on-TCP communication service can access the data in the controller via the configured TSAP.

Defining the name of the data structure

In the following dialog of the Open Communication Wizard you specify the data block in which you wish to save the connection parameters. Furthermore the already selected STEP 7 project and the already selected block folder is displayed by generating the data block with the connection parameters. If you have not yet selected a STEP 7 project and block folder (see Figure 5-11 select STEP 7 project and block folder), then you can do this in the following dialog.

Figure 5-17 generate data block with connection parameters
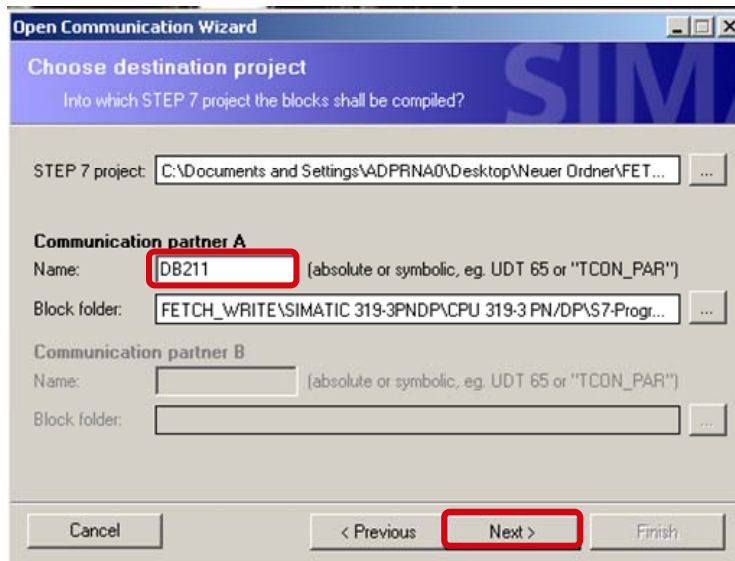
According to the configuration of the connection parameters with the Open Communication Wizard the specified data block is generated with the correct connection parameters (see Table 4-4 Data structure of the connection parameters for ISO-on-TCP connections).

If the given data block already exists in the STEP 7 project, the connection parameters are attached to the already existing data in the data block. In this case you acknowledge the following message with "OK".

Figure 5-18 Data block already exists

5.1.3 Structure of DB211 "Connect_Param"

The following table illustrates the structure of DB211 "ConnectParam", in which the connection parameters of the ISO-on-TCP and TCP connections are saved.

Table 5-5 Structure of DB211 "ConnectParam"

Byte	Description
0-63	Parameter set of the ISO-on-TCP connection for the FETCH job
64-127	Parameter set of the ISO-on-TCP connection for the WRITE job
128-191	Parameter set of the TCP connection for the FETCH job
192-255	Parameter set of the TCP connection for the WRITE job

5.2 Configuring the communication connections in the PC station

After programming the communication connections in the controller, these connections have to be configured in the PC station.

For the PC station the communication connections are configured in STEP 7 or NCM PC. In STEP 7 / NCM PC, open NetPro and insert two TCP connections and two ISO-on-TCP connections in the PC station.

In the Properties window of the individual communication connections, the following settings have to be made:

Specifying the connection name

The connection name is specified in the "General Information" tab of the Properties window of the communication connection. This name can be freely selected.

Then click the button "Estab. Details..." So the dialog "Details of Connection Establishment" is opened.
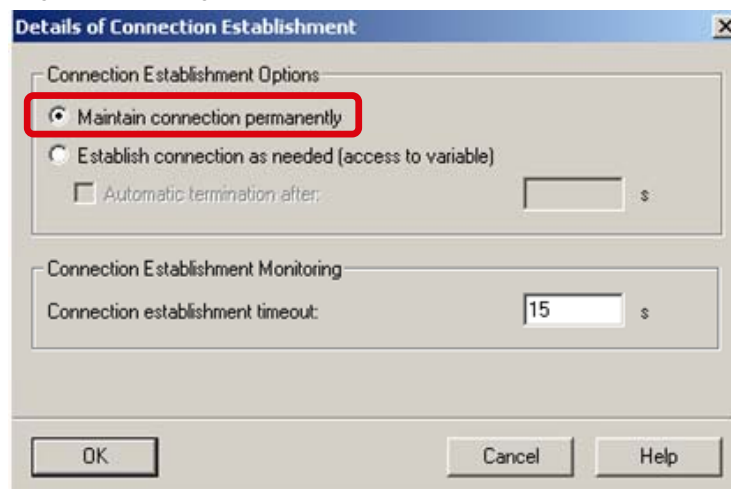
Figure 5-19 Properties dialog of the communication connection → "General Information" tab
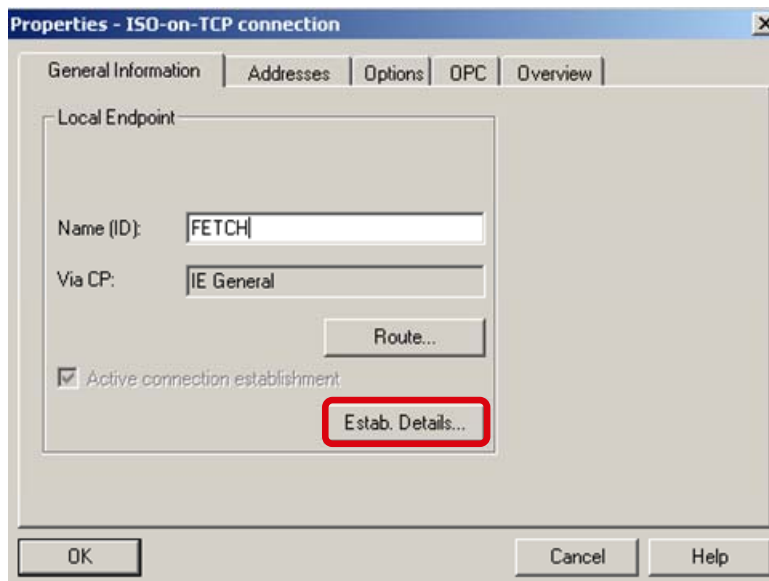
In the dialog "Details of Connection Establishment" you select the function "Maintain connection permanently".

