SIEMENS

CPU-CPU Communication with SIMATIC Controllers

SIMATIC S7

http://support.automation.siemens.com/WW/view/de/78028908

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.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Table of Contents

1 Preliminary Remarks on the Document

1.1 Colored text passages

For better orientation in the document, blue text passages have been used in some parts of the document. These blue parts either discuss the system of the document, or they contain examples for using the document. This prevents confusion with the areas only describing the technology (black text).

1.2 Subject of the document

Communication tasks

In the field of automation technology, the communication of controllers plays a crucial part. Controllers perform various communication tasks. The table below illustrates these communication tasks.

Table 1-1

Communication task	Communication partner	Communication	Data (examples)	Network (examples)
Field and process communication	• Controller • Distributed I/O (actuator, sensor)	within a network	Limit-switch positions Temperature values	PROFINET/ Industrial Ethernet PROFIBUS
Data communication	• Controller 1: • Controller 2:	within a network or across network boundaries	Setpoint values Recipes	PROFINET/ Industrial Ethernet PROFIBUS
IT communication	• Controller • PC	worldwide	E-mail File	PROFINET/ Industrial Ethernet Internet

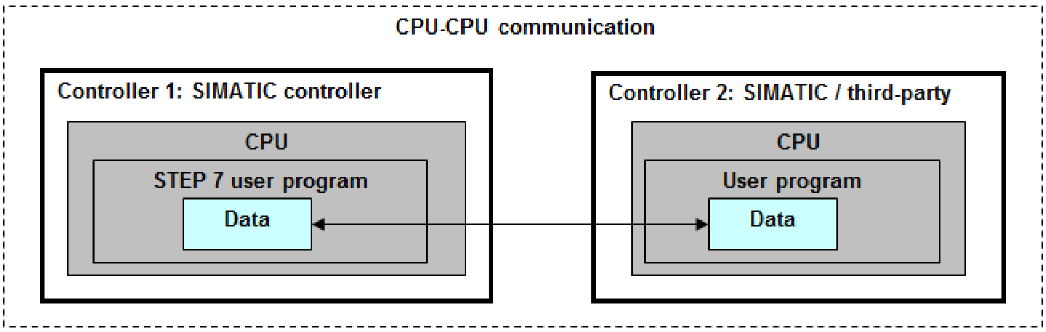
Subject of the document on hand is the data communication between the following communication partners:

- SIMATIC controller / SIMATIC controller
- SIMATIC controller / other controller

CPU-CPU communication

During data communication, data is exchanged between controllers (data blocks, flags, ...). This data is located in the user programs of the CPUs. For clarity reasons, the term "CPU-CPU communication" is used for the term "data communication". The following figure illustrates this.

Figure 1-1



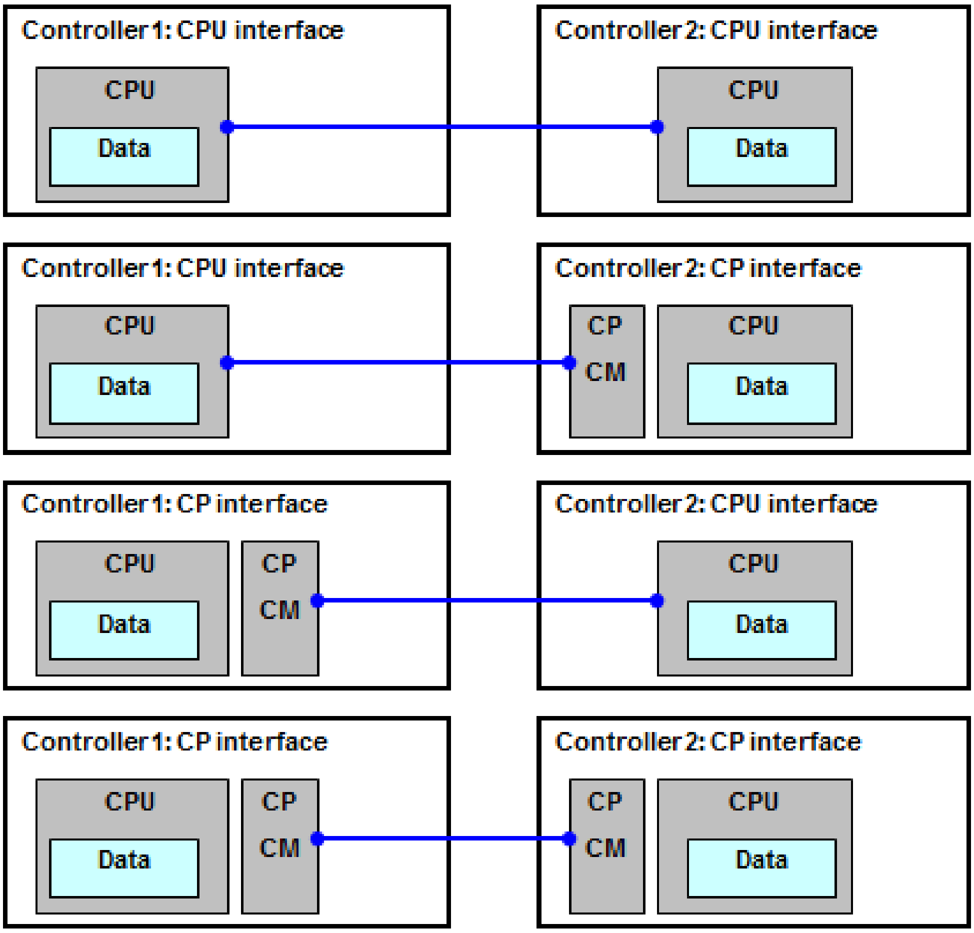
Communication paths

The following interfaces can be used for communication:

- Integrated interface: interface of a SIMATIC CPU
- External interface: interface of a SIMATIC CP of CM

The figure below shows a schematic layout of the interface combinations of Controller 1 and Controller 2 discussed in the document.

Figure 1-2



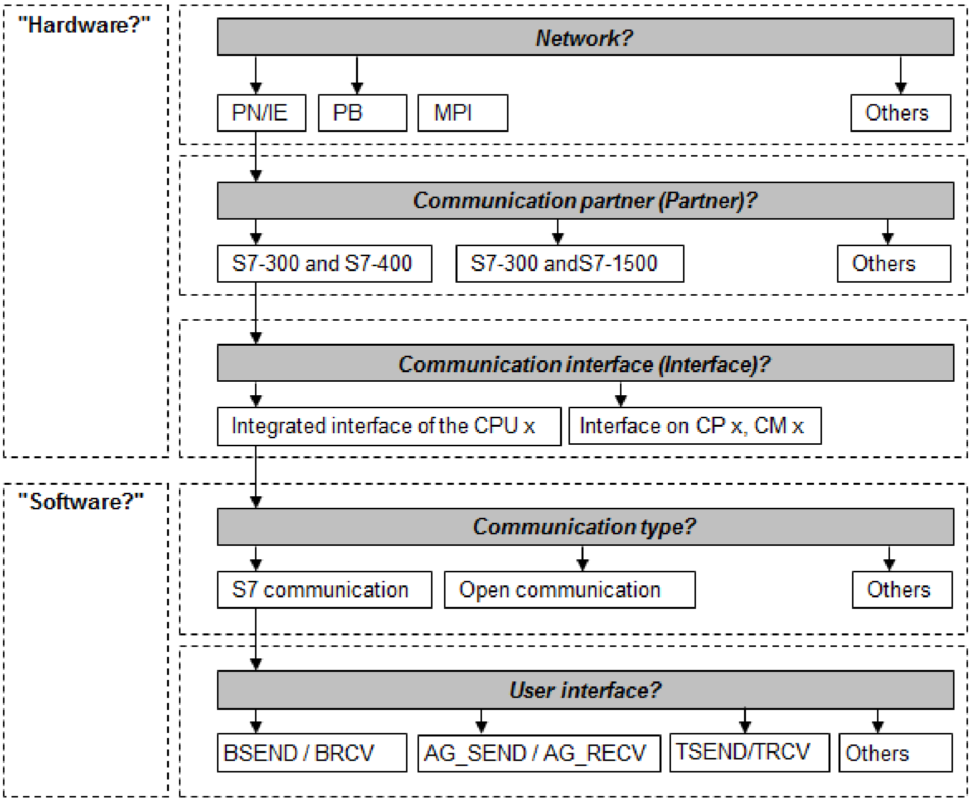
1.3 Purpose and objective of the document

Purpose

To realize a CPU-CPU communication there are a number of different options. In the course of searching for an optimal solution the user is faced with the following questions:

- Which solutions exist?
- In which way do the solutions differ?

Figure 1-3



Objective

The document helps you find an optimal solution for CPU-CPU communication between two SIMATIC controllers, or between a SIMATIC controller and a third-party controller (“other controller”).

In summary, the document answers the question:

Who can communicate with whom, and how?

1.4 Properties and benefits of the document

Properties

The document has the following properties:

- Clear and compact structure
- Content in keywords
- No details also available in other documents are described here. The exact working of communication blocks (BSEND, TSEND, ...), for example, is not described (*1).

(*1)

Details on the communication blocks are available in the

- STEP 7 online help
- device manuals on the S7-CPUs and S7-CPs.
- system and standard functions for SIMATIC S7-CPUs (/6/)
- functions and function blocks for SIMATIC NET S7-CPs (/13/)

Benefits

The document provides the following benefits to the user:

- Support for planning and configuration
- Quick finding of information (reference guide)
- Conveying basic knowledge
- Reference to further information (manual, application example, FAQ, ...)

1.5 Scope of validity of this document

All statements in the document refer exclusively to the most recent SIMATIC controllers (with delivery release):

- **as of March 2013**

In the document, the following communication options are not discussed:

- Field and process communication (sensors, ...)
- IT communication (e-mail, …)
- Communication with standard PC (OPC, …)
- Communication via modem
- F communication
- H communication

In the document, the following components are not discussed:

- SIMATIC Controller LOGO!
- H-CPUs, T-CPUs
- Components to be phased out

1.6 Coverage of the document

Overview

Medium:

- Network: PN/IE, PB, MPI
- SIMATIC backplane bus
- Serial interface (PtP)

SIMATIC families:

- SIMATIC modular controllers:
  ET 200 CPU, S7-300, S7-400, S7-1200, S7-1500
- SIMATIC PC-based controllers:
  WinAC RTX, S7-mEC, Box PC, Panel PC

Communication types:

- SIMATIC-specific
- Open standard

Components

Chapter 65 contains an overview of all the components (CPU, CP, CM) referred to in the document.

Considering the versions of STEP 7

When statements in the document depend on the variant of the engineering tool, then this is referred in the document as follows:

- "STEP 7": STEP 7 up to V5.5 and/or from V10
- "STEP 7 (not TIA)": STEP 7 only up to V5.5
- "STEP 7 (TIA)": STEP 7 only from V10

1.7 Document classification

For SIMATIC, there are a number of documents on the topic of communication. The following table shows how to classify the document on hand.

Table 1-2

Document	Objective of the document	Reference
Document on hand: CPU-CPU communication with SIMATIC controllers	Selection aid for CPU-CPU communication	---
Manuals on the components (S7-CPU, S7-CP, ...)	Technical documentation of the components. (Properties of the interfaces, ...)	/0/
Application examples on the communication	Solutions on specific tasks (documentation and STEP 7 project)	/200/
FAQs on communication	Answers to FAQs	/0/
System manual Communication with SIMATIC	Basic knowledge on industrial communication	/3/
Catalog Products for Totally Integrated Automation and Micro Automation	Ordering document for SIMATIC Controller	/4/
Catalog Industrial Communication	Ordering document for SIMATIC Net products	/5/

2 Structure of the Document

The document consists of several parts (part 1 to part 5). Objective and content of each of the parts is described briefly below.

2.1 PART 1: Introduction

Objective

Part 1 serves as an introduction into the topic of CPU-CPU communication:

- Compact introduction into the topic of communication with SIMATIC
- Explanation of terms and correlations necessary for understanding the document.

Content

The following topics are addressed:

- Function models on CPU-CPU communication
- Connections for SIMATIC
- Data consistency with SIMATIC
- Overview of SIMATIC controllers
- Media for SIMATIC communication
- Interfaces of the SIMATIC families

The end of Part 1 contains a chapter with references to further information.

Details on the content of Part 1: see chapter 4.1.

2.2 PART 2: Selection aid

Objective

Part 2 is the central part of the document:

- Clear representation of all options for a CPU-CPU communication with SIMATIC controllers

Content

Pro Medium (PN/IE, PB, MPI, …) is described:

- For each SIMATIC family (S7-300, …):
  - what interfaces (CPU, CP, CM) are there?
  - what communication types (S7 communication, ...) are there?
- How can the SIMATIC families communicate with each other?
  - via which interfaces?
  - with which communication types?
  - what are the distinctive characteristics (client, server, ...) to be observed?
- Comparison of all available communication types

The end of Part 2 contains a chapter with references to further information.

Details on the content of Part 2: see chapter 18.1.

2.3 PART 3: Communication types

Objective

Part 3 provides in-depth information:

- Detailed information on all communication types

Content

The following is described for each communication type (S7 communication, ...):

- Characteristics
- Properties (table with uniform criteria)
- User interface (communication blocks, ...)

The end of Part 3 contains a chapter with references to further information.

Details on the content of Part 3: See chapter 27.1

2.4 PART 4: Other controllers

Part 4 describes examples for the communication via open protocols between the partners:

- SIMATIC controller
- Third-party controller (other controller)

Example: communication with third-party controllers via Modbus/TCP.

The end of Part 4 contains a chapter with references to further information.

Details on the content of Part 4: see chapter 56.1.

2.5 PART 5: Appendix

Contents of Part 5:

- Related literature
- Terms and abbreviations
- Background information on selected topics
- Overview of the viewed SIMATIC components (CPU, CP, CM)
- History of the document

Details on the content of Part 5: see chapter 60.1.

3 Application of the Document

This document can be used in different ways:

- Reading the document directly on the PC (online)
- Reading the print-out of the document (offline)

The following description considers this. It is noted in brackets whether the described action is possible online or offline.

3.1 Notes on handling the document

Navigation in the document

Since the document is very extensive, provisions have been made to facilitate handling the document.

Table of contents

The detailed table of contends enables specific selection of chapters (online, offline).

Jump distributor

At the beginning of Part 2 of the documentation, there is the chapter jump distributor (chapter 20). For each medium, there is a page with a collection of cross-references. It lists all important chapters on the respective medium. Cross-references are marked in the document (shaded in gray or framed).

Clicking a cross-reference (online) leads to the respective chapter. At the end of this chapter there is a back jump which enables returning quickly to the jump distributor (online). A back jump is marked as a blue and underlined text.

Example: Back to jump distributor PN/IE

Related literature

Bibliographic references in the text are labeled with /x/. Chapter 61 contains a collection of links to the respective sources. Clicking on a link (online) takes you directly to the desired information.

Terms and abbreviations

In chapter 62, important terms are explained. Chapter 63 contains a description of important abbreviations.

Background information

Important correlations are explained in chapter 64.

3.2 Example for the application of the document

The application of the document is illustrated using a specific example.

The objective of this chapter is to illustrate the principle. Therefore, no details are explained in this chapter.

3.2.1 Task description

Known

Two SIMATIC controllers shall communicate via PN/IE network:

- Controller 1: from the S7-300 family
- Controller 2: from the S7-400 family

Sought

The answers to the following questions are sought:

<u>Question 1:</u>

Which interfaces and communication types are available per family?

<u>Question 2:</u>

Which components can communicate with each other, and which communication types are possible?

<u>Question 3:</u>

Which properties have the available communication types?

<u>Question 4:</u>

What do user interfaces (communication blocks) specifically look like?

Solution

Using the jump distributor for the PN/IE network (chapter 20.1) can easily answer the above questions.

This is shown in the following chapters.

3.2.2 Jump distributor

For each medium (PNIE, PB, MPI, ...) there is a so-called jump distributor in the document. The jump distributor contains links (shaded gray) to information in the document.

The figure below shows the jump distributor for PN/IE.

Figure 3-1

20.1 Jump distributor: PN/IE network

1

20.1.1 Overview of interfaces and communication types

Table 20-1 Links to the Interfaces tables

Family	Chapter
ET 200 CPU	21.2.1
S7-300	21.2.2
S7-400	21.2.3
S7-1200	21.2.4
S7-1500	21.2.5
S7-mEC	21.2.6
Box PC	21.2.7
Panel PC	21.2.8
WinAC RTX	21.2.9

2

20.1.2 Combination controller 1 / controller 2

Table 20-2 Links to the Communications tables

Controller 2	Controller 1					
	ET 200 CPU	S7-300	S7-400	S7-1200	S7-1500	WinAC RTX
ET 200 CPU	21.3.1	21.3.2	21.3.3	21.3.4	21.3.5	21.3.6
S7-300	21.3.2	21.4.2	21.4.3	21.4.4	21.4.5	21.4.6
S7-400	21.3.3	21.4.3	21.5.3	21.5.4	21.5.5	21.5.6
S7-1200	21.3.4	21.4.4	21.5.4	21.6.4	21.6.5	21.6.6
S7-1500	21.3.5	21.4.5	21.5.5	21.6.5	21.7.5	21.7.6
WinAC RTX	21.3.6	21.4.6	21.5.6	21.6.6	21.7.6	21.8.6

Note: The SIMATIC controllers S7-mEC, Box PC and Panel PC behave like WinAC RTX with regards to CPU-CPU communication.

3

20.1.3 Communication types

Table 20-3 Links to the "Communication types" tables

Communication type		Chapter
All communication types (compact table)		21.9
Table with details	S7 communication	32.2
	Open communication with send/receive blocks	34.2
	Open communication with T blocks	35.2
	PNIO	37.2
Modbus/TCP (SIMATIC / third-party controller)		57

The jump distributor consists of the following sections:

(1): Overview of interfaces and communication types

(2): Combination controller 1 / controller 2

(3): Communication types

In the following chapters, the application of the sections is demonstrated:

3.2.3 Overview of interfaces and communication types

The following figure shows the “Overview interfaces and communication types” section from the jump distributor for PN/IE.

Figure 3-2

Table 20-1 Links to the Interfaces tables

Family	Chapter
ET 200 CPU	21.2.1
S7-300	21.2.2
S7-400	21.2.3
S7-1200	21.2.4
S7-1500	21.2.5
S7-mEC	21.2.6
Box PC	21.2.7
Panel PC	21.2.8
WinAC RTX	21.2.9

<u>Procedure for answering Question 1: which interfaces and communication types are available per family?</u>

Click on the chapter for the searched SIMATIC family (online), or open the respective chapter (offline).

<u>Result:</u>

The chapter contains a table which lists all interfaces and communication types of the respective family. The following figure shows a section of this table for the S7-300 family.

Figure 3-3

Controller to PN/IE: S7-300			Communication type	
			SIMATIC-specific	Open standard
			S7 communication	Open communication
CPU	Interface: PN (2 ports)		(1)	(3) (IoT, TCP, UDP) (13) (IoT, TCP) (*1)
CP	343-1 Lean	1 x PN (2 ports)	"PUT, GET" Server	(8) (IoT, TCP, UDP) (13) (IoT, TCP)
	343-1	1 x PN (2 ports)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (ISO, IoT, TCP)
	343-1 Advanced	1 x PN (2 ports)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (IoT, TCP, ISO)
		1 x IE (1 port)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (IoT, TCP, ISO)
	343-1 ERPC	1 x IE (1 port)	(1)	(8) (IoT, TCP, UDP) (13) (TCP)

<u>Back to jump distributor PN/IE</u>

<u>Communication blocks</u>

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"   (2) Load/transfer commands, D
(3) TSEND/TRCV, TUSEND/TURCV   (6) PNIO_SEND, PNIO_RECV
(8) AG_SEND/AG_RECV
(13) Server for Fetch, Write (communication blocks required in the server: FW_TCP or F

This is how the table is read (example in the red frame):

- Using CP 343-1, an S7-300 can be operated at the PN/IE.
- One possible communication type is the S7 communication (server and client).
- The communication blocks listed in index (1) can be employed: "USEND/URCV", BSEND/BRCV, "PUT, GET"
- Available protocols: ISO on TCP, ISO

Note: only the principle of the table is shown here. A detailed description of the table structure is available in chapter (19.2).

3.2.4 Combination controller 1 / controller 2

The following figure shows the "Combination controller 1 controller 2" section from the jump distributor for PN/IE.

Figure 3-4

Controller 2	Controller 1					
	ET 200 CPU	S7-300	S7-400	S7-1200	S7-1500	WinAC RTX
ET 200 CPU	21.3.1	21.3.2	21.3.3	21.3.4	21.3.5	21.3.6
S7-300	21.3.2	21.4.2	21.4.3	21.4.4	21.4.5	21.4.6
S7-400	21.3.3	21.4.3	21.5.3	21.5.4	21.5.5	21.5.6
S7-1200	21.3.4	21.4.4	21.5.4	21.6.4	21.6.5	21.6.6
S7-1500	21.3.5	21.4.5	21.5.5	21.6.5	21.7.5	21.7.6
WinAC RTX	21.3.6	21.4.6	21.5.6	21.6.6	21.7.6	21.8.6

Procedure for answering Question 2: which components can communicate with each other and which communication types are possible?

Click the chapter for the searched combination of two SIMATIC families (online), or open the respective chapter (offline).

Result:

The chapter contains a table which lists all combinations of the interfaces (CPU, CP) of both families. The possible communication types have been entered there for each combination. The following figure shows an extract of this table for the families S7-300 and S7-400.

Figure 3-5

Controller 2: S7-400				**Controller 1: S7-300 to PN/IE**							
				CPU				CP			
				Interface: PN				343-1 Lean			
				IOC, IOD				IOD			
				S7	**OC**	**PN**		**S7**	**OC**	**PN**	
						PNIO	CBA			PNIO	CBA
CPU	Interface: PN		IOC, IOD	(1)	(3)	(2)	x	(21)	(8)/(3)	(6)/(2)	---
CP	443-1		IOC, IOD	(1)	(3)/(4)+(9)	(2)	---	(21)	(8)/(4)+(9)	(6)/(2)	---
	443-1 Advanced	X: PN	IOC, IOD	(1)	(3)/(4)+(9)	(2)	x	(21)	(8)/(4)+(9)	(6)/(2)	---
		X: IE	---	(1)	(3)/(4)+(9)	---	---	(21)	(8)/(4)+(9)	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

(21) Controller 1 is server (for "PUT, GET")

This is how the table is read (example in the red frame):

- All S7-300 CPUs with PN interface (controller 1) can communicate with CP 443-1 Advanced (controller 2).
- Possible communication types via the PN interface of the CP:
  - S7 communication (S7)
  - Open communication (OC)
  - PN communication (PN)
- Possible communication types via the IE interface of the CP:
  - S7 communication (S7)
  - Open communication (OC)
- Possible communication blocks for S7 communication:

  Controller 1 and controller 2: communication blocks (1)
- Possible communication blocks for open communication:
  - Controller 1: communication blocks (3)
  - Controller 2: communication blocks (4) and (9)

- Possible communication types for PN communication:
  - PNIO with communication blocks (2)
  - CBA

Note: Only the principle of the table shall be shown here. A detailed description of the table structure is available in chapter (19.3).

3.2.5 Communication types

The following figure shows the "Communication types" section from the jump distributor for PN/IE.

Figure 3-6

Table 20-3 Links to the "Communication types" tables

Communication type		Chapter
All communication types (compact table)		21.9
Table with details	S7 communication	32.2
	Open communication with send/receive blocks	34.2
	Open communication with T blocks	35.2
	PNIO	37.2
Modbus/TCP (SIMATIC / third-party controller)		57

1

2

Overview of all communication types

Procedure for answering Question 3: which properties have the available communication types?

Click on (online) or open (offline) chapter (1).

Result:

The table in this chapter shows a comparison of all communication types possible via PN/IE. The following figure shows a section from the table.

Figure 3-7

	SIMATIC-specific	Open standard	
	S7 communication	Open communication	
		Send/Receive blocks	T blocks T-Compact bocks
Protocols	ISO (only CP), IoT	ISO, IoT, TCP, UDP	IoT, TCP, UDP
Interfaces	CPU, CP, CM	CP	CPU, CP, CM
Communication blocks (max. data)	BSEND (≤ 64 Kbytes) Type "USEND/URCV" (≥ 160 bytes) Type "PUT, GET" (≥ 160 bytes)	AG_xSEND (ISO, IoT, TCP ≤ 8 Kbytes) (UDP ≤ 2 Kbytes) Server for FETCH, WRITE (not UDP)	TSEND, TSEND_C, ... (IoT ≤ 32 Kbytes) (TCP ≤ 64 Kbytes) (UDP = 1472 bytes)
Remote confirmation	BSEND: application Type "USEND / URCV": transport Type "PUT, GET": application	ISO, IoT, TCP: transport UDP: none	IoT, TCP: transport UDP: none
Connections?	yes	ISO, IoT, TCP: yes UDP: no	IoT, TCP: yes UDP: no

Details on a specific communication type

Procedure for answering Question 4: what do user interfaces (communication blocks) specifically look like?

Click on (online) or open (offline) chapter (2).

Result:

The table in this chapter shows all important properties of communication type “S7 communication”. The figure below shows the table.

Figure 3-8

Communication type:		**S7 communication**		
Protocol:		**S7 protocol**		
General				
Media		MPI, PB, PN/IE, backplane bus (only for SIMATIC S7-400, multicomputing)		
Interfaces		CPU, CP, CM		
Connection	SIMATIC S5	no		
	third-party (open standards)	no		
Protocol				
Dynamic data length		yes		
Multicast / broadcast		no		
Connections	to the remote partner?	yes		
	dynamic / static	static		
User interface				
Communication blocks		**BSEND / BRCV**	**Type "USEND / URCV"**	**Type "PUT, GET"**
Maximal number of data **(*1)**		<= 64 Kbytes	>= 160 bytes	>= 160 bytes
Dynamic addressing of data		S7-300: yes	S7-300: yes	S7-300: yes
		other: no	other: no	other: no
Remote confirmation		Application	Transport	Application
Model		Client / Client	Client / Client	Client / Server

If further information on the communication blocks is required, then these can be read up in the chapter on S7 communication.

Example: parameters of the BSEND communication block

Figure 3-9

Communication type:		**S7 communication**		
Protocol:		**S7 protocol**		
General				
Media		MPI, PB, PN/IE, backplane bus (only for SIMATIC S7-400, multicomputing)		
Interfaces		CPU, CP, CM		
Connection	SIMATIC S5	no		
	third-party (open standards)	no		
Protocol				
Dynamic data length		yes		
Multicast / broadcast		no		
Connections	to the remote partner?	yes		
	dynamic / static	static		
User interface				
Communication blocks		**BSEND / BRCV**	**Type "USEND / URCV"**	**Type "PUT, GET"**
Maximal number of data **(*1)**		<= 64 Kbytes	>= 160 bytes	>= 160 bytes
Dynamic addressing of data		S7-300: yes	S7-300: yes	S7-300: yes
		other: no	other: no	other: no
Remote confirmation		Application	Transport	Application
Model		Client / Client	Client / Client	Client / Server

3.2.6 Summary

The following figure illustrates how the jump distributor works in the document using the example of the PN/IE network.

The jump distributor contains links to all chapters possibly of interest for CPU-CPU communication via PN/IE:

(1): link to the Interfaces table

(2): link to the Combinations table

(3): link to the “Communication types – compact” table

(4): link to the “Communication types – detailed” table

(5): link to the Description

Figure 3-10

PART2	Link to:	PN/IE Net	Overview interfaces ans communication types		1
			Combinations of Controller 1 / Controller 2		2
			Communication Types	Overview (all)	3
				Details (specific)	4
				Third party Contr.	5
		PB Net	*above*		
		MPI Net	*above*		
		SIMATIC Backplane Bus	*above*		
		Serial Interface	*above*		
	Selection aid for:	PN/IE	Table Interfaces		
			Table Combinations		
			Table Types of Communications		
		PB	*above*		
		MPI	*above*		
		SIMATIC Backplane Bus	*above*		
		Serial Interface	*above*		
PART3	Communi-cation Types:	S7-Communikation	Table Types of Communication (Details)		
			User Interfaces (FBs, FCs, ...)		
		... others ...	*above*		
PART4	Third party controller:	Modbus/TCP	Descriptions		
		Modbus Serial (RTU)	*above*		

4 ********** PART 1: Introduction ***************

4.1 Structure and content

Table 4-1

<table>
<tr><th>Chapt.</th><th colspan="2">Structure</th><th>Content</th></tr>
<tr><td>5</td><td colspan="2">Models on CPU-CPU communication</td><td>Overview of the function models</td></tr>
<tr><td>6</td><td colspan="2">Connections for SIMATIC</td><td>The most important information on connections</td></tr>
<tr><td>7</td><td colspan="2">Data consistency with SIMATIC</td><td>Explanations on data consistency</td></tr>
<tr><td>8</td><td colspan="2">SIMATIC controller</td><td>Overview of SIMATIC Controller and families</td></tr>
<tr><td>9</td><td colspan="2">Media for SIMATIC communication</td><td>Overview of all media on SIMATIC communication</td></tr>
<tr><td>10</td><td></td><td>PROFINET/Industrial Ethernet (PN/IE)</td><td rowspan="5">The following is described per medium:
• Characteristics
• ISO/OSI reference model</td></tr>
<tr><td>11</td><td></td><td>PROFIBUS (PB)</td></tr>
<tr><td>12</td><td></td><td>MPI</td></tr>
<tr><td>13</td><td></td><td>SIMATIC backplane bus</td></tr>
<tr><td>14</td><td></td><td>Serial Interface (PtP)</td></tr>
<tr><td>15</td><td></td><td>Comparison of the Media</td><td>Table comparison of the media</td></tr>
<tr><td>16</td><td colspan="2">Interfaces of the SIMATIC families</td><td>Overview of all SIMATIC interfaces</td></tr>
<tr><td>17</td><td colspan="2">Information</td><td>Notes regarding further information</td></tr>
</table>

5 Models on CPU-CPU Communication

5.1 Definition of controller

The following definitions are used in this document:

A controller is a central or decentralized automation station (station) with the components CPU, CP (optional), CM (optional) and distributed I/O. Within the station, the components are connected via the backplane bus.

Central station:

- contains a distributed I/O
- communicates with distributed stations via PROFINET IO or PROFIBUS DP

Decentralized station:

- contains a distributed I/O
- communicates with the central station
  via PROFINET IO or PROFIBUS DP

5.2 Definition CPU-CPU communication

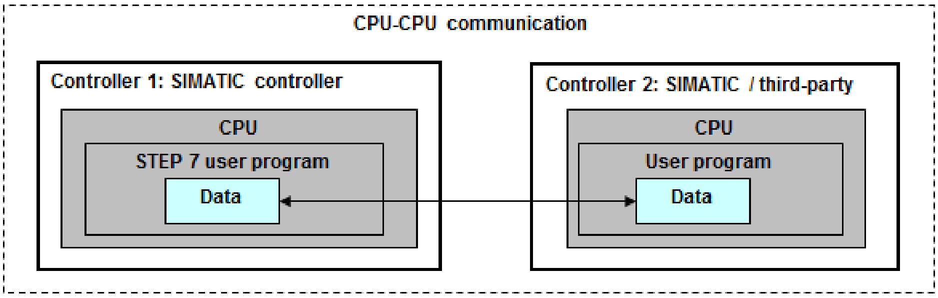
For CPU-CPU communication, data is exchanged between the CPUs of two controllers:

- Controller 1: SIMATIC controller
- Controller 2: SIMATIC controller or other controller

Source or destination of the data is the user data area of the CPU of the controller:

- data block, flag, inputs, outputs, ...

Figure 5-1



For CPU-CPU communication, the following cases are differentiated:

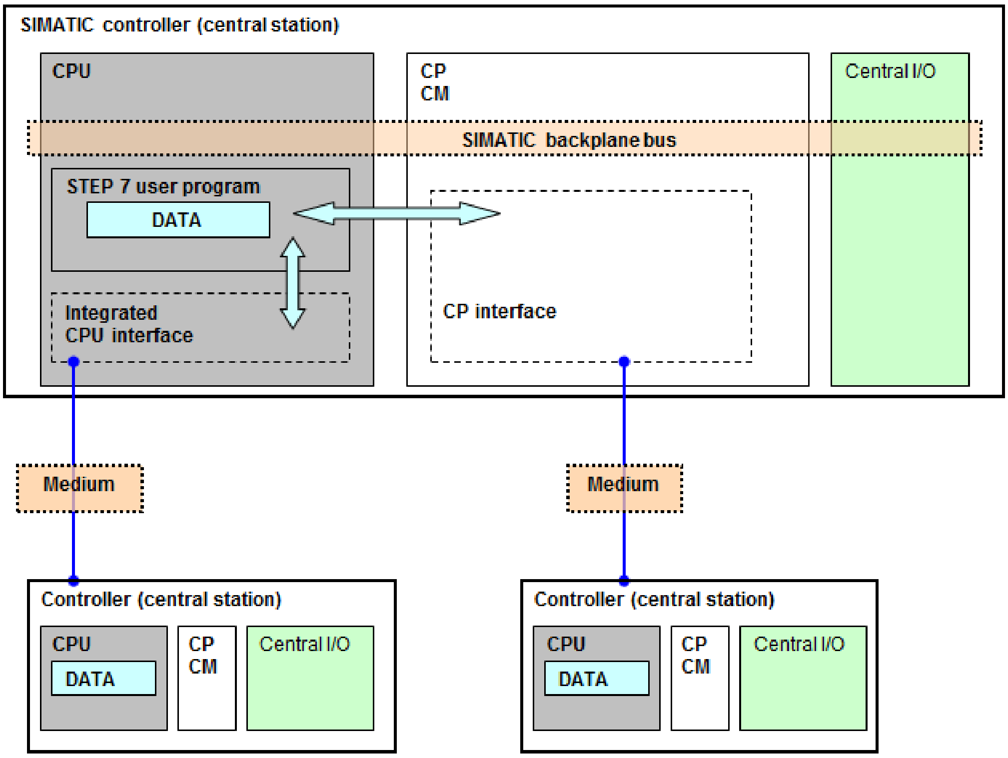
- CPUs in different central stations
- CPUs in central and decentralized station (*1)
- CPUs within a central SIMATIC station

(*1): a decentralized station with CPU is also referred to as I-slave (for PROFIBUS) or I-device (for PROFINET).

5.3 CPUs in different central stations

The figure shows the function model for the CPU-CPU communication between distributed stations.

Figure 5-2



Interfaces for communication:

- Integrated interface: interface to CPU
- External interface: interface to CP or CM

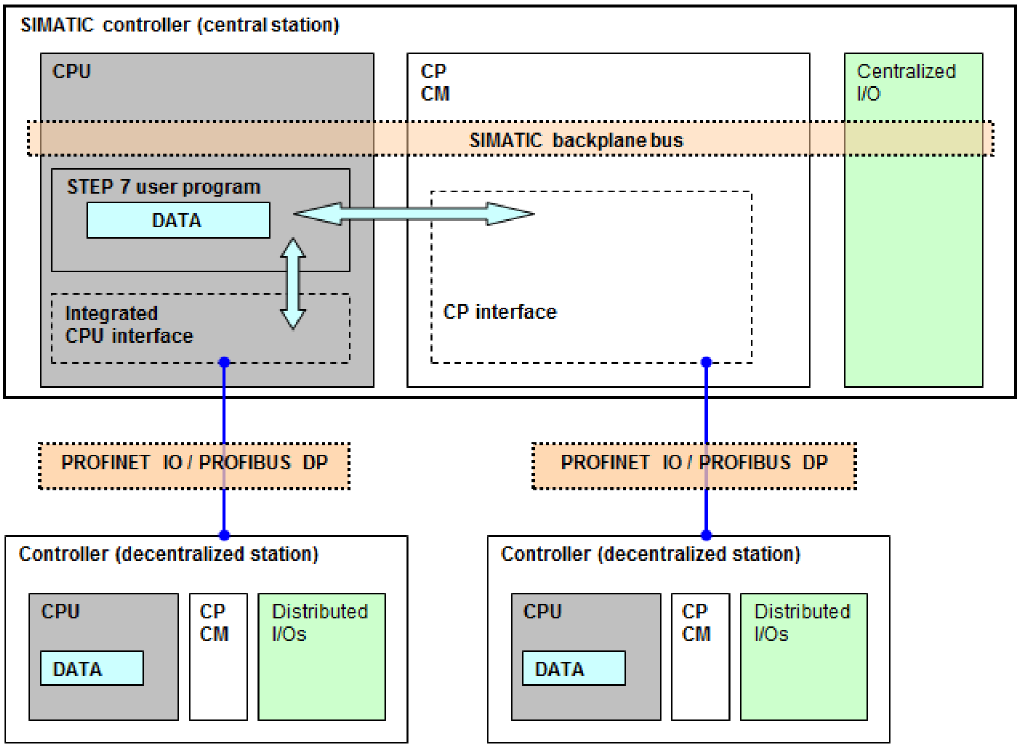
Media for communication:

- Network (PROFINET/Industrial Ethernet, PROFIBUS, MPI)
- Serial interface (*ASCII*, 3964(R), RK 512, …)

5.4 CPUs in central and decentralized station

The figure shows the functional model for the CPU-CPU communication between central and decentralized station.