Accept the settings with "OK".

Figure 5-20 dialog "Details of Connection Establishment"

Defining the connection parameters

In the Properties dialog of the communication connection, select the "Addresses" tab. In this tab, define the connection parameters. Specify the IP address of the CPU 319-3 PN/DP as remote IP address.

In addition, define the TSAPs or ports between which the communication connection is to be established. The same ports or TSAPs are used locally and remotely.

Note   The first two bytes "E0.02" of the remote TSAP have to be entered manually.

Figure 5-21 Properties dialog ISO-on-TCP connection → "Addresses" tab
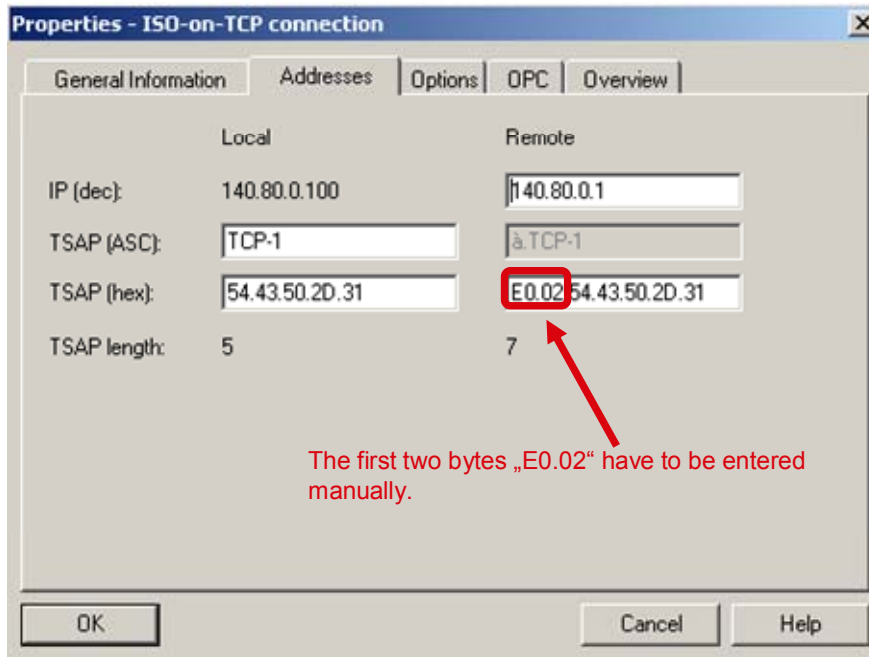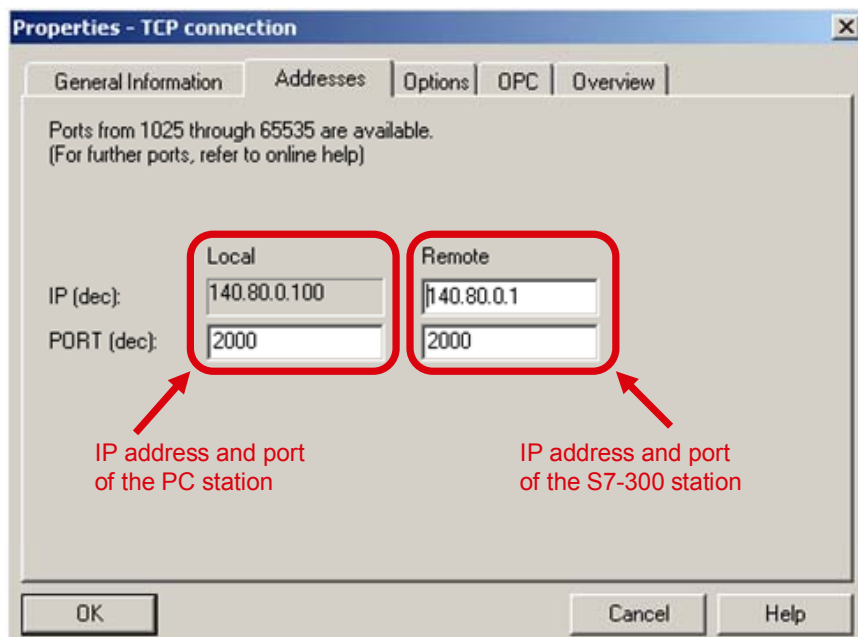
Figure 5-22 Properties dialog TCP connection → "Addresses" tab

In the following table you find an overview of the local and remote TSAPs or ports, which are used in this application.

Table 5-6 Overview of the local and remote TSAPs or ports

| Connection ID	Local TSAP / port in the PC station	Remote TSAP / port in the S7-300 station
W#16#1	TCP-1 (ASC) = 54.43.50.2D.31 (hex)	TCP-1 (ASC) = **E0.02**.54.43.50.2D.31 (hex)
W#16#2	TCP-2 (ASC) = 54.43.50.2D.32 (hex)	TCP-2 (ASC) = **E0.02**.54.43.50.2D.32 (hex)
W#16#3	2000 (dec) = 7D0 (hex)	2000 (dec) = 7D0 (hex)
W#16#4	2001 (dec) = 7D1 (hex)	2001 (dec) = 7D1 (hex)

Note The configuration of the TCP and ISO-on-TCP connections, as in this application software, must be partially specified in the S7-300 station and fully specified in the PC station.