Figure 5-3



Interfaces for communication:

- Integrated interface: interface to CPU
- External interface: interface to CP or CM

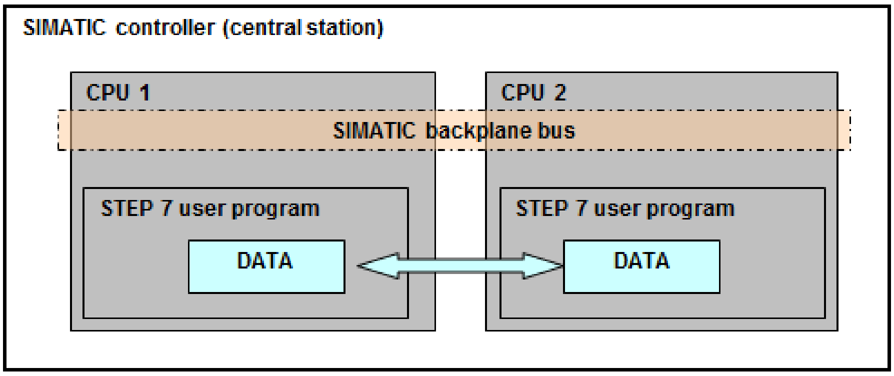
Media for communication:

- PROFINET/Industrial Ethernet (PROFINET IO)
- PROFIBUS (PROFIBUS DP)

5.5 CPUs within a central station

The figure shows the functional model for the CPU-CPU communication between CPUs within a central SIMATIC station.

Figure 5-4



Medium for communication:

- SIMATIC backplane bus

Note

This is only possible for S7-400, and is referred to as “multi-computing” there. Up to 4 S7-CPUs in a central SIMATIC station can be operated simultaneously.

6 Connections for SIMATIC

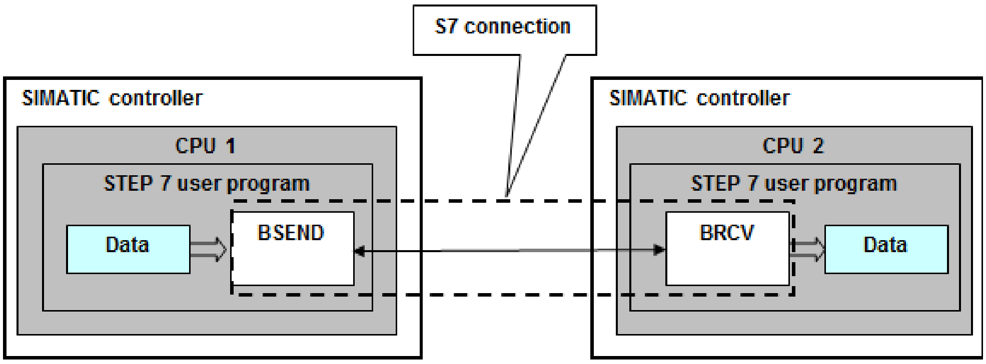
6.1 Connections

Introduction

Generally, CPU-CPU communication with SIMATIC occurs via connections. A connection defines the location of the end points of the communication.

The figure below shows an example of a connection:
CPU 1 communicates with CPU 2 via an S7 connection.

Figure 6-1



Connections must be established (chapter 6.4). This can be performed automatically, by configuring, or by programming.

Properties of a connection

A connections is defined by the following properties (examples):

- involved CPUs (CPU 1, CPU 2)
- used protocol (TCP, ISO on TCP, ...)
- behavior after data transmission (connection remains established, or is cancelled)

Function of a connection

During active connection establishment, the interfaces of both communication partners (integrated interface of the CPU, or external interface of a CP or CM) negotiate the connection parameters (maximal message frame length, ...).

The active communication partner suggests a value. The other communication partner confirms or suggests something else. The negotiated value then applies for the duration of the connection.

During an established connection, the following tasks are performed by the operating system of the CPU, CP or CM:

- Flow control (avoiding overload in the partners, ...)
- Monitoring the connection (verification whether partner can still be reached, ...)
- Exchange of acknowledgements (confirmation that data has arrived without error, …)

6.2 Protocols

A protocol defines the rules of the communication between two communication partners. The ISO/OSI reference model is often used to classify a protocol (chapter 64.1). Two classes are described for the transport protocols.

Connection-based protocols

These protocols establish a connection between both communication partners (examples: TCP, ISO on TCP). The protocols are used if the priority lies on a reliable communication between the control programs of two CPUs.

Protocols without connection

These protocols do not establish a connection between both communication partners (example: UDP). The protocols are used if the priority lies on a fast communication between the control programs of two CPUs.

6.3 Connection resources

Meaning

Connections use CPU, CP or CM resources (example: buffer for received data). With SIMATIC, these resources are referred to as “Connection resources”.

The number of maximal possible connections per CPU, CP or CM is limited. It depends on the available connection resources of CPU, CP or CM. The technical data in the manuals for the modules therefore states the maximum number of connections possible for each communication type. The following figure shows an example from the S7-300 manual.

Figure 6-2

Technical specifications of CPU 31x

9.7 CPU 317-2 PN/DP

Technical specifications	
TCP/IP	Yes (via integrated PROFINET interface and loadable FBs)
• Maximum number of connections	16
• Data length for connection type 01_H, max.	1460 bytes
• Data length for connection type 11_H, max.	32768 bytes
• Multiple passive connections per port (multiport), Supported	Yes
ISO on TCP	Yes (via integrated PROFINET interface and loadable FBs)
• Maximum number of connections	16
• Data length, max.	32768 bytes
UDP	Yes (via integrated PROFINET interface and loadable FBs)
• Maximum number of connections	16
• Data length, max.	1472 bytes

Assignment and release

Connection resources are assigned when establishing the connection, and released again when the connection is terminated.

If connections are configured in STEP 7, then STEP 7 monitors meeting the technical data regarding the maximum number of connections possible.

If connections are established or terminated in the STEP 7 user program, the user must manage the number of connections used by himself. This means, the user must ensure that the maximum number of possible connections of the modules involved in the communication is met.

6.4 Establishing connections

Connections must be established. There are two cases to be distinguished:

- configured connection
- non-configured connection

6.4.1 Configured connection

Configured connections are connections configured in the hardware configuration of STEP 7. Configured connections are, for example, used for the following communication types:

- S7 communication
- Open communication with send/receive blocks

Steps for realizing a communication:

- Configuration with STEP 7 (hardware configuration):
  connection between the communication partners
- Programming in STEP 7 (user program):
  calling the communication block (for data transmission)

Connecting, disconnecting and interrupting a connection:

- Connecting: automatic connection establishment when starting up the communication-capable modules (CPU, CP, CM). When establishing the connection, the required connection resources are assigned by the operating system of the modules.
- Disconnecting: the connection is <u>not</u> cancelled after the data transfer, i.e. the connection resources remain permanently assigned.
- Interrupting (e.g. due to a failure): after the interruption, the connection is automatically reestablished by the operating system. Possibly released connection resources are reassigned.

6.4.2 Non-configured connection

Non-configured connections are connections which are not configured in the hardware configuration of STEP 7.

There are two cases to be distinguished:

- Automatic establishing of a connection (example: S7 basic communication)
- Programmed establishing of a connection (example: open communication)

Automatic establishing of a connection

Step for realizing a communication:

- Programming in STEP 7 (user program):
  calling the communication block (for data transmission)

Connecting, disconnecting and interrupting a connection:

- Connecting: at the first call of the communication block, the operating system of the modules (CPU, CP, CM) establishes a connection. During establishing the connection, the required connection resources are assigned by the operating system of the modules.
- Disconnecting: the configuration at the communication block defines whether the connection remains after completing the data transmission, or whether the connection is cancelled. Disconnecting an existing connection releases previously assigned connection resources.
- Interruption (e.g. due to a failure): there are two possibilities for the behavior after an interrupted connection:
  - the operation system keeps re-establishing the connection until the connection is cancelled by the control program.
  - the control program detects the interruption and reestablishes the connection.

Programmed establishing of a connection

Two different options are available:
connection management with connection blocks or with communication blocks.

Connection management with **connection** blocks

Steps for realizing a communication:

- Programming in STEP 7 (user program):
calling the connection block (example: T_CONFIG)
- Programming in STEP 7 (user program):
calling the communication block (for data transmission, example TSEND)

Connecting, disconnecting and interrupting the connection:

- Connecting: when calling the connection block, the operating system of the modules (CPU, CP, CM) establishes a connection.
- Disconnecting: the connections can be disconnected by calling a connection block. Connection resources are released again after the disconnection.
- Interruption (e.g. due to a failure): there are two possibilities for the behavior after an interrupted connection:
  - the operation system keeps re-establishing the connection until the connection is cancelled by the control program.
  - the control program detects the interruption and reestablishes the connection.

Connection management with **communication** blocks

Steps for realizing a communication:

- Programming in STEP 7 (user program):
calling the communication block (for connection management and for data transmission) (example: TSEND_C)
- Programming in STEP 7 (user program):
Parameterization of a communication block (defining the connection parameters).

Connecting, disconnecting and interrupting the connection:

- Connecting: when calling the communication block, the operating system of the modules (CPU, CP, CM) establishes a connection.
- Disconnecting: the configuration at the communication block defines whether the connection remains after completing the data transmission, or whether the connection is cancelled. Disconnecting an existing connection releases previously assigned connection resources.
- Interruption (e.g. due to a failure): there are two possibilities for the behavior after an interrupted connection:
  - the operation system keeps re-establishing the connection until the connection is cancelled by the control program.
  - the control program detects the interruption and reestablishes the connection.

7 Data Consistency with SIMATIC

In this chapter, the topic of data consistency is discussed from the point of view of CPU-CPU communication with SIMATIC controllers.

7.1 Definitions

Data area

A data area is a coherent area of data in the user memory of a SIMATIC CPU (for example: MW100 to MW200).

Consistent data

A data area which cannot be changed simultaneously by competing processes (user programs, operating system programs, ...) is referred to as a consistent (connected) data area.

This data area contains consistent data. In this document, the size of this data area is referred to as “consistent data volume”.

Inconsistent data

A data area larger than the consistent data area can be falsified. The data area can then at a certain time consist of new and partially of old data areas.

<u>Example</u>

Inconsistent data can occur if in the STEP 7 user program a running communication block is interrupted by a process alarm OB with higher priority. If in this OB, the user program then changes the data, which partially has already been processed by the communication block, then inconsistent data may occur. The data in the example is inconsistent (not belonging together), because:

- a part of the data is taken from the time before the process alarm processing (“old data area”)
- a part of the data is taken from the time after the process alarm processing (“new data area”)

Data consistency

The following two options are distinguished for data consistency:

- system-related data consistency
- advanced data consistency

System-related data consistency

Data consistency for a maximum number of data ("consistent data volume") always guaranteed by the operating system of a CPU or a CP. Data areas smaller than "consistent data volume" are required in the STEP 7 user program don’t require any measures in order to guarantee data consistency.

Advanced data consistency

If data consistency is required for a data area larger than "consistent data volume", then this can be achieved by additional measures in the STEP 7 user program (example: blocking the process alarm OB for the duration of the data transfer).

7.2 System-related data consistency

The operating system of the SIMATIC controller guarantees a system-related data consistency. This system-related data consistency depends on:

- CPU type (if communication with the CPU)
- CP type or CM and CPU type (if communication with the support of a CP or CM)
- Communication block type

Statements on the various SIMATIC families are given below.

S7-300

For S7-300 the data is copied consistently from the operating system to the STEP 7 user memory in blocks of x bytes (see table below).

Copying is performed in the cycle checkpoint of the operating system. Data consistency is not guaranteed by the system for larger data areas.

If a certain data consistency is demanded, the data in the user program must not be larger than these x bytes.

Table 7-1

Case discrimination	Consistent data volume
Communication via integrated interface of the CPU	64 bytes up to 240 bytes (*1)
Communication via CP	32 bytes

(*1): The concrete values for the “consistent data volume” are available in the manuals of the CPUs or CPs.

S7-400

With S7-400 the data are, in contrast to the S7-300, not processed in the cycle checkpoint of the operating system, but infixed time disks during the cycle (OB1).

The data consistency of a tag is ensured.

S7-1200

The CPU guarantees the data consistency for all elementary data types (example: Word or DWord) and all system-defined structures (example: IEC_TIMERS or DTL).

S7-1500

The CPU ensures the data consistency for a tag.

7.3 Advanced data consistency

Additional measures

In order to ensure advanced data consistency, additional measures must be taken in the STEP 7 user program of the sender and receiver.

Additional measures in the sender

Access to the send area (data block, flag, …) only after the data was transferred completely. This can be read at the control parameters of the communication blocks (example: DONE = 1).

Additional measures in the receiver

Access to the receive area (data block, flag, …) only after the data was received completely. This can be read at the control parameters of the communication blocks (example: NDR = 1).

Subsequently, blocking the receive area until the data was processed. This can be read at the control parameters of the communication blocks (example: EN_R = 0).

Case discrimination

Two cases must be distinguished:

Client Client communication

Examples for communication blocks: BSEND / BRCV

If advanced data consistency is to be ensured, then the data must not be modified during the transmission.

Client Server communication

Examples for communication blocks: PUT, GET

The STEP 7 user program of the server does not contain any communication block. Therefore, the access to the data in the user program cannot be coordinated. The system-related size of the consistent data areas (system-related data consistency) must already be considered during the programming or configuration process.

8 SIMATIC Controller

An overview of the SIMATIC families viewed in the document is given below (/1/).

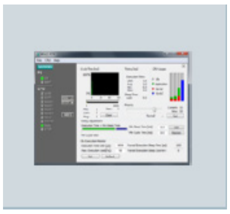
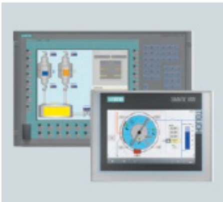
SIMATIC modular controller

Table 8-1

SIMATIC family	
ET 200 CPU Modular, distributed I/O system with intelligence on site	
S7-300 Modular controller in manufacturing industry	
S7-400 Modular controller for manufacturing and process industry	
S7-1200 Modular, compact controller	
S7-1500 Modular controller for manufacturing and process industry	

SIMATIC PC-based controller

Table 8-2

SIMATIC family	
WinAC RTX Software controller	
S7 Modular Embedded Controller (S7-mEC) S7-300 design	
Embedded Box PC Bundles (Box PC) Top hat rail PC, ready to be switched on (bundles with WINAC RTX)	
Embedded Panel PC Bundles (Panel PC) Panel PC ready to be switched on (bundles with WINAC RTX)	

9 Media for SIMATIC Communication

For SIMATIC, there are various options to realize a CPU-CPU communication. Data transmission can be performed via different media.

The table below shows which media are available.

Table 9-1

Medium		Communication partners: SIMATIC S7 controller	Communication partners: Other controller
Network	PROFINET/Industrial Ethernet (PN/IE)	x	x
	PROFIBUS (PB)	x	x
	MPI	x	---
SIMATIC backplane bus		x	---
Serial Interface (PtP)		x	x

In the following chapters, the media are described briefly. Details on the media can be read up in the literature (chapter 17).

Overview of the following chapters:

Table 9-2

Media for SIMATIC Communication		Chapter
PROFINET/Industrial Ethernet (PN/IE)		10
	Preliminary remarks	10.1
	Ethernet	10.2
	Industrial Ethernet (IE)	10.3
	PROFINET (PN)	10.4
PROFIBUS (PB)		11
MPI		12
SIMATIC backplane bus		13
Serial Interface (PtP)		14
Comparison of the media		15
Interfaces of the SIMATIC families		16

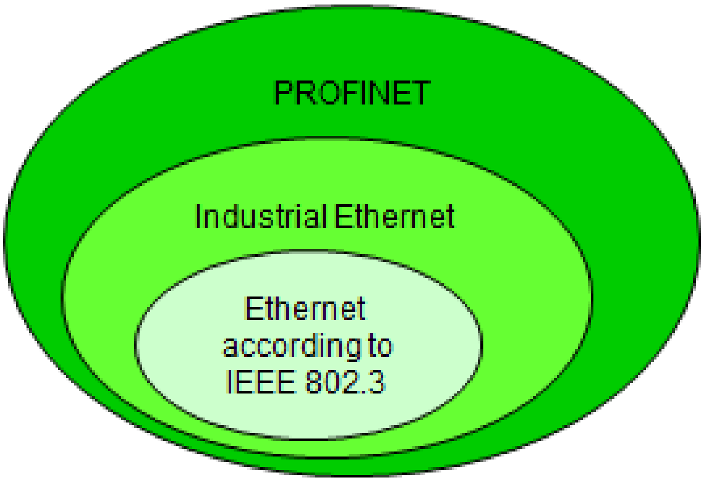
10 PROFINET/Industrial Ethernet (PN/IE)

10.1 Preliminary remarks

The terms PROFINET, Industrial Ethernet and Ethernet are all related to each other:

- PROFINET is based on Industrial Ethernet
- Industrial Ethernet is based on Ethernet.

Figure 10-1



The following chapters briefly describe Ethernet, Industrial Ethernet and PROFINET.

10.2 Ethernet

Introduction

Ethernet is the standard for networks (LAN) in the office sector.

Characteristics

- International standard: IEEE 802.3
- Worldwide use
- Simple and standardized wiring
- Basis for higher level protocols (TCP/IP, UDP, …)
- Fail-safe networks through redundancy
- Simple connection to wireless networks
  (Industrial Wireless LAN, according to IEEE 802.11)
- Scalable performance through switched Ethernet (*1)

(*1): Switched-Ethernet divides the network into subnets connected by switches.

This makes it possible to realize the following functionality:

- Several pairs of stations are connected with each other at the same time. Each connection has the full data throughput.
- Local data traffic remains local. Only the data of another subnet is forwarded by switches.

Advantage of Switched-Ethernet:

- Increased data throughput through structuring the data traffic

With SIMATIC, a switch can be realized in different ways:

- as an independent component (SCALANCE X)
- integrated in SIMATIC components (CPU, CP or CM with PN/IE connection)

ISO/OSI reference model

Ethernet comprises layer 1 and layer 2 of the ISO/OSI reference model:

- Layer 2: access control and addressing (MAC addresses)
- Layer 1: transmission technology (physics)

Figure 10-2

Layer 7	Application Layer		
Layer 6	Presentation Layer		
Layer 5	Session Layer		
Layer 4	Transport Layer		
Layer 3	Network Layer		
Layer 2	Data Link Layer	IEEE 802.2 Logical Link Control (LLC)	
		IEEE 802.1 Media Access Control (MAC)	**Ethernet**
Layer 1	Physical Layer	IEEE 802.3 Ethernet	

10.3 Industrial Ethernet (IE)

Introduction

IE is the Ethernet variant suitable for industrial applications.

Characteristics

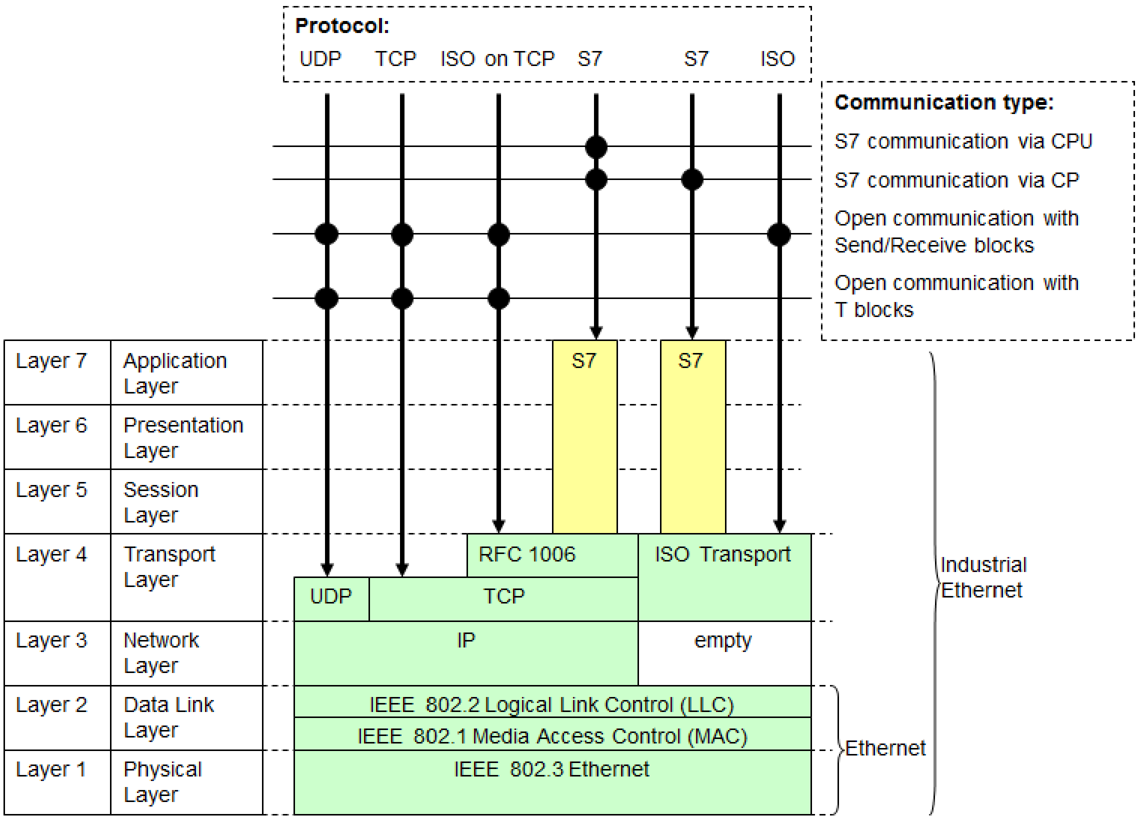
Apart from the characteristics for Ethernet (chapter 10.2), the following additional characteristics exist for Industrial Ethernet:

- connects different application areas: office and production
- uses the possibilities of the IT standard (known from the office sector) in automation (browser, e-mail, …)
- optimized communication between automation components and simultaneous communication according to TCP/IP (open standard)
- network components for the application in harsh industrial environments (dust, dampness, vibrations, …)
- simple connection system on site.

ISO/OSI reference model

Industrial Ethernet comprises layer 1 to layer 7 of the ISO/OSI reference model: The figure below shows all protocols and communication types supported by the SIMATIC controllers.

Figure 10-3



10.4 PROFINET (PN)

Introduction

PN is the open Industrial Ethernet standard for automation. PN is based on Industrial Ethernet.

Variants

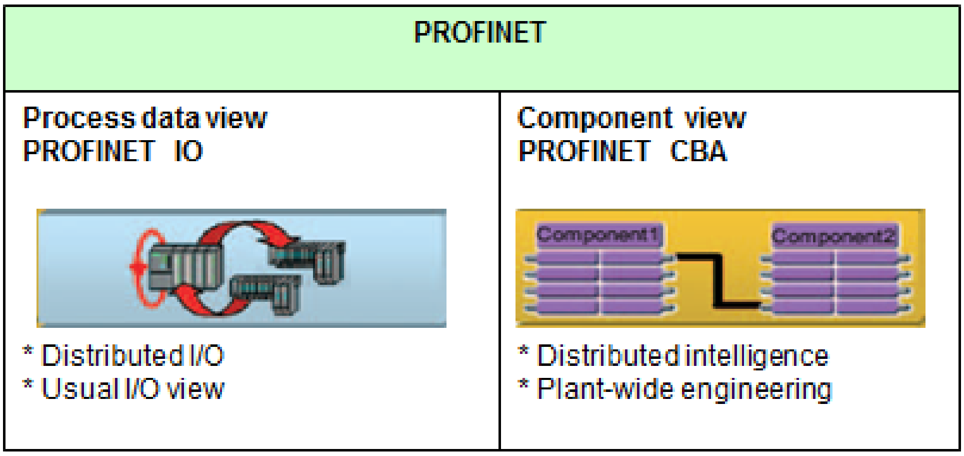
Within the framework of PROFINET there are two variants:

- PROFINET IO:
  Automation concept for realizing modular applications through integration of distributed I/O with real-time communication.
- PROFINET CBA:
  Components model for automation solutions based on distributed components and sub-functions.

Views

PROFINET IO and PROFINET CBA are two different views for automation devices in Industrial Ethernet.

Figure 10-4



PROFINET IO provides a picture of the automation plant which is similar to that of the PROFIBUS DP view. The individual automation devices are configured and programmed.

PROFINET CBA divides and entire automation plant into different functions. These functions are configured and programmed.

Characteristics

PROFINET

- International standard: IEC 61158, IEC 61784
- Integrated communication via fieldbus and Ethernet
- Integration of existing fieldbus systems (PROFIBUS, ASi)
- Using the TCP/IP protocol
- Communication in real-time
- Clock-synchronized drive control for motion control applications

PROFINET IO

- Communication of field devices (IO device) with controllers (IO controller)
- IO view, as for PROFIBUS DP

PROFINET CBA

- Communication between CBA components
- Communication is configured (with Tool iMap), not programmed

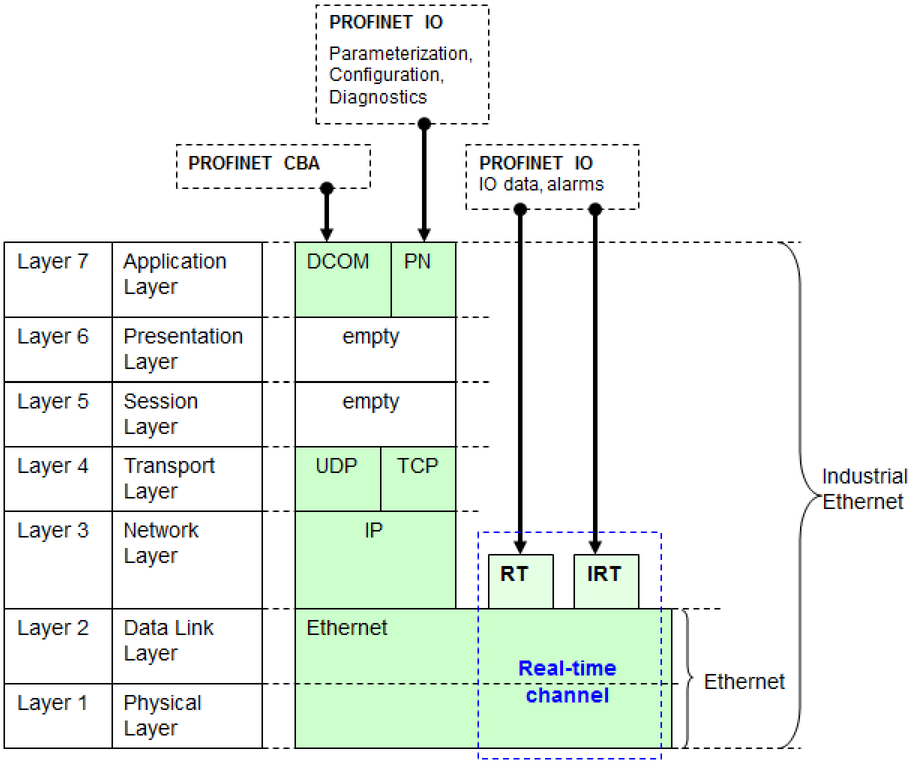
ISO/OSI reference model

PROFINET communication is based on Ethernet communication. It distinguishes between three communication channels, or respectively three performance levels:

Table 10-1

Communication channel		Application	Examples
TCP/IP		non-time-critical communication	• Acyclic reading and writing of data records • Parameterization • Configuration • Diagnostics
Real-time	Real Time (RT)	time-critical communication	• Cyclic process image of the field devices (IO data) • Alarms
	Isochronous Real Time (IRT)	high-performance, deterministic and clock-synchronized communication	• Process data in the Motion Control sector

Figure 10-5



11 PROFIBUS (PB)

Introduction

PROFIBUS is an international standardized, electrical field bus system.

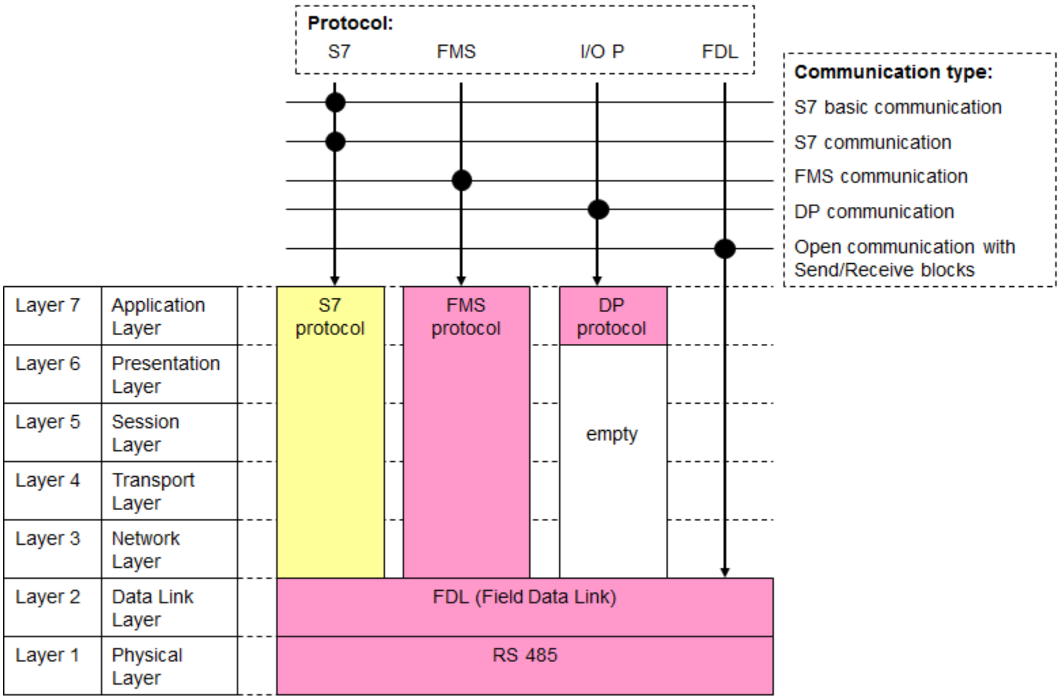
Characteristics

- International standard: IEC61158, IEC61784
- Communication mechanism between controllers:
  Token passing
- Communication mechanism between field devices and controller:
  Master/slave principle

ISO/OSI reference model

The figure below shows all protocols and communication types supported by the SIMATIC controllers.

Figure 11-1



12 MPI

Introduction

MPI is the network of SIMATIC for the communication with PG/OP and CPU-CPU communication. An MPI interface is integrated on CPUs of the modular SIMATIC controller.

The following controllers have no MPI interface: S7-1200 and S7-1500.

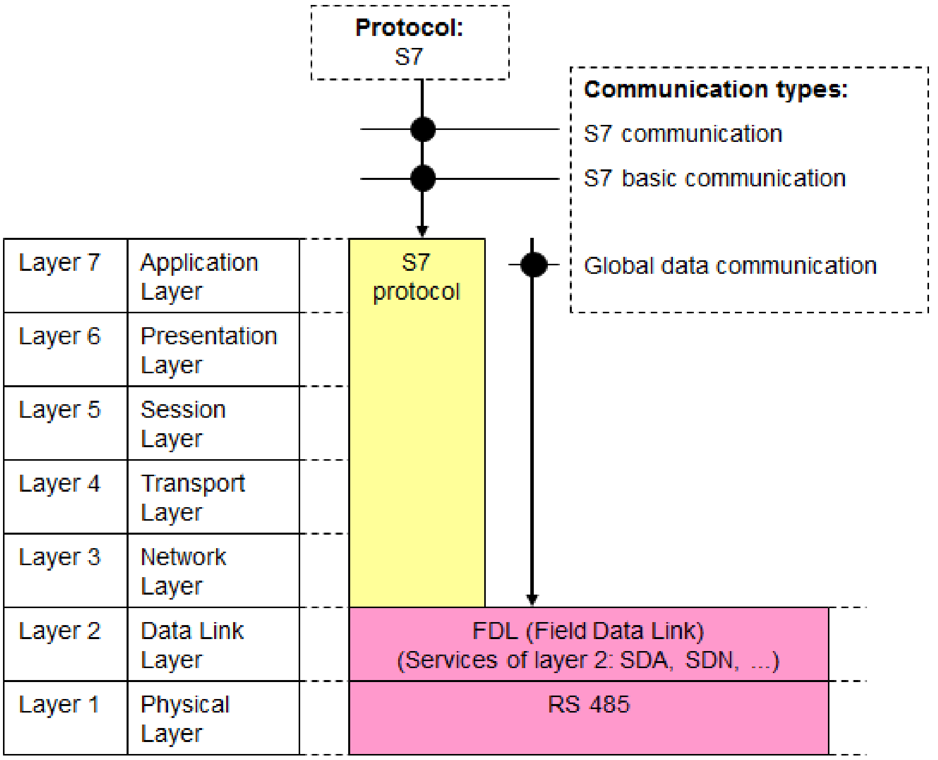
Characteristics

- Manufacturer-specific interface
- Network with low expansion and low number of stations
- MPI is based on PROFIBUS

ISO/OSI reference model

The figure below shows all protocols and communication types supported by the SIMATIC controllers.

Figure 12-1



13 SIMATIC Backplane Bus

Introduction

CPU-CPU communication within a SIMATIC station is possible via the backplane bus

Note

- Only possible for SIMATIC S7-400 (multi-computing, chapter 5.5)
- Not supported by STEP 7 (TIA).

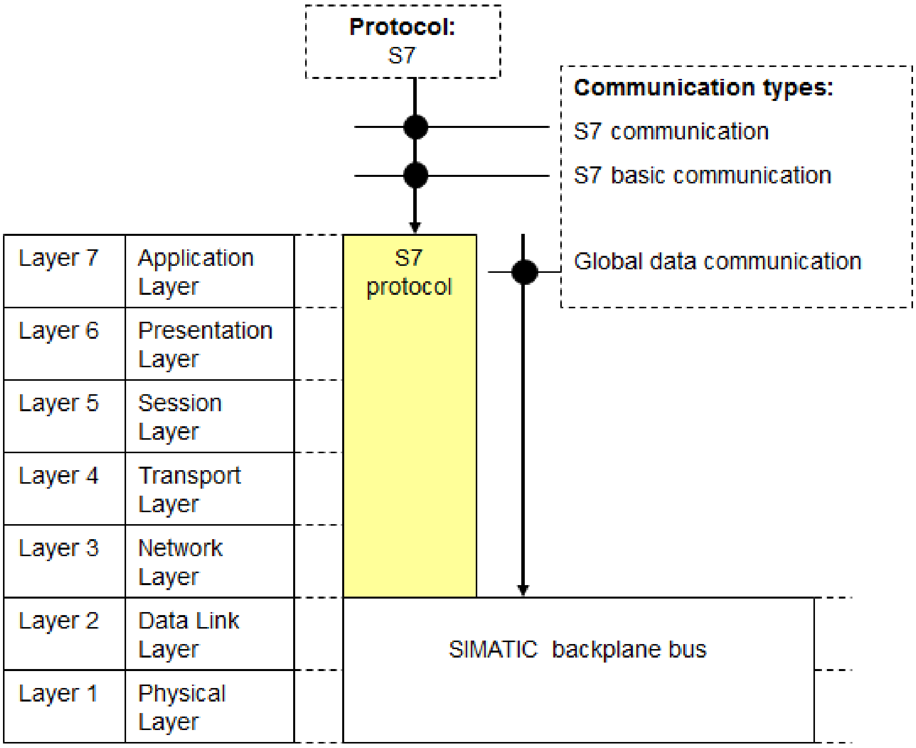
Characteristics

- Manufacturer-specific interface
- SIMATIC-specific communication types:
  Global data communication, S7 communication

ISO/OSI reference model

The figure below shows all protocols and communication types supported by the SIMATIC controllers.

Figure 13-1



14 Serial Interface (PtP)

Introduction

Communication via a “serial interface” provides a simple option of exchanging data between two communication partners.