Configuring the FETCH and WRITE service

In the Properties dialog → "Options" tab of the communication connection the FETCH active or WRITE active service is defined for the TCP and ISO-on-TCP connections. Since the PC station has to establish the communication connections, the FETCH and WRITE service is active.

Figure 5-23 Properties dialog of the communication connection → "Options" tab
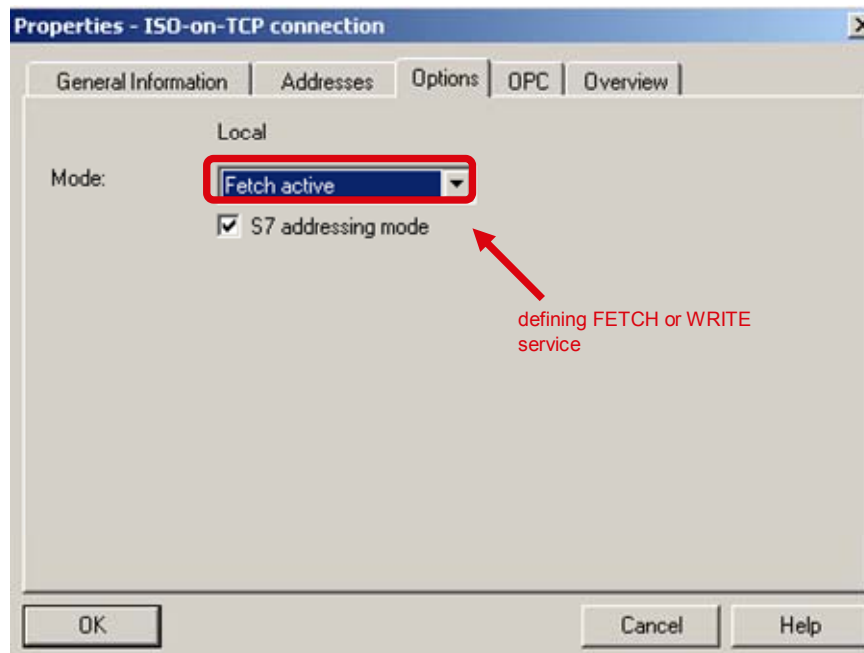
Structure, Configuration and Operation of the Application

Content

This part leads you step by step through the structure, important configuration steps, commissioning and operation of the application.

6 Installation and Commissioning

6.1 Installation of the application software

The application software is a STEP 7 project that contains the complete hardware configuration, including the user program of the S7-300 station and the configuration of the PC station.

This STEP 7 project can be downloaded from the internet as a ZIP file.

Extract the "23592043_FetchWrite_PNCPU_V21.zip" file to a separate directory. The STEP7 project is then automatically unzipped with all respective subdirectories. Subsequently, you can open and edit the extracted STEP 7 project with the SIMATIC Manager.

6.2 Startup of the S7 program

Configuring the S7 station

In the application software, adjust the hardware configuration of the S7 station in such a way that it corresponds to the hardware configuration in your plant.

Subsequently, open the Properties dialog of the PN IO interface of the configured CPU in the hardware configuration of the S7 station.

Figure 6-1 Properties dialog of the PN IO interface
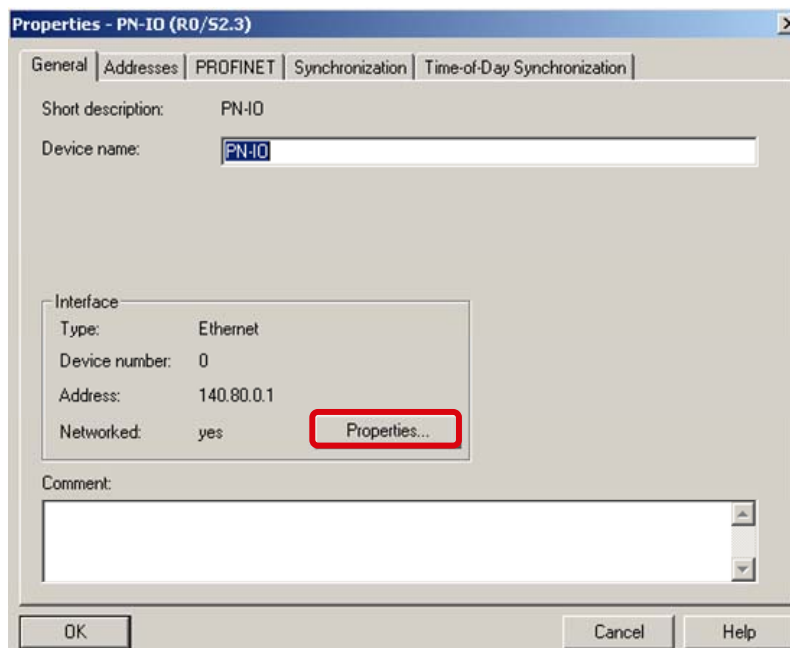
Click the button "Properties". Now enter the IP address and subnet mask for the CPU in the "Properties – Ethernet interface PN-IO" dialog → "Parameters" tab.