SIMATIC controllers can communicate via the “serial interface” with different partners:

- simple devices, such as printer, barcode reader
- Drives (USS protocol, …)
- SIMATIC controller, other controller

<u>Number of nodes</u>

Generally, precisely two communication partners participate in the communication (point-to-point coupling).

For RS 422/485, however, more than two communication peers are also possible (multipoint link).

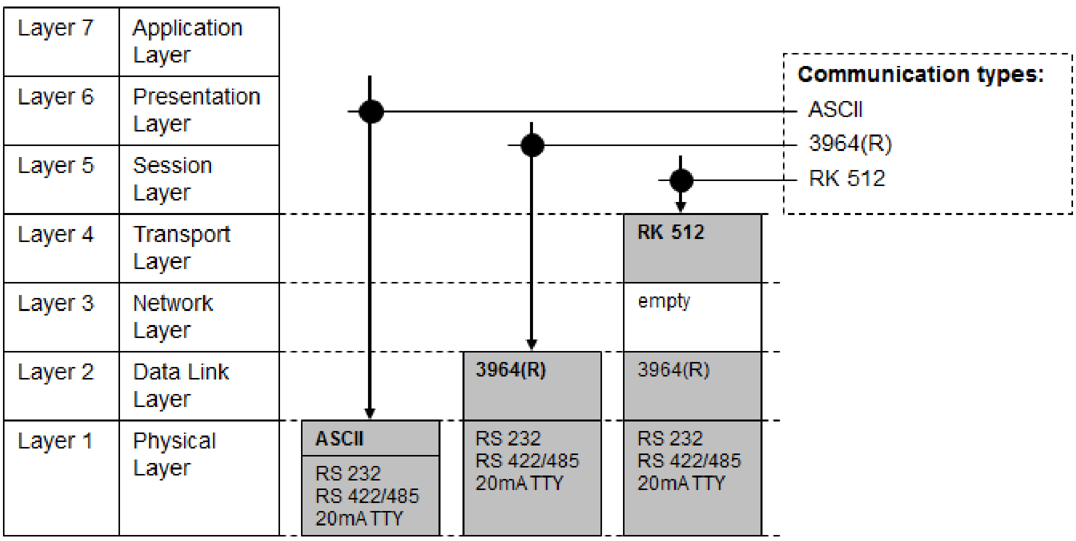
Characteristics

- Communication is mostly open
- Communication via standardized interface (physics):
  RS 232C, RS 422/485, 20mA TTY

ISO/OSI reference model

The figure below shows all protocols and communication types supported by the SIMATIC controllers.

Figure 14-1



15 Comparison of the Media

The table below shows how the media are principally distinguished from each other. The comparison is an oversimplification. In individual cases, deviations from the entered values may occur. In concrete application cases, the respective manuals must therefore be consulted.

Table 15-1

Medium		Transmission rate (from …to)	Maximal number of peers	Maximal distance between two peers	Maximal data volume per job	Redundant setup possible?
Net-work	PROFINET/ Industrial Ethernet	PN: 10/100 MBit/s IE: 1/10 GBit/s	over 1000	electrical: 100 m optical: 5 km (multimode) or 120 km (single mode)	64 Kbytes	yes
	PROFIBUS	9.6 Kbit/s to 12 Mbit/s	126	electrical: 1 km (without repeater) or resp. 10 km (with repeater) optical: 1875 km (with OLM)	64 Kbytes	yes
	MPI	187.5 Kbit/s to 12 Mbit/s	126	electrical: 50 m	64 Kbytes	no
SIMATIC backplane bus		10 Mbit/s	4 CPUs in the SIMATIC station	---	64 Kbytes	---
Serial interface		110 bit/s up to 115.2 bit/s (20mA-TTY: up to 19.2 Kbit/s	Point-to-point: 2 Multipoint: 32	RS232C: 15 m RS422/485: 1,200 m 20mA-TTY: 1,000 m	4 Kbytes	---

16 Interfaces of the SIMATIC Families

The table gives an overview of the media via which SIMATIC families can communicate (CPU-CPU communication). For PROFIBUS and PROFINET/Industrial (PN/IE) it is also specified which functionality the interface can adopt.

Table 16-1

SIMATIC controller	SIMATIC family	MPI	PROFIBUS			PN/IE				Serial interface
				Functionality			Functionality			
				DP master	DP slave		PROFINET IO controller	PROFINET IO device	PROFINET CBA	
Modular controllers	ET 200 CPU	X	X	X	X	X	X	X	X	X
	S7-300	X	X	X	X	X	X	X	X	X
	S7-400	X	X	X	X	X	X	X	X	X
	S7-1200	---	X	X	X	X	X	---	---	X
	S7-1500	---	X	X	X	X	X	X	---	X
PC-based controllers	WinAC RTX	---	X	X	---	X	X	---	X	X
	S7-mEC (with WinAC RTX)									
	Box PC (with WinAC RTX)									
	Panel PC (with WinAC RTX)									

Meaning of the entries in the table:

"x": interface / functionality existing
"---" interface / functionality not existing

Note

"x" does not mean that all interfaces of a SIMATIC family have this functionality.
"x" means that there is at least one interface which has this functionality.

17 Information on Part 1

The table contains references to information on the topics in Part 1 (introduction). All references /x/ are stored centrally in chapter 61. There you also find the respective internet links.

Table 17-1

/x/	Title	Information on
/0/	Siemens Industry Online Support: FAQs, manuals	Connections Connection resources Communication types Data consistency
/6/	SIMATIC system and standard functions for S7-300/400, reference manual	
/15/	SIMATIC / Configuring hardware and communication connections STEP 7 V5.5, manual	
/33/	SIMATIC STEP 7 V5.5, manual	
/32/	SIMATIC STEP 7 Professional V12.0, system manual	
/1/	SIMATIC controller / The innovative solution for all automation tasks, brochure.	SIMATIC controller
/4/	Catalog ST 70, Products for Totally Integrated Automation and Micro Automation	
/2/	SIMATIC NET, industrial communication, brochure	Media
/3/	SIMATIC, Communication with SIMATIC, system manual	
/5/	Catalog IK PI, industrial communication	

18 ******** PART 2: Selection Aid *********

18.1 Structure and content

Table 18-1

Chapt.	Structure	Content
19	Preliminary remarks	Explanations on the used tables
20	Jump distributor	There is one page per medium with cross-references to the central chapters of the documentation. This makes it possible to select information quickly.
21	Selection aid PN/IE	The following is described per medium: • All interfaces (CPU, CP,CM) and communication types per SIMATIC family (Interfaces table). • All possible ways of how SIMATIC families can communicate with each other (Combinations table) • Comparison of all communication types ("Communication types – compact" table)
22	Selection aid PB	
23	Selection aid MPI	
24	Selection aid SIMATIC backplane bus	
25	Selection aid Serial interface	The following is described per SIMATIC family: • All interfaces (CPU, CP) and communication types (*ASCII, 3964(R), …). • Properties of the communication types
26	Information	Notes regarding further information

<u>Explanation of terms in the above table</u>

Medium:

- Network: PN/IE, PB, MPI
- SIMATIC backplane bus
- Serial interface

SIMATIC families:

- Modular controllers:
  ET 200 CPU, S7-300, S7-400, S7-1200, S7-1500
- SIMATIC PC-based controllers:
  WinAC RTX, S7-mEC, Box PC, Panel PC

Communication types:

- SIMATIC-specific
- Open standard

19 Preliminary Remarks

The following is discussed below:

- Principle of the display of all combinations of SIMATIC families
- Explanation of the tables used in part 2:
  - Interfaces table
  - Combinations table
  - “Communication types – compact” table

19.1 Display principle of all combinations

The communication possibilities between two SIMATIC families (x, y) are displayed in the document.

Both SIMATIC families (x, y) are referred to as pairs.

One pair makes two combinations:

- x/y (x communicates with y)
- y/x (x communicates with x)

For each pair (x,y) there are two chapters in the document:

- one chapter for the combination x/y (chapter x/y)
- one chapter for the combination y/x (chapter y/x)

The description of the possible communication types for one pair is of course only given in one single chapter (chapter x/y). The other chapter (chapter y/x) only contains a reference to this chapter (chapter x/y).

This procedure affects the structure of the document (table of contents) and the representation of the combinations (jump distributor).

<u>Example</u>

Pair:

- ET 200 CPU, S7-300

Combinations:

- ET 200 CPU / S7-300
- S7-300 / ET 200 CPU

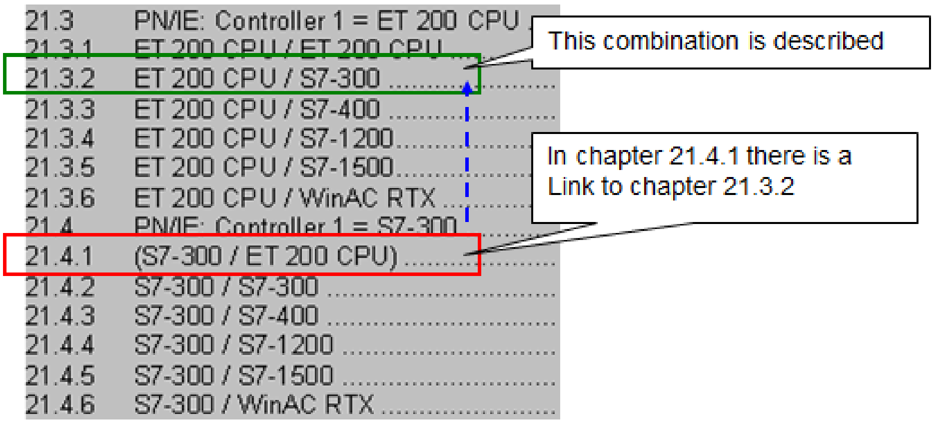
19.1.1 Structure of the document (table of contents)

If one chapter (chapter y/x) only contains a reference to another chapter (chapter x/y), then the header of this chapter (chapter y/x) is written in brackets. The following example explains this in more detail:

- Medium PN/IE.
- Pair: ET 200 CPU, S7-300

The following figure shows an extract from the table of contents.

Figure 19-1



19.1.2 Combinations (jump distributor)

In the document the combinations are represented in tables (jump distributor). For each combination, the chapter is given in which this combination is described specifically. The following example explains this in more detail:

- PN/IE medium
- Pair: ET 200 CPU, S7-300

The figure below shows the jump distributor for PN/IE.

Figure 19-2

Controller 2	Controller 1					
	ET 200 CPU	S7-300	S7-400	S7-1200	S7-1500	WinAC RTX
ET 200 CPU	21.3.1	21.3.2	21.3.3	21.3.4	21.3.5	21.3.6
S7-300	21.3.2	21.4.2	21.4.3	21.4.4	21.4.5	21.4.6
S7-400	21.3.3	21.4.3	21.5.3	21.5.4	21.5.5	21.5.6
S7-1200	21.3.4	21.4.4	21.5.4	21.6.4	21.6.5	21.6.6
S7-1500	21.3.5	21.4.5	21.5.5	21.6.5	21.7.5	21.7.6
WinAC RTX	21.3.6	21.4.6	21.5.6	21.6.6	21.7.6	21.8.6

The table contains two entries for the pair (Controller 1 / Controller 2)

- Controller 1: ET 200 CPU / Controller 2: S7-300 (green frame)
- Controller 1: S7-300 / Controller 2: ET 200 CPU (red frame)

Both entries contain identical chapter numbers.

19.2 Interfaces table

19.2.1 Purpose of this table

The Interfaces table contains a compact representation of the interfaces provided by the SIMATIC families. For each medium and each SIMATIC family there is one table. This table is the basis for the Combinations tables (chapter 19.3).

The table answers the following questions:

- Which interfaces (CPU, CP, CM) are available?
- Which communication types are possible?

19.2.2 Structure of the table

The structure of the tables for the PN/IE and PB media is described below.

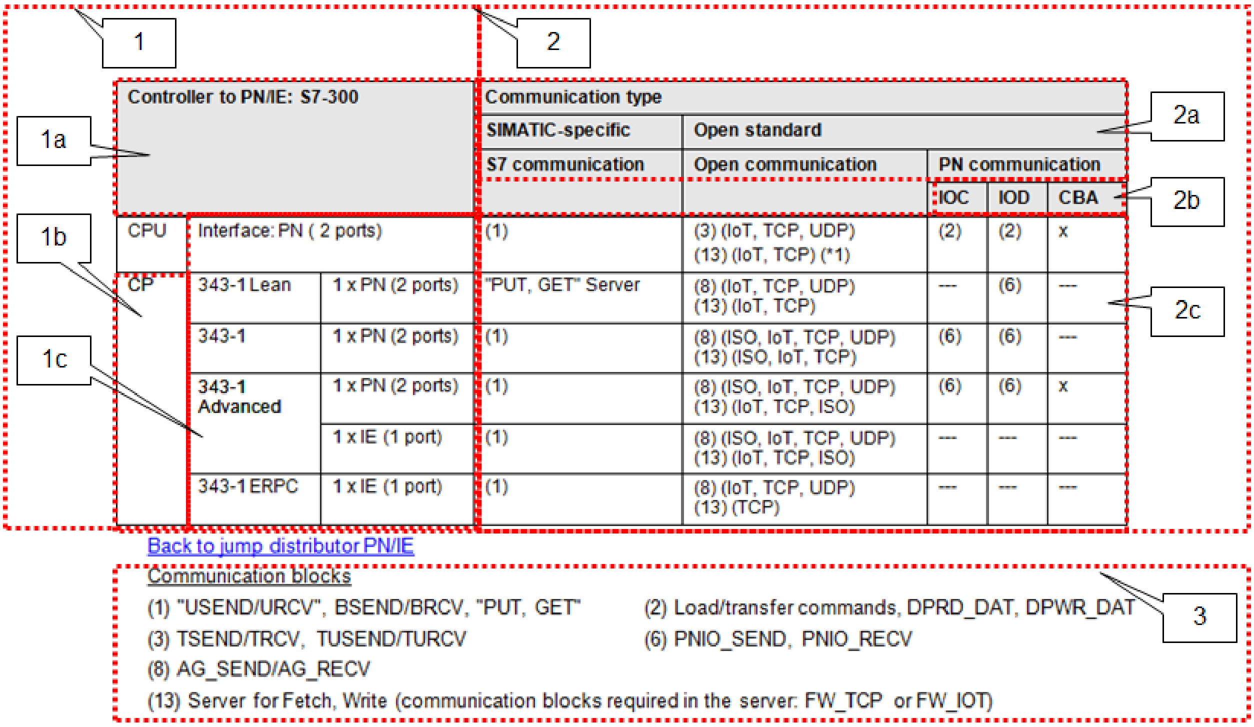
PN/IE medium

The structure is explained by means of a direct example:

- S7-300 to PN/IE

The figure shows the respective Interfaces table (Table 21-2).

Figure 19-3



Controller to PN/IE: S7-300			Communication type				
			SIMATIC-specific	Open standard			
			S7 communication	Open communication	PN communication		
					IOC	IOD	CBA
CPU	Interface: PN (2 ports)		(1)	(3) (IoT, TCP, UDP) (13) (IoT, TCP) (*1)	(2)	(2)	x
CP	343-1 Lean	1 x PN (2 ports)	"PUT, GET" Server	(8) (IoT, TCP, UDP) (13) (IoT, TCP)	---	(6)	---
	343-1	1 x PN (2 ports)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (ISO, IoT, TCP)	(6)	(6)	---
	343-1 Advanced	1 x PN (2 ports)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (IoT, TCP, ISO)	(6)	(6)	x
		1 x IE (1 port)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (IoT, TCP, ISO)	---	---	---
	343-1 ERPC	1 x IE (1 port)	(1)	(8) (IoT, TCP, UDP) (13) (TCP)	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET" (2) Load/transfer commands, DPRD_DAT, DPWR_DAT
(3) TSEND/TRCV, TUSEND/TURCV (6) PNIO_SEND, PNIO_RECV
(8) AG_SEND/AG_RECV
(13) Server for Fetch, Write (communication blocks required in the server: FW_TCP or FW_IOT)

The table consists of several areas, which are explained below.

Area 1: controller interfaces

Here, the properties of the SIMATIC family are described:

Table 19-1

	Meaning	Example
1a	Designation of the medium and the SIMATIC family	S7-300 to PN/IE
1b	Type of interface: integrated interface of a CPU, or external interface of a CP or CM	CPU, CP
1c	Designation of the modules	CPU with PN interface CP 343-1 Lean CP 343-1 Advanced CP 343-1 ERPC

Area 2: Communication types

Here, the communication options are described:

Table 19-2

	Meaning	Example
2a	Designation of the communication types, divided into: • SIMATIC-specific • open standard	S7 communication Open communication PN communication
2b	Functionality of the interface	Profinet IO controller PROFINET IO device PROFINET CBA
2c	Here you enter the existing communication options for the modules. The communication blocks possible here are either entered directly, or represented via an index (x). The index (x) is explained in area 3.	direct entry **(*1)**: "PUT, GET" Server (IoT) Index: (1), (2), (3), (6), (8), (13)
	The possible protocols are given in brackets.	IoT (***2**), ISO, TCP, UDP
	Entries in the table: --- communication is not possible x communication is possible	

(*1):

If only the server functionality is possible, then this is described with a respective suffix.
Example: "PUT, GET", Server

If client as well as server are possible, then there is no suffix.
Example: "PUT, GET"

(*2):

IoT refers to ISO-on-TCP. In the document, the term IoT was selected in order to save space.

Area 3: communication blocks for the index (x)

Here, the communication blocks possible for an index are listed.

PB medium

The structure is explained by means of a direct example:

- S7-300 at PB

The figure shows an extract from the Interfaces table (Table 22-2).

Figure 19-4

Controller to PB: S7-300		Functionality of the interface		Communication type	
2b				SIMATIC-specific	
				S7 basic communication	S7 communication
CPU	Interface: DP, MPI/DP (*2)	DP master		I_PUT, I_GET	"PUT, GET", Server
		DP slave (*1)	active	I_PUT, I_GET, Server	"PUT, GET", Server
			passive	I_PUT, I_GET, Server	---
CP	342-5 (*3)	no DP operation		---	(1)
		DP master		---	(1)
		DP slave (*1)	active	---	"PUT, GET", Server
			passive	---	---
	343-5	no DP operation		---	"PUT, GET", Server

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"
(2) Load/transfer commands, DPRD_DAT, DP
(7) DP_SEND, DP_RECV
(8) AG_SEND/AG_RECV
(10) READ, WRITE, REPORT

The structure of the table is the same as for the PN/IE medium (see above), with the exception of area 2b:

Area 2b: functionality of the interface

Here, the functionalities of the interface are specified.

- DP master or DP slave
- DP slave: active or passive

There are various communication types depending on the functionality.

19.2.3 Abbreviations and indices

An overview of abbreviations is given in chapter 63. An overview of all indices (x) of the Interfaces table is given in chapter 19.5.

19.3 Combinations table

19.3.1 Purpose of this table

The Combinations table contains a compact representation of how SIMATIC controllers can communicate with each other. For each medium and each combination of two SIMATIC families, there is one table in the document.

The table answers the following questions:

- Which interfaces (CPU, CP, CM) are available?
- Which communication types are possible?

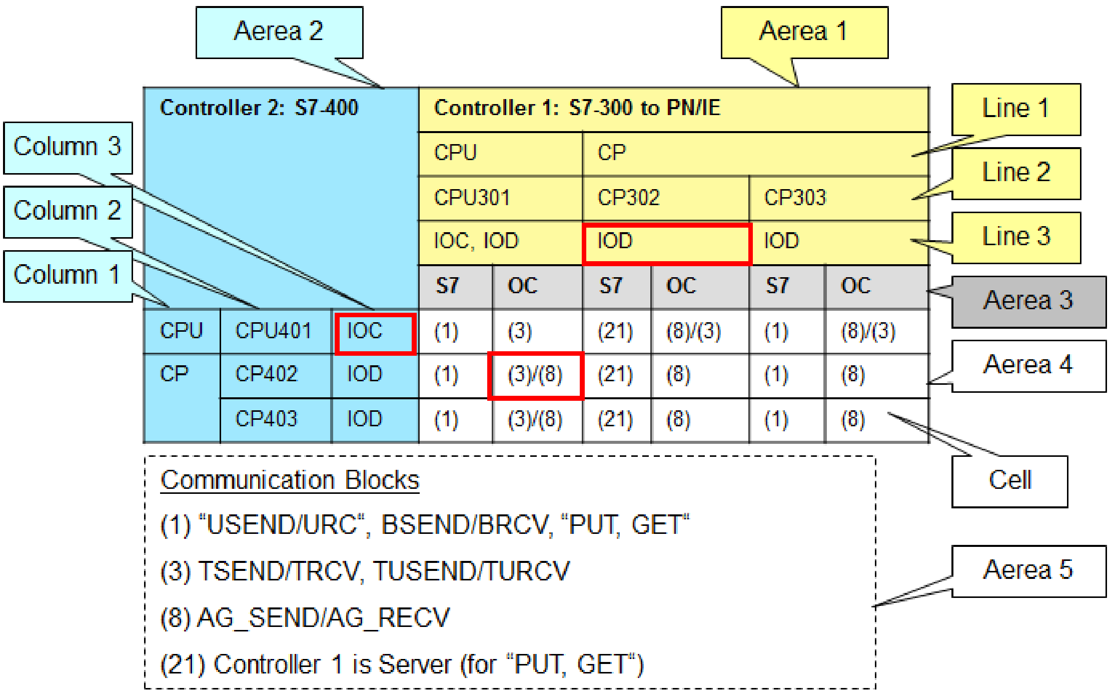
19.3.2 Structure of the table

The structure of the table is described by means of a simplified (not real) example. The following boundary conditions for the example:

- Controller 1 is a SIMATIC family with a CPU interface (CPU301) and two CP interfaces (CP302, CP303)
- Controller 2 is a SIMATIC family with a CPU interface (CPU401) and two CP interfaces (CP402, CP403)
- It is assumed that two communication types are possible: S7 communication (S7) and open communication (OC)

Under the above boundary conditions the Combinations table looks like this:

Figure 19-5



The table is structured as matrix. It shows all combinations for all interfaces of both controllers. A description of the individual areas is given below.

Area 1

This area shows all communication-capable modules of controller 1.

A column represents an interface:

- Row 1: CPU, CP or CM
- Row 2: designation of the module
- Row 3: functionality of the interface on the module

Example (red frame)

CP302 has an interface with PN IO Device functionality (IOD)

Area 2

This area shows all communication-capable modules of controller 2.

A row represents an interface:

- Column 1: CPU, CP or CM
- Column 2: designation of the module
- Column 3: functionality of the interface on the module

Example (red frame)

CP401 has an interface with PN IO Controller functionality (IOC)

Area 3

Area 3 lists the communication types possible for both controllers. They are repeated for each interface (column) of controller 1 (area 1):

In this example:

- S7 (S7 communication)
- OC (open communication)

Any jointly possible communication type between controller 1 and controller 2 is considered in the table (see explanations on area 4).

Area 4

Each cell in the area represents a combination of two interfaces.

The area provides the following information per combination:

- jointly possible communication types (displayed in area 3)
- respective communication blocks (represented via an index (x))

The notation of the indices (x) in the cells indicates the following information:

Table 19-3

Notation in the cell	Communication type according to area 3 possible?	Which communication blocks can be used?
(1)	yes	See area 5: index (1)
(1)+(2)	yes	See area 5: (1) or (2)
(1) / (2)	yes	Controller 1 with (1) / Controller 2 with (2)
(1) / (2)+(3)	yes	Two different cases are possible here: • Controller 1 with (1) / Controller 2 with (2) • Controller 1 with (1) / Controller 2 with (3)
x	yes	No communication blocks are required.
---	no	Not applicable, since the communication type is not possible.

Area 5

In area 5 the indices (x) from area 4 are explained. It is specified for each index which communication blocks are possible and which characteristics must be observed.

Summary

Each cell in area 4 provides the following information:

- Module x (area 1) can communicate with module y (area 2) via communication type z (area 3).
- The possible communication blocks are described in area 5.

Example

The red framed cells in area 4 of Figure 19-5 mean:

Combination:

- CPU301 (functionality of the interface: IOC, IOD)
- CP402 (functionality of the interface: IOD)

Communication type:

- Open communication (OC)

Communication blocks:

- For controller 1: TSEND/TRCV, TUSEND/TURCV
- For controller 2: AG_SEND / AG_RECV

Further examples on area 4:

(1) means:

Communication blocks:

- Both controllers: “USEND/URC”, BSEND/BRCV, “PUT, GET”
- Both controllers: client or server for "PUT, GET"

(21) means:

Communication blocks:

- For controller 1: can only be server for "PUT, GET"

19.3.3 Real example

Using the table is explained here by means of a real example:

- Communication via PN/IE
- Controller 1: ET 200 CPU
- Controller 2: S7-300

The figure shows the respective Combinations table.

Figure 19-6

Controller 2: S7-300				Controller 1: ET 200 CPU to PN/IE			
				CPU			
				IM 151-8(F) PN/DP, IM 154-8(F) PN/DP CPU			
				IOC, IOD			
				S7	OC	PN	
						PNIO	CBA
CPU	Interface: PN		IOC, IOD	(1)	(3)	(2)	x
CP	343-1 Lean		IOD	(22)	(3) / (8)	(2) / (6)	---
	343-1		IOC, IOD	(1)	(3) / (8)	(2) / (6)	---
	343-1 Advanced	Interface: PN	IOC, IOD	(1)	(3) / (8)	(2) / (6)	x
		Interface: IE	---	(1)	(3) / (8)	---	---
	343-1 ERPC		---	(1)	(3) / (8)	---	---

a b c d

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(22) Controller 2 is server (for "PUT, GET")

The table below describes the examples for the picture above. Explanations of the syntax are given in chapter 19.3.2 (section "Area 4").

Table 19-4

In picture	Controller 1:		Controller 2:		Communication type	Communication blocks
	Interface	Functionality	Interface	Functionality		
a	IM151-8, IM154-8	IOC, IOD	343- 1	IOC, IOD	S7	Both controllers: (1)
b	IM151-8, IM154-8	IOC, IOD	343-1 ERPC	---	OC	Controller 1: (3) Controller 2: (8)
c	IM151-8, IM154-8	IOC, IOD	343-1 Lean	IOD	PNIO	Controller 1: (2) Controller 2: (6)
d	IM151-8, IM154-8	IOC, IOD	S7-300 CPU PN	IOC, IOD	CBA	none

19.3.4 Abbreviations and indices

An overview of abbreviations is given in chapter 63. An overview of all indices (x) of the Combinations table is given in chapter 19.5.

19.4 “Communication types – compact” table

19.4.1 Purpose of this table

The “Communication types – compact” table answers the following questions for each medium (PN/IE, PB, MPI):

- Which communication types exist per medium?
- What are the main differences?

The “Communication types – compact” table is a summary (compact version) of the “Communication types – detailed” tables.

Not each SIMATIC family or module fulfills all values given in the “Communication types – compact” table. The values must be considered as supersets.

19.4.2 Structure of the table

Example: extract from the table for PN/IE (Table 21-34).

Figure 19-7

1

2

	SIMATIC-specific	**Open standard**	
	S7 communication	**Open communication**	
		Send/Receive blocks	**T blocks T-Comp**
Protocols	ISO (only CP), IoT	ISO, IoT, TCP, UDP	IoT, TCP
Interfaces	CPU, CP, CM	CP	CPU, CP
Communication blocks (max. data)	BSEND (≤ 64 Kbytes) Type "USEND/URCV" (≥ 160 bytes) Type "PUT, GET" (≥ 160 bytes)	AG_xSEND (ISO, IoT, TCP ≤ 8 Kbytes) (UDP ≤ 2 Kbytes) Server for FETCH, WRITE (not UDP)	TSEND, (IoT ≤ (TCP ≤ (UDP ≤
Remote confirmation	BSEND: application Type "USEND / URCV": transport Type "PUT, GET": application	ISO, IoT, TCP: transport UDP: none	IoT, TCP UDP: no
Connections?	yes	ISO, IoT, TCP: yes UDP: no	IoT, TCP UDP: no

The table consists of two areas:

Area 1

This area contains the designation of the communication types, divided into the classes “SIMATIC-specific” and “Open standard”.

Area 2

It contains the most important properties of the communication types.

The criteria (protocols, interfaces, …) are described in chapter 28.2.3.

Note: The criteria of the “Communication types – compact” table are a subset of the “Communication types – detailed” tables.

19.5 Indices overview in the tables

Interfaces table and Combinations table

The following indices are used in the Interfaces tables and in the Combinations tables:

Table 19-5

(x)	User interfaces (communication blocks)	
(1)	"USEND/URCV", BSEND/BRCV, "PUT, GET"	**(*1)**
(2)	Load/transfer commands, DPRD_DAT, DPWR_DAT	
(3)	TSEND/TRCV, TUSEND/TURCV	**(*2)**
(4)	TSEND/TRCV	**(*2)**
(5)	TSEND_C/TRCV_C	
(6)	PNIO_SEND, PNIO_RECV	
(7)	DP_SEND, DP_RECV	
(8)	AG_SEND / AG_RECV	
(9)	AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV	
(10)	READ, WRITE, REPORT	
(11)	X_SEND/X_RCV, X_PUT, X_GET	
(12)	GD_SND/GD_RCV	
(13)	Server for FETCH, WRITE	

Combinations table

The following indices are only used in the Combinations tables:

Table 19-6

(x)	User interfaces (communication blocks)	
Communication type: S7 communication		
(21)	Controller 1 is the server (for "PUT, GET")	**(*1)**
(22)	Controller 2 is the server (for "PUT, GET")	**(*1)**
(40)	"PUT, GET" (both controllers can be client or server) **(*1)**	
Communication type: S7 basic communication		
(61)	Controller 1 is the server (for I_PUT, I_GET)	
(62)	Controller 2 is the server (for I_PUT, I_GET)	

Explanations

(*1):

The term "USEND/URCV" refers to all variants of the communication blocks (USEND, USEND_E, USEND_S, ...). See chapter 32.4 .

The term " PUT, GET " refers to all variants of the communication blocks (PUT, PUT_E, PUT_S, ...). See chapter 32.6.

An overview of all variants is available in chapter 32.3.

(*2):

With (3), TSEND/TRCV and TUSEND/TURCV are possible.

With (4), TUSEND/TURCV is not possible.

.

20 >>> JUMP DISTRIBUTOR >>>

This chapter contains cross-references to all the important information in the document.

Notes on handling the jump distributor: See chapter 3.

20.1 Jump distributor: PN/IE network

20.1.1 Overview of interfaces and communication types

Table 20-1 Links to the Interfaces tables

Family	Chapter
ET 200 CPU	21.2.1
S7-300	21.2.2
S7-400	21.2.3
S7-1200	21.2.4
S7-1500	21.2.5
S7-mEC	21.2.6
Box PC	21.2.7
Panel PC	21.2.8
WinAC RTX	21.2.9

20.1.2 Combination controller 1 / controller 2

Table 20-2 Links to the Communications tables

Controller 2	Controller 1					
	ET 200 CPU	S7-300	S7-400	S7-1200	S7-1500	WinAC RTX
ET 200 CPU	21.3.1	21.3.2	21.3.3	21.3.4	21.3.5	21.3.6
S7-300	21.3.2	21.4.2	21.4.3	21.4.4	21.4.5	21.4.6
S7-400	21.3.3	21.4.3	21.5.3	21.5.4	21.5.5	21.5.6
S7-1200	21.3.4	21.4.4	21.5.4	21.6.4	21.6.5	21.6.6
S7-1500	21.3.5	21.4.5	21.5.5	21.6.5	21.7.5	21.7.6
WinAC RTX	21.3.6	21.4.6	21.5.6	21.6.6	21.7.6	21.8.6

Note: The SIMATIC controllers S7-mEC, Box PC and Panel PC behave like WinAC RTX with regards to CPU-CPU communication.

20.1.3 Communication types

Table 20-3 Links to the “Communication types” tables

Communication type		Chapter
All communication types (compact table)		21.9
Table with details	S7 communication	32.2
	Open communication with send/receive blocks	34.2
	Open communication with T blocks	35.2
	PNIO	37.2
Modbus/TCP (SIMATIC / third-party controller)		57

20.2 Jump distributor: Network PB

20.2.1 Overview of interfaces and communication types

Table 20-4 Links to the Interfaces tables

Family	Chapt.
ET 200 CPU	22.2.1
S7-300	22.2.2
S7-400	22.2.3
S7-1200	22.2.4
S7-1500	22.2.5
S7-mEC	22.2.6
Box PC	22.2.7
Panel PC	22.2.8
WinAC RTX	22.2.9

20.2.2 Combination controller 1 / controller 2

Table 20-5 Links to the Communications tables

Controller 2	Controller 1					
	ET 200 CPU	S7-300	S7-400	S7-1200	S7-1500	WinAC RTX
ET 200 CPU	22.3.1	22.3.2	22.3.3	22.3.4	22.3.5	22.3.6
S7-300	22.3.2	22.4.2	22.4.3	22.4.4	22.4.5	22.4.6
S7-400	22.3.3	22.4.3	22.5.3	22.5.4	22.5.5	22.5.6
S7-1200	22.3.4	22.4.4	22.5.4	22.6.4	22.6.5	22.6.6
S7-1500	22.3.5	22.4.5	22.5.5	22.6.5	22.7.5	22.7.6
WinAC RTX	22.3.6	22.4.6	22.5.6	22.6.6	22.7.6	22.8.6

Note: The SIMATIC controllers S7-mEC, Box PC and Panel PC behave like WinAC RTX with regards to CPU-CPU communication.