Figure 6-2 "Properties – Ethernet interface PN-IO" dialog
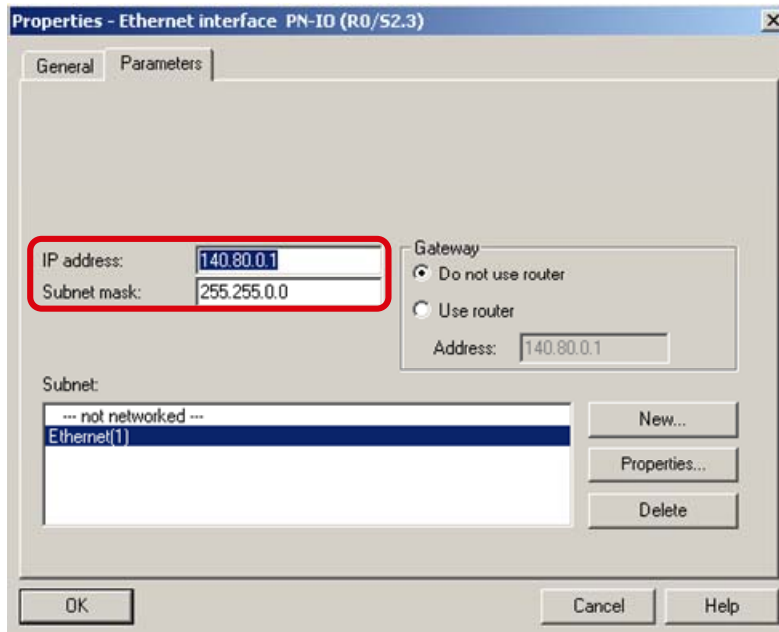
Subsequently, save and compile the hardware configuration of the S7-300 station and download it to your S7 CPU.

Adjust connection parameters

You have to adjust the connection parameter "local_device_id" according to your used and configured hardware as follows.

Table 6-1 "local_device_id" parameter

Module	"local_device_id" parameter
CPU 315(F)-2 PN/DP **Note:** from 6ES7315-2EH14-0AB0 and 6ES7315-2FJ14-0AB0	B#16#02
CPU 317(F)-2 PN/DP	B#16#02
CPU 319(F)-3 PN/DP	B#16#03
CPU 41x	B#16#05
Industrial Ethernet CP	B#16#00

You can now download the S7 program to the CPU.

## 6.3   Commissioning the PC station (FETCH/WRITE client)

Determining the IP address of the PC station

First determine IP address and subnet mask of the PC station acting as a FETCH/WRITE client. To do this, open the "Configuration Console" on this PC station (Windows START menu → SIMATIC → SIMATIC NET → Configuration Console).

In the Configuration Console Explorer you find the Industrial Ethernet PC CP that you have connected in the PC station and that you are using for the communication in "Modules". In "Address", you find the current IP address of the Industrial Ethernet PC CP (see Figure 6-3 determining the IP address of the PC station).

Configure this IP address in STEP 7 / NCM PC for the PC station.

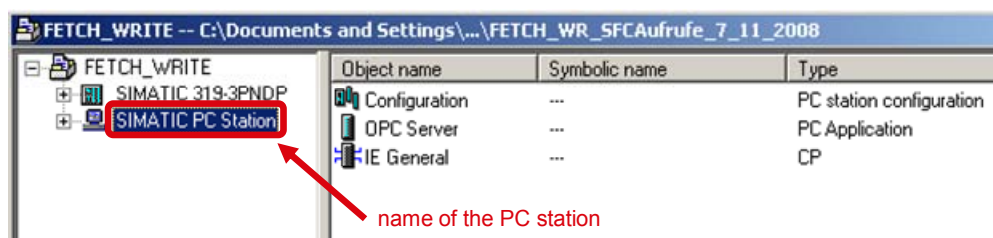Figure 6-3 determining the IP address of the PC station
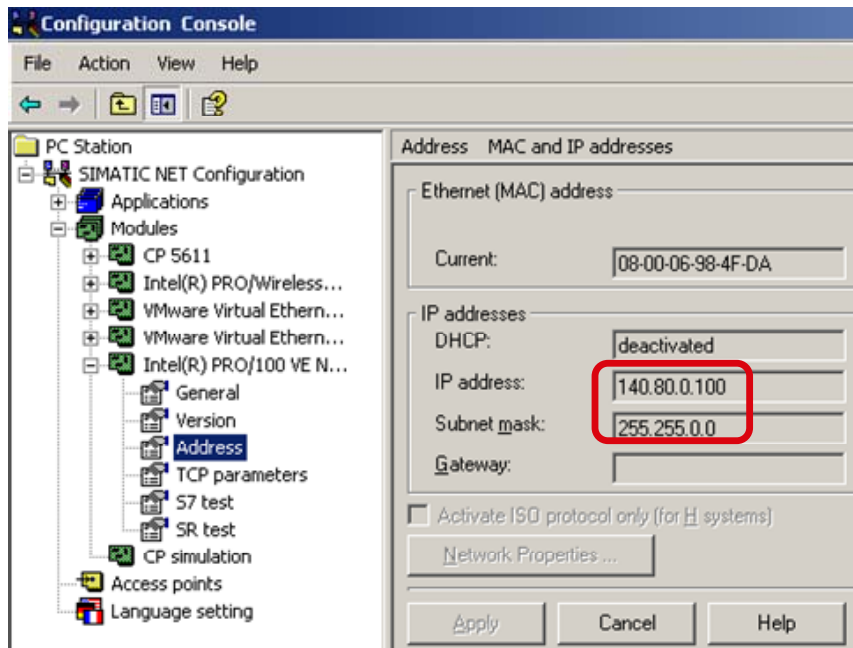
Configuring the PC station in STEP 7 / NCM PC

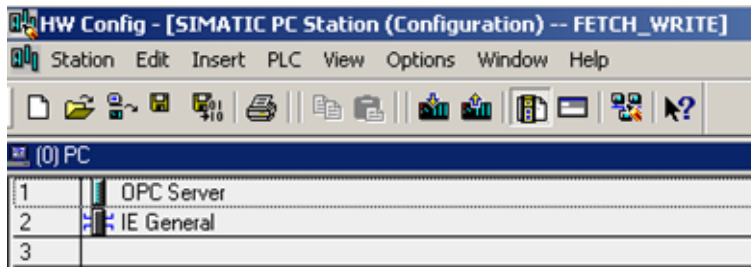Among other things, the STEP 7 project of this application includes the configuration of the PC station.