20.2.3 Communication types

Table 20-6 Links to the “Communication types” tables

Communication type		Chapter
All communication types (compact table)		22.9
Table with details	S7 basic communication	31.2
	S7 communication	32.2
	Open communication with send/receive blocks	39.2
	FMS communication	40.2
	DP communication	41.2

20.3 Jump distributor: Network MPI

20.3.1 Overview of interfaces and communication types

Table 20-7 Links to the Interfaces tables

Family	Chapt.
ET 200 CPU	23.2.1
S7-300	23.2.2
S7-400	23.2.3

20.3.2 Combination controller 1 / controller 2

Table 20-8 Links to the Combinations tables

Controller 2	Controller 1		
	ET 200 CPU	S7-300	S7-400
ET 200 CPU	23.3.1	23.3.2	23.3.3
S7-300	23.3.2	23.4.2	23.4.3
S7-400	23.3.3	23.4.3	23.5.3

20.3.3 Communication types

Table 20-9 Links to the “Communication types” tables

Communication type		Chapt.
All communication types (compact table)		23.6
Table with details	Global data	30.2
	S7 basic communication	31.2
	S7 communication	32.2

20.4 Jump distributor: SIMATIC backplane bus

20.4.1 Overview of interfaces and communication types

Table 20-10 Links to the Interfaces tables

Family	Chapter
S7-400	24.2

20.4.2 Combination controller 1 / controller 2

Table 20-11 Links to the Combinations tables

Controller 2	Controller 1
	S7-400
S7-400	24.3

20.4.3 Communication types

Table 20-12 Links to the “Communication types” tables

Communication type		Chapt.
All communication types (compact table)		24.4
Table with details	Global data	30.2
	S7 basic communication	31.2
	S7 communication	32.2

20.5 Jump distributor: Serial interfaces

20.5.1 Overview of interfaces and communication types

Table 20-13 Links to the Interfaces tables

Family	Chapt.
ET 200 CPU	25.2
S7-300	25.3
S7-400	25.4
S7-1200	25.5
S7-1500	25.6
S7-mEC	25.7
Box PC	25.8
Panel PC	25.9
WinAC RTX	25.10
Distributed station ET 200	25.11

20.5.2 Communication types

Table 20-14 Links to the “Communication types” tables

Communication type	Chapter
ASCII, 3964(R), RK 512	42.2
Overview of user interfaces	43
Modbus serial (RTU format)	58

21 Selection Aid: PROFINET/Industrial Ethernet (PN/IE)

21.1 PN/IE: Content of the chapter

The following is described for the PN/IE medium:

- Which interfaces (modules) and communication types are available?
  (-> Interfaces table)
- Which partners can communicate via which communication types?
  (-> Combinations table)
- Overview of all available Communication types
  (-> “Communication types – compact” table)

21.2 PN/IE: Interfaces and communication types

21.2.1 ET 200 CPU to PN/IE

Table 21-1

Controller to PN/IE: ET 200 CPU			Communication types				
			SIMATIC-specific	Open standard			
			S7 communication	Open communication	PN communication		
					IOC	IOD	CBA
ET 200S	CPU	IM 151-8(F) PN/DP CPU	(1)	(3) (IoT, TCP, UDP)	(2)	(2)	x
ET 200Pro	CPU	IM 154-8(F) PN/DP CPU	(1)	(3) (IoT, TCP, UDP)	(2)	(2)	x

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

21.2.2 S7-300 to PN/IE

Table 21-2

Controller to PN/IE: S7-300			**Communication type**				
			SIMATIC-specific	**Open standard**			
			S7 communication	**Open communication**	**PN communication**		
					IOC	**IOD**	**CBA**
CPU	Interface: PN (2 ports)		(1)	(3) (IoT, TCP, UDP) (13) (IoT, TCP) (*1)	(2)	(2)	x
CP	343-1 Lean	1 x PN (2 ports)	"PUT, GET" Server	(8) (IoT, TCP, UDP) (13) (IoT, TCP)	---	(6)	---
	343-1	1 x PN (2 ports)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (ISO, IoT, TCP)	(6)	(6)	---
	343-1 Advanced	1 x PN (2 ports)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (IoT, TCP, ISO)	(6)	(6)	x
		1 x IE (1 port)	(1)	(8) (ISO, IoT, TCP, UDP) (13) (IoT, TCP, ISO)	---	---	---
	343-1 ERPC	1 x IE (1 port)	(1)	(8) (IoT, TCP, UDP) (13) (TCP)	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(13) Server for Fetch, Write (communication blocks required in the server: FW_TCP or FW_IOT)

21.2.3 S7-400 to PN/IE

Table 21-3

Controller to PN/IE: S7-400			**Communication type**				
			SIMATIC-specific	**Open standard**			
			S7 communication	**Open communication**	**PN communication**		
					IOC	**IOD**	**CBA**
CPU	Interface: PN (2 ports)		(1)	(3) (IoT, TCP, UDP) (13) (IoT, TCP) (*1)	(2)	(2)	x
CP	443-1	1 x PN (2 ports)	(1)	(4) (IoT) (9) (ISO, IoT, TCP, UDP) (13) (ISO, IoT, TCP)	(2)	(2)	---
	443-1 Advanced	1 x PN (4 ports)	(1)	(4) (IoT) (9) (ISO, IoT, TCP, UDP) (13) (ISO, IoT, TCP)	(2)	(2)	x
		1 x IE (1 port)	(1)	(4) (IoT) (9) (ISO, IoT, TCP, UDP) (13) (ISO, IoT, TCP)	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

(13) Server for Fetch, Write (communication blocks required in the server: FW_TCP or FW_IOT)

21.2.4 S7-1200 to PN/IE

Table 21-4

Controller to PN/IE: S7-1200		**Communication type**				
		SIMATIC-specific	**Open standard**			
		S7 communication	**Open communication**	**PN communication**		
				IOC	**IOD**	**CBA**
CPU	Interface: PN	"PUT, GET"	(3) (IoT, TCP, UDP) (5) (IoT, TCP)	(2)	---	---

Back to jump distributor PN/IE

Communication blocks

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

21.2.5 S7-1500 to PN/IE

Table 21-5

Controller to PN/IE: S7-1500			Communication type				
			SIMATIC-specific	Open standard			
			S7 communication	Open communication	PN communication		
					IOC	IOD	CBA
CPU	1511-1 PN 1513-1 PN	1 x PN (2 ports)	(1)	(3) (IoT, TCP, UDP) (5) (IoT, TCP, UDP)	(2)	(2)	---
	1516-3 PN/DP	1 x PN (2 ports)	(1)	(3) (IoT, TCP, UDP) (5) (IoT, TCP, UDP)	(2)	(2)	---
		1 x IE (1 port)	(1)	(3) (IoT, TCP, UDP) (5) (IoT, TCP, UDP)	---	---	---
CP	1543-1	1 x IE (1 port)	(1)	(3) (IoT, TCP, UDP, ISO) (5) (IoT, TCP, UDP, ISO) (13) (ISO, IoT, TCP)	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(13) Server for Fetch, Write (communication blocks are not required in the server)

21.2.6 S7-mEC to PN/IE

Table 21-6

Controller to PN/IE: S7-mEC with WinAC RTX (F) 2010			**Communication type**				
			SIMATIC-specific	**Open standard**			
			S7 communication	**Open communication**	**PN communication**		
					IOC	**IOD**	**CBA**
CPU	EC31-RTX (F)	X1: control by RTX (submodule)	(1)	(3) (IoT, TCP, UDP)	(2)	---	x
		X2: controlled by Windows	(1)	---	---	---	---
CP	EM PC	X1: controlled by Windows	(1)	---	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

21.2.7 Box PC to PN/IE

Table 21-7

Controller to PN/IE: Box PC SIMATIC embedded bundles: IPC227D, IPC427C with WinAC RTX (F) 2010		**Communication type**				
		SIMATIC-specific	**Open standard**			
		S7 communication	**Open communication**	**PN communication**		
				IOC	**IOD**	**CBA**
CPU / CP (*1)	under the control of RTX (submodule): • “CP1616/CP1604”	(1)	(3) (IoT,TCP, UDP)	(2)	---	x
	under the control of Windows: • “IE General”	(1)	---	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

Explanations for the table:

(*1): Meaning of the interfaces CPU or resp. CP: CPU = Onboard interface of the PC, CP = plug-in PC card

The designations of the interfaces in the table are collective names. This is explained in the following table.

Table 21-8

Collective name	**PC card (corresponds to the CP interface)**	**Onboard interface of the PC (corresponds to the CPU interface)**
“CP1616/CP1604”	CP 1616, CP 1604	Onboard PN interface of the SIMATIC IPC and S7-mEC: CP 1616 integrated
“IE General”	Standard Ethernet card (Intel Chipset, non shared IRQ)	Onboard Ethernet interface of the SIMATIC IPC

21.2.8 Panel PC to PN/IE

Table 21-9

Controller to PN/IE: Box PC SIMATIC embedded bundles: IPC277D, IPC477C with WinAC RTX (F) 2010		**Communication type**				
		SIMATIC-specific	**Open standard**			
		S7 communication	**Open communication**	**PN communication**		
				IOC	**IOD**	**CBA**
CPU / CP (*1)	under the control of RTX (submodule): • “CP1616/CP1604”	(1)	(3) (IoT, TCP, UDP)	(2)	---	x
	under the control of Windows: • “IE General”	(1)	---	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

Explanations for the table

(*1): Meaning of the interfaces CPU or resp. CP: CPU = Onboard interface of the PC, CP = plug-in PC card

The designations of the interfaces in the table are collective names. This is explained in the following table.

Table 21-10

Collective name	**PC card (corresponds to the CP interface)**	**Onboard interface of the PC (corresponds to the CPU interface)**
“CP1616/CP1604”	CP 1616, CP 1604	Onboard PN interface of the SIMATIC IPC and S7-mEC: CP 1616 integrated
“IE General”	Standard Ethernet card (Intel Chipset, non shared IRQ)	Onboard Ethernet interface of the SIMATIC IPC

21.2.9 WinAC RTX to PN/IE

Table 21-11

Controller to PN/IE: WinAC RTX WinAC RTX (F) 2010		**Communication type**				
		SIMATIC-specific	**Open standard**			
		S7 communication	**Open communication**	**PN communication**		
				IOC	**IOD**	**CBA**
CPU / CP (*1)	under the control of RTX (submodule): • “CP1616/CP1604” • “IE General”	(1)	(3) (IoT, TCP, UDP)	(2)	---	x
	under the control of Windows: • “CP1616/CP1604” • “IE General”	(1)	---	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

Explanations for the table

(*1): Meaning of the interfaces CPU or resp. CP: CPU = onboard interface of the PC, CP = plug-in PC card

The designations of the interfaces in the table are collective names. This is explained in the following table.

Table 21-12

Collective name	**PC card (corresponds to the CP interface)**	**Onboard interface of the PC (corresponds to the CPU interface)**
“CP1616/CP1604”	CP 1616, CP 1604	Onboard PN interface of the SIMATIC IPC and S7-mEC: CP 1616 integrated
“IE General”	Standard Ethernet card (Intel Chipset, non shared IRQ)	Onboard Ethernet interface of the SIMATIC IPC

21.3 PN/IE: Controller 1 = ET 200 CPU

21.3.1 ET 200 CPU / ET 200 CPU

Table 21-13

<table>
<tr><td colspan="3" rowspan="6">Controller 2: ET 200 CPU</td><td colspan="4">Controller 1: ET 200 CPU to PN/IE</td></tr>
<tr><td colspan="4">CPU</td></tr>
<tr><td colspan="4">IM 151-8(F) PN/DP CPU, IM 154-8(F) PN/DP CPU</td></tr>
<tr><td colspan="4">IOC, IOD</td></tr>
<tr><td rowspan="2">S7</td><td rowspan="2">OC</td><td colspan="2">PN</td></tr>
<tr><td>PNIO</td><td>CBA</td></tr>
<tr><td>CPU</td><td>IM 151-8(F) PN/DP CPU
IM 154-8(F) PN/DP CPU</td><td>IOC, IOD</td><td>(1)</td><td>(3)</td><td>(2)</td><td>x</td></tr>
</table>

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

21.3.2 ET 200 CPU / S7-300

Table 21-14

<table>
<tr><td colspan="4" rowspan="6">Controller 2: S7-300</td><td colspan="4">Controller 1: ET 200 CPU to PN/IE</td></tr>
<tr><td colspan="4">CPU</td></tr>
<tr><td colspan="4">IM 151-8(F) PN/DP, IM 154-8(F) PN/DP CPU</td></tr>
<tr><td colspan="4">IOC, IOD</td></tr>
<tr><td rowspan="2">S7</td><td rowspan="2">OC</td><td colspan="2">PN</td></tr>
<tr><td>PNIO</td><td>CBA</td></tr>
<tr><td>CPU</td><td colspan="2">Interface: PN</td><td>IOC, IOD</td><td>(1)</td><td>(3)</td><td>(2)</td><td>x</td></tr>
<tr><td rowspan="5">CP</td><td colspan="2">343-1 Lean</td><td>IOD</td><td>(22)</td><td>(3) / (8)</td><td>(2) / (6)</td><td>---</td></tr>
<tr><td colspan="2">343-1</td><td>IOC, IOD</td><td>(1)</td><td>(3) / (8)</td><td>(2) / (6)</td><td>---</td></tr>
<tr><td rowspan="2">343-1 Advanced</td><td>Interface: PN</td><td>IOC, IOD</td><td>(1)</td><td>(3) / (8)</td><td>(2) / (6)</td><td>x</td></tr>
<tr><td>Interface: IE</td><td>---</td><td>(1)</td><td>(3) / (8)</td><td>---</td><td>---</td></tr>
<tr><td colspan="2">343-1 ERPC</td><td>---</td><td>(1)</td><td>(3) / (8)</td><td>---</td><td>---</td></tr>
</table>

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(22) Controller 2 is server (for "PUT, GET")

21.3.3 ET 200 CPU / S7-400

Table 21-15

Controller 2: S7-400				**Controller 1: ET 200 CPU to PN/IE**			
				CPU			
				IM 151-8(F) PN/DP CPU, IM 154-8(F) PN/DP CPU			
				IOC, IOD			
				S7	**OC**	**PN**	
						PNIO	CBA
CPU	Interface: PN		IOC, IOD	(1)	(3)	(2)	x
CP	443-1		IOC, IOD	(1)	(3) / (4)+(9)	(2)	---
	443-1 Advanced	Interface: PN	IOC, IOD	(1)	(3) / (4)+(9)	(2)	x
		Interface: IE	---	(1)	(3) / (4)+(9)	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

21.3.4 ET 200 CPU / S7-1200

Table 21-16

Controller 2: S7-1200			**Controller 1: ET 200 CPU to PN/IE**			
			CPU			
			IM 151-8(F) PN/DP CPU, IM 154-8(F) PN/DP CPU			
			IOC, IOD			
			S7	**OC**	**PN**	
					PNIO	CBA
CPU	Interface: PN	IOC	(40)	(3) / (3)+(5)	(2)	---

Back to jump distributor PN/IE

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(40) "PUT, GET" (both controllers can be client or server)

21.3.5 ET 200 CPU / S7-1500

Table 21-17

Controller 2: S7-1500			**Controller 1: ET 200 CPU to PN/IE**			
			CPU			
			IM 151-8(F) PN/DP, IM 154-8(F) PN/DP CPU			
			IOC, IOD			
			S7	**OC**	**PN**	
					PNIO	CBA
CPU	Interface: PN	IOC, IOD	(1)	(3) / (3)+(5)	(2)	---
	Interface: IE	---	(1)	(3) / (3)+(5)	---	---
CP	CP 1543-1 (Interface IE)	---	(1)	(3) / (3)+(5)	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

21.3.6 ET 200 CPU / WinAC RTX

Table 21-18

Controller 2: WinAC RTX to PC WinAC RTX (F) 2010			**Controller 1: ET 200 CPU to PN/IE**			
			CPU			
			IM 151-8(F) PN/DP CPU, IM 154-8(F) PN/DP CPU			
			IOC, IOD			
			S7	**OC**	**PN**	
					PNIO	CBA
CPU/CP	Controlled by RTX (submodule): • "CP1616/CP1604" • "IE General"	IOC	(1)	(3)	(2)	x
	Controlled by Windows: • "CP1616/CP1604" • "IE General"	---	(1)	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

21.4 PN/IE: Controller 1 = S7-300

21.4.1 (S7-300 / ET 200 CPU)

See ET 200 CPU / S7-300: 21.3.2

21.4.2 S7-300 / S7-300

Table 21-19

Controller 2: S7-300				**Controller 1: S7-300 to PN/IE**																			
				CPU				CP															
				Interface: PN				343-1 Lean				343-1				343-1 Advanced						343-1 ERPC	
																Interface: PN				Interface: IE			
				IOC, IOD				IOD				IOC, IOD				IOC, IOD				---		---	
				S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**S7**	**OC**
						PNIO	CBA			PNIO	CBA			PNIO	CBA			PNIO	CBA				
CPU	Interface: PN		IOC, IOD	(1)	(3)	(2)	x	(21)	(8)/(3)	(6)/(2)	---	(1)	(8)/(3)	(6)/(2)	---	(1)	(8)/(3)	(6)/(2)	x	(1)	(8)/(3)	(1)	(8)/(3)
CP	343-1 Lean		IOD	(22)	(3)/(8)	(2)/(6)	---	---	(8)	---	---	(22)	(8)	(6)	---	(22)	(8)	(6)	---	(22)	(8)	(22)	(8)
	343-1		IOC, IOD	(1)	(3)/(8)	(2)/(6)	---	(21)	(8)	(6)	---	(1)	(8)	(6)	---	(1)	(8)	(6)	---	(1)	(8)	(1)	(8)
	343-1 Advanced	X: PN	IOC, IOD	(1)	(3)/(8)	(2)/(6)	x	(21)	(8)	(6)	---	(1)	(8)	(6)	---	(1)	(8)	(6)	x	(1)	(8)	(1)	(8)
		X: IE	---	(1)	(3)/(8)	---	---	(21)	(8)	---	---	(1)	(8)	---	---	(1)	(8)	---	---	(1)	(8)	(1)	(8)
	343-1 ERPC		---	(1)	(3)/(8)	---	---	(21)	(8)	---	---	(1)	(8)	---	---	(1)	(8)	---	---	(1)	(8)	(1)	(8)

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"
(2) Load/transfer commands, DPRD_DAT, DPWR_DAT
(3) TSEND/TRCV, TUSEND/TURCV
(6) PNIO_SEND, PNIO_RECV
(8) AG_SEND/AG_RECV
(21) Controller 1 is server (for "PUT, GET")
(22) Controller 2 is server (for "PUT, GET")

21.4.3 S7-300 / S7-400

Table 21-20

Controller 2: S7-400				**Controller 1: S7-300 to PN/IE**																			
				CPU				CP															
				Interface: PN				343-1 Lean				343-1				343-1 Advanced						343-1 ERPC	
																Interface: PN				X: IE			
				IOC, IOD				IOD				IOC, IOD				IOC, IOD				---		---	
				S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**S7**	**OC**
						PNIO	CBA			PNIO	CBA			PNIO	CBA			PNIO	CBA				
CPU	Interface: PN		IOC, IOD	(1)	(3)	(2)	x	(21)	(8)/(3)	(6)/(2)	---	(1)	(8)/(3)	(6)/(2)	---	(1)	(8)/(3)	(6)/(2)	x	(1)	(8)/(3)	(1)	(8)/(3)
CP	443-1		IOC, IOD	(1)	(3)/(4)+(9)	(2)	---	(21)	(8)/(4)+(9)	(6)/(2)	---	(1)	(8)/(4)+(9)	(6)/(2)	---	(1)	(8)/(4)+(9)	(6)/(2)	---	(1)	(8)/(4)+(9)	(1)	(8)/(4)+(9)
	443-1 Advanced	X: PN	IOC, IOD	(1)	(3)/(4)+(9)	(2)	x	(21)	(8)/(4)+(9)	(6)/(2)	---	(1)	(8)/(4)+(9)	(6)/(2)	---	(1)	(8)/(4)+(9)	(6)/(2)	x	(1)	(8)/(4)+(9)	(1)	(8)/(4)+(9)
		X: IE	---	(1)	(3)/(4)+(9)	---	---	(21)	(8)/(4)+(9)	---	---	(1)	(8)/(4)+(9)	---	---	(1)	(8)/(4)+(9)	---	---	(1)	(8)/(4)+(9)	(1)	(8)/(4)+(9)

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

(21) Controller 1 is server (for "PUT, GET")

21.4.4 S7-300 / S7-1200

Table 21-21

Controller 2: S7-1200			Controller 1: S7-300 to PN/IE																			
			CPU				CP															
			all with interface: PN				343-1 Lean				343-1				343-1 Advanced						343-1 ERPC	
															Interface: PN				X: IE			
			IOC, IOD				IOD				IOC, IOD				IOC, IOD				---		---	
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**S7**	**OC**
					PNIO	CBA			PNIO	CBA			PNIO	CBA			PNIO	CBA				
CPU	Interface: PN	IOC	(40)	(3)/(3)+(5)	(2)	---	(21)	(8)/(3)+(5)	(2)	---	(40)	(8)/(3)+(5)	(2)	---	(40)	(8)/(3)+(5)	(2)	---	(40)	(8)/(3)+(5)	(40)	(8)/(3)+(5)

[Back to jump distributor PN/IE]

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(8) AG_SEND/AG_RECV

(21) Controller 1 is server (for "PUT, GET")

(40) "PUT, GET" (both controllers can be client or server)

21.4.5 S7-300 / S7-1500

Table 21-22

Controller 2: S7-1500			**Controller 1: S7-300 to PN/IE**																			
			CPU				CP															
			Interface: PN				343-1 Lean				343-1				343-1 Advanced						343-1 ERPC	
															Interface: PN				X: IE			
			IOC, IOD				IOD				IOC, IOD				IOC, IOD				---		---	
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**S7**	**OC**
					PNIO	CBA			PNIO	CBA			PNIO	CBA			PNIO	CBA				
CPU	Interface: PN	IOC, IOD	(1)	(3)/(3)+(5)	(2)	---	(21)	(8)/(3)+(5)	(6)/(2)	---	(1)	(8)/(3)+(5)	(6)/(2)	---	(1)	(8)/(3)+(5)	(6)/(2)	---	(1)	(8)/(3)+(5)	(1)	(8)/(3)+(5)
	Interface: IE	---	(1)	(3)/(3)+(5)	---	---	(21)	(8)/(3)+(5)	---	---	(1)	(8)/(3)+(5)	---	---	(1)	(8)/(3)+(5)	---	---	(1)	(8)/(3)+(5)	(1)	(8)/(3)+(5)
CP	CP 1543-1	---	(1)	(3)/(3)+(5)	---	---	(21)	(8)/(3)+(5)	---	---	(1)	(8)/(3)+(5)	---	---	(1)	(8)/(3)+(5)	---	---	(1)	(8)/(3)+(5)	(1)	(8)/(3)+(5)

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(21) Controller 1 is server (for "PUT, GET")

21.4.6 S7-300 / WinAC RTX

Table 21-23

Controller 2: WinAC RTX on PC WinAC RTX (F) 2010			**Controller 1: S7-300 to PN/IE**																			
			CPU				CP															
			Interface: PN				343-1 Lean				343-1				343-1 Advanced						343-1 ERPC	
															Interface: PN				X: IE			
			IOC, IOD				IOD				IOC, IOD				IOC, IOD				---		---	
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**S7**	**OC**
					PNIO	CBA			PNIO	CBA			PNIO	CBA			PNIO	CBA				
CPU/CP	Controlled by RTX (submodule): • "CP1616/CP1604" • "IE General"	IOC	(1)	(3)	(2)	x	(21)	(8)/(3)	(6)/(2)	---	(1)	(8)/(3)	(6)/(2)	---	(1)	(8)/(3)	(6)/(2)	x	(1)	(8)/(3)	(1)	(8)/(3)
	Controlled by Windows: • "CP1616-CP1604" • "IE General"	---	(1)	---	---	---	(21)	---	---	---	(1)	---	---	---	(1)	---	---	---	(1)	---	(1)	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(6) PNIO_SEND, PNIO_RECV

(8) AG_SEND/AG_RECV

(21) Controller 1 is server (for "PUT, GET")

21.5 PN/IE: Controller 1 = S7-400

21.5.1 (S7-400 / ET 200 CPU)

See ET 200 CPU / S7-400: 21.3.3

21.5.2 (S7-400 / S7-300)

See S7-300 / S7-400: 21.4.3

21.5.3 S7-400 / S7-400

Table 21-24

Controller 2: S7-400				**Controller 1: S7-400 to PN/IE**													
				CPU				CP									
				Interface: PN				443-1				443-1 Advanced					
												Interface: PN				X: IE	
				IOC, IOD				IOC, IOD				IOC, IOD				---	
				S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**
						PNIO	CBA			PNIO	CBA			PNIO	CBA		
CPU	Interface: PN		IOC, IOD	(1)	(3)	(2)	x	(1)	(4)+(9)/(3)	(2)	---	(1)	(4)+(9)/(3)	(2)	x	(1)	(4)+(9)/(3)
CP	443-1		IOC, IOD	(1)	(3)/(4)+(9)	(2)	---	(1)	(4)+(9)	(2)	---	(1)	(4)+(9)	(2)	---	(1)	(4)+(9)
	443-1 Advanced	Interface: PN	IOC, IOD	(1)	(3)/(4)+(9)	(2)	x	(1)	(4)+(9)	(2)	---	(1)	(4)+(9)	(2)	x	(1)	(4)+(9)
		Interface: IE	---	(1)	(3)/(4)+(9)	---	---	(1)	(4)+(9)	---	---	(1)	(4)+(9)	---	---	(1)	(4)+(9)

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

21.5.4 S7-400 / S7-1200

Table 21-25

Controller 2: S7-1200			**Controller 1: S7-400 to PN/IE**													
			CPU				CP									
			Interface: PN				443-1				443-1 Advanced					
											Interface: PN				X: IE	
			IOC, IOD				IOC, IOD				IOC, IOD				---	
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**
					PNIO	CBA			PNIO	CBA			PNIO	CBA		
CPU	Interface: PN	IOC	(40)	(3)/(3)+(5)	(2)	---	(40)	(4)+(9)/(3)+(5)	(2)	---	(40)	(4)+(9)/(3)+(5)	(2)	---	(40)	(4)+(9)/(3)+(5)

Back to jump distributor PN/IE

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(5) TSEND_C/TRCV_C

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

(40) "PUT, GET" (both controllers can be client or server)

21.5.5 S7-300 / S7-400

Table 21-26

Controller 2: S7-1500			**Controller 1: S7-400 to PN/IE**													
			CPU				CP									
			Interface: PN				443-1				443-1 Advanced					
											Interface: PN				X: IE	
			IOC, IOD				IOC, IOD				IOC, IOD				---	
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**
					PNIO	CBA			PNIO	CBA			PNIO	CBA		
CPU	Interface: PN	IOC, IOD	(1)	(3)/(3)+(5)	(2)	---	(1)	(4)+(9)/(3)+(5)	(2)	---	(1)	(4)+(9)/(3)+(5)	(2)	---	(1)	(4)+(9)/(3)+(5)
	Interface: IE	---	(1)	(3)/(3)+(5)	---	---	(1)	(4)+(9)/(3)+(5)	---	---	(1)	(4)+(9)/(3)+(5)	---	---	(1)	(4)+(9)/(3)+(5)
CP	CP 1543-1	---	(1)	(3)/(3)+(5)	---	---	(1)	(4)+(9)/(3)+(5)	---	---	(1)	(4)+(9)/(3)+(5)	---	---	(1)	(4)+(9)/(3)+(5)

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

21.5.6 S7-400 / WinAC RTX

Table 21-27

Controller 2: WinAC RTX to PC WinAC RTX (F) 2010			**Controller 1: S7-400 to PN/IE**													
			CPU				CP									
			Interface: PN				443-1				443-1 Advanced					
											Interface: PN				X: IE	
			IOC, IOD				IOC, IOD				IOC, IOD				---	
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**
					PNIO	CBA			PNIO	CBA			PNIO	CBA		
CPU/CP	Controlled by RTX (submodule): • "CP1616/CP1604" • "IE General"	IOC	(1)	(3)	(2)	x	(1)	(4)+(9)/(3)	(2)	---	(1)	(4)+(9)/(3)	(2)	x	(1)	(4)+(9)/(3)
	Controlled by Windows: • "CP1616-CP1604" • "IE General"	---	(1)	---	---	---	(1)	---	---	---	(1)	---	---	---	(1)	---

[Back to jump distributor PN/IE](#)

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(4) TSEND/TRCV

(9) AG_SEND/AG_RECV, AG_LSEND/AG_LRECV, AG_SSEND/AG_SRECV

21.6 PN/IE: Controller 1 = S7-1200

21.6.1 (S7-1200 / ET 200 CPU)

See ET 200 CPU / S7-1200: 21.3.4

21.6.2 (S7-1200 / S7-300)

See S7-300 / S7-1200: 21.4.4

21.6.3 (S7-1200 / S7-400)

See S7-400 / S7-1200: 21.5.4

21.6.4 S7-1200 / S7-1200

Table 21-28

Controller 2: S7-1200			**Controller 1: S7-1200 to PN/IE**			
			CPU			
			Interface: PN			
			IOC			
			S7	**OC**	**PN**	
					PNIO	CBA
CPU	Interface: PN	IOC	(40)	(3)+(5)	---	---

Back to jump distributor PN/IE

Communication blocks

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(40) "PUT, GET" (both controllers can be client or server)

21.6.5 S7-1200 / S7-1500

Table 21-29

<table>
<tr><td colspan="3" rowspan="6">Controller 2: S7-1500</td><td colspan="4">Controller 1: S7-1200 to PN/IE</td></tr>
<tr><td colspan="4">CPU</td></tr>
<tr><td colspan="4">Interface: PN</td></tr>
<tr><td colspan="4">IOC</td></tr>
<tr><td rowspan="2">S7</td><td rowspan="2">OC</td><td colspan="2">PN</td></tr>
<tr><td>PNIO</td><td>CBA</td></tr>
<tr><td rowspan="2">CPU</td><td>Interface: PN</td><td>IOC, IOD</td><td>(40)</td><td>(3)+(5)</td><td>(2)</td><td>---</td></tr>
<tr><td>Interface: IE</td><td>---</td><td>(40)</td><td>(3)+(5)</td><td></td><td>---</td></tr>
<tr><td>CP</td><td>1543-1</td><td>---</td><td>(40)</td><td>(3)+(5)</td><td></td><td>---</td></tr>
</table>

Back to jump distributor PN/IE

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(40) "PUT, GET" (both controllers can be client or server)

21.6.6 S7-1200 / WinAC RTX

Table 21-30

<table>
<tr><td colspan="3" rowspan="6">Controller 2: WinAC RTX to PC
WinAC RTX (F) 2010</td><td colspan="4">Controller 1: S7-1200 to PN/IE</td></tr>
<tr><td colspan="4">CPU</td></tr>
<tr><td colspan="4">Interface: PN</td></tr>
<tr><td colspan="4">IOC</td></tr>
<tr><td rowspan="2">S7</td><td rowspan="2">OC</td><td colspan="2">PN</td></tr>
<tr><td>PNIO</td><td>CBA</td></tr>
<tr><td rowspan="2">CPU/CP</td><td>Controlled by RTX (submodule):
• “CP1616/CP1604”
• “IE General”</td><td>IOC</td><td>(40)</td><td>(3)+(5)/(3)</td><td>---</td><td>---</td></tr>
<tr><td>Controlled by Windows:
• “CP1616/CP1604”
• “IE General”</td><td>---</td><td>(40)</td><td>---</td><td>---</td><td>---</td></tr>
</table>

[Back to jump distributor PN/IE]

Communication blocks

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

(40) "PUT, GET" (both controllers can be client or server)

21.7 PN/IE: Controller 1 = S7-1500

21.7.1 (S7-1500 / ET 200 CPU)

See ET 200 CPU / S7-1500: 21.3.5

21.7.2 (S7-1500 / S7-300)

See S7-300 / S7-1500: 21.4.5

21.7.3 (S7-1500 / S7-400)

See S7-400 / S7-1500: 21.5.5

21.7.4 (S7-1500 / S7-1200)

See S7-1200 / S7-1500: 21.6.5

21.7.5 S7-1500 / S7-1500

Table 21-31

Controller 2: S7-1500			**Controller 1: S7-1500 to PN/IE**											
			CPU								CP			
			Interface: PN				Interface: IE				1543-1			
			IOC, IOD				---				---			
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**	
					PNIO	CBA			PNIO	CBA			PNIO	CBA
CPU	Interface: PN	IOC, IOD	(1)	(3)+(5)	(2)	---	(1)	(3)+(5)	---	---	(1)	(3)+(5)	---	---
	Interface: IE	---	(1)	(3)+(5)	---	---	(1)	(3)+(5)	---	---	(1)	(3)+(5)	---	---
CP	CP 1543-1	---	(1)	(3)+(5)	---	---	(1)	(3)+(5)	---	---	(1)	(3)+(5)	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

21.7.6 S7-1500 / WinAC RTX

Table 21-32

Controller 2: WinAC RTX			**Controller 1: S7-1500 to PN/IE**											
			CPU								CP			
			Interface: PN				Interface: IE				1543-1			
			IOC, IOD				---				---			
			S7	**OC**	**PN**		**S7**	**OC**	**PN**		**S7**	**OC**	**PN**	
					PNIO	CBA			PNIO	CBA			PNIO	CBA
CPU	RTX	IOC	(1)	(3)+(5)/(3)	(2)	---	(1)	(3)+(5)/(3)	---	---	(1)	(3)+(5)/(3)	---	---
	Windows	---	(1)	---	---	---	(1)	---	---	---	(1)	---	---	---

Back to jump distributor PN/IE

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(3) TSEND/TRCV, TUSEND/TURCV

(5) TSEND_C/TRCV_C

21.8 PN/IE: Controller 1 = WinAC RTX

21.8.1 (WinAC RTX / ET 200 CPU)

See ET 200 CPU / WinAC RTX: 21.3.6

21.8.2 (WinAC RTX / S7-300)

See S7-300 / WinAC RTX: 21.4.6

21.8.3 (WinAC RTX / S7-400)

See S7-400 / WinAC RTX: 21.5.6

21.8.4 (WinAC RTX / S7-1200)

See S7-1200 / WinAC RTX: 21.6.6

21.8.5 (WinAC RTX / S7-1500)

See S7-1500 / WinAC RTX: 21.7.6

21.8.6 WinAC RTX / WinAC RTX

Table 21-33

Controller 2: WinAC RTX to PC WinAC RTX (F) 2010			**Controller 1: WinAC RTX to PN/IE** WinAC RTX (F) 2010							
			CPU/CP							
			Controlled by RTX (submodule): • “CP1616/CP1604” • “IE General”				Controlled by Windows: • “CP1616/CP1604” • “IE General”			
			IOC				---			
			S7	**OC**	**PN**		**S7**	**OC**	**PN**	
					PNIO	CBA			PNIO	CBA
CPU/CP	Controlled by RTX (submodule): • “CP1616/CP1604” • “IE General”	IOC	(1)	(3)	---	x	(1)	---	---	---
	Controlled by Windows: • “CP1616/CP1604” • “IE General”	---	(1)	---	---	---	(1)	---	---	---

[Back to jump distributor PN/IE]

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(3) TSEND/TRCV, TUSEND/TURCV

21.9 PN/IE: Overview of communication types

Comparison of all communication types with PN/IE.

Table 21-34 “Compact” communication type

	SIMATIC-specific	**Open standard**		
	S7 communication	**Open communication**		**PN communication**
		Send/Receive blocks	**T blocks** **T-Compact bocks**	**PNIO**
Protocols	ISO (only CP), IoT	ISO, IoT, TCP, UDP	IoT, TCP, UDP	PN
Interfaces	CPU, CP, CM	CP	CPU, CP, CM	CPU, CP, CM
Communication blocks (max. data)	BSEND (≤ 64 Kbytes) Type "USEND/URCV" (≥ 160 bytes) Type "PUT, GET" (≥ 160 bytes)	AG_xSEND (ISO, IoT, TCP ≤ 8 Kbytes) (UDP ≤ 2 Kbytes) Server for FETCH, WRITE (not UDP)	TSEND, TSEND_C, ... (IoT ≤ 32 Kbytes) (TCP ≤ 64 Kbytes) (UDP = 1472 bytes)	Load commands / transfer commands DPR_DAT, DPWR_DAT PNIO_SEND, PNIO_RECV
Remote confirmation	BSEND: application Type "USEND / URCV": transport Type "PUT, GET": application	ISO, IoT, TCP: transport UDP: none	IoT, TCP: transport UDP: none	Application
Connections?	yes	ISO, IoT, TCP: yes UDP: no	IoT, TCP: yes UDP: no	no

Back to jump distributor PN/IE

The data is taken from the “Communication types – detailed” table:

- S7 communication (Table 32 -1), PN communication (Table 37 -1)
- Open communication with Send/Receive blocks (Table 34-3), open communication with T bocks (Table 35 -3)

22 Selection Aid: PROFIBUS (PB)

22.1 PB: Content of the chapter

The following is described for the PB medium:

- Which interfaces (modules) and communication types are available?
(-> Interfaces table)
- Which partners can communicate via which communication types?
(-> Combinations table)
- Overview of all available Communication types
(-> "Communication types – compact" table)

22.2 PB: Interfaces and communication types

22.2.1 ET 200 CPU to PB

Table 22-1

Controller to PB: ET 200 CPU			Functionality of the interface		Communication type		
					SIMATIC-specific		Open standard
					S7 basic communication	S7 communication	DP communication
ET 200 S	CPU	IM151-7(F) CPU	DP slave (*1)	active	I_PUT, I_GET, Server	"PUT, GET", Server	(2)
				passive	I_PUT, I_GET, Server	---	(2)
	CP	DP master module (*2)	DP master		I_PUT, I_GET	"PUT, GET", Server	(2)
ET 200 Pro	CPU	IM154-8(F) PN/DP CPU	DP master		I_PUT, I_GET	"PUT, GET", Server	(2)
			DP slave (*1)	active	I_PUT, I_GET, Server	"PUT, GET", Server	(2)
				passive	I_PUT, I_GET, Server	---	(2)
	CP	DP master module	DP master		I_PUT, I_GET	"PUT, GET", Server	(2)

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

Explanations for the table

(*1): A DP slave can be active or passive. An active DP slave receives the token and is therefore also an active node at PROFIBUS. A passive DP slave receives no token. The settings are made by configuring the DB interface in STEP 7.

(*2): For IM151-7(F) CPU or IM151-8(F) PN/DP CPU

22.2.2 S7-300 to PB

Table 22-2

Controller to PB: S7-300		Functionality of the interface		Communication type				
				SIMATIC-specific		Open standard		
				S7 basic communication	S7 communication	Open communication	FMS communication	DP communication
CPU	Interface: DP, MPI/DP (*2)	DP master		I_PUT, I_GET	"PUT, GET", Server	---	---	(2)
		DP slave (*1)	active	I_PUT, I_GET, Server	"PUT, GET", Server	---	---	(2)
			passive	I_PUT, I_GET, Server	---	---	---	(2)
CP	342-5 (*3)	no DP operation		---	(1)	(8) (FDL)	---	---
		DP master		---	(1)	(8) (FDL)	---	(7)
		DP slave (*1)	active	---	"PUT, GET", Server	(8) (FDL)	---	(7)
			passive	---	---	---	---	(7)
	343-5	no DP operation		---	"PUT, GET", Server	(8) (FDL)	(10) (FMS)	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(8) AG_SEND/AG_RECV

(10) READ, WRITE, REPORT

Explanations for the table

(*1): A DP slave can be active or passive. An active DP slave receives the token and is therefore also an active node at PROFIBUS. A passive DP slave receives no token. The settings are made by configuring the DB interface in STEP 7.