The name of the PC station is "SIMATIC PC Station".

Figure 6-4 specifying the name for the PC station

Open the hardware configuration of the PC station.

Figure 6-5 Hardware configuration of the PC station

| **Note** | When you are using a CP1613 (A2), CP1623 or CP1512 in your PC station, you have to configure the corresponding module in the hardware configuration. |

In the hardware configuration you open the Properties window of the configured Industrial Ethernet PC CP.
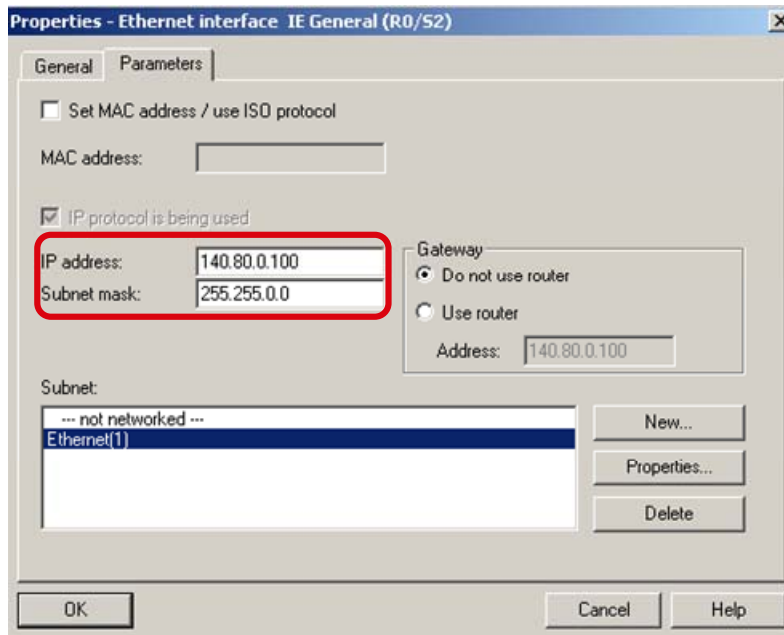
Figure 6-6 Properties window of the Industrial Ethernet PC CP
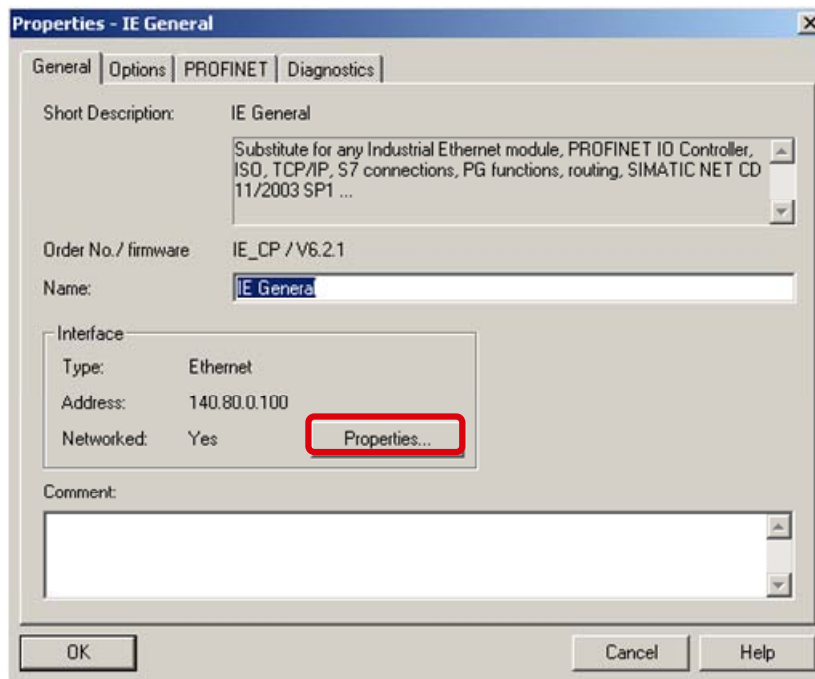
Click the button "Properties". Now enter IP address and subnet mask of your PC station acting as a FETCH/WRITE client in the "Properties – Ethernet interface IE General" dialog.

Figure 6-7 "Properties – Ethernet interface IE General" dialog

Note	The IP address configured in STEP 7 for the PC station must correspond to the IP address set in Windows. Furthermore, it has to be ensured that the IP addresses of the PC station and of the S7 station are located in the same IP subnet.

Via the Windows taskbar, open the Station Configuration Editor.

Figure 6-8 Windows taskbar

Insert the modules, i.e. the OPC server and the Industrial Ethernet PC CP, according to the hardware configuration of the PC station.

Subsequently, adapt the station name in the Station Configuration Editor. The PC station name must be identical in STEP 7 / NCM PC and in the Station Configuration Editor.