(*2): MPI/DP interface in DP operating mode

(*3): 342-5 stands for the variants: CP 342-5, CP 342-5 FO

22.2.3 S7-400 to PB

Table 22-3

Controller to PB: S7-400		**Functionality of the interface**		**Communication type**				
				SIMATIC-specific		**Open standard**		
				S7 basic communication	**S7 communication**	**Open communication**	**FMS communication**	**DP communication**
CPU	Interface: DP (*3), MPI/DP (*2)	DP master		I_PUT, I_GET	(1)	---	---	(2)
		DP slave (*1)	active	I_PUT, I_GET, Server	"PUT, GET", Server	---	---	(2)
			passive	I_PUT, I_GET, Server	---	---	---	(2)
CP	443-5 Basic	no DP operation		---	(1)	(8) (FDL)	(10) (FMS)	---
	443-5 Extented	no DP operation		---	(1)	(8) (FDL)	---	---
		DP master		---	(1)	(8) (FDL)	---	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(8) AG_SEND/AG_RECV

(10) READ, WRITE, REPORT

Explanations for the table

(*1): A DP slave can be active or passive. An active DP slave receives the token and is therefore also an active node at PROFIBUS. A passive DP slave receives no token. The settings are made by configuring the DB interface in STEP 7.

(*2): MPI/DP interface in DP operating mode

(*3): Integrated interface (X1, ...) or plug-in interface module (IF1, ...)

22.2.4 S7-1200 to PB

Table 22-4

Controller to PB: S7-1200		**Functionality of the interface**	**Communication type**				
			SIMATIC-specific		**Open standard**		
			S7 basic communication	**S7 communication**	**Open communication**	**FMS communication**	**DP communication**
CM	CM 1242-5	DP slave	---	---	---	---	(2)
	CM 1243-5	DP master	---	"PUT, GET"	---	---	(2)

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

22.2.5 S7-1500 to PB

Table 22-5

Controller to PB: S7-1500			Functionality of the interface		Communication type				
					SIMATIC-specific		Open standard		
					S7 basic communication	S7 communication	Open communication	FMS communication	DP communication
CPU	1516-3 PN/DP	Interface: DP	DP master		---	(1)	---	---	(2)
CM	CM 1542-5		DP master		---	(1)	---	---	(2)
			DP slave (*1)	active	---	(1)	---	---	(2)
				passive	---	"PUT, GET", Server	---	---	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

Explanations for the table

(*1): A DP slave can be active or passive. An active DP slave receives the token and is therefore also an active node at PROFIBUS. A passive DP slave receives no token. The settings are made by configuring the DB interface in STEP 7.

22.2.6 S7-mEC to PB

Table 22-6

Controller to PB: S7-mEC with WinAC RTX (F) 2010			**Functionality of the interface**	**Communication type**	
				SIMATIC-specific	**Open standard**
				S7 communication	**DP communication**
CP	EM PCI-104	under the control of RTX (submodule): CP 5603	DP master	(1)	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

22.2.7 Box PC to PB

Table 22-7

Controller to PB: Box PC SIMATIC embedded bundles: HMI IPC427C with WinAC RTX (F) 2010		**Functionality of the interface**	**Communication type**	
			SIMATIC-specific	**Open standard**
			S7 communication	**DP communication**
CPU/CP (*1)	under the control of RTX (submodule): • "CP5611/CP5621" • "CP5613/CP5603/CP5623"	DP master	(1)	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

Explanations for the table

(*1): Meaning of the interfaces CPU or resp. CP: CPU = Onboard interface of the PC, CP = plug-in PC card

The designations of the interfaces in the table are collective names. This is explained in the following table.

Table 22-8

Interface	CP (plug-in to PC)	CPU (onboard interface of the PC)
"CP5611/CP5621"	CP 5611, CP 5611-A2, CP5621	Onboard PB interface of the SIMATIC IPC: CP 5611 integrated
"CP5613/CP5603/CP5623"	CP 5613, CP 5613-A2, CP 5603, CP 5623	---

22.2.8 Panel PC to PB

Table 22-9

Controller to PB: Box PC SIMATIC embedded bundles: HMI IPC477C with WinAC RTX (F) 2010		**Functionality of the interface**	**Communication type**	
			SIMATIC-specific	**Open standard**
			S7 communication	**DP communication**
CPU/CP (*1)	under the control of RTX (submodule): • "CP5611/CP5621" • "CP5613/CP5603/CP5623"	DP master	(1)	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

Explanations for the table

(*1): Meaning of the interfaces CPU or resp. CP: CPU = Onboard interface of the PC, CP = plug-in PC card

The designations of the interfaces in the table are collective names. This is explained in the following table.

Table 22-10

Interface	CP (plug-in to PC)	CPU (onboard interface of the PC)
"CP5611/CP5621"	CP 5611, CP 5611-A2, CP5621	Onboard PB interface of the SIMATIC IPC: CP 5611 integrated
"CP5613/CP5603/CP5623"	CP 5613, CP 5613-A2, CP 5603, CP 5623	---

22.2.9 WinAC RTX to PB

Table 22-11

Controller to PB: WinAC RTX WinAC RTX (F) 2010		**Functionality of the interface**	**Communication type**	
			SIMATIC-specific	**Open standard**
			S7 communication	**DP communication**
CPU / CP (*1)	under the control of RTX (submodule): • "CP5611/CP5621" • "CP5613/CP5603/CP5623"	DP master	(1)	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

Explanations for the table

(*1): Meaning of the interfaces CPU or resp. CP: CPU = onboard interface of the PC, CP = plug-in PC card

The designations of the interfaces in the table are collective names. This is explained in the following table.

Table 22-12

Collective name	PC card (corresponds to the CP interface)	Onboard interface of the PC (corresponds to the CPU interface)
"CP5611/CP5621"	CP 5611, CP 5611-A2, CP5621	Onboard PB interface of the SIMATIC IPC: CP 5611 integrated
"CP5613/CP5603/CP5623"	CP 5613, CP 5613-A2, CP 5603, CP 5623	---

22.3 PB: Controller 1 = ET 200 CPU

22.3.1 ET 200 CPU / ET 200 CPU

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-13

Controller 2: ET 200 CPU				**Controller 1: ET 200 CPU to PB**														
				ET200 S						ET 200 Pro								
				CPU			CP			CPU						CP		
				IM151-7 (F) CPU			DP master module			IM154-8 (F) PN/DP CPU						DP master module		
				DP slave passive			DP master			DP master			DP slave passive			DP master		
				S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7-B**	**S7**	**DP**	**S7 B**	**S7**	**DP**
ET200 S	CPU	IM151-7 (F) CPU	DP slave passive	---	---	---	(62)	---	(2)	(62)	---	(2)	---	---	---	(62)	---	(2)
	CP	DP master module	DP master	(61)	---	(2)	---	---	---	---	---	---	(61)	---	(2)	---	---	---
ET200 Pro	CPU	IM154-8(F) PN/DP CPU	DP master	(61)	---	(2)	---	---	---	---	---	---	(61)	---	(2)	---	---	---
			DP slave passive	---	---	---	(62)		(2)	(62)	---	(2)	---	---	---	(62)	---	(2)
	CP	DP master module	DP master	(61)	---	(2)	---	---	---	---	---	---	(61)	---	(2)	---	---	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(61) Controller 1 is server (for I_PUT, I_GET)

(62) Controller 2 is server (for I_PUT, I_GET)

22.3.2 ET 200 CPU / S7-300

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-14

Controller 2: S7-300			**Controller 1: ET 200 CPU to PB**														
			ET200 S						ET 200 Pro								
			CPU			CP			CPU						CP		
			IM151-7 (F) CPU			DP master module			IM154-8 (F) PN/DP CPU						DP master module		
			DP slave passive			DP master			DP master			DP slave passive			DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7-B**	**S7**	**DP**	**S7-B**	**S7**	**DP**
CPU	Interface: DP, MPI/DP	DP master	(61)	---	(2)	---	---	---	---	---	---	(61)	---	(2)	---	---	---
		DP slave passive	---	---	---	(62)	---	(2)	(62)	---	(2)	---	---	---	(62)	---	(2)
CP	342-5	no DP	---	---	---	---	(21)	---	---	(21)	---	---	---	---	---	(21)	---
		DP master	---	---	(2) / (7)	---	(21)	---	---	(21)	---	---	---	(2) / (7)	---	(21)	---
		DP slave passive	---	---	---	---	---	(2) / (7)	---	---	(2) / (7)	---	---	---	---	---	(2) / (7)
	343-5	no DP	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(21) Controller 1 is server (for "PUT, GET")

(61) Controller 1 is server (for I_PUT, I_GET)

(62) Controller 2 is server (for I_PUT, I_GET)

22.3.3 ET 200 CPU / S7-400

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-15

Controller 2: S7-400			**Controller 1: ET 200 CPU to PB**														
			ET200 S						ET 200 Pro								
			CPU			CP			CPU						CP		
			IM151-7 (F) CPU			DP master module			IM154-8 (F) PN/DP CPU						DP master module		
			DP slave passive			DP master			DP master			DP slave passive			DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7-B**	**S7**	**DP**
CPU	Interface: DP, MPI/DP	DP master	(61)	---	(2)	---	(21)	---	---	(21)	---	(61)	---	(2)	---	(21)	---
		DP slave passive	---	---	---	(62)	---	(2)	(62)	---	(2)	---	---	---	(62)	---	(2)
CP	443-5 Basic	no DP	---	---	---	---	(21)	---	---	(21)	---	---	---	---	---	(21)	---
	443-5 Ext.	no DP	---	---	---	---	(21)	---	---	(21)	---	---	---	---	---	(21)	---
		DP master	---	---	(2)	---	(21)	---	---	(21)	---	---	---	(2)	---	(21)	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(21) Controller 1 is server (for "PUT, GET")

(61) Controller 1 is server (for I_PUT, I_GET)

(62) Controller 2 is server (for I_PUT, I_GET)

22.3.4 ET 200 CPU / S7-1200

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-16

Controller 2: S7-1200			**Controller 1: ET 200 CPU to PB**														
			ET200 S						ET 200 Pro								
			CPU			CP			CPU						CP		
			IM151-7 (F) CPU			DP master module			IM154-8 (F) PN/DP CPU						DP master module		
			DP slave passive			DP master			DP master			DP slave passive			DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7-B**	**S7**	**DP**
CM	CM 1242-5	DP slave	---	---	---	---	---	(2)	---	---	(2)	---	---	---	---	---	(2)
	CM 1243-5	DP master	---	---	(2)	---	(21)	---	---	(21)	---	---	---	(2)	---	(21)	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(21) Controller 1 is server (for "PUT, GET")

22.3.5 ET 200 CPU / S7-1500

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-17

Controller 2: S7-1500			**Controller 1: ET 200 CPU to PB**														
			ET200 S						ET 200 Pro								
			CPU			CP			CPU						CP		
			IM151-7 (F) CPU			DP master module			IM154-8 (F) PN/DP CPU						DP master module		
			DP slave passive			DP master			DP master			DP slave passive			DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7-B**	**S7**	**DP**
CPU	Interface: DP	DP master	---	---	(2)	---	(21)	---	---	(21)	---	---	---	(2)	---	(21)	---
CM	1542-5 (DP)	DP master	---	---	(2)	---	(21)	---	---	(21)	---	---	---	(2)	---	(21)	---
		DP slave passive	---	---	---	---	---	(2)	---	---	(2)	---	---	---	---	---	(2)

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(21) Controller 1 is server (for "PUT, GET")

22.3.6 ET 200 CPU / WinAC RTX

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-18

Controller 2: WinAC RTX to PC WinAC RTX (F) 2010			**Controller 1: ET 200 CPU to PB**														
			ET200 S						ET 200 Pro								
			CPU			CP			CPU						CP		
			IM151-7 (F) CPU			DP master module			IM154-8 (F) PN/DP CPU						DP master module		
			DP slave passive			DP master			DP master			DP slave passive			DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7 B**	**S7**	**DP**
CPU/CP	Controlled by RTX (submodule): • “CP5611/CP5621” • “CP5613CP5603/CP5623”	DP master	---	---	(2)	---	(21)	---	---	(21)	---	---	---	(2)	---	(21)	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(21) Controller 1 is server (for "PUT, GET")

22.4 PB: Controller 1 = S7-300

22.4.1 (S7-300 / ET 200 CPU)

See ET 200 CPU / S7-300: 22.3.2

22.4.2 S7-300 / S7-300

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-19

Controller 2: S7-300			**Controller 1: S7-300 to PB**																
			CPU						CP										
			Interface: DP, MPI/DP						342-5								343-5		
			DP master			DP slave passive			no DP		DP master			DP slave passive			no DP		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**FMS**
CPU	Interface: DP, MPI/DP	DP master	---	---	---	(61)	---	(2)	(22)	---	(22)	---	---	---	---	(7) / (2)	---	---	---
		DP slave passive	(62)	---	(2)	---	---	---	---	---	---	---	(7) / (2)	---	---	---	---	---	---
CP	342-5	no DP	---	(21)	---	---	---	---	(1)	(8)	(1)	(8)	---	---	---	---	(21)	(8)	---
		DP master	---	(21)	---	---	---	(2) / (7)	(1)	(8)	(1)	(8)	---	---	---	(7)	(21)	(8)	---
		DP slave passive	---	---	(2) / (7)	---	---	---	---	---	---	---	(7)	---	---	---	---	---	---
	343-5	no DP	---	---	---	---	---	---	(22)	(8)	(22)	(8)	---	---	---	---	---	(8)	(10)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(8) AG_SEND/AG_RECV

(10) READ, WRITE, REPORT

(21) Controller 1 is server (for "PUT, GET")

(22) Controller 2 is server (for "PUT, GET")

(61) Controller 1 is server (for I_PUT, I_GET)

(62) Controller 2 is server (for I_PUT, I_GET)

22.4.3 S7-300 / S7-400

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-20

Controller 2: S7-400			**Controller 1: S7-300 to PB**																
			CPU						CP										
			Interface: DP, MPI/DP						342-5								343-5		
			DP master			DP slave passive			no DP		DP master			DP slave passive			no DP		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**FMS**
CPU	Interface: DP, MPI/DP	DP master	---	(21)	---	(61)	---	(2)	(1)	---	(1)	---	---	---	---	(7)/(2)	(21)	---	---
		DP slave passive	(62)	---	(2)	---	---	---	---	---	---	---	(7)/(2)	---	---	---	---	---	---
CP	443-5 Basic	no DP	---	(21)	---	---	---	---	(1)	(8)	(1)	(8)	---	---	---	---	(21)	(8)	(10)
	443-5 Ext.	no DP	---	(21)	---	---	---	---	(1)	(8)	(1)	(8)	---	---	---	---	(21)	(8)	---
		DP master	---	(21)	---	---	---	(2)	(1)	(8)	(1)	(8)	---	---	---	(7)/(2)	(21)	(8)	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(8) AG_SEND/AG_RECV

(10) READ, WRITE, REPORT

(21) Controller 1 is server (for "PUT, GET")

(61) Controller 1 is server (for I_PUT, I_GET)

(62) Controller 2 is server (for I_PUT, I_GET)

22.4.4 S7-300 / S7-1200

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-21

Controller 2: S7-1200			**Controller 1: S7-300 to PB**																
			CPU						CP										
			Interface: DP, MPI/DP						342-5								343-5		
			DP master			DP slave passive			no DP		DP master			DP slave passive			no DP		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**FMS**
CM	CM 1242-5	DP slave	---	---	(2)	---	---	---	---	---	---	---	(7)/(2)	---	---	---	---	---	---
	CM 1243-5	DP master	---	(21)	---	---	---	(2)	(40)	---	(40)	---	---	---	---	(7)/(2)	(21)	---	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(21) Controller 1 is server (for "PUT, GET")

(40) "PUT, GET" (both controllers can be client or server)

22.4.5 S7-300 / S7-1500

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-22

Controller 2: S7-1500			**Controller 1: S7-300 to PB**																
			CPU						CP										
			Interface: DP, MPI/DP						342-5								343-5		
			DP master			DP slave passive			no DP		DP master			DP slave passive			no DP		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**FMS**
CPU	Interface: DP	DP master	---	(21)	---	---	---	(2)	(1)	---	(1)	---	---	---	---	(7)/(2)	(21)	---	---
CM	1542-5	DP master	---	(21)	---	---	---	(2)	(1)	---	(1)	---	---	---	---	(7)/(2)	(21)	---	---
		DP slave passive	---	---	(2)	---	---	---	---	---	(22)	---	(7)/(2)	---	---	---	---	---	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(21) Controller 1 is server (for "PUT, GET")

(22) Controller 2 is server (for "PUT, GET")

22.4.6 S7-300 / WinAC RTX

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-23

Controller 2: WinAC RTX WinAC RTX (F) 2010			**Controller 1: S7-300 to PB**																
			CPU						CP										
			Interface: DP, MPI/DP						342-5								343-5		
			DP master			DP slave passive			no DP		DP master			DP slave passive			no DP		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**FMS**
CPU/CP	Controlled by RTX (submodule): • “CP5611/CP5621” • “CP5613CP5603/CP5623”	DP master	---	(21)	---	---	---	(2)	(1)	---	(1)	---	---	---	---	(7)/(2)	(21)	---	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(7) DP_SEND, DP_RECV

(21) Controller 1 is server (for "PUT, GET")

22.5 PB: Controller 1 = S7-400

22.5.1 (S7-400 / ET 200 CPU)

See ET 200 CPU / S7-400: 22.3.3

22.5.2 (S7-400 / S7-300)

See S7-300 / S7-400: 22.4.3

22.5.3 S7-400 / S7-400

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-24

Controller 2: S7-400			**Controller 1: S7-400 to PB**													
			CPU						CP							
			Interface: DP, MPI/DP						443-5 Basic			443-5 Ext.				
			DP master			DP slave passive			no DP			no DP		DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**FMS**	**S7**	**OC**	**S7**	**OC**	**DP**
CPU	Interface: DP, MPI/DP	DP master	---	(1)	---	(21)	---	(2)	(1)	---	---	(1)	---	(1)	---	---
		DP slave passive	(22)	---	(2)	---	---	---	---	---	---	---	---	---	---	(2)
CP	443-5 Basic	no DP	---	(1)	---	---	---	---	(1)	(8)	(10)	(1)	(8)	(1)	(8)	---
	443-5 Ext.	no DP	---	(1)	---	---	---	---	(1)	(8)	---	(1)	(8)	(1)	(8)	---
		DP master	---	(1)	---	---	---	(2)	(1)	(8)	---	(1)	(8)	(1)	(8)	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(8) AG_SEND/AG_RECV

(10) READ, WRITE, REPORT

(21) Controller 1 is server (for "PUT, GET")

(22) Controller 2 is server (for "PUT, GET")

22.5.4 S7-400 / S7-1200

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-25

Controller 2: S7-1200			**Controller 1: S7-400 to PB**													
			CPU						CP							
			Interface: DP, MPI/DP						443-5 Basic			443-5 Ext.				
			DP master			DP slave passive			no DP			no DP		DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**FMS**	**S7**	**OC**	**S7**	**OC**	**DP**
CM	CM 1242-5	DP slave	---	---	(2)	---	---	---	---	---	---	---	---	---	---	(2)
	CM 1243-5	DP master	---	(40)	---	---	---	(2)	(40)	---	---	(40)	---	(40)	---	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(40) "PUT, GET" (both controllers can be client or server)

22.5.5 S7-400 / S7-1500

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-26

Controller 2: S7-1500			**Controller 1: S7-400 to PB**													
			CPU						CP							
			Interface: DP, MPI/DP						443-5 Basic			443-5 Ext.				
			DP master			DP slave passive			no DP			no DP		DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**FMS**	**S7**	**OC**	**S7**	**OC**	**DP**
CPU	Interface: DP	DP master	---	(1)	---	---	---	(2)	(1)	---	---	(1)	---	(1)	---	---
CP	1542-5	DP master	---	(1)	---	---	---	(2)	(1)	---	---	(1)	---	(1)	---	---
		DP slave passive	---	(22)	(2)	---	---	---	(22)	---	---	(22)	---	(22)	---	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(22) Controller 2 is server (for "PUT, GET")

22.5.6 S7-400 / WinAC RTX

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-27

Controller 2: WinAC RTX WinAC RTX (F) 2010			**Controller 1: S7-400 to PB**													
			CPU						CP							
			Interface: DP, MPI/DP						443-5 Basic			443-5 Ext.				
			DP master			DP slave passive			no DP			no DP		DP master		
			S7 B	**S7**	**DP**	**S7 B**	**S7**	**DP**	**S7**	**OC**	**FMS**	**S7**	**OC**	**S7**	**OC**	**DP**
CPU/CP	Controlled by RTX (submodule): • “CP5611/CP5621” • “CP5613CP5603/CP5623”	DP master	---	(1)	---	---	---	(2)	(1)	---	---	(1)	---	(1)	---	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

22.6 PB: Controller 1 = S7-1200

22.6.1 (S7-1200 / ET 200 CPU)

See ET 200 C PU / S7-1200: 22.3.4

22.6.2 (S7-1200 / S7-300)

See S7-300 / S7-1200: 22.4.4

22.6.3 (S7-1200 / S7-400)

See S7-400 / S7-1200: 22.5.4

22.6.4 S7-1200 / S7-1200

Table 22-28

Controller 2: S7-1200			**Controller 1: S7-1200 to PB**			
			CM			
			CM 1243-5		CM 1242-5	
			DP master		DP slave	
			S7	**DP**	**S7**	**DP**
CM	CM 1242-5	DP slave	---	(2)	---	---
	CM 1243-5	DP master	(40)	---	---	(2)

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(40) "PUT, GET" (both controllers can be client or server)

22.6.5 S7-1200 / S7-1500

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-29

Controller 2: S7-1500			**Controller 1: S7-1200 to PB**			
			CM			
			CM 1243-5		CM 1242-5	
			DP master		DP slave	
			S7	**DP**	**S7**	**DP**
CPU	Interface: DP	DP master	(40)	---	---	(2)
CM	1542-5	DP master	(40)	---	---	(2)
		DP slave passive	(22)	(2)	---	---

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(22) Controller 2 is server (for "PUT, GET")

(40) "PUT, GET" (both controllers can be client or server)

22.6.6 S7-1200 / WinAC RTX

Table 22-30

<table>
<tr><td colspan="3" rowspan="5">Controller 2: WinAC RTX
WinAC RTX (F) 2010</td><td colspan="4">Controller 1: S7-1200 to PB</td></tr>
<tr><td colspan="4">CM</td></tr>
<tr><td colspan="2">CM 1243-5</td><td colspan="2">CM 1242-5</td></tr>
<tr><td colspan="2">DP master</td><td colspan="2">DP slave</td></tr>
<tr><td>S7</td><td>DP</td><td>S7</td><td>DP</td></tr>
<tr><td>CPU/CP</td><td>Controlled by RTX (submodule):
• “CP5611/CP5621”
• “CP5613CP5603/CP5623”</td><td>DP master</td><td>(40)</td><td>---</td><td>---</td><td>(2)</td></tr>
</table>

Back to jump distributor PB

Communication blocks

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(40) "PUT, GET" (both controllers can be client or server)

22.7 PB: Controller 1 = S7-1500

22.7.1 (S7-1500 / ET 200 CPU)

See ET 200 CPU / S7-1500: 22.3.5

22.7.2 (S7-1500 / S7-300)

See S7-300 / S7-1500: 22.4.5

22.7.3 (S7-1500 / S7-400)

See S7-400 / S7-1500: 22.5.5

22.7.4 (S7-1500 / S7-1200)

See S7-1200 / S7-1500: 22.6.5

22.7.5 S7-1500 / S7-1500

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-31

Controller 2: S7-1500			**Controller 1: S7-1500 to PB**								
			CPU			CM					
			Interface: DP			1542-5					
			DP master			DP master			DP slave passive		
			S7 B	**S7**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**
CPU	Interface: DP	DP master	---	(1)	---	(1)	---	---	(21)	---	(2)
CP	1542-5	DP master	---	(1)	---	(1)	---	---	(21)	---	(2)
		DP slave passive	---	(22)	(2)	(22)	---	(2)	---	---	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(21) Controller 1 is server (for "PUT, GET")

(22) Controller 2 is server (for "PUT, GET")

22.7.6 S7-1500 / WinAC RTX

In the following table it is assumed that the DP slave interface is “passive”. If the DP slave interface is operated as “active”, then there are additional communication options (see chapter 22.2).

Table 22-32

Controller 2: WinAC RTX			**Controller 1: S7-1500 to PB**								
WinAC RTX (F) 2010			CPU			CM					
			Interface: DP			1542-5					
			DP master			DP master			DP slave passive		
			S7 B	**S7**	**DP**	**S7**	**OC**	**DP**	**S7**	**OC**	**DP**
CPU/CP	Controlled by RTX (submodule): • “CP5611/CP5621” • “CP5613CP5603/CP5623”	DP master	---	(1)	---	(1)	---	---	(21)	---	(2)

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(2) Load/transfer commands, DPRD_DAT, DPWR_DAT

(21) Controller 1 is server (for "PUT, GET")

22.8 PB: Controller 1 = WinAC RTX

22.8.1 (WinAC RTX / ET 200 CPU)

See ET 200 CPU / WinAC RTX: 22.3.6

22.8.2 (WinAC RTX / S7-300)

See S7-300 / WinAC RTX: 22.4.6

22.8.3 (WinAC RTX / S7-400)

See S7-400 / WinAC RTX: 22.5.6

22.8.4 (WinAC RTX / S7-1200)

See S7-1200 / WinAC RTX: 22.6.6

22.8.5 (WinAC RTX / S7-1500)

See S7-1500 / WinAC RTX: 22.7.6

22.8.6 WinAC RTX / WinAC RTX

Table 22-33

Controller 2: WinAC RTX WinAC RTX (F) 2010			**Controller 1: WinAC RTX to PB** WinAC RTX (F) 2010	
			CPU/CP	
			Controlled by RTX (submodule): • “CP5611/CP5621” • “CP5613CP5603/CP5623”	
			DP master	
			S7	**DP**
CPU/CP	Controlled by RTX (submodule): • “CP5611/CP5621” • “CP5613CP5603/CP5623”	DP master	(1)	---

Back to jump distributor PB

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

22.9 PB: Overview of communication types

Comparison of all communication types with PB.

Table 22-34 “Compact” communication type

	SIMATIC-specific		**Open standard**		**DP communication**
	S7 basic communication	**S7 communication**	**Open communication**	**FMS communication**	
Protocols	S7	S7	FDL	FMS	DP
Interfaces	CPU	CPU, CP, CM	CP	CP	CPU, CP, CM
Communication blocks (max. data)	I_PUT (= 84 bytes) I_GET (= 94 bytes)	BSEND (<= 64 Kbytes) Type "USEND/URCV" (>= 160 bytes) Type "PUT, GET" (>= 160 bytes)	AG_SEND (=240 bytes) AG_LSEND (=240 bytes)	READ (<= 237 bytes) WRITE (<= 233 bytes) REPORT (<= 233 bytes)	Load commands / transfer commands DPR_DAT, DPWR_DAT DP_SEND, DP_RECV
Remote confirmation	Application	BSEND: Application Type "USEND/URCV": Transport Type "PUT, GET": Application	Transport	READ: Application WRITE: Application REPORT: none	Application
Connections?	yes	yes	yes	yes	no

Back to jump distributor PB

The data is taken from the “Communication types – detailed” table:

- S7 basic communication (Table 31 -1), S7 communication (Table 32-2)
- Open communication with send/receive blocks (Table 39 -1)
- FMS communication (Table 40 -1), DP communication (Table 41 -1)

23 Selection Aid: MPI

23.1 MPI: Content of the chapter

The following is described for the MPI medium:

- Which interfaces (modules) and communication types are available? (-> Interfaces table)
- Which partners can communicate via which communication types? (-> Combinations table)
- Overview of all available Communication types (-> “Communication types – compact” table)

23.2 MPI: Interfaces and communication types

23.2.1 ET 200 CPU to MPI

Table 23-1

Controller to MPI: ET 200 CPU			**Communication type**		
			SIMATIC-specific		
			Global data	**S7 basic communication**	**S7 communication**
ET 200 S	CPU	IM151-7(F) CPU	cyclically	(11)	"PUT, GET", Server
ET 200 Pro	CPU	IM154-8(F) PN/DP CPU	cyclically	(11)	"PUT, GET", Server

Back to jump distributor MPI

Communication blocks

(11) X_SEND/X_RCV, X_PUT, X_GET

23.2.2 S7-300 to MPI

Table 23-2

Controller to MPI: S7-300		Communication type		
		SIMATIC-specific		
		Global data	**S7 basic communication**	**S7 communication**
CPU	Interface: MPI, MPI/DP (*1)	cyclically	(11)	"PUT, GET", Server

Back to jump distributor MPI

Communication blocks

(11) X_SEND/X_RCV, X_PUT, X_GET

Explanations for the table

(*1): MPI/DP interface in MPI operating mode

23.2.3 S7-400 to MPI

Table 23-3

Controller to MPI: S7-400		Communication type		
		SIMATIC-specific		
		Global data	S7 basic communication	S7 communication
CPU	Interface: MPI/DP (*1)	cyclically acyclic (12)	(11)	(1)

Back to jump distributor MPI

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(11) X_SEND/X_RCV, X_PUT, X_GET

(12) GD_SND/GD_RCV

Explanations for the table

(*1): MPI/DP interface in MPI operating mode

23.3 MPI: Controller 1 = ET 200 CPU

23.3.1 ET 200 CPU / ET 200 CPU

Table 23-4

Controller 2: ET 200 CPU		**Controller 1: ET 200 CPU to MPI**		
		CPU		
		IM151-7 (F) CPU, IM154-8 (F) PN/DP CPU		
		GD	**S7 Basis**	**S7**
CPU	151-7, 154-8	cyclically	(11)	---

Back to jump distributor MPI

Communication blocks

(11) X_SEND/X_RCV, X_PUT, X_GET

23.3.2 ET 200 CPU / S7-300

Table 23-5

Controller 2: S7-300		**Controller 1: ET 200 CPU to MPI**		
		CPU		
		IM151-7 (F) CPU, IM154-8 (F) PN/DP CPU		
		GD	**S7 basic**	**S7**
CPU	Interface: MPI, MPI/DP	cyclically	(11)	---

Back to jump distributor MPI

Communication blocks

(11) X_SEND/X_RCV, X_PUT, X_GET

23.3.3 ET 200 CPU / S7-400

Table 23-6

Controller 2: S7-400		**Controller 1: ET 200 CPU to MPI**		
		CPU		
		IM151-7 (F) CPU, IM154-8 (F) PN/DP CPU		
		GD	**S7 basic**	**S7**
CPU	Interface: MPI/DP	cyclically + C2 acyclic (12)	(11)	(21)

Back to jump distributor MPI

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(11) X_SEND/X_RCV, X_PUT, X_GET

(12) GD_SND/GD_RCV

(21) Controller 1 is server (for "PUT, GET")

23.4 MPI: Controller 1 = S7-300

23.4.1 (S7-300 / ET 200 CPU)

See ET 200 CPU / S7-300: 23.3.2

23.4.2 S7-300 / S7-300

Table 23-7

Controller 2: S7-300		**Controller 1: S7-300 to MPI**		
		CPU		
		all with interface: MPI, MPI/DP		
		GD	**S7 basic**	**S7**
CPU	Interface: MPI, MPI/DP	cyclically	(11)	---

Back to jump distributor MPI

Communication blocks

(11) X_SEND/X_RCV, X_PUT, X_GET

23.4.3 S7-300 / S7-400

Table 23-8

Controller 2: S7-400		Controller 1: S7-300 to MPI		
		CPU		
		all with interface: MPI, MPI/DP		
		GD	**S7 basic**	**S7**
CPU	Interface: MPI/DP	cyclically + C2 acyclic (12)	(11)	(21)

Back to jump distributor MPI

Communication blocks

(11) X_SEND/X_RCV, X_PUT, X_GET

(12) GD_SND/GD_RCV

(21) Controller 1 is server (for "PUT, GET")

23.5 MPI: Controller 1 = S7-400

23.5.1 (S7-400 / ET 200 CPU)

See ET 200 CPU / S7-400: 23.3.3

23.5.2 (S7-400 / S7-300)

See S7-300 / S7-400: 23.4.3

23.5.3 S7-400 / S7-400

Table 23-9

Controller 2: S7-400		**Controller 1: S7-400 to MPI**		
		CPU		
		all with interface: MPI/DP		
		GD	**S7 Basis**	**S7**
CPU	Interface: MPI/DP	cyclically + acyclic (12)	(11)	(1)

Back to jump distributor MPI

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(11) X_SEND/X_RCV, X_PUT, X_GET

(12) GD_SND/GD_RCV

23.6 MPI: Overview of communication types

Comparison of all communication types with MPI.

Table 23-10 “Compact” communication type

	SIMATIC-specific		
	Global data communication	**S7 basic communication**	**S7 communication**
Protocols	S7	S7	FDL
Interfaces	CPU	CPU	CPU, CP
Communication blocks (max. data)	cyclical: no communication blocks acyclic: GD_SND, GD_RCV (22 bytes per GD package)	X_PUT (= 76 bytes) X_GET (= 76 bytes)	BSEND (<= 64 Kbytes) Type "USEND/URCV" (>= 160 bytes) Type "PUT, GET" (>= 160 bytes)
Remote confirmation	no	Application	BSEND: Application Type "USEND/URCV": Transport Type "PUT, GET": Application
Connections?	no	yes	yes

Back to jump distributor MPI

The data is taken from the “Communication types – detailed” table:

- Global data communication (Table 30-1)
- S7 basic communication (Table 31 -1), S7 communication (Table 32 -1)

24 Selection Aid: SIMATIC Backplane Bus

24.1 Content of the chapter

The following is described for the SIMATIC backplane bus medium:

- Which interfaces (modules) are available?
- Which partners can communicate via which communication types? (*1)
- Overview of all available communication types

(*1)

Discussed combinations controller 1 / controller 2:

for S7-400, up to 4 CPUs can be operated in one single controller (multi-computing, chapter 5.5). This means, in the following tables there is no differentiation between controller 1 and controller 2. The CPUs are in the same controller!

24.2 SIMATIC backplane bus: interfaces and communication types

Table 24-1

CPU at backplane bus: S7-400		**Communication type**	
		SIMATIC	
		Global data communication	**S7 communication**
CPU	all	acyclic:	(1)

Back to jump distributor backplane bus

Communication blocks

(1) USEND/URCV, BSEND/BRCV, PUT, GET

24.3 Controller 1 = S7-400 / Controller 2 = S7-400

Table 24-2

Controller: S7-400		**Controller: S7-400**	
		CPU	
		all	
		GD	**S7**
CPU	all	cyclic, acyclic (12)	(1)

Back to jump distributor backplane bus

Communication blocks

(1) "USEND/URCV", BSEND/BRCV, "PUT, GET"

(12) GD_SND/GD_RCV

24.4 Overview of communication types

Comparison of all communication types available via the SIMATIC backplane bus.

Table 24-3 “Compact” communication types

Communication type	**SIMATIC**	
	Global data communication	**S7 communication**
Protocols	S7	S7
Interfaces	CPU	CPU
Communication blocks (max. data)	Cyclic transmission: no communication blocks necessary acyclic transmission GD_SND, GD_RCV (54 bytes per GD package)	BSEND (<= 64 Kbytes) Type "USEND/URCV" (>= 160 bytes) Type "PUT, GET" (>= 160 bytes)
Remote confirmation	none	BSEND: Application Type "USEND/URCV": Transport Type: "PUT, GET": Application
Connections?	no	yes

Back to jump distributor backplane bus

The data is taken from the “Communication types – detailed” table:

- Global data communication (Table 30-1)
- S7 basic communication (Table 31 -1)

25 Selection Aid: Serial Interface (PtP)

25.1 Content of the chapter

This chapter describes:

- What options the SIMATIC families offer for communication via a “serial interface”?
- The combination of SIMATIC families (such as for PN/IE, PB and MPI networks) is not discussed here.

25.2 ET 200 CPU to PtP

Table 25-1

Controller to PtP: ET 200 CPU			Protocol		Maximal number of data	Physics (max. distance communication partner)	Transmission rate
			Type	**integrated / loadable**			
ET 200S	CP	Modbus/USS module	Modbus serial (RTU format), Modbus master and Modbus slave USS protocol (*1)	integrated	224 bytes	RS 232C (15 m) RS 422/485 (1200 m)	110 Bit/s up to 115.2 Kbit/s
		1SI module	3964(R), *ASCII*	integrated			

Additional option: in decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

Explanations for the table

(*1): not further discussed in the document since no CPU-CPU communication can be realized with it.

25.3 S7-300 to PtP

Table 25-2

Controller to PtP: S7-300		Protocol: Type	Protocol: integrated / loadable	Maximal number of data	Physics (max. distance communication partner)	Transmission rate
CPU	313C-2 PtP	3964(R), *ASCII*	integrated	1024 bytes	RS 422/485 (1200 m)	300 bps up to 38.4 Kbit/s
	314C-2 PtP	3964(R), *ASCII*, RK512				
CP	340 (*2)	3964(R), *ASCII* printer (*3)	integrated	1024 bytes	RS 422/485 (1200 m) RS 232C (15 m) 20mA TTY (active: 100 m, passive 1000 m)	2.4 kBit/s up to 19.2 kBit/s
	341 (*2)	3964R, *ASCII*, RK512 printer (*3)	integrated	4096 bytes	RS 422/485 (1200m) RS 232C (15m) 20mA TTY (active and passive 1000m)	300 bit/s up to 115.2 Kbit/s at 20mA: up to 19.2 Kbit/s
		Modbus serial (RTU format), Modbus master and Modbus slave	loadable	dependent on function code (*1)		

Additional option: in decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

Explanations for the table

(*1): see respective manuals.

(*2): select the physics of the interface via the module variant

(*3): not further discussed in the document since no CPU-CPU communication can be realized with it.

25.4 S7-400 to PtP

Table 25-3

Controller to PtP: S7-400		Protocols		Max. number of data	Physics (max. distance communication partner)	Transmission rate
		Type	integrated / loadable			
CP	440	3964, *ASCII*	integrated	400 bytes	RS 422/485 (1200m)	300 Bit/s up to 115.2 kBit/s
	441-1 (*1)	3964, *ASCII* printer (*2)	integrated	ASCCII, 3964: 4096 bytes RK 512, send: 4096 bytes RK 512, fetch: 450 bytes	RS 232C (10m) RS 422/485 (1200m) 20mA-TTY (1000m)	300 Bit/s up to 115.2 kBit/s at 20mA-TTY: up to 19.2 kBit/s
	441-2 (*1)	3964, *ASCII*, RK512 printer (*2)	integrated			
		Modbus serial (RTU format), Modbus master and Modbus slave	loadable	dependent on function code (*3)		

Additional option: in decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

Explanations for the table

(*1): selecting the physics of the interface via the plug-in IF module

(*2): not further discussed in the document since no CPU-CPU communication can be realized with it.

(*3): see respective manuals.

25.5 S7-1200 to PtP

Table 25-4

Controller to PtP: S7-1200		Protocol		Maximal number of data	Physics (max. distance communication partner)	Transmission rate
		Type	integrated / loadable			
CPU	Interface: PN	Modbus/TCP, Modbus Client and Server	integrated			
	CB 1241 (*3)	Modbus serial (RTU format), Modbus master and slave	integrated		1000 m	300 Bit/s up to 115.2 kBit/s
CP	CM 1241	USS drive protocol (*2) User-defined protocol Modbus serial (RTU format), Modbus master and slave	integrated	1024 bytes	RS 485 (1000 m) RS 232 (10 m) (*1)	300 Bit/s up to 115.2 kBit/s

Additional option: in decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

Explanations for the table

(*1): selecting the physics: various module variants

(*2): not further discussed in the document since no CPU-CPU communication can be realized with it.

(*3): the module is not plugged into the CPU.

25.6 S7-1500 to PtP

Table 25-5

Controller to PtP: S7-1500			Protocol		Maximal number of data	Physics (max. distance communication partner)	Transmission rate
			Type	integrated / loadable			
CP	CM PtP RS232	BA	3964(R), Freeport, USS (*1)	integrated	1024 bytes	RS 232 incl. secondary signals (15 m)	300 to 19200 bit/s
		HF	3964(R), Freeport, USS (*1) Modbus RTU	integrated	4096 bytes	RS 232 incl. secondary signals (15 m)	300 to 115200 bit/s
	CM PtP RS422/485	BA	3964(R), Freeport, USS (*1)	integrated	1024 bytes	RS 422/485 (1200 m)	300 to 19200 bit/s
		HF	3964(R), Freeport, USS (*1) Modbus RTU	integrated	4096 bytes	RS 422/485 (1200 m)	300 to 115200 bit/s

Additional option: in decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

Explanations for the table

(*1): not further discussed in the document since no CPU-CPU communication can be realized with it.

25.7 S7-mEC to PtP

Table 25-6

Controller to PtP: S7-mEC		**Properties**
CP	CP 340	See chapter 25.3: S7-300 / CP 340

Additional option: in decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

25.8 Box PC to PtP

In decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

25.9 Panel PC to PtP

In decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

25.10 WinAC RTX to PtP

In decentralized ET 200 stations serial interface modules can be used (chapter 25.11).

Back to jump distributor SERIAL

25.11 Distributed station ET 200

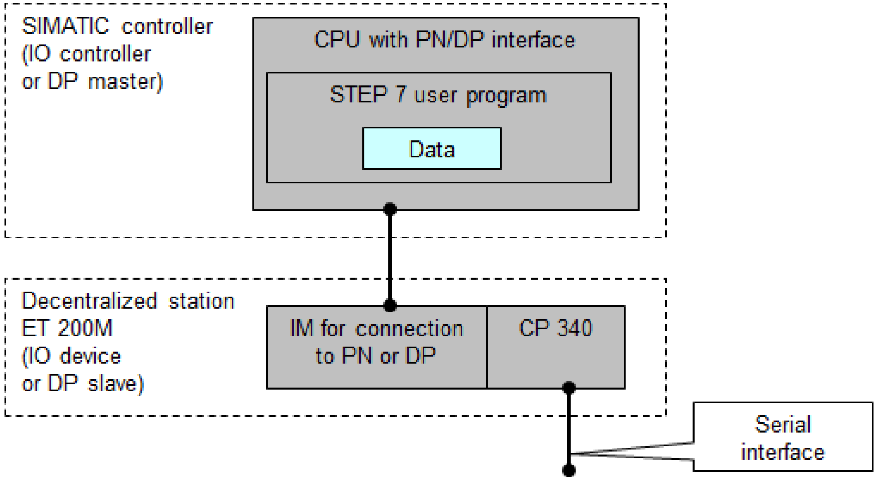
In decentralized ET 200 stations serial interface modules can be used.

Table 25-7

Serial interface module	Decentralized station	Medium
1-SI module	ET 200S	PROFINET, PROFIBUS
CP340, CP341	ET 200M	PROFINET, PROFIBUS
CM PtP RS232 BA CM PtP RS232 HF CM PtP RS422/485 BA CM PtP RS422/485 HF	ET 200MP	PROFINET
ET 200SP CM PtP	ET 200SP	PROFINET

This results in further options for a CPU-CPU communication via serial interface. All controllers with a PN or DP interface can use this option as IO controller or DP master. The figure shows one example.

Figure 25-1



Back to jump distributor SERIAL

26 Information on Part 2

The tables contain references to information on the topics in Part 2 (selection aid). All references /x/ are stored centrally in chapter 61. There you also find the respective internet links.

Communication

The following table contains links to manuals with details on the interfaces of the communication partners.

Table 26-1

/x/	Title	Information on
/7/	CPU 31xC and CPU 31 x technical data, device manual	Communication via: MPI, PB, IE
/8/	Automation system S7-400 CPU data, device manual	
/34/	SIMATIC S7 S7-1200 automation system, system manual	
/35/	SIMATIC S7 S7-1500 automation system, system manual	
/36/	SIMATIC WinAC RTX (F) 2010, operating instructions	
/37/	PROFINET system connection for SIMATIC S7, manuals	
/38/	PROFIBUS system connection for SIMATIC S7, manuals	
/39/	Serial communication	Communication via: serial interface
/21/	S7-300 CPU 31xC Equipment Modules, Manual	

Application examples

The following table contains references to application examples.

Table 26-2

/x/	Title / content
/200/	Applications examples on the topic of communication

Performance data

The following table contains a reference to the results of measurements:

Table 26-3

/x/	Title / content
/18/	Performance data on the communication

Examples for measurements:

- Transmission time for typical configurations on Industrial Ethernet
- PN response time for typical configurations in the PROFITNET IO

27 ****** PART 3: Communication Types ******

27.1 Structure and content

Table 27-1

Chapt.	Structure			Content
29	SIMATIC S7-specific communication			Overview
30		Global data communication		Description
31		S7 basic communication		
32		S7 communication		
33	PROFINET/Industrial Ethernet			Overview
34		Open communication with send/receive blocks		Description
35		Open communication with T blocks		
36		PN communication:	CBA	
37			PNIO	
38	PROFIBUS			Overview
39		Open communication with send/receive blocks		Description
40		FMS communication		
41		DP communication		
42	Serial interface			Overview
43		Overview of user interfaces		Tables with all interfaces
43 to 54		ET 200S, S7-300, S7-400, S7-1200, S7-1500		Description of the interfaces
55	Information			Manuals, FAQs, applications

Connecting to controllers with Modbus interface is described in Part 4:

Table 27-2

Chapt.	Structure	Content
57	Modbus/TCP	Communication via PN/IE network
58	Modbus serial (RTU format)	Communication via serial interface

28 Preliminary Remarks

The communication types are described according to a uniform schematic layout. For each communication type there are the following chapters:

- Characteristics
- Details of communication type
- Overview of user interfaces
- User interfaces

The contents of these chapters is described below.

28.1 Chapter: Characteristics

This chapter gives a brief characterization of the communication type.

28.2 Chapter: Details of communication type

This chapter contains a table named “Communication types – detailed”.

The most important properties of a communication type are described in this table.

28.2.1 Purpose of this table

The table answers the following questions:

- What are the properties of this communication type?
- What are the properties of the user interfaces (communication blocks) of this communication type?

Note

The “Communication type - detailed” tables are the basis for a further table named “Communication types – compact” (19.4).

In the “Communication types – compact” table, all communication types of a medium (PN/IE, ...) are summarized.

28.2.2 Structure of the table

The structure of the “Communication types – detailed” table are explained using the following example:

- Medium: PN/IE
- Communication type: open communication with T blocks.

The figure shows an extract of the table (Table 35 -3).

Figure 28-1

1

2

Communication type:		Open communication with T-blocks (PN/IE network)		
Protocol:		ISO on TCP	TCP	
General				
Interfaces		CPU, CP, CM	CPU, CP, CM	CPU, CP
Connection	SIMATIC S5	yes	yes	no
	third-party (open standards)	yes	yes	yes
Protocol				
Dynamic data length		yes	no	yes
Multicast / broadcast		no	no	no
Connections	to the remote partner?	yes	yes	no
	dynamic / static	TSEND/TRCV: dynamic + static TSEND_C/TRCV_C: dynamic	TSEND/TRCV: dynamic + static TSEND_C/TRCV_C: dynamic	dynamic
User interface				
Communication blocks		**TSEND / TRCV TSEND_C / TRCV_C**	**TSEND / TRCV TSEND_C / TRCV_C**	**TUSEND**
Maximal number of data **(*1)**		<= 64 Kbytes	<= 64 Kbytes	= 1472
Dynamic addressing of data		yes	yes	yes
Remote confirmation		Transport	Transport	no
Model		Client / Client	Client / Client	Client / C

The table consists of two areas:

Area 1

The area contains the designation of the communication type and the possible protocols.

Area 2

The area contains the criteria used for describing the communication type. The criteria are summarized under the following headings:

- General
- Protocol
- User interface

The individual criteria are described in the chapter below. A subset of these criteria appears in the “Communication types – compact” table. These criteria are underlined (example: Interfaces).

28.2.3 Criteria of the table

Note: The "---" entry in the value range of the table means: "the criterion is not relevant here."

Table 28-1

Criterion		Meaning	Value range
General			
Media		What are the media via which both communication peers can exchange data?	PN/IE, PB, MPI, backplane bus, serial interface
Interfaces		Here you enter the interfaces to which the media are connected. Integrated interface: CPU External interface: CP, CM	CPU, CP, CM
Connection	SIMATIC S5	Communication with SIMATIC S5 possible?	yes, no
	third-party	Is it possible to communicate with third-party controllers via open standards ?	yes, no
Protocol			
Dynamic data length		Can the data length be changed at the communication block during runtime (RUN of the CPU)?	yes, no
Multicast / broadcast		Here it is entered whether multicast or broadcast are possible. Multicast: simultaneous sending to several communication partners Broadcast: simultaneous sending to all communication partners	Multicast / broadcast
Connection	to the remote peer	Is a connection to the remote peer established for the communication?	yes, no
	dynamic / static	dynamic: the connection is cancelled after data transmission. static: the connection remains after data transmission. dynamic + static: both above cases are possible.	dynamic, static, dynamic + static

Criterion	Meaning	Value range
User interface		
Communication blocks	All communication blocks (FB, SFB, FC, SFC) available for data transmission are listed here. Blocks which might be required for the connecting and disconnecting process are not mentioned here (connection blocks). Meaning of the notation: a / b: data transmission requires both blocks (example: BSEND / BRCV) a, b: each individual block can perform a data transmission (example: PUT, GET)	BSEND / BRCV AG_SEND / AG_RECV etc.
Maximal data volume	The maximum data volume that can be transferred at once by a communication block is entered here: From "trigger job", to "job finished" Meaning of the notation: = x bytes: always x bytes (under all circumstances) <= x bytes: x bytes at the most, however it can also be less (depending on CPU, CP, ...) >= x bytes: x bytes at least, however it can also be more (depending on CPU, CP, ...)	= x bytes <= x bytes >= x bytes
Dynamic addressing of data	Is it possible to change the addressing of data areas during runtime (RUN of the CPU)?	yes, no
Remote confirmation	Here it is specified whether a data transmission of remote communication partners (CPU, CP, CM) is confirmed and what the confirmation means. no: no confirmation from remote • These data were sent and have left the local partner. • The sender is not notified whether the data has arrived in the remote user area (in the remote application of the CPU). Transport: transport confirmation from remote • Data was sent and received by the remote partner (CPU, CP, CM). • The sender is not notified whether the data has arrived in the remote user area (in the remote application of the CPU). Application: application confirmation from remote • Data was sent and the data has arrived in the remote user area (in the remote application of the CPU).	no, transport, application
Model	Here it is specified which communication model the data transmission is based on. Description of the configuration models: see chapter 64.2.	Client / Client, Client / Server, S7 only Server, Master / Slave, Consumer / Provider

28.3 Chapter: Overview of user interfaces

This chapter gives an overview of all communication blocks of the respective communication type.

The following questions are answered:

- For which family and interface (CPU, CP, CM) are the communication blocks suitable?
- Where in STEP 7 can these communication blocks be found?

28.4 Chapter: User interfaces

The user interfaces of the respective communication types are briefly described:

- Functionality of the communication blocks
- Parameters of the communication blocks

In the document on hand, uniform terms are used for the description of the communication blocks. In this context, both cases can be distinguished:

- the remote partner is not given any address information of the data
- the remote partner is given the address information of the data

The following figures show the terms used in this context.

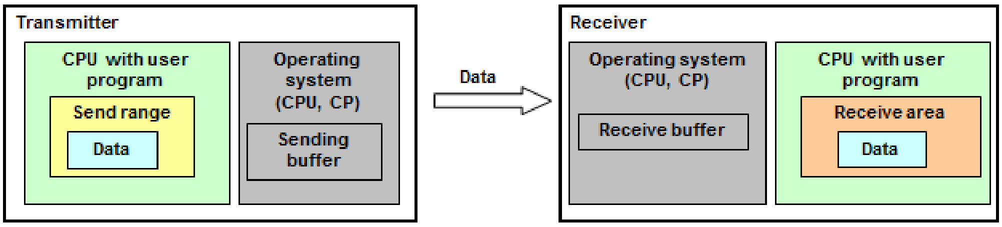
The remote partner is <u>not</u> given any address information of the data

Used terms:

- send range, receive range
- send buffer, receive buffer
- send data
- receive data

<u>Example: send data</u>

Figure 28-2



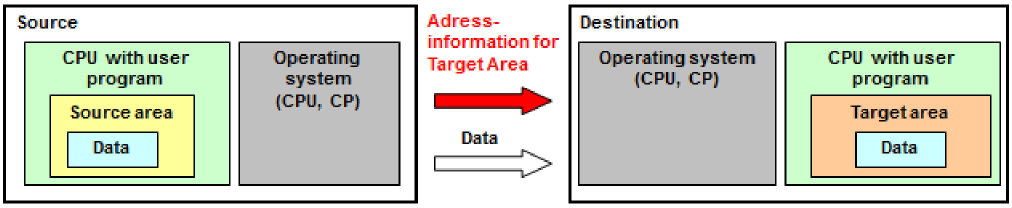
The remote partner is given the address information of the data

Used terms:

- source area, target area
- write data
- read data, fetch

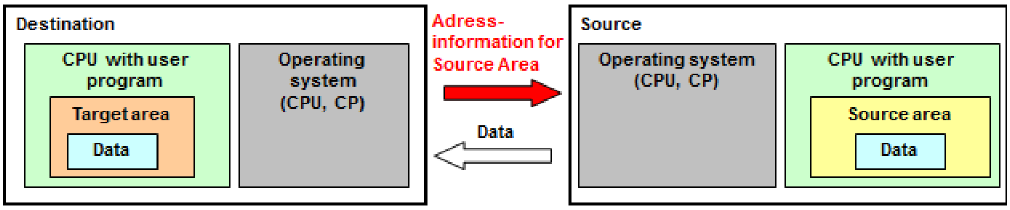
<u>Write data</u>

Figure 28-3



<u>Read data, fetch</u>

Figure 28-4



29 SIMATIC S7-specific Communication

29.1 Characteristics

The SIMATIC-specific communication is characterized by the following characteristics:

- Optimized communication between SIMATIC S7 controllers
- Manufacturer-specific communication

29.2 Overview

Two following communication types and media are available:

Table 29-1

Communication type	Media				
	Network			Backplane bus	Serial interface
	PN/IE	PB	MPI		
Global data communication	---	---	x	x	---
S7 basic communication	---	x (DP)	x	---	---
S7 communication	x	x	x	x	---

In the following chapters the communication types are described:

Table 29-2

	Chapt.
Global data communication	30
S7 basic communication	31
S7 communication	32

30 Global Data Communication

30.1 Characteristics

Global data

Data exchanged in this form of communication are referred to as global data (GD).

Global data can be:

- inputs, outputs (process image)
- flag
- areas of data blocks
- times, counter

Data exchange occurs cyclic during updating the process image of the inputs and outputs. For S7-400 event-controlled data exchange is possible.

Global data are transferred in GD packages. A GD package is a frame sent from a SIMATIC CPU to one or several other SIMATIC CPUs.

Characteristics

The GD communication is characterized by the following characteristics:

- very simple application
- data volume: <= 54 bytes

30.2 Details communication type

Table 30-1 Communication types - detailed

Communication type:		**GD communication**
Protocol:		**S7 protocol**
General		
Media		MPI, backplane bus (only for SIMATIC S7-400, multicomputing)
Interfaces		CPU
Connection	SIMATIC S5	no
	third-party (open standards)	no
Protocol		
Dynamic data length		no
Multicast / broadcast		Multicast
Connections	to the remote peer	no
	dynamic / static	---
User interface		
Communication blocks		Cyclic transmission: no communication blocks necessary acyclic transmission (only S7-400): **GD_SND, GD_RCV**
Maximal data volume		S7-300: Data volume per GD packet: = 22 bytes
		S7-400: Data volume per GD packet: = 54 bytes
Dynamic addressing of data		no
Remote confirmation		no
Model		---

Back to jump distributor MPI
Back to jump distributor backplane bus

30.3 Overview of user interfaces

Cyclic transmission

No communication blocks are required.

Acyclic transmission

Overview of communication blocks:

Table 30-2

Communication blocks	S7-400
	CPU
GD_SND	SFC 60
GD_RCV	SFC 61

Communication blocks in STEP 7:

Table 30-3

Interface		available in STEP 7	
S7-400	CPU	STEP 7 (not TIA)	Library: Standard library / System function blocks
		STEP 7 (TIA)	---

30.4 User interface GD_SND, GD_RCV

30.4.1 Description

The communication blocks enable an acyclic GD communication between the CPUs of S7-400.

GD_SND

Programmed sending of a GD package

GD_RCV

Programmed receiving of a GD package

30.4.2 Parameters for GD_SND

Table 30-4

INPUT	Type	Remark
CIRCLE_ID	BYTE	Number of GD group in which the GD package to be send is located.
BLOCK_ID	BYTE	Number of GD packet to be sent in the selected GD circle.
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information

30.4.3 Parameters for GD_RCV

Table 30-5

INPUT	Type	Remark
CIRCLE_ID	BYTE	Number of GD group to which the arrived GD package is to be entered.
BLOCK_ID	BYTE	Number of GD package to which the arrived data is to be entered.
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information

31 S7 Basic Communication

31.1 Characteristics

Communication partners

Possible communication partners for a CPU-CPU communication using the S7 basic communication:

- CPU outside the own station (*1)
- CPU within the own station:
  - CPU in the central device or expansion unit
  - CPU decentralized

(*1):

"Own station" refers to the following:

- central station with CPU, CP, central I/O, and
- decentralized station.

In a decentralized station a CPU can be plugged in. In this case, this decentralized station is also referred to as "intelligent slave (I-slave)".

CPU outside the own station

Communication is realized via the MPI network.

The required communication blocks are referred to as "X blocks".

CPU within the own station

Communication is realized via the PB (DP) network.

The required communication blocks are referred to as "I-blocks".

Data is exchanged between DP master and DP slave:

- with I-block the DB master reads data from the DP slave
- with I-block the DB master writes data to the DP slave

No communication blocks are required in the DP slave.

Characteristics

The S7 basic communication is characterized by the following characteristics:

- Communication only via the integrated interface of the CPU
- Communication via connections. The connections are established by the communication block
  (Configuration is not necessary in STEP 7).
- Data volume per communication job: < 94 bytes
- The user program in the sender contains an item of information if the data in the user data area of the receiver have arrived (confirmation by the remote application).

31.2 Details communication type

Table 31 -1 Communication types - detailed

Communication type:		**S7 basic communication**		
Protocol:		**S7 protocol**		
General				
Media		PB (DP)	MPI	MPI
Interfaces		CPU	CPU	CPU
Connection	SIMATIC S5	no	no	no
	third-party (open standards)	no	no	no
User interface				
Communication blocks		**I_PUT, I_GET**	**X_PUT, X_GET**	**X_SEND / X_RCV**
Maximal data volume		I_PUT: = 84 bytes I_GET: = 94 bytes	= 76 bytes	= 76 bytes
Dynamic addressing of data		yes	yes	yes
Remote confirmation		Application	Application	Application
Model		Client / Server	Client / Server	Client / Client
Protocol				
Dynamic data length		yes		
Multicast / broadcast		no		
Connections	to the remote peer	yes		
	dynamic / static	dynamic + static		

Back to jump distributor MPI

Back to jump distributor backplane bus

Back to jump distributor PB

31.3 Overview of user interfaces

Overview of communication blocks:

Table 31-2

Communication block		S7-300	S7-400
		CPU	CPU
X-blocks (Network MPI)	X_SEND / X_RCV	SFC 65 / SFC 66	SFC 65 / SFC 66
	X_PUT	SFC 68	SFC 68
	X_GET	SFC 67	SFC 67
I-blocks (Network PB(DP))	I_PUT	SFC 73	SFC 73
	I_GET	SFC 72	SFC 72

Communication blocks in STEP 7:

Table 31-3

Interface		available in STEP 7	
S7-300, S7-400	CPU	STEP 7 (not TIA)	Library: Standard library / System function blocks
		STEP 7 (TIA)	Instructions: Communication

31.4 User interface X_SEND/ X_RCV

31.4.1 Description

With the communication blocks a CPU communicates with another CPU outside the own station (client / client communication).

X_SEND

Send data to CPU outside the own station

X_RCV

Receive data from CPU outside the own station.

31.4.2 Parameters for X_SEND

Table 31-4

INPUT	Type	Remark
REQ	BOOL	Trigger send job
CONT	BOOL	Connection after completing the job: hold / disconnect
DEST_ID	WORD	MPI address of communication partner
REQ_ID	DWORD	Job ID for receiver
SD	ANY	Send area
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information
BUSY	BOOL	Job running / job finished (*1)

(*1): Remote confirmation: Application

31.4.3 Parameters for X_RCV

Table 31-5

INPUT	Type	Remark
EN_DT	BOOL	Triggering receive job
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information
REQ_ID	DWORD	Job ID of sender
NDA	BOOL	Data in receive buffer: yes / no
INOUT	**Type**	**Remark**
RD	ANY	Receive area

31.5 User interface X_PUT, X_GET

31.5.1 Description

With the communication blocks a CPU communicates with another CPU outside the own station (client / server communication).

X_PUT

Writing data to CPU outside the own station.

X_GET

Reading data from CPU outside the own station.

31.5.2 Parameters for X_PUT

Table 31-6

INPUT	Type	Remark
REQ	BOOL	Trigger write job
CONT	BOOL	Connection after completing the job: hold / disconnect
DEST_ID	WORD	MPI address of the communication partner
VAR_ADDR	DWORD	Destination area
SD	ANY	Source area
OUTPUT	**Type**	**Remark**
RET_VAL	RET_VAL	Error information
BUSY	BUSY	Job running / job finished (*1)

(*1): Remote confirmation: application

31.5.3 Parameters for X_GET

Table 31-7

INPUT	Data type	Remark
REQ	BOOL	Triggering read job
CONT	BOOL	Connection after completing the job: hold / disconnect
DEST_ID	WORD	MPI address of the communication partner
VAR_ADDR	DWORD	Source area
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information
BUSY	BOOL	Job running / job finished (*1)
INOUT	**Type**	**Remark**
RD	ANY	Destination area (E, A, M, D)

(*1): Remote confirmation: application

31.6 User interface I_PUT, I_GET

31.6.1 Description

With the communication blocks a CPU communicates with another CPU within the own station (client / server communication).

I_PUT

Writing data to CPU within the own station.

I_GET

Reading data from CPU within the own station.

31.6.2 Parameters for I_PUT

Table 31-8

INPUT	Type	Remark
REQ	BOOL	Trigger write job
CONT	BOOL	Connection after completing the job: hold / disconnect
IOID	BYTE	Address area of the partner module (PE, PA)
LADDR	WORD	Logic address of the partner module
VAR_ADDR	ANY	Destination area
SD	ANY	Source area
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information
BUSY	BOOL	Job running / job finished (*1)

(*1): Remote confirmation: application

31.6.3 Parameters for I_GET

Table 31-9

INPUT	Type	Remark
REQ	BOOL	Trigger read job
CONT	BOOL	Connection after completing the job: hold / disconnect
IOID	BYTE	Address area of the partner module (PE, PA)
LADDR	WORD	Logic address of the partner module
VAR_ADDR	ANY	Source area
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Error information
BUSY	BOOL	Job running / job finished (*1)
RD	ANY	Destination area

(*1): Remote confirmation: application

32 S7 Communication

32.1 Characteristics

The S7 communication is characterized by the following characteristics:

- Network-independent user interface:
  Identical handling for PN/IE, PB and MPI
- Communication via integrated interface of the CPU
  or via external interface of CP or CM.
- Communication via configured connections (S7 connection).
- Data volume per communication job: <= 64 Kbytes
- The user program in the sender contains an item of information if the data in the user data area of the receiver have arrived (confirmation by the remote application).

32.2 Details communication type

Table 32 -1 Communication types - detailed

Communication type:		S7 communication		
Protocol:		**S7 protocol**		
General				
Media		MPI, PB, PN/IE, backplane bus (only for SIMATIC S7-400, multicomputing)		
Interfaces		CPU, CP, CM		
Connection	SIMATIC S5	no		
	third-party (open standards)	no		
Protocol				
Dynamic data length		yes		
Multicast / broadcast		no		
Connections	to the remote partner?	yes		
	dynamic / static	static		
User interface				
Communication blocks		**BSEND / BRCV**	**Type "USEND / URCV"**	**Type "PUT, GET"**
Maximal number of data **(*1)**		<= 64 Kbytes	>= 160 bytes	>= 160 bytes
Dynamic addressing of data		S7-300: yes	S7-300: yes	S7-300: yes
		other: no	other: no	other: no
Remote confirmation		Application	Transport	Application
Model		Client / Client	Client / Client	Client / Server

Back to jump distributor PN/IE

Back to jump distributor PB

Back to jump distributor MPI

Back to jump distributor backplane bus

(*1): The maximal data volume per communication job depends on:

- network (MPI, PB, PN/IE)
- Interface (communication via CPU, CP or CM)
- Communication partner (S7-300, S7-400, ...)
- Communication block (BSEND, …)

For some communication blocks it is specified how much data can transferred at a **minimum** (≥ x).
If this minimum number (≥ x) is insufficient, the exact maximal number can be determined via (STEP 7online help).

Table 32-2

Communication block	S7-300		S7-400	S7-1200	S7-1500
	CPU	CP	CPU, CP	CPU, CM	CPU, CP, CM
BSEND / BRCV	= 64 Kbytes	≤ 32 Kbytes	= 64 Kbytes	---	= 64 Kbytes
"USEND / URCV"	≥ 160 bytes	≤ 160 bytes	≥ 440 bytes	---	≥ 920 bytes
"PUT, GET"	≥ 160 bytes	≤ 160 bytes	≥ 400 bytes	≥ 160 bytes	≥ 880 bytes

32.3 Overview of user interfaces

Overview of communication blocks

Table 32-3

Type of the user interface			S7-300 CPU	S7-300 CP	S7-400, WinAC	S7-1200	S7-1500
BSEND/BRCV	**Block:**		**FB12/ FB13**	**FB12 / FB13**	**SFB12 / SFB13**	**---**	**SFB12 / SFB13**
	STEP 7	TIA	BSEND /BRCV	BSEND /BRCV	BSEND /BRCV	---	BSEND /BRCV
	STEP 7	Not TIA	BSEND /BRCV	BSEND /BRCV	BSEND /BRCV	---	---
"USEND/URCV" (*1)	**Block:**		**FB8 / FB9**	**FB8 / FB9**	**SFB8 / SFB9**	**---**	**SFB8 / SFB9**
	STEP 7	TIA	USEND_**S**/ URCV_**S**	USEND_**S**/ URCV_**S**	USEND/ URCV	---	USEND/ URCV
	STEP 7	Not TIA	USEND /URCV	USEND /URCV	USEND /URCV	---	---
	Block:		**FB28 / FB29**	---	---	---	---
	STEP 7	TIA	USEND /URCV	---	---	---	---
	STEP 7	Not TIA	USEND_**E** /URCV_**E**	---	---	---	---
"PUT, GET" (*2)	**Block:**		**FB15, FB14**	**FB15, FB14**	**SFB15, SFB14**	**SFB15, SFB14**	**SFB15, SF 14**
	STEP 7	TIA	PUT_**S**, GET_**S**	PUT_**S**, GET_**S**	PUT, GET	PUT, GET	PUT, GET
	STEP 7	Not TIA	PUT, GET	PUT, GET	PUT, GET	---	---
	Block:		**FB35, FB34**	---	---	---	---
	STEP 7	TIA	PUT, GET	---	---	---	---
	STEP 7	Not TIA	PUT_**E**, GET_**E**	---	---	---	---

(*1): Designation "USEND / URCV" comprises all variants of the blocks (FBx, SFBx) and of the designations in STEP 7 (USEND, USEND_E, USEND_S).

(*2): Designation "PUT, GET" comprises all variants of the blocks (FBx, SFBx) and of the designations in STEP 7 (PUT, PUT_E, PUT_S).

Communication blocks in STEP 7

Table 32-4

Interface		available in STEP 7	
S7-300	CPU	STEP 7 (not TIA)	Library: Standard Library / Communication Blocks
	CP	STEP 7 (not TIA)	Library: SIMATIC_NET_CP / CP300
S7-400, WinAC		STEP 7 (not TIA)	Library: Standard library / System function blocks
S7-1200, S7-1500		STEP 7 (TIA)	Instruction: Communication

32.4 User interface: Type "USEND / URCV"

32.4.1 Preliminary remarks

Designation "USEND / URCV" comprises all variants of the communication blocks (FBx, SFBx) and of the designations in STEP 7 (USEND, USEND_E, USEND_S).

The communication blocks differ in the maximum number of send areas or receive areas.

Table 32-5

		S7-300		**S7-400, WinAC**	**S7-1200, S7-1500**
		CPU, CP	**CPU**		
Maximal number of areas		1	4	4	4
Designation in STEP 7	TIA	USEND_**S**, URCV_**S**	USEND, URCV	USEND, URCV	USEND, URCV
	Not TIA	USEND, URCV	USEND_**E**, URCV_**E**	USEND, URCV	---

32.4.2 Description

The communication block of type "USEND" sends data to a communication block type "URCV". "URCV" receives data from "USEND", and copies it to the configured receive areas.

"USEND" and "URCV" are <u>not</u> coordinated with each other:

- A send job is complete if data has arrived in the receive buffer. The data is then generally not yet in the receive area. For a completed send job it is therefore not ensured that the data has been copied from the receive buffer to the receive area with URCV.
- A new send job ("USEND") can be started, even if the previously sent data has not been copied to the receive area yet by "URCV". Data can therefore be overwritten in the receive buffer unnoticed by the sender. In this case the "URCV", in the receiver, turns out an Overrun warning.

Type "USEND"

Uncoordinated sending of data

Type "URCV"

Uncoordinated receiving of data

32.4.3 Parameters for type "USEND"

Table 32-6

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Trigger send job
ID	WORD	CONN_PRG	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	CONN_R_ID	Assigning the send SFB/FB and the receive SFB/FB. This enables communication of several SFB/FB pairs via the same logic connection.
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
SD_i	ANY	VARIANT	Send area (i=1,2,3,4)

(*1): Remote confirmation: transport

32.4.4 Parameters for type "URCV"

Table 32-7

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
EN_R	BOOL	BOOL	Triggering receive job
ID	WORD	CONN_PRG	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	CONN_R_ID	Assigning the send SFB/FB and the receive SFB/FB. This enables communication of several SFB/FB pairs via the same logic connection.
OUTPUT			**Remark**
NDR	BOOL	BOOL	Data in receive buffer
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
RD_i	ANY	VARIANT	Receive area (i=1,2,3,4)

32.5 User interface: BSEND / BRCV

32.5.1 Description

Communication block BSEND sends data to a BRCV communication block. BRCV receives data from BSEND, and copies it to the configured receive areas.

BSEND and BRCV need to be used in pairs for data transmission.

The data transmission is coordinated as follows:

- A send job is complete if data has arrived in the receive area. For a completed send job it is therefore ensured that the data has been written to the receive area with BRCV.
- A new send job (BSEND) can only be started if the previously sent data has not been entered to the receive area by BRCV.

<u>Notes on the internal mode of operation</u>

The data to be sent is divided into data blocks. Each data block is sent to the communication partner individually (block-oriented sending). After each received data block a confirmation is sent to BSEND, and parameter LEN is updated (block-oriented receiving).

BSEND

Block-oriented sending of data

BRCV

Block-oriented receiving of data

32.5.2 Parameters for BSEND

Table 32-8

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Trigger send job
R	BOOL	BOOL	Abort send job
ID	WORD	CONN_PRG	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	CONN_R_ID	Assigning the send SFB/FB and the receive SFB/FB. This enables communication of several SFB/FB pairs via the same logic connection.
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
SD_1	ANY	VARIANT	Send area
LEN	WORD	WORD	Length of the data to be sent

(*1): Remote confirmation: application

32.5.3 Parameters for BRCV

Table 32-9

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
EN_R	BOOL	BOOL	Triggering receive job
ID	WORD	BOOL	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	CONN_PRG	Assigning the send SFB/FB and the receive SFB/FB. This enables communication of several SFB/FB pairs via the same logic connection.
OUTPUT		CONN_R_ID	**Remark**
NDR	BOOL		Data in the receive area
ERROR	BOOL	BOOL	Error information
STATUS	WORD	BOOL	
IN_OUT		WORD	**Remark**
RD_1	ANY	VARIANT	Receive area
LEN	WORD	WORD	Length of received data

32.6 User interface: Type "PUT, GET"

32.6.1 Preliminary remarks

Designation "PUT, GET" comprises all variants of the communication blocks (FBx, SFBx) and of the designations in STEP 7 (PUT, PUT_E, PUT_S).