Figure 6-9 Station Configuration Editor
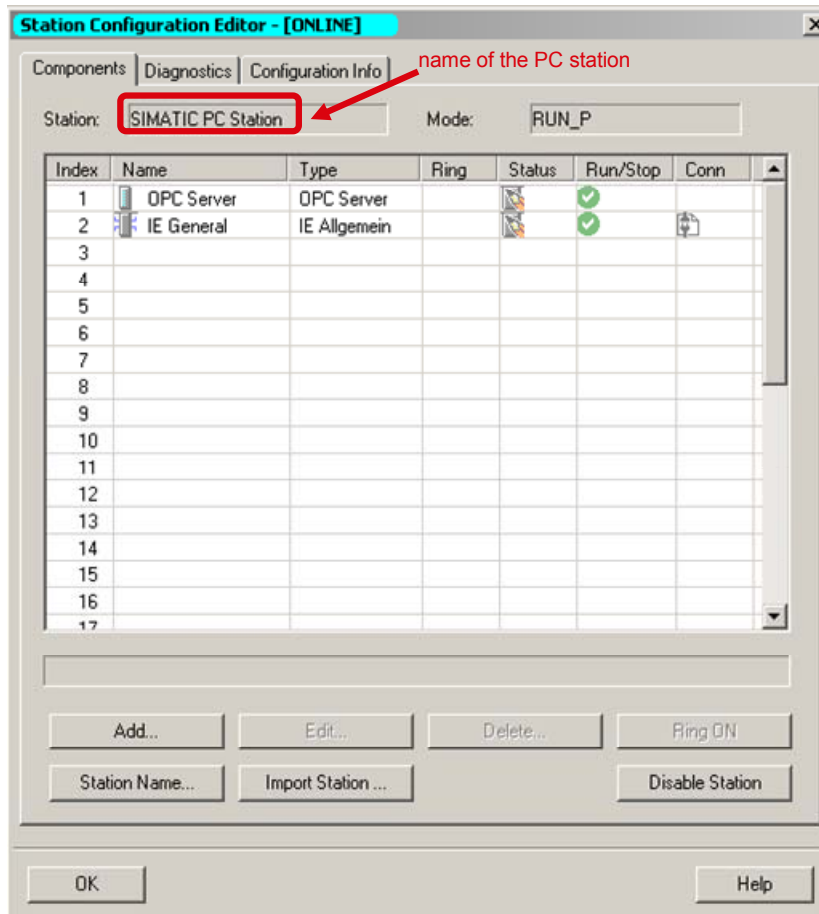
Downloading the configuration to the PC station

In the SIMATIC Manager in the "Options → Set PG/PC Interface" menu, select the TCP/IP interface.

In NetPro, save and compile the configuration of the PC station. To do this, select the "PC Station" object and click the button "Save and Compile".

If the configuration has been successfully compiled, click the button "Download". You thus download the entire PC station configuration, including the configured connections.

Figure 6-10 saving, compiling and downloading the PC station configuration
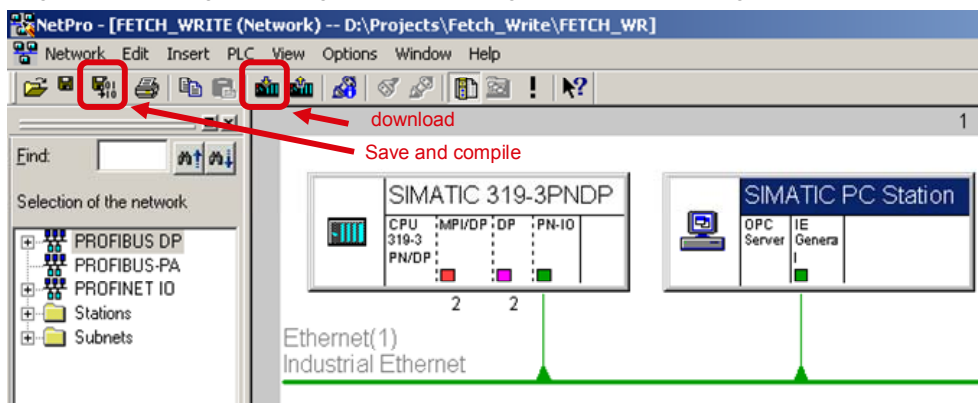
After downloading the configuration, the commissioning of the PC station is completed. It now acts as a FETCH/WRITE client.

7 Operating the Application

7.1 Initiating the terminating the communication connections in the S7 controller

A total of four communication connections are programmed for the data exchange between CPU 319-3 PN/DP and PC station.

The input parameter "ENABLE" of FB210 or FB220 is preassigned with "TRUE". This automatically initiates the setup of the communication connections in the controller.

Figure 7-1 Call of the FB220 in OB1

If the connection parameters of a communication connection are changed, then the establishment of the connection is automatically reinitiated.

7.2 Starting OPC Scout

Start OPC Scout in the Windows START menu by selecting "SIMATIC → SIMATIC NET → OPC Scout".

Open the "T_CON.opp" file that is attached to the application software.

Click the created "Group_1" OPC group. You now see the value and status of the defined OPC items.

When OPC Scout is connected to the SIMATIC NET OPC server and when the communication connections between PC station and CPU 319-3 PN/DP have been

established, the PC station (FETCH/WRITE client) can access the process data of the CPU 319-3 PN/DP by means of the FETCH and WRITE services. With this application software, 8176 bytes = 4088 words of data are transmitted with the FETCH and WRITE services.

Permissible memory areas are the following:

- data blocks
- memory bits
- process-image partition (part process image) for inputs and outputs

Figure 7-2 OPC Scout

In the "T_CON.opp" file, corresponding OPC items are defined to:

- Read out the status of the connections.
  When the name of the OPC item SR:[connection name]&statepath() = UP, the communication connection has been established.

- Write values to data blocks of the CPU 319-3 PN/DP that are evaluated in the S7 program, for example, to control processes. The quality of these items has the value "bad" since these OPC items are only provided with write accesses. By double-clicking the field of the value, you can write values.

- Read values from data blocks of the CPU 319-3 PN/DP that are visualized in the OPC client. If the "quality" of the value is good, your connection has been established and the OPC items can be read.

In this application, the data that are sent from the PC station to the CPU 319-3 PN/DP via the WRITE job are written to DB2. In the S7 station, you can monitor the values of this data via the variable table.

In addition, the data stored in DB1 is read out of the CPU 319-3 PN/DP via the FETCH job. In the S7 station, you can set the values of the data via the variable table.

Figure 7-3 Variable table

		Address	Symbol	Display format	Status value
1		DB1.DBW 0		DEC	-26368
2		DB1.DBW 2		DEC	-26368
3		DB2.DBW 0		DEC	555
4		DB2.DBW 2		DEC	333
5					

VAT_FW -- @FETCH_WRITE\SIMATIC 319-3PNDP\CPU 319-3 PN/DP\S7-Programm

8　Diagnostics

8.1　Structure of the STATUS output parameter

The "STATUS" output parameter of FB210 and FB220 provides information on the status of the communication between S7 station and PC station. The following table shows the structure of this status word.

Table 8-1 Structure of the STATUS parameter

Bit	Instance DB.FBStatus…	comment
High Byte		
0	NotConnected	No connection established or interrupted • FETCH/WRITE client hasn't initialized the connection establishment yet • error during connection establishment
1	ErrorConnect	Error during connection establishment (e.g. incorrect data or structure in the connection data block)
2	ErrorDisConnect	Error during connection clearing
3	ErrorConType	Incorrect connection type in the connection data block
4	Not assigned	
5	BusyConnect	Connection establishment active
6	BusyDisConnect	Connection clearing active
7	Not assigned	
Low Byte		
8	ErrorRec1	TCP connection • Error during receive job • FETCH/WRITE Header telegram isn't received ISO-on-TCP connection • Error during receive job • FETCH/WRITE Header telegram inclusive user data isn't received
9	ErrorRec2	TCP connection • Error during receive job • User data isn't received
10	ErrorSend	TCP and ISO-on-TCP connection • Error during send job • FETCH/WRITE acknowledgement inclusive user data isn't sent
11	Not assigned	
12	BusyRec1	receive job active
13	BusyRec2	receive job active
14	BusySend	send job active
15	ErrorHeader	invalid FETCH/WRITE Header telegram is received

FETCH or WRITE job or user data is received

As soon as the communication connection has been established and the FETCH or WRITE job including user data is received in the S7 station, the status word has the following value:

0001 0000 0000 0000 (bin) = 1000 (hex), i.e.

bit 0 = 0 communication connection has been established

bit 12 = 1 receive job active

Acknowledgement message or user data are sent

When the S7 station sends the acknowledgement message and the user data of the FETCH job or the acknowledgement message of the WRITE job, the status word has the following value:

0100 0000 0000 0000 (bin) = 4000 (hex), i.e.

bit 0 = 0 communication connection has been established

bit 14 = 1 send job active

Communication connection could not be established

If the communication connection could not be established the status word has the following value:

0000 0000 0011 (bin) = 3 (hex), i.e.

bit 0 = 1 no connection established or interrupted

bit 1 = 1 error during connection establishment

The FETCH and WRITE jobs cannot be received in the S7 station if the communication connection between S7 station and PC station has been cleared. No data can be sent or received.

Incorrect connection type

If an incorrect connection is defined in the connection parameters the status word has following value:

0000 0000 0000 1001 (bin) = 9 (hex), i.e.

bit 0 = 1 no connection established or interrupted

bit 3 = 1 incorrect connection type in the connection data block

Invalid FETCH/WRITE Header telegram

If an invalid Header telegram is received for the FETCH and WRITE job respectively the value of the status word changes cyclically.

- 1001 0000 0000 0000 (bin) = 9 (hex), i.e.

bit 12 = receive job active (FETCH/WRITE Header telegram is received)

bit 15 = invalid FETCH/WRITE Header telegram

- 1100 0000 0000 0000 (bin) = C000 (hex), i.e.

bit 14 = send job active (negative acknowledgment of the FETCH and WRITE job respectively)

bit 15 = invalid FETCH/WRITE Header telegram

8.2 Error status of connection establishment and clearing as well as the error status of the send and receive job

The Instance data block of the FB210 „FW_TCP" includes 8348 Byte. The Instance data block of the FB220 „FW_IOT" includes 16544 Byte.

The error status of the connection establishment and clearing as well as the error status of the send and receive job is saved in the Instance data blocks FB210 „FW_TCP" and FB220 „FW_IOT".

The following table shows an overview about the addresses and variables of the Instance data block where the error status are saved.

Table 8-2 addresses and variables of Instance data block

Address	Data type	Variable	Description
22.0	WORD	DoneError.STATUS_Connect	Status connection establishment
24.0	WORD	DoneError.STATUS_REC_1	Status of the first receive job
26.0	WORD	DoneError.STATUS_REC_2	Status of the second receive job (only with TCP)
28.0	WORD	DoneError.STATUS_SEND	Status of the send job
30.0	WORD	DoneError.STATUS_DisConnect	Status connection clearing

Status value connection establishment

The following table shows an overview about the status value of the connection establishment.