The communication blocks differ in the maximum number of send areas or receive areas.

Table 32-10

		S7-300		S7-400, WinAC	S7-1200, S7-1500
		CPU, CP	**CPU**		
Maximal number of areas		1	4	4	4
Designation in STEP 7	TIA	PUT_**S**, GET_**S**	PUT, GET	PUT, GET	PUT, GET
	Not TIA	PUT, GET	PUT_**E**, GET_**E**	PUT, GET	---

32.6.2 Description

With the communication blocks a CPU (client) communicates with another CPU (server).

In the other CPU (server) no communication block is required in the user program (client / server communication).

Type "PUT"

Writing data to CPU

Type "GET"

Reading data from CPU

32.6.3 Parameters for type "PUT"

Table 32-11

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Trigger write job
ID	WORD	CONN_PRG	Reference to the respective connection (from the configured connection in STEP 7)
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
ADDR_i	ANY	REMOTE	Destination area (i=1,2,3,4)
SD_i	ANY	VARIANT	Source area (i=1,2,3,4)

(*1): Remote confirmation: application

32.6.4 Parameters for type "GET"

Table 32-12

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Trigger read job
ID	WORD	CONN_PRG	Reference to the respective connection (from the configured connection in STEP 7)
OUTPUT			**Remark**
NDR	BOOL	BOOL	Job finished (*1)
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
ADDR_i	ANY	REMOTE	Source area (i=1,2,3,4)
RD_i	ANY	VARIANT	Destination area (i=1,2,3,4)

(*1): Remote confirmation: application

33 PROFINET/Industrial Ethernet (PN/IE)

33.1 Characteristics

The communication via PN/IE is characterized by the following characteristics:

- Communication between SIMATIC controllers
- Communication with third-party controllers

33.2 Overview

Via PN/IE the following communication types are possible:

Table 33-1

	Communication type		**Chapt.**
SIMATIC S7-specific communication	S7 communication		32
Open standard	Open communication with send/receive blocks		34
	Open communication with T blocks		35
	PN communication	CBA	36
		PNIO	37

The communication types for the “open standard” are discussed below.

34 PN/IE: Open Communication with Send/Receive Blocks

34.1 Characteristics

Open communication with send/receive blocks is characterized by the following characteristics:

- Open standard (communication with third-party controllers possible)
- Communication via CP
- Communication via configured connections
- Communication via protocols: TCP, ISO on TCP, UDP, ISO
- Data volume per communication job: <= 8 Kbytes
- Conformation from the remote transport system (not for UDP protocol).
- No confirmation from the remote application

Configuration of connections

Connections are configured in STEP 7. Connecting and disconnecting is performed via the operating system. Since the connections are configured, the connection parameters cannot be modified during runtime.

Connection types

Table 34-1

Protocol	Connection types	
TCP	B#16#01, B#16#11	to RFC 793
ISO on TCP	B#16#12	to RFC 1006
UDP	B#16#13	to RFC 768

Exceptions for protocol UDP: a “UDP connection” must be configured in STEP 7. However, an “UDP connection” is not a connection in the sense of the document (chapter 6). For UDP a connection with the remote communication partner is not established.

Characteristics of protocols

Table 34-2

Protocol	Connecting to the remote partner	Transferring length and end of the data
TCP	yes	no (*1)
ISO on TCP	yes	yes
UDP	no	yes

(*1): This is not a problem during sending, since the sender knows how much data it can send. However, the receiver has no option detecting where the data ends

34.2 Details communication type

Table 34-3 Communication types - detailed

Communication type:		Open communication with send/receive blocks (PN/IE network)						
Protocol:		**ISO**		**ISO on TCP**		**TCP**		**UDP**
General								
Interfaces		CP		CP		CP		CP
Connection	SIMATIC S5	yes		yes		yes		yes
	third-party (open standards)	no		yes		yes		yes
Protocol								
Dynamic data length		yes		yes		no		yes
Multicast / broadcast		no		no		no		yes / yes
Connections	to the remote partner?	yes		yes		yes		no
	dynamic / static	static		static		static		static
User interface								
Communication blocks **(*1)**		**AG_xSEND / AG_xREC**	**FETCH, WRITE**	**AG_xSEND / AG_xREC**	**FETCH, WRITE**	**AG_xSEND / AG_xREC**	**FETCH, WRITE**	**AG_xSEND / AG_xREC**
Maximal number of data **(*1)**		<= 8 Kbytes		<= 8 Kbytes		<= 8 Kbytes		<= 2 Kbytes
Dynamic addressing of data		yes	---	yes	---	yes	---	yes
Remote confirmation		Transport	---	Transport	---	Transport	---	no
Model		Client / Client	S7 only Server	Client / Client	S7 only Server	Client / Client	S7 only Server	Client / Client

Back to jump distributor PN/IE

Explanations for the table:

(*1): The maximal data volume depends on:

- communication partner (S7-300, S7-400, ...)
- communication block (AG_SEND, AG_**L**SEND, AG_**S**SEND)
- protocol (TCP, ...)

Overview:

Table 34-4

Communication block	Protocol	S7-300	S7-400	Remark
		CP	**CP**	
AG_SEND / AG_RECV	ISO, TCP, ISO on TCP	8 Kbytes	240 bytes	----
	UDP	2 Kbytes	240 bytes	including IP header and UDP header
AG_**L**SEND / AG_**L**RECV	ISO, TCP, ISO on TCP	---	8 Kbytes	---
	UDP	---	2 Kbytes	including IP header and UDP header
AG_**S**SEND / AG_**S**RECV	ISO, TCP, ISO on TCP	---	1452 bytes	---
	UDP	---	1452 bytes	including IP header and UDP header

34.3 Overview of user interfaces

Send/Receive blocks

Overview of communication blocks:

Table 34-5

Communication block	S7-300	S7-400
	CP	CP
AG_SEND / AG_RECV	FC 5 / FC 6	FC 5 / FC 6
AG_LSEND / AG_LREC	---	FC 50 / FC 60
AG_SSEND / AG_SREC	---	FC 53 / FC 63

Depending on the family (S7-300, S7-400) different communication blocks must be used. The communication blocks are stored in STEP 7 under different libraries.

Communication blocks in STEP 7:

Table 34-6

Interface		available in STEP 7 (not TIA)
S7-300	CP	Library: SIMATIC_NET_CP / CP300
S7-400	CP	Library: SIMATIC_NET_CP / CP400

Server for Fetch/Write

A SIMATIC S7-CP can be server for FETCH/WRITE jobs of another controller (third-party controller, SIMATIC S5). In the S7-CPU of the server no communication blocks are required for the data exchange.

34.4 User interface AG_xSEND, AG_xRECV

For the communication blocks the following abbreviations are used:

- AG_**x**SEND stands for: AG_SEND, AG_LSEND, AG_SSEND
- AG_**x**RECV stands for: AG_RECV, AG_LRECV, AG_SSRECV

Meaning of x = L

The communication blocks are optimized for transferring extensive data (L stands for “long”).

Meaning of x = S

The communication blocks are time-optimized (S stands for “speed”):

- Optimized communication between CPU and CP in the station
- Without effect on the communication via the network.

34.4.1 Description

Communication block AG_xRECV sends data to communication block AG_xRECV. The mode of operation of the communication blocks depends on the used CP (/13/).

AG_xSEND

The communication block transfers data to the CP which are sent via a configured connection.

AG_xRECV

The communication block receives data from the CP which were received via a configured connection.

34.4.2 Parameters for AG_SEND, AG_LSEND, AG_SSEND

Table 34-7

INPUT	Type	Remark
ACT	BOOL	Trigger send job
ID	INT	Reference to the respective connection (from the configured connection in STEP 7)
LADDR	WORD	Address of the module (from the hardware configuration in STEP 7)
SEND	ANY	Send area
LEN	INT	Length of the data to be sent
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Case discrimination:

Different meaning depending on the used protocol:

- data lies on the remote CP: ISO, ISO on TCP, TCP
- data was sent by the local CP: UDP

34.4.3 Parameters for AG_RECV, AG_LRECV, AG_SSRECV

Table 34-8

INPUT	Type	Remark
ID	INT	Reference to the respective connection (from the configured connection in STEP 7)
LADDR	WORD	Address of the module (from the hardware configuration in STEP 7)
RECV	ANY	Receive area
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data in the receive area
LEN	INT	Length of received data
ERROR	BOOL	Error information
STATUS	WORD	

(*1): data was copied from the receive buffer (CP) into the receive area (CPU).

<u>Special case TCP connection:</u>

Here, the length specified in the RECV parameter rules.

A receive job is completed with the NDR=1 display as soon as a data volume equal to the specified length has been written to the receive area. This means, NDR is only set if the receive area has been filled up completely. LEN always shows the length of the receive area.

34.5 User interface FETCH, WRITE (Server)

In the S7-CPU of the server no communication blocks are required for the data exchange.

FETCH

The connection partner (third-party controller, SIMATIC S5) has read access to the data in the S7-CPU (server) (fetch data).

WRITE

The connection partner (third-party controller, SIMATIC S5) has write access to the data in the SIMATIC S7 (write data).

Data

The following data can be accessed in the S7-CPU (server):

- data blocks
- flags
- process image of inputs and outputs
- distributed I/O
- counter, times

Connection types

For access with FETCH or WRITE a connection in the S7-CPU (server) must be configured to FETCH passive or WRITE passive mode.

The following connection types are possible: ISO, ISO on TCP, TCP

Access coordination via the STEP 7 user program

The blocks (FC) AG_LOCK and AG_UNLOCK are available for the access coordination. With these blocks the access to data can be coordinated by blocking or enabling the connections.

35 PN/IE: Open Communication with “T Blocks”

35.1 Characteristics

The open communication with T-blocks is characterized by the following characteristics:

- Open standard (communication with third-party controllers possible)
- Communication via
  - configured or programmed connections: TCP, IoT, UDP
  - configured connections: ISO
- Communication via protocols: TCP, ISO on TCP, UDP, ISO
- Data volume per communication job: <= 64 Kbytes
- Conformation from the remote transport system (not for UDP protocol).
- No confirmation from the remote application.

Programming of connections

The connecting and disconnecting process is programmed in the STEP 7 user program. There are two options:

- calling the connection blocks (TCON, ...)
- calling communication blocks with integrated connecting or disconnecting feature (TSEND_C, ...)

Due to the programming of the connections the connection parameters can be modified during runtime (in RUN of the CPU).

Connection types

Table 35-1

Protocol	Connection types	
TCP	B#16#01, B#16#11	to RFC 793
ISO on TCP	B#16#12	to RFC 1006
UDP	B#16#13	to RFC 768

Exceptions for protocol UDP: here, the local communication access point is parameterized with the connection blocks. For UDP, a connection with the remote communication partner is not established.

Characteristics of protocols

Table 35-2

Protocol	Connecting to the remote partner	Transferring length and end of the data
TCP	yes	no (*1)
ISO on TCP	yes	yes
UDP	no	yes

(*1): This is not a problem during sending, since the sender knows how much data it can send. However, the receiver has no option detecting where the data ends.

35.2 Details communication type

Table 35 -3 Communication types - detailed

Communication type:		Open communication with T-blocks (PN/IE network)		
	Protocol:	**ISO on TCP**	**TCP**	**UDP**
General				
Interfaces		CPU, CP, CM	CPU, CP, CM	CPU, CP, CM
Connection	SIMATIC S5	yes	yes	no
	third-party (open standards)	yes	yes	yes
Protocol				
Dynamic data length		yes	no	yes
Multicast / broadcast		no	no	no
Connections	to the remote partner?	yes	yes	no
	dynamic / static	TSEND/TRCV: dynamic + static TSEND_C/TRCV_C: dynamic	TSEND/TRCV: dynamic + static TSEND_C/TRCV_C: dynamic	dynamic + static
User interface				
Communication blocks		**TSEND / TRCV** **TSEND_C / TRCV_C**	**TSEND / TRCV** **TSEND_C / TRCV_C**	**TUSEND / TURCV**
Maximal number of data **(*1)**		<= 64 Kbytes	<= 64 Kbytes	= 1472 bytes
Dynamic addressing of data		yes	yes	yes
Remote confirmation		Transport	Transport	no
Model		Client / Client	Client / Client	Client / Client

Back to jump distributor PN/IE

Explanations for the table:

(*1): The maximal data volume depends on:

- Communication partner (S7-300, S7-400, ...) and used interface (CPU, CP, CM)
- Protocol (TCP, …) and connection type

Overview:

Table 35-4

Protocol	Connection type	S7-1200	S7-1500		S7-300	S7-400		WinAC RTX 2010
		CPU, CM	**CPU, CM**	**CP**	**CPU**	**CPU**	**CP**	**CPU / CP (submodule)**
TCP	B#16#01	---	---	---	1460 bytes	---	---	---
	B#16#11	8 Kbytes	64 Kbytes	64 Kbytes	32 Kbytes	32 Kbytes	---	64 Kbytes
ISO on TCP	B#16#12	8 Kbytes	64 Kbytes	64 Kbytes	32 Kbytes	32 Kbytes	1452 bytes	64 Kbytes
UDP	B#16#13	---	1472 **bytes**	1472 bytes	1472 bytes	1472 bytes	---	1472 bytes

35.3 Overview of user interfaces

T-blocks

Overview of communication blocks

Table 35-5

Communication block	S7-300	S7-400	S7-1200	S7-1500
	CPU	CPU, CP		
TSEND / TRCV	FB 63 / FB 64	FB 63 / FB 64	SFB 100 / SFB 101	SFB 150 / SFB 151
TUSEND / TURCV	FB 67 / FB 68	FB 67 / FB 68	SFB 100 / SFB 101	SFB 150 / SFB 151
TSEND_C / TRCV_C	---	---	FB 1030 / FB 1031	FB 1030 / FB 1031

Overview of communication blocks:

Table 35-6

Connection block	S7-300	S7-400	S7-1200	S7-1500
	CPU	CPU, CP		
TCON	FB 65	FB 65	SFB 102	SFB 109
TDISCON	FB 66	FB 66	SFB 103	SFB 103

Communication blocks in STEP 7:

Table 35-7

Interfaces	available in STEP 7	
S7-300, S7-400	STEP 7 (not TIA)	Library: Standard Library / Communication Blocks
S7-1200, S7-1500	STEP 7 (TIA)	Instruction: communication

Server for Fetch/Write

Overview: Communication blocks

Table 35-8

Communication block	S7-300	S7-400
	CPU	CPU
FW_TCP	FB 210	FB 210
FW_IOT	FB 220	FB 220

Communication blocks in STEP 7:

Table 35-9

Interfaces		available in STEP 7	
S7-300, S7-400	CPU	STEP 7 (not TIA)	Library: Standard Library / Communication Blocks

Note: A SIMATIC S7-CPU can be server for FETCH/WRITE jobs of another controller (third-party controller, SIMATIC S5).

The communication blocks of the server internally use the T-blocks:

- TSEND / TRCV (sending and receiving of data)
- TCON, TDISCON (connecting and disconnecting)

35.4 User interface TSEND / TRCV

35.4.1 Description

Communication block **TSEND** sends data to communication block **TRCV**. Data can be transmitted via configured connections or via programmed connections (chapter 6).

35.4.2 Parameters for TSEND

Table 35-10

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Trigger send job
ID	WORD	CONN_OUC	Reference to the respective connection
LEN	INT	UINT	Length of the data to be sent
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
DATA	ANY	VARIANT	Send area

(*1): Remote confirmation: transport

35.4.3 Parameters for TRCV

Table 35-11

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
EN_R	BOOL	BOOL	Trigger: receive job
ID	WORD	CONN_OUC	Reference to the respective connection
LEN	INT	UINT	Length of the receive area
OUTPUT			**Remark**
NDR	BOOL	BOOL	Data in the receive area
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error display
STATUS	WORD	WORD	
RCVD_LEN	INT	UINT	Volume of actually received data.
IN_OUT			**Remark**
DATA	ANY	VARIANT	Receive area

35.5 User interface TUSEND / TURCV

35.5.1 Description

Communication block **TUSEND** sends data to communication block **TURCV**. The data is transferred via programmed with connections (calling connection blocks). Here, only the UDP protocol (unacknowledged datagram service) is possible.

35.5.2 Parameters for TUSEND

Table 35-12

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Trigger send job
ID	WORD	WORD	Reference to the respective connection
LEN	INT	UINT	Length of the data to be sent
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
DATA	ANY	VARIANT	Send area
ADDR	ANY	TADDR_Param	Address of the receiver

(*1): Remote confirmation: none

35.5.3 Parameters for TURCV

Table 35-13

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
EN_R	BOOL	BOOL	Triggering receive job
ID	WORD	WORD	Reference to the respective connection
LEN	INT	UINT	Length of the receive area
OUTPUT			**Remark**
NDR	BOOL	BOOL	Data in the receive area
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
RCVD_LEN	INT	UINT	Volume of actually received data.
IN_OUT			**Remark**
DATA	ANY	VARIANT	Receive area
ADDR	ANY	TADDR_Param	Address of the sender

35.6 User interface for connection blocks

35.6.1 Description

Connection block TCON is used for establishing a connection, connection block TDISCON for terminating a connection.

35.6.2 Parameters for TCON

Table 35-14

INPUT	Type			Remark
	S7-300,S 7-400	S7-1200	S7-1500	
REQ	BOOL	BOOL	BOOL	Triggering the connecting process
ID	WORD	CONN_UOC	CONN_UOC	Reference to the connection
OUTPUT				**Remark**
DONE	BOOL	BOOL	BOOL	Job finished
BUSY	BOOL	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	BOOL	Error information
STATUS	WORD	WORD	WORD	
IN_OUT				**Remark**
CONNECT	ANY	TCON_Param	VARIANT	Connection description

35.6.3 Parameters for TDISCON

Table 35-15

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Triggering the disconnecting process
ID	WORD	CONN_OUC	Reference to the connection
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	

35.7 User interface TSEND_C / TRCV_C

To distinguish them from the other T-blocks these blocks are referred to as T-compact blocks below.

35.7.1 Description

The communication block TSEND_C sends data to the communication block TRCV_C. Data can be transmitted via configured connections or via programmed connections (chapter 6).

TSEND_C

Establishes a connection with the partner, sends data and can also terminate the connection again.

TSEND_C connects the functions of TCON, TDISCON and TSEND.

TRCV_C

Establishes a connection with the partner, receives data and can also terminate the connection again.

TRCV_C connects the functions of TCON, TDISCON and TRCV.

35.7.2 Parameters for TSEND_C

Table 35-16

INPUT	Type		Remark
	S7-1200	S7-1500	
REQ	BOOL	BOOL	Trigger send job
CONT	BOOL	BOOL	Connecting / disconnecting process
LEN	UINT	UINT	Length of the byte to be sent
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
DATA	VARIANT	VARIANT	Send area
ADDR	---	VARIANT	Optional parameter (only S7-1500): for the UDP protocol it points to the address of the receiver.
CONNECT	TCON_Param	VARIANT	Connection description
COM_RST	BOOL	BOOL	New start of the block

(*1): Remote confirmation: transport

35.7.3 Parameters for TRCV_C

Table 35-17

INPUT	Type		Remark
	S7-1200	S7-1500	
EN_R	BOOL	BOOL	Trigger: receive job
CONT	BOOL	BOOL	Connecting / disconnecting process
LEN	UINT	UINT	Length of the receive area
OUTPUT			**Remark**
DONE	BOOL	BOOL	Data in the receive area
BUSY	BOOL	BOOL	Job is processed
RCVD_LEN	UINT	UINT	Volume of received data
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
RCVD_LEN	INT	INT	Volume of actually received data.
IN_OUT			**Remark**
DATA	VARIANT	VARIANT	Receive area
ADDR	---	VARIANT	Optional parameter (only S7-1500): for the UDP protocol it points to the address of the sender.
CONNECT	TCON_Param	VARIANT	Connection description
COM_RST	BOOL	BOOL	New start of the block

35.8 User interface FETCH, WRITE (Server)

35.8.1 Description

A SIMATIC S7-CPU can be server for FETCH/WRITE jobs of another controller (third-party controller, SIMATIC S5).

The communication blocks of the server internally use the T-blocks:

- TSEND / TRCV (sending and receiving of data)
- TCON, TDISCON (connecting and disconnecting)

FW_TCP

Communication block for server, used protocol is TCP

FW_IOT

Communication block for server, used protocol is ISO on TCP

35.8.2 Parameters for FW_TCP

Table 35-18

INPUT	Type	Remark
ENABLE	BOOL	Connecting and disconnecting process
CONNECT	ANY	Description of TCP connection
ADDRMODE	INT	Addressing mode S5 or S7
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data of the WRITE job were adopted
ERROR	BOOL	Error display
MODE	BYTE	Performing FETCH or WRITE job
STATUS	WORD	Status display

35.8.3 Parameters for FW_IOT

Table 35-19

INPUT	Type	Remark
ENABLE	BOOL	Connecting and disconnecting process
CONNECT	ANY	Description of IoT connection
ADDRMODE	INT	Addressing mode S5 or S7
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data of the WRITE job were adopted
ERROR	BOOL	Error display
MODE	BYTE	Performing FETCH or WRITE job
STATUS	WORD	Status display

36 PN/IE: CBA

The following abbreviations are used in the document:

- CBA for PROFINET CBA
- PNIO for PROFINET IO

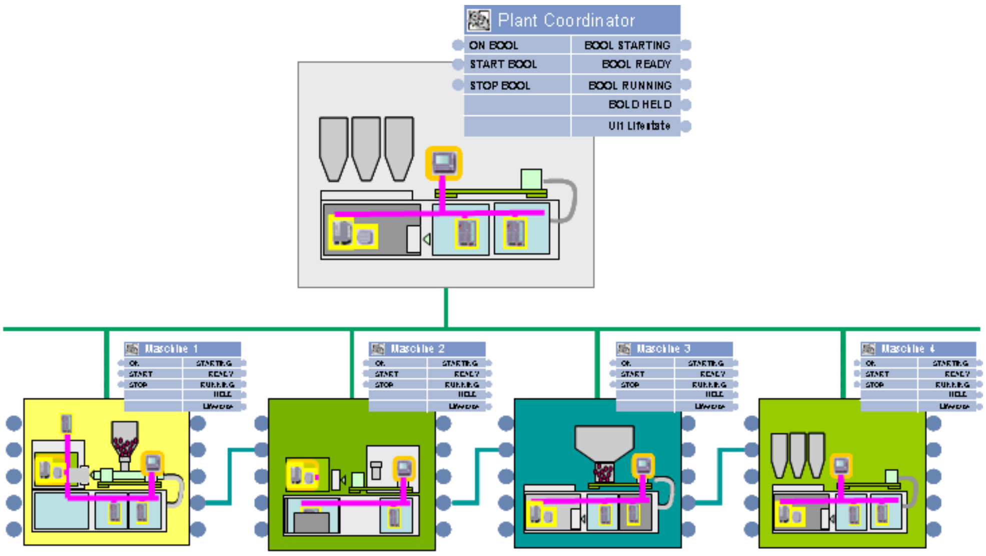
36.1 Characteristics

CBA

CBA (Component Based Automation) is an automation concept for realizing modular applications on the basis of the open PROFINET standard:

- Simple modularization of plants and production lines through decentralized intelligence. Modularization occurs with PROFINET components.
- Machine-machine communication along the production line
- Graphic configuration of the communication

Figure 36-1

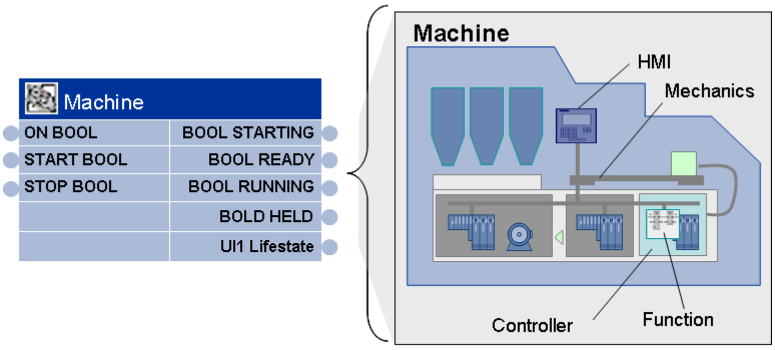


PROFINET component

A PROFINET component is a reusable functional unit:

- encapsulation of the automation functionality in a software program
- unique component interfaces for the data exchange with other components.

Figure 36-2



Characteristics

- Graphic configuration of the communication (configuring instead of programming)
- The performance of the communication can be calculated offline.
- Cross-vendor communication

CBA and PNIO

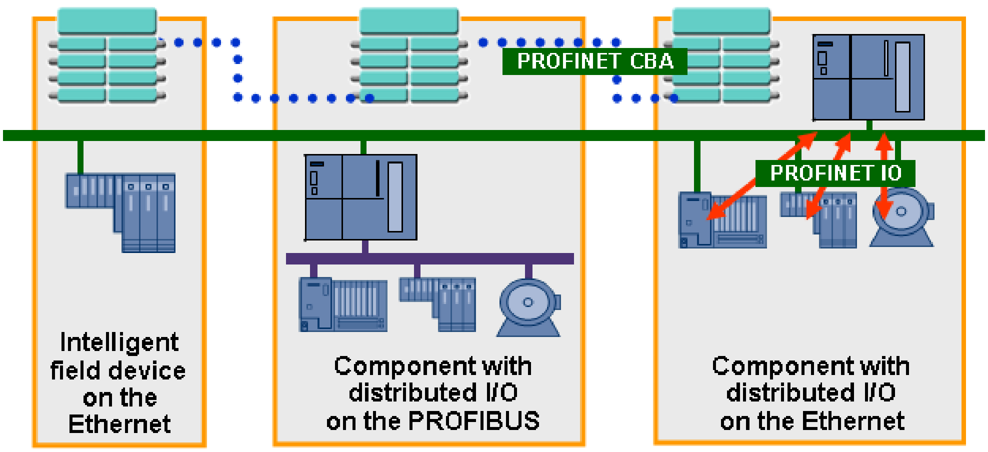
CBA (PROFINET CBA):

- Cyclic and acyclic data exchange between controllers

PNIO (PROFINET IO):

- Cyclic data exchange between a PN controller and the respective PN devices.

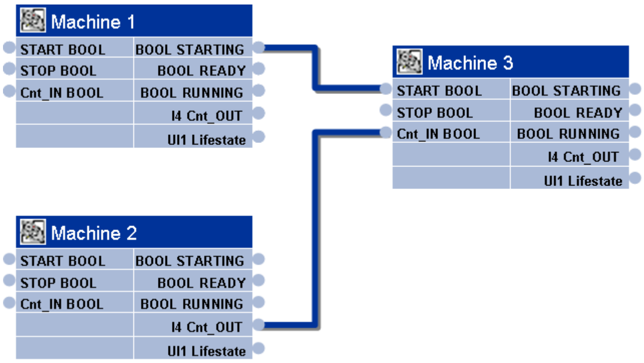
Figure 36-3



36.2 User interfaces

The communication between PROFINET components occurs via graphic interconnection. Therefore it is not required to program anything in the STEP 7 user program for the communication, which makes user interfaces unnecessary.

Figure 36-4



37 PN/IE: PNIO

The following abbreviations are used in the document:

- CBA for PROFINET CBA
- PNIO for PROFINET IO

37.1 Characteristics

Preliminary remarks

The communication with PNIO is a special case of CPU-CPU communication.

Here, the communication mechanisms of the distributed I/O are used for CPU-CPU communication:

- one CPU is plugged in at the central station
- the other CPU is plugged in at the decentralized station

Please refer to the functional model in chapter 5.4:

Characteristics

The communication with PNIO is characterized by the following characteristics:

- Cyclic exchange of data between IO controller and IO device via the PROFINET IO protocol:
  - IO controller sends data to IO device
  - IO device sends data to IO controller
- The data exchange is performed consistent over the entire length (system-related data consistency)

37.2 Communication type - detailed

Table 37 -1 Communication types - detailed

Communication type:		PNIO (PN/IE network)
Protocol:		PN
General		
Interfaces		CPU, CP, CM
Connection	SIMATICS5	yes
	third-party (open standards)	yes
Protocol		
Dynamic data length		no
Multicast / broadcast		no
Connections	to the remote partner?	no
	dynamic / static	---
User interface		
Communication blocks		Load commands / transfer commands DPRD_DAT, DPWR_DAT PNIO_SEND, PNIO_RECV (only S7-300 CP)
Maximal data volume		See technical data of the CPU
Dynamic addressing of data		no
Remote confirmation		Application
Model		Consumer / Provider

Back to jump distributor PN/IE

37.3 Overview of user interfaces

Overview:

Table 37-2

User interface	S7-300		S7-400	S7-1200	S7-1500
	CPU	CP	CPU, CP		
PNIO_SEND	---	FC 11 (*3)	---	---	---
PNIO_RECV	---	FC 12 (*3)	---	---	---
DPRD_DAT	SFC 14 (*2)	---	SFC 14 (*2)	(*1)	(*1)
DPWR_DAT	SFC 15 (*2)	---	SFC 15 (*2)	(*1)	(*1)

The user interface is available in STEP 7:

(*1): STEP 7 (TIA): Expanded instruction

(*2): STEP 7 (not TIA): Library: Standard library / System function blocks

(*3): STEP 7 (not TIA): Library: SIMATIC_NET_CP / CP300

37.4 User interface PNIO_SEND, PNIO_RECV

37.4.1 Description

The communication blocks are used for communication between local CPU and local CP. The CP is IO controller or IO device

PNIO_SEND

The communication block transfers data to the CP.

Case discrimination:

<u>CP is IO controller:</u>

The data transferred to the CP with PNIO_SEND,
are sent to the IO devices by the CP.

<u>CP is IO device:</u>

The data transferred to the CP with PNIO_SEND,
are sent to an IO controller.

PNIO_RECV

The communication block receives on data from the CP.

Case discrimination:

<u>CP is IO controller:</u>

The data received by the CP with PNIO_RECV,
were sent to the CP by the IO devices.

<u>CP is IO device:</u>

The data received by the CP with PNIO_RECV,
were sent to the CP by an IO controller.

37.4.2 Parameters for PNIO_SEND

Table 37-3

INPUT	Type	Remark
CPLADDR	WORD	Configured start address of the CP
MODE	BYTE	CP is IO controller or IO device
LEN	INT	Length of the data to be sent
OUTPUT	**Type**	**Remark**
CHECK_IOCS	BOOL	IOCS status area
IOCS	ANY	
ERROR	BOOL	Error information
STATUS	WORD	
DONE	BOOL	Display whether job was executed without errors (*1).
IN_OUT	**Type**	**Remark**
SEND	ANY	Send area

(*1): Remote confirmation: application

37.4.3 Parameters for PNIO_RECV

Table 37-4

INPUT	Type	Remark
CPLADDR	WORD	Configured start address of the CP
MODE	BYTE	CP is IO controller or IO device
LEN	INT	Length of data to be received
OUTPUT	**Type**	**Remark**
CHECK_IOPS	BOOL	IOCS status area
IOPS	ANY	
NDR	BOOL	Display whether job was executed without errors (*1).
ERROR	BOOL	Error information
STATUS	WORD	
ADD_INFO	WORD	
IN_OUT	**Type**	**Remark**
RECV	ANY	Receive area

(*1): Remote confirmation: application

37.5 User interface DPRD_DAT, DPWR_DAT

37.5.1 Description

The user interfaces give an IO controller access to the data of an IO device.

DPRD_DAT

Reading consistent data of an IO device

DPWR_DAT

Writing consistent data to IO device

37.5.2 Parameters for DPRD_DAT

Table 37-5

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
LADDR	WORD	HW_IO	Configured start address in the input area to be read from
OUTPUT			**Remark**
RET_VAL	INT	DINT, INT, LREAL, REAL	Display whether job was executed without errors (*1).
RECORD	ANY	VARIANT	Destination area

(*1): Remote confirmation: application

37.5.3 Parameters for DPWR_DAT

Table 37-6

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
LADDR	WORD	HW_IO	Configured start address in the output area to be written to
RECORD	ANY	VARIANT	Source area
OUTPUT			**Remark**
RET_VAL	INT	DINT, INT, LREAL, REAL	Display whether job was executed without errors (*1).

(*1): Remote confirmation: application

38 PROFIBUS (PB)

38.1 Characteristics

The communication via PROFIBUS is characterized by the following characteristics:

- Communication between SIMATIC controllers
- Communication with third-party controllers

38.2 Overview

PB enables the following communication types:

Table 38-1

	Communication type	Chapt.
SIMATIC S7-specific communication	S7 basic communication	31
	S7 communication	32
Open standard	Open communication with send/receive blocks	39
	FMS communication	40
	DP communication	41

The communication types for the “open standard” are discussed below.

39 PB: Open Communication with Send/Receive Blocks

39.1 Characteristics

Open communication with send/receive blocks is characterized by the following characteristics:

- Open standard (communication with third-party controllers possible)
- Communication via CP
- Communication via protocol: FDL
- Communication via configured connections
- Data volume per communication job: <= 240 bytes
- Confirmation from remote transport system
- No confirmation from the remote application

FDL connections

During configuration the following properties can be assigned to an FDL connection:

- Unspecified
- Specified
- Multicast / broadcast

Unspecified

An unspecified FDL connection can be used in 2 ways:

- Connection with a station in another STEP 7 project
- Open Layer 2 access

Open Layer 2 access

The remote communication partner is not established during configuration but in the STEP 7 user program.

This requires installing a job header (4 bytes) in the data. The job header specifies the address of the destination station, and the service (SDA or SDN).

Specified

For a specified FDL connection the remote communication partner is defined during the configuration.

The data has no job header.

Multicast / broadcast

The data contains a job header.

The job header has no function.

39.2 Communication type - detailed

Table 39 -1 Communication types - detailed

Communication type:		Open communication with send/receive blocks (PB network)
Protocol:		FDL
General		
Interfaces		CP
Connection	SIMATIC S5	yes
	third-party (open standards)	yes
Protocol		
Dynamic data length		yes
Multicast / broadcast		yes / yes
Connections	to the remote partner?	yes
	dynamic / static	static
User interface		
Communication blocks		AG_SEND / AG_REC, AG_LSEND / AG_LREC
Maximal data volume		= 240 bytes (includes the possibly contained job header (4 bytes)).
Dynamic addressing of data		yes
Remote confirmation		Transport
Model		Client / Client

Back to jump distributor PB

39.3 Overview of user interfaces

Overview of communication blocks:

Table 39-2

Communication block	S7-300	S7-400
	CP	CP
AG_SEND / AG_RECV	FC 5 / FC 6	FC 5 / FC 6
AG_LSEND / AG_LREC (*1)	------	FC 50 / FC 60

(*1): AG_LSEND / AG_LREC can be used, however, it has no other function like AG_SEND / AG_RECV

Depending on the family (S7-300, S7-400), different communication blocks must be used. The communication blocks are stored in STEP 7 under different libraries.

Communication blocks in STEP 7:

Table 39-3

Interface	available in STEP 7	
S7-300, CP	STEP 7 (not TIA)	Library: SIMATIC_NET_CP / CP300
S7-400, CP	STEP 7 (not TIA)	Library: SIMATIC_NET_CP / CP400

39.4 User interface: AG_xSEND, AG_xRECV

For the names of the communication blocks the following abbreviations are used:

- AG_**x**SEND stands for: AG_SEND, AG_LSEND
- AG_**x**RECV stands for: AG_RECV, AG_LRECV

Meaning of x = L

The communication blocks are optimized for transferring extensive data (L stands for “long”).

39.4.1 Description

Communication block AG_xRECV sends data to communication block AG_xRECV. The mode of operation of the communication blocks depends on the used CP (/13/).

AG_xSEND

The communication block transfers data to the CP which are sent via a configured connection.

AG_xRECV

The communication block receives data from the CP which were received via a configured connection.

39.4.2 Parameters for AG_SEND, AG_LSEND

Table 39-4

INPUT	Type	Remark
ACT	BOOL	Trigger send job
ID	INT	Reference to the respective connection (from the configured connection in STEP 7)
LADDR	WORD	Address of the module (from the hardware configuration in STEP 7)
SEND	ANY	Send area
LEN	INT	Length of the data to be sent
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): different meaning, depending on the FDL connection type:

- data lies on the remote CP: specified
- data was sent by the local CP:
  - broadcast / multicast
  - unspecified and SDN service

39.4.3 Parameters for AG_RECV, AG_LRECV

Table 39-5

INPUT	Type	Remark
ID	INT	Reference to the respective connection (from the configured connection in STEP 7)
LADDR	WORD	Address of the module (from the hardware configuration in STEP 7)
RECV	ANY	Receive area
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data in the receive area
LEN	INT	Length of received data
ERROR	BOOL	Error information
STATUS	WORD	

(*1): data was copied from the receive buffer (CP) into the receive area (CPU).