Table 8-3 Status value of connection establishment

STATUS (W#16#...)	Description
0000	Connection was established successfully.
8086	The ID parameter is outside the permitted range.
8087	Maximal number of connections reached; no additional connection possible
8089	The parameter CONNECT does not point to a data block.
809A	The parameter CONNECT points to a field that does not match the length of the connection description.
809B	The local_device_id in the connection description does not match the CPU.
80A1	Connection or port is already occupied by user.
80A2	Local or remote port is occupied by the system.
80A4	IP address of the remote connection end point is invalid, it may match the

STATUS (W#16#...)	Description
	local IP address
80B2	The parameter CONNECT parameter points to a data block that was generated with the keyword UNLINKED.
80B3	Inconsistent parameter assignment.
80B4	When using the protocol variant ISO on TCP (connection_type = B#16#12) for passive establishment of a connection (active_est = FALSE), you violated one or both of the following conditions: • local_tsap_id_len >= B#16#02 • local_tsap_id[1] = B#16#E0
80B6	Parameter assignment error relating to the connection_type parameter.
80B7	Error in connection description.
80C3	All connection resources are in use.
80C4	Temporary communications error: • The connection cannot be established at this time. • The interface is receiving new parameters. • The configured connection is currently being removed by a TDISCON.
82xx	General error information

Status value connection clearing

The following table shows an overview about the status value of connection clearing.

Table 8-4 Status value of connection clearing

STATUS (W#16#...)	Description
0000	Connection terminated successfully.
8086	The ID parameter is outside the permitted range.
80A3	Attempt being made to terminate a non-existent connection.
80C4	Temporary communications error: The interface is receiving new parameters or the connection is currently being established.

Status value send job

The following table shows an overview about the status value of the send job.

Table 8-5 Status value of send job

STATUS (W#16#...)	Description
0000	Send job completed without error.
8086	The ID parameter is outside the permitted range.
80A1	Communications error: • The specified connection was not yet established. • The specified connection is currently being terminated. Transmission over this connection is not possible. • The interface is being reinitialized.
80C3	• A block with this ID is already being processed in a different priority class. • Internal lack of resources
80C4	Temporary communications error:

STATUS (W#16#...)	Description
	• The connection to the communications partner cannot be established at this time. • The interface is receiving new parameters or the connection is currently being established.
82xx	General error information

Status value of receive job

The following table shows an overview about the status value of the receive job.

STATUS (W#16#...)	Description
0000	New data were accepted.
8086	The ID parameter is outside the permitted range.
80A1	Communications error: • The specified connection has not yet been established. • The specified connection is currently being terminated. A receive job over this connection is not possible. • The interface is receiving new parameters.
80C3	• A block with this ID is already being processed in a different priority class. • Internal lack of resources
80C4	Temporary communications error: • The connection to the partner cannot be established at the moment. • The interface is receiving new parameter settings or the connection is currently being established.
82xx	General error information

8.3 Negative response of the FETCH or WRITE job

In chapter 3.2.5 "Error number response telegram" you can find an overview about the error numbers in the response telegram of the FETCH or WRITE job.

The FTECH or WRITE job is receipt with error number 2 "Requested block does not exist" when:

• The data block, from which the data are read or to which the data are written, isn't loaded in the CPU.

The FETCH job is receipt with error number 3 "Requested block is to small" when:

• More than 8176 Bytes user data are requested.

• Area boundaries of the memory area are exceeded (start address + FETCH/WRITE Header telegram + user data ≤ size of memory area)
Exception: When the area boundaries of the bit memory address area or PII/PIQ are exceeded with a WRITE job on TCP connection the connection is cleared.

The FTECH or WRITE job is receipt with error number 6 "No valid ORG ID" when the received ORG ID isn't valid, i.e. access to an invalid memory area.

Valid memory areas are:

• Data blocks

• Bit memory address area

• Process image of the inputs or outputs (PII / PIQ)

8.4 Connection clearing

The TCP or ISO-on-TCP connection is cleared when

• More than 8176 bytes are written with the WRITE job.

8 Diagnostics

- An invalid FETCH/WRITE Header-Telegram is received, i.e. the values of the following parameters do not correspond to the structure of the FETCH/WRITE request telegram (see 3.2.1 "Write job"/ 3.2.2 "FETCH job").
  - System ID
  - Length of the request telegram
  - OP-Code ID
  - Length of the OP-Code
  - OP-Code
  - ORG field
  - Length of the ORG field
  - Empty block
  - Length of the empty block

Appendix and List of Further Literature

9 Bibliography

9.1 Internet Links

This list is not complete and only represents a selection of relevant literature.

Table 9-1 Internet links

	Topic	Title
\1\	Chapter 23	System Software for S7-300/400 System and Standard Functions 1214574
\2\	Appendix F – FETCH/WRITE header	S7-CPs for Industrial Ethernet Configuring and Commissioning 8777865
\3\	Configuring PC station	SIMATIC NET Commissioning PC Stations – Manual and Quick Start 13542666
\4\	Siemens I IA/DT Customer Support	http://support.automation.siemens.com

10 History

Table 10-1 History

Version	Date	Modification
V1.0	31.12.07	First issue
V1.4	28.11.08	Complete rework
V2.1	02.04.09	Addition in chapter 7.2 • 8176 bytes = 4088 words of data are transmitted with the FETCH and WRITE services • Permissible memory areas are data blocks, memory bits and process-image partition (part process image) for inputs and outputs Addition in chapter 8.1 • New bit 15 "ErrorHeader" in STATUS output parameter New chapter: • 3.2 "Structure FETCH/WRITE Header telegram" • 8.2 Error status of connection establishment and clearing as well as the error status of send and receive job • 8.3 "Negative response of the FETCH or WRITE job" • 8.4 "Connection clearing"
V2.1	10.07.09	Only modifications in documentation, New chapter: • 2.4 "Performance Data"
V2.1	20.10.09	Change the name of the function blocks FB210 and FB220 • S5FW_TCP → FW_TCP • S5FW_IOT → FW_IOT
V2.2	05.10.10	Addition in chapter 2.5: • Table 2-5 is updated with the new CPUs Addition in chapter 6.2: • Table 6-1 is updated with a note for CPU 315(F)-2 PN/DP