40 PB: FMS Communication

40.1 Characteristics

The FMS communication is characterized by the following characteristics:

- Open standard (communication with third-party controllers possible)
- Data is transferred in a device-neutral form (FMS variable). Conversion of FMS variables into device-specific form, and vice versa, occurs in the communication partners.
- Additional configuration workload for defining the FMS variables
- Communication via CP
- Communication via connections
- Connections are configured

40.2 Communication type - detailed

Table 40 -1 Communication types - detailed

	Communication type:	FMS communication (PB network)	
	Protocol:	FMS	
General			
Interfaces		CP	
Connection	SIMATIC S5	yes	
	third-party (open standards)	yes	
Protocol			
Dynamic data length		yes	
Multicast / broadcast		no / yes	
Connections	to the remote partner?	yes	
	dynamic / static	static	
User interface			
Communication blocks		READ, WRITE	REPORT
Maximal number of data **(*1)**		READ <= 237 bytes WRITE <= 233 bytes	<= 233 bytes
Dynamic addressing of data		yes	yes
Remote confirmation		Application	no
Model		Client / Server	Client / Server

Back to jump distributor PB

Explanations for the table:
(*1): Contains information which describes the data (FMS variable) (/12/).

40.3 Overview of user interfaces

Overview: Communication blocks

Table 40-2

Communication block	S7-300	S7-400
	CP	CP
READ	FB 3	FB 3
WRITE	FB 6	FB 6
REPORT	FB 4	FB 4

Depending on the family (S7-300, S7-400), different communication blocks must be used. The communication blocks are stored in STEP 7 under different libraries.

Communication blocks in STEP 7:

Table 40-3

Interface	available in STEP 7	
S7-300, CP	STEP 7 (not TIA)	Library: SIMATIC_NET_CP / CP300
S7-400, CP	STEP 7 (not TIA)	Library: SIMATIC_NET_CP / CP400

40.4 User interface: READ, WRITE, REPORT

40.4.1 Description

READ Reading the variable

With the communication block, data is read from the remote communication partner.

The structure description of the FMS variable lies in the remote communication partner (FMS server). When establishing the FMS connection, the local communication partner reads the structure description from the remote communication partner, which the local communication partner uses to convert the data accordingly.

WRITE Writing the variable

With the communication block data is written to the remote communication partner.

The structure description of the FMS variable lies in the remote communication partner (FMS server). When establishing the FMS connection, the local communication partner reads the structure description from the remote communication partner, which the local communication partner uses to convert the data accordingly.

REPORT Reporting the variable

The communication block enables unconfirmed transmission of variables to an FMS client. The communication block is also used for transferring the broadcast to FMS connections.

40.4.2 Parameters for READ

Table 40-4

INPUT	Type	Remark
REQ	BOOL	Trigger read job
ID	DWORD	Reference to the respective connection (from the configured connection in STEP 7)
VAR_1	ANY	Variable to be read remotely
RD_1	ANY	Destination area
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data in the destination area (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: Application

40.4.3 Parameters for WRITE

Table 40-5

INPUT	Type	Remark
REQ	BOOL	Trigger write job
ID	DWORD	Reference to the respective connection (from the configured connection in STEP 7)
VAR_1	ANY	Variable to be written remotely
SD_1	ANY	Source area
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job is processed / Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: application

40.4.4 Parameters for REPORT

Table 40-6

INPUT	Data type	Remark
REQ	BOOL	Triggering report job
ID	DWORD	Reference to the respective connection (from the configured connection in STEP 7)
SD_1	ANY	Source area local
VAR_1	ANY	Variable to be written remotely (*2)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job is processed / Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: none

41 PB: DP Communication

41.1 Characteristics

Preliminary remarks

DP communication is a special case of CPU-CPU communication.

Here, the communication mechanisms of the distributed I/O are used for CPU-CPU communication:

- one CPU is plugged in at the central station
- the other CPU is plugged in at the decentralized station

Please refer to the functional model in chapter 5.4:

Characteristics

The DP communication is characterized by the following characteristics:

- Cyclic exchange of data between IO master
  and IO slave via the PROFIBUS IO protocol:
  - DB master reads data from DP slaves
  - DB master writes data to DP slaves
- Data exchange is performed consistently over the entire length
  (system-related data consistency)

41.2 Communication type - detailed

Table 41 -1 Communication types - detailed

Communication type:		DP communication (PB network)
Protocol:		DP
General		
Interfaces		CPU, CP, CM
Connection	SIMATIC S5	yes
	third-party (open standards)	yes
Protocol		
Dynamic data length		no
Multicast / broadcast		no
Connections	to the remote partner?	no
	dynamic / static	---
User interface		
Communication blocks		Load commands / transfer commands DPRD_DAT, DPWR_DAT DP_SEND, DP_RECV (only S7-300 CP)
Maximal data volume		See technical data of the CPU
Dynamic addressing of data		no
Remote confirmation		Application
Model		Master / slave

Back to jump distributor PB

41.3 Overview of user interfaces

Overview:

Table 41-2

User interface	S7-300		S7-400	S7-1200	S7-1500
	CPU	CP	CPU, CP		
DP_SEND	---	FC 1 (*3)	---	---	---
DP_RECV	---	FC 2 (*3)	---	---	---
DPRD_DAT	SFC 14 (*2)	---	SFC 14 (*2)	(*1)	(*1)
DPWR_DAT	SFC 15 (*2)	---	SFC 15 (*2)	(*1)	(*1)

The user interface is available in STEP 7:

(*1): STEP 7 (TIA) : Expanded instruction

(*2): STEP 7 (not TIA): Library: Standard library / System function blocks

(*3): STEP 7 (not TIA): Library: SIMATIC_NET_CP / CP300

41.4 User interface DP_SEND, DP_RECV

41.4.1 Description

The communication blocks are used for communication between local CPU and local CP. The CP is DP master or DP slave.

DP_SEND

The communication block transfers data to the CP.

Case discrimination:

CP is DP master:

The data transferred to the CP with DP_SEND is written to the DP slaves cyclically.

CP is DP slave:

The data transferred to the CP with DP_SEND are read from the CP cyclically by a DP master.

DP_RECV

The communication block receives on data from the CP.

Case discrimination:

CP is DP master:

The data received by the CP with DP_RECV,
were read cyclically from the DP slaves by the CP.

CP is DP slave:

The data received by the CP with DP_RECV,
were written to the CP cyclically by a DP master.

41.4.2 Parameters for DP_SEND

Table 41-3

INPUT	Type	Remark
CPLADDR	WORD	Configured start address of the CP
SEND	ANY	Send area
OUTPUT	**Type**	**Remark**
DONE	BOOL	Display whether job was executed without errors (*1).
ERROR	BOOL	Error display
STATUS	WORD	Status display

(*1): Remote confirmation: application

41.4.3 Parameters for DP_RECV

Table 41-4

INPUT	Type	Remark
CPLADDR	WORD	Configured start address of the CP
RECV	ANY	Receive area
OUTPUT	**Type**	**Remark**
NDR	BOOL	Display whether job was executed without errors (*1).
ERROR	BOOL	Error display
STATUS	WORD	Status display
DPSTATUS	BYTE	Status display

(*1): Remote confirmation: application

41.5 User interface DPRD_DAT, DPWR_DAT

41.5.1 Description

The communication blocks give the DP master access to the data of a DP slave.

DPRD_DAT

Reading consistent data of a DP standard slave

DPWR_DAT

Writing consistent data to DP standard slave

41.5.2 Parameters for DPRD_DAT

Table 41-5

INPUT	Type	Remark
LADDR	WORD	Configured start address in the receive area to be read from
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Display whether job was executed without errors (*1).
RECORD	ANY	Destination area

(*1): Remote confirmation: application

41.5.3 Parameters for DPWR_DAT

Table 41-6

INPUT	Type	Remark
LADDR	WORD	Configured start address in the output area to be written to
RECORD	ANY	Source area
OUTPUT	**Type**	**Remark**
RET_VAL	INT	Display whether job was executed without errors (*1).

(*1): Remote confirmation: application

42 Serial Interface

Note

Connecting to controllers with the serial Modbus interface (RTU format) is described in Part 4 (chapter 58):

42.1 Characteristics

The communication via a serial interface is characterized by the following characteristics:

- simple option of a CPU-CPU coupling with nodes (point-to-point link)
- multipoint coupling is also possible (for RS 422/485)

42.2 Comparing the protocols: *ASCII* / 3964(R) / RK 512

42.2.1 Delimitation

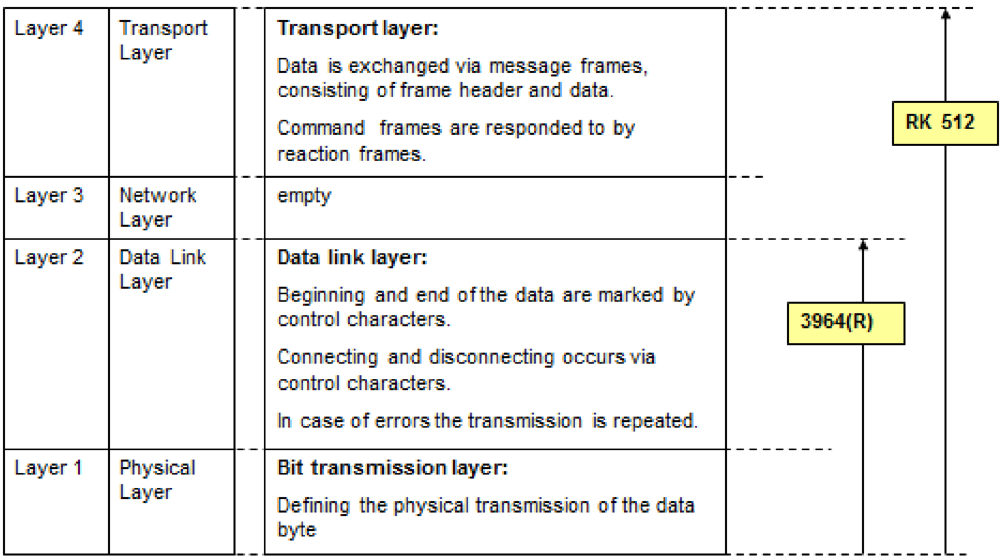
The communication types can be classified best by using the ISO/OSI reference model:

Table 42-1

Communication type	ISO/OSI reference model	Transmission security in comparison
ASCII	uses only layer 1	---
3964(R)	uses layer 1 and 2	higher as for *ASCII*
RK 512	uses layer 1, 2 and 4 layer 1 and 2 correspond to 3964(R)	higher as for 3964(R)

The picture shows the ISO/OSI reference model:

Figure 42-1



42.2.2 *ASCII* characteristics

Functionality

The receiver detects the end of the data transmission ("end of the data") via a configurable end criterion (end of character delay time, receiving end characters, receiving fixed data volume).

Code transparency

A protocol is code transparent if any character (00H to FFH) can occur in the data to be transferred.

In the following cases, *ASCII* is not code transparent:

- using the flow control
- using the end character as end criterion

Error recognition

Detected errors:

- parity errors

Undetected errors:

- no storage of received data
- incomplete reception (missing data)

Remote confirmation / Feedback message at the communication block

The user cannot detect whether the sent data has arrived in the user data area of the remote CPU (application) without error.

42.2.3 Characteristics 3964(R)

Functionality

- During the sending process control characters are added to the data (start character, end character, block check character).
- Connecting and disconnecting occurs via control characters.
- In case of errors the transmission is repeated.

Code transparency

A protocol is code transparent if any character (00H to FFH) can occur in the data to be transferred.

3964(R) is code transparent.

Error recognition

Detected errors:

- parity errors
- incomplete reception (missing data)
  exception: byte with “00H” (see below)

Undetected errors:

- byte with “00H” has been lost.

Remote confirmation / Feedback message at the communication block

The user cannot detect whether the sent data has arrived in the user data area of the remote CPU (application) without error.

42.2.4 Characteristics RK 512

Functionality

The protocol works with message frames. The frames contain the data and an automatically added frame header. A command frame (SEND/PUT frame, GET frame) follows a reaction frame (with or without data).

Frame header of a command frame:

- Frame identifier (SEND/PUT frame, GET frame)
- Data destination for SEND/PUT job
- Data source for GET job
- Length of data to be transferred

Frame header of a reaction frame:

- Frame identifier (reaction frame, continuation frame)
- Error number

Sequence of sending/writing data:

- Communication partner sends command frame (SEND/PUT) with data
- Communication partner responds with reaction frame without data

Sequence for fetching data:

- Communication partner sends a command frame (GET) without data
- Communication partner responds with reaction frame with data

Error recognition

Detected errors:

- Parity errors
- Incomplete reception (missing data)

Remote confirmation / Feedback message at the communication block

The user can detect whether the sent data has arrived in the user data area of the remote CPU (application) without error, or whether the data were fetched from the user data area.

42.2.5 Communication type - detailed

Table 42 -2 Communication types - detailed

Communication type:		Serial interface		
Protocol:		***ASCII***	**3964(R)**	**RK 512**
General				
Interfaces		CPU, CP	CPU, CP	CPU, CP
Connection	SIMATIC S5	yes	yes	yes
	third-party (open standards)	yes	yes	no
User interface				
Communication block		see chapter 43	see chapter 43	see chapter 43
Maximal data volume		<= 4096 bytes	<= 4096 bytes	<= 4096 bytes
Dynamic addressing of data		yes, exception: S7-400 and SFB	yes, exception: S7-400 and SFB	yes, exception: S7-400 and SFB
Remote confirmation		no	Transport	Application
Model		Master / Master	Master / Master	Master / Master
Protocol				
Dynamic data length		yes		
Connections	to the remote partner?	no **(*2)**		
	dynamic / static	---		

Back to jump distributor SERIAL

Explanations for the table:

(*1): Exception: CP441: BSEND / receiving without BRCV. A communication block in the receiver is not necessary there.

(*2): For S7-400, a "PtP connection" is configured in STEP 7. This is not a connection to the remote partner.

43 Overview of User Interfaces

43.1 Application in stations with CPU

The table contains combinations for stations which a CPU is plugged into.

The option of operating CPs or CMs in distributed ET 200 stations, is considered in the subsequent chapter 43.2.

Table 43-1

Interface		Protocol	User interface		(*x)
ET200 CPU	**1SI (*101)**	*ASCII*, 3964R	S_SEND / S_RCV	FB3 / FB2	(*1)
S7-300	**CPU**	*ASCII*, 3964R	SEND_PTP / RCV_PTP	SFB60 /SFB61	(*2)
		RK 512	SEND_RK / SERVE_RK	SFB 63 / SFB 65	(*2)
			FETCH_RK / SERVE_RK	SFB 64 / SFB 65	(*2)
	CP 340 (*102)	*ASCII*, 3964R	P_SEND / P_RCV	FB3 / FB2	(*3)
	CP 341 (*102)	*ASCII*, 3964R	P_SND_RK / P_RCV_RK	FB8 / FB7	(*4)
		RK 512	P_SND_RK / P_RCV_RK	FB 8 / FB 7	(*4)
S7-400	**CP 440**	*ASCII*, 3964R	SEND_440 / REC_440	FB10 / FB9	(*5)
	CP 441	*ASCII*, 3964R	BSEND / BRCV	SFB12 / SFB13	(*6)
		ASCII, 3964R	BSEND / receive buffer	SFB12 / ---	(*6)
		RK512	BSEND / BRCV	SFB 12 / SFB 13	(*6)
			BSEND / ---	SFB 12 / ---	(*6)
			PUT	SFB 15 / ---	(*6)
			GET	SFB 14 / ---	(*6)
S7-1200	**CPU**	Freeport	SEND_PTP / RCV_PTP MB_MASTER / MB_SLAVE	SFB113 / SFB114	(*10)
	CM 1241				
S7-1500	**CM PtP (*103)**	Freeport, 3964R	Send_P2P / Receive_P2P Modbus_Master/ Modbus_Slave	FB613 / FB614 FB641 / FB642	(*10)

Note

Both protocols, *ASCII* and Freeport, are comparable

Explanations for the table

The user interface is available in STEP 7:

Table 43-2

(*x)	
(*1)	STEP7 (not TIA): Library: ET200sSI / ET200S Serial Interface
(*2)	STEP7 (not TIA): Library: Standard library / System function blocks
(*3)	STEP 7 (not TIA): Library: CP PtP / CP 340
(*4)	STEP 7 (not TIA): Library: CP PtP / CP 341
(*5)	STEP 7 (not TIA): Library: CP PtP / CP 440
(*6)	STEP 7 (not TIA): Library: Standard library / System function blocks
(*10)	STEP 7 (TIA): Instructions / Communication
(*101) to (*103)	see the following chapter

43.2 Application in distributed stations without CPU

CPs or CMs with serial interface can also be operated in distributed stations to which no CPU is plugged.

CPs or CMs from the table in chapter 43

(*101): also in distributed ET 200S station

(*102): also in distributed ET 200M station

(*103): also in distributed ET 200P station

Additional options

Apart from the CPs or CMs listed above, there are also the following options:

Table 43-3

Decentralized station	CM	Protocol	User interface	
ET 200SP	CM PtP	Freeport, 3964R	Send_P2P / Receive_P2P	FB613 / FB614

Note

CM PtP for ET 200SP has the same functionality as the CM PtP for S7-1500 (central) or ET 200MP (distributed).

44 ET 200S: *ASCII* and 3964(R)

44.1 Description

The communication block S_SEND sends data to the communication block S_RCV.

S_SEND Sending data

S_RCV Receiving data

44.2 Parameters for S_SEND

Table 44-1

INPUT	Type	Remark
REQ	BOOL	Trigger send job
R	BOOL	Abort job
LADDR	INT	Base address ET 200S 1SI
DB_NO	INT	Send area
DBB_NO	INT	
LEN	INT	Length of data to be received (*2)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
COM_RST	BOOL	New start of the FB

(*1): Remote confirmation: for *ASCII*: none / for 3964(R): transport
(*2): Maximal data volume: = 224 bytes

44.3 Parameters for S_RCV

Table 44-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
R	BOOL	Cancelling the job
LADDR	INT	Base address ET 200S 1SI
DB_NO	INT	Receive area
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
LEN	INT	Length of data to be received (*2)
NDR	BOOL	Data in the receive area
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
COM_RST	BOOL	New start of the FB

(*1): Data copied from the receive buffer into the receive area.
(*2): Maximal data volume: = 224 bytes

45 S7-300 CPU: *ASCII* / 3964(R)

45.1 Description

Communication block SEND_PTP sends data to communication block RCV_PTP.

SEND_PTP Sending data

RCV_PTP Receiving data

45.2 Parameters for SEND_PTP

Table 45-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
LADDR	WORD	I/O address of the submodule
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
SD_1	ANY	Send area
LEN	INT	Length of data to be received (*2)

(*1): Remote confirmation: for *ASCII*: none / for 3964(R): transport
(*2): Maximal data volume: = 1024 bytes

45.3 Parameters for RCV_PTP

Table 45-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
R	BOOL	Cancelling the job
LADDR	WORD	I/O address of the submodule
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data in the receive area (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
RD_1	ANY	Receive area
LEN	INT	Length of received data (*2)

(*1): Data copied from the receive buffer into the receive area.
(*2): Maximal data volume: = 1024 bytes

46 S7 300 CPU: RK 512

46.1 Description

Case discrimination:

- Sending data with SEND_RK / SERVE_RK
- Fetching data with FETCH_RK / SERVE_RK

These two communication partners are referred to as:

- CPU_1
- CPU_2

46.2 Sending data

CPU_1 sends data to CPU_2.
CPU_1 determines where the data is stored in CPU_2.
Communication blocks:

- CPU_1: SEND_RK
- CPU_2: SERVE_RK

SEND_RK

Sending data, with specifying the receive area

SERVE_RK

Receiving data

46.3 Fetching data

CPU_1 fetches data from CPU_2.
CPU_1 determines which data is fetched from CPU_2.
Communication blocks:

- CPU_1: FETCH_RK
- CPU_2: SERVE_RK

FETCH_RK

Fetching data with specifying the source area

SERVE_RK

Data provision

46.4 Parameters for SEND_RK

Communication block in remote CPU: SERVE_RK

Table 46-1

INPUT	Type	Remark
SYNC_DB	INT	Data block for synchronization
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
LADDR	WORD	I/O address of the submodule
R_CPU	INT	Number of the remote CPU
R_TYPE	CHAR	Destination area in remote CPU
R_DBNO	INT	
R_OFFSET	INT	
R_CF_BYT	INT	Communication flag of the remote CPU
R_CF_BIT	INT	
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
SD_1	ANY	Source area in local CPU
LEN	INT	Length of data to be received (*4)

(*1): Remote confirmation: application
(*2): Maximal data volume: 1024 bytes

46.5 Parameters for SERVE_RK

Communication block in remote CPU: SEND_RK

Table 46-2

INPUT	Type	Remark
SYNC_DB	INT	Data block for synchronization
EN_R	BOOL	Triggering receiving of data
R	BOOL	Cancelling the job
LADDR	WORD	I/O address of the submodule
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
L_TYPE	CHAR	Destination area on local CPU
L_DBNO	INT	
L_OFFSET	INT	
L_CF_BYT	INT	Communication flag
L_CF_BIT	INT	
IN_OUT	**Type**	**Remark**
LEN	INT	Length of data to be received (*2)

(*1): Data received
(*2): Maximal data volume: = 1024 bytes

46.6 Parameters for FETCH_RK

Communication block in remote CPU: SERVE_RK

Table 46-3

INPUT	Type	Remark
SYNC_DB	INT	Data block for synchronization
REQ	BOOL	Triggering fetching of data
R	BOOL	Cancelling the job
LADDR	EORD	I/O address of the submodule
R_CPU	INT	Number of the remote CPU
R_TYPE	CHAR	Source area in remote CPU
R_DBNO	INT	
R_OFFSET	INT	
R_CF_BYT	INT	Communication flag of the remote CPU
R_CF_BIT	INT	
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
RD_1	ANY	Destination area in local CPU
LEN	INT	Length of data to be received (*2)

(*1): Remote confirmation: application
(*2): Maximal data volume: = 1024 bytes

46.7 Parameters for SERVE_RK

Communication block in remote CPU: FETCH_RK

Table 46-4

INPUT	Type	Remark
SYNC_DB	INT	Data block for synchronization
EN_R	BOOL	Triggering provision of data
R	BOOL	Cancelling the job
LADDR	WORD	I/O address of the submodule
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
L_TYPE	CHAR	Source area on local CPU
L_DBNO	INT	
L_OFFSET	INT	
L_CF_BYT	INT	Communication flag
L_CF_BIT	INT	
IN_OUT	**Type**	**Remark**
LEN	INT	Length of supplied data (*2)

(*1): Data has been fetched.
(*2): Maximal data volume: = 1024 bytes

47 CP 340: *ASCII* / 3964(R)

47.1 Description

The communication block P_SEND sends data to the communication block P_RCV.

P_SEND Sending data

P_RCV Receiving data

47.2 Parameters for P_SEND

Table 47-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 340
DB_NO	INT	Send area
DBB_NO	INT	
LEN	INT	Length of data to be received (*2)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: for *ASCII*: none / for 3964(R): transport
(*2): Maximal data volume: = 1024 bytes

47.3 Parameters for P_RCV

Table 47-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 340
DB_NO	INT	Receive area
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
LEN	INT	Length of received data (*2)
NDR	BOOL	Data in the receive area (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Data copied from the receive buffer into the receive area.
(*2): Maximal data volume: = 1024 bytes

48 CP 341: *ASCII* / 3964(R)

48.1 Description

The communication block P_SND_RK sends data to the communication block P_RCV_RK.

P_SND_RK Sending data

P_RCV_RK Receiving data

Note: The communication blocks (P_SND_RK, P_RCV_RK) are universally used for *ASCII*, 3964(R), and RK 512. Listed below are only those parameters which are relevant for *ASCII* and 3964(R).

48.2 Parameters for P_SND_RK

Table 48-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Send area
DBB_NO	INT	
LEN	INT	Length of data to be received (*2)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: for *ASCII*: none / for 3964(R): transport
(*2): Maximal data volume: = 1024 bytes

48.3 Parameters for P_RCV_RK

Table 48-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Receive area
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
LEN	INT	Length of data to be received (*2)
NDR	BOOL	Data in the receive area (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Data copied from the receive buffer into the receive area.
(*2): Maximal data volume: = 1024 bytes

49 CP 341: RK 512

49.1 Description

Case discrimination:

- Sending data with P_SND_RK / P_RCV_RK
- Fetching data with P_SND_RK / P_RCV_RK

These two communication partners are referred to as:

- CPU_1
- CPU_2

Note
The communication blocks (P_SND_RK, P_RCV_RK) are universally used for *ASCII*, 3964(R), and RK 512. Listed below are only those parameters which are relevant for RK 512.

49.2 Sending data

CPU_1 sends data to CPU_2.

CPU_1 determines where the data is stored in CPU_2.

Communication blocks:

- CPU_1: P_SND_RK
- CPU_2: P_RCV_RK

P_SND_RK

Sending data, with specifying the receive area

P_RCV_RK

Receiving data

49.3 Fetching data

CPU_1 fetches data from CPU_2.

CPU_1 determines which data is fetched from CPU_2.

Communication blocks:

- CPU_1: P_SND_RK
- CPU_2: P_RCV_RK

P_SND_RK

Fetching data with specifying the source area

P_RCV_RK

Data provision

49.4 Parameters for P_SND_RK

Communication block in remote CPU: P_RCV_RK

Table 49-1

INPUT	Type	Remark
SF	**CHAR**	**SF = "S" (parameter for "sending data")**
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Source area in local CPU
DBB_NO	INT	
LEN	INT	Length of data to be received (*2)
R_CPU_NO	INT	Number of the remote CPU
R_TYP	CHAR	Destination area in remote CPU
R_NO	INT	
R_OFFSET	INT	
R_CF_BYT	INT	Communication flag of the remote CPU
R_CF_BIT	INT	
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: application
(*4): Maximal data volume: 4096 bytes

49.5 Parameters for P_RCV_RK

Communication block in remote CPU: P_SND_RK, with parameter SF = "S"

Table 49-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receiving of data
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Destination area on local CPU, if data destination "DX" has been configured at the sender
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
L_TYP	CHAR	Destination area on local CPU, if data destination "DX" has been configured at the sender
L_NO	INT	
L_OFFSET	INT	
LEN	INT	Length of data to be received (*2)
L_CF_BYT	INT	Communication flag
L_CF_BIT	INT	

(*1): Data received
(*2): Maximal data volume: = 4096 bytes

49.6 Parameters for P_SND_RK

Communication block in remote CPU: P_RCV_RK

Table 49-3

INPUT	Type	Remark
SF	**CHAR**	**SF = "S" (parameter for fetching data)**
REQ	BOOL	Triggering fetching of data
R	BOOL	Cancelling the job
LADDR	INT	Base address of the CP 341
DB_NO	INT	Destination area local CPU
DBB_NO	INT	
LEN	INT	Data length (*2)
R_CPU_NO	INT	Number of the remote CPU
R_TYP	CHAR	Source area remote CPU
R_NO	INT	
R_OFFSET	INT	
R_CF_BYT	INT	Communication flag of the remote CPU
R_CF_BIT	INT	
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: application
(*2): Maximal data volume: 1024 bytes

49.7 Parameters for P_RCV_RK

Communication block in remote CPU: P_SND_RK, with parameter SF = "F"

Table 49-4

INPUT	Type	Remark
EN_R	BOOL	Triggering provision of data
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Not applicable
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
L_TYP	CHAR	Source area on local CPU
L_NO	INT	
L_OFFSET	INT	
LEN	INT	Data length (*2)
L_CF_BYT	INT	Communication flag
L_CF_BIT	INT	

(*1): Data has been fetched
(*2): Maximal data volume: = 4096 bytes

50 CP 440: *ASCII* / 3964(R)

50.1 Description

Communication block SEND_440 sends data to communication block REC_440.

SEND_440 Sending data

REC_440 Receiving data

50.2 Parameters for SEND_440

Table 50-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 440
DB_NO	INT	Send area
DBB_NO	INT	
LEN	INT	Length of data to be received (*2)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Remote confirmation: for *ASCII*: none / 3964(R): transport
(*2): Maximal data volume: = 400 bytes

50.3 Parameters for REC_440

Table 50-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 440
DB_NO	INT	Receive area
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
LEN	INT	Length of received data (*2)
NDR	BOOL	Data in the receive area (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Data copied from the receive buffer into the receive area.
(*2): Maximal data volume: = 400 bytes

51 CP 441: *ASCII* / 3964(R)

51.1 Description

Two options are available:

- Sending with BSEND / receiving with BRCV
- Sending with BSEND / receiving with receive buffer

In both cases the receive buffer is established in the receiver. The receive area is not transmitted during sending.

Sending with BSEND / receiving with BRCV

Communication blocks are required in sender and receiver.

Advantage of using BRCV in the receiver:

- The application (user program) recognizes the complete reception of the data
- Preventing overwriting of data in the receive buffer which have not yet been fetched by the application (user program).

Sending with BSEND / receiving with receive buffer

A communication block in the receiver is not necessary. The receive buffer (data block) is configured in the receiver.

Disadvantage of using a receive buffer in the receiver:

- The application (user program) cannot recognize when a data transfer takes place.

51.2 Parameters for BSEND

Table 51-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	Here, data length
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
SD_1	ANY	Send area
LEN	WORD	Length of data to be received (*2)

(*1): Remote confirmation: for *ASCII*: none / for 3964(R): transport
(*2): Maximal data volume: = 4096 bytes

51.3 Parameters for BRCV

Table 51-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
ID	WORD	Reference to local connection description (given by configured connection in STEP 7)
R_ID	DWORD	Here, data length
OUTPUT	**Type**	**Remark**
NDR	BOOL	Data in the receive area (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
RD_1	ANY	Receive area
LEN	WORD	Length of data to be received (*2)

(*1): Data copied from the receive buffer into the receive area.
(*2): Maximal data volume: = 4096 bytes

52 CP 441: RK 512

52.1 Description

Case discrimination

- Sending data with BSEND / BRCV
- Sending data with BSEND / ---
- Sending data with PUT
- Fetching data with GET

These two communication partners are referred to as:

- CPU_1
- CPU_2

52.2 Sending data

Sending data from CPU_1, and receiving in CPU_2.
CPU_1 determines where the data is stored.
Communication blocks:

- CPU_1: BSEND
- CPU_2: BRCV

BSEND

Sending data

BRCV

Receiving data, specifying the receive area

52.3 Sending data

CPU_1 sends data to CPU_2.
CPU_1 determines where the data is stored.
Communication blocks:

- CPU_1: BSEND
- CPU_2: ---

BSEND

Sending data, specifying the receive area

Note: CPU_2 cannot detect when a data transfer takes place.

Sending data PUT / ---

CPU_1 sends data to CPU_2.

Communication blocks:

- CPU_1: PUT
- CPU_2: ---

PUT: Sending data, specifying a maximum of four receive areas

Note: CPU_2 cannot detect when a data transfer takes place.

Fetching data GET / ---

CPU_1 fetches data from CPU_2.

Communication blocks:

- CPU_1: GET
- CPU_2: ---

GET: Fetching data, specifying a maximum of four source areas.

Note: CPU_2 cannot detect when a data transfer takes place.

52.4 Parameters for BSEND

Communication block in remote CPU: BRCV

Table 52-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	Assigning the send SFB/FB and the receive SFB/FB. This enables communication of several SFB/FB pairs via the same logic connection.
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
SD_1	ANY	Source area in local CPU
LEN	WORD	Length of data to be received (*2)

(*1): Remote confirmation: application
(*2): Maximal data volume of sent data: 4096 bytes

52.5 Parameters for BRCV

Communication block in remote CPU: BRCV

Table 52-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	Assigning the send SFB/FB and the receive SFB/FB. This enables communication of several SFB/FB pairs via the same logic connection.
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
RD_1	ANY	Destination area in local CPU
LEN	WORD	Length of data to be received (*2)

(*1): Data has been received
(*2): Maximal data volume: 4096 bytes

52.6 Parameters for BSEND

Communication block in remote CPU: ---

Table 52-3

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	Destination area in remote CPU
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
SD_1	ANY	Source area in local CPU
LEN	WORD	Length of data to be received (*2)

(*1): Remote confirmation: application
(*2): Maximal data volume: <= 450 bytes (depending on remote CPU)

52.7 Parameters for PUT

Communication block in remote CPU: ---

Table 52-4

INPUT	Type	Remark
REQ	BOOL	Trigger write job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT		
ADDR_i	ANY	Destination areas in the remote CPU (i=1, 2, 3, 4) (*2)
SD_i	ANY	Source areas in the local CPU (i=1, 2, 3, 4) (*2)

(*1): Remote confirmation: application
(*2): Maximal data volume: <= 450 bytes (depending on remote CPU)

52.8 Parameters for GET

Communication block in remote CPU: ---

Table 52-5

INPUT	Type	Remark
REQ	BOOL	Trigger read job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT		
ADDR_i	ANY	Source area in the remote CPU (i=1, 2, 3, 4) (*2)
RD_i	ANY	Destination areas in the local CPU (i=1, 2, 3, 4) (*2)

(*1): Remote confirmation: application
(*2): Maximal data volume: <= 450 bytes (depending on remote CPU)

53 S7-1200: Freeport

Note: only STEP 7 (TIA).

53.1 Description

Using the instructions, the user program can send data to the local communication interfaces (CM, CP) or receive data from them.

The communication interface (CM, CP) performs the actual data transmission with the remote communication partner.

SEND_PTP Sending data

RCV_PTP Receiving data

53.2 Parameters for SEND_PTP

Table 53-1

INPUT	Type	Remark
REQ	BOOL	Triggering send job
PORT	PORT	Identifier of communication port
BUFFER	VARIANT	Send area
LENGTH	UINT	Length of data to be received (*2)
PTRCL	BOOL	---
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Data was transferred to the local CM
(*2): Maximal data volume: =1024 bytes

53.3 Parameters for RCV_PTP

Table 53-2

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
PORT	PORT	Identifier of communication port
BUFFER	VARIANT	Receive area
OUTPUT	**Type**	**Remark**
NDR	BOOL	Job finished (*1)
LENGTH	UINT	Length of received data (*2)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Data fetched from the local CM and written to the buffer of the CPU.
(*2): Maximal data volume: =1024 bytes

54 S7-1500, S7-300, S7-400: Freeport / 3964(R)

Note: only STEP 7 (TIA).

54.1 Description

Using the instructions, the user program can send data to the local communication interfaces (CM, CP) or receive data from them.

The communication interface (CM, CP) performs the actual data transmission with the remote communication partner.

Send_P2P Sending data

Receive_P2P Receiving data

54.2 Parameters for Send_P2P

Table 54-1

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Triggering send job
PORT	WORD	PORT	Identifier of communication port
BUFFER	ANY	VARIANT	Send area
LENGTH	WORD	UINT	Length of data to be received (*2)
OUTPUT			**Remark**
DONE	BOOL	BOOL	Job finished (*1)
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			**Remark**
COM_RST	BOOL	---	Initialization

(*1): Data transferred to the local CM and sent successfully.
(*2): Maximal data volume: ≤ 4096 (depending on CPU, CM)

54.3 Parameters for Receive_P2P

Table 54-2

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
PORT	WORD	PORT	Identifier of communication port
BUFFER	ANY	VARIANT	Receive area
OUTPUT			**Remark**
NDR	BOOL	BOOL	Job finished (*1)
LENGTH	WORD	UINT	Length of received data (*2)
ERROR	BOOL	BOOL	Error information
STATUS	BOOL	WORD	
IN_OUT			**Remark**
COM_RST	BOOL	---	Initialization

(*1): Data fetched from the local CM and written to the buffer of the CPU.
(*2): Maximal data volume: ≤ 4096 (depending on CPU, CM)

55 Information on Part 3

The tables contain references to information on the topics in Part 3 (communication types). All references /x/ are stored centrally in chapter 61. There you also find the respective internet links.

Table 55-1

/x/	Title	Information on
---	STEP 7 Online Help	Communication types, communication blocks
/6/	System and standard functions for S7-300/400	
/13/	SIMATIC NET functions (FC) and function blocks (FB) for SIMATIC NET S7-CPs	
/40/	SIMATIC S7-1500 ET 200MP, ET 200SP communication	
/100/	FAQs for S7-300 CPUs	Using communication blocks
/101/	FAQs for S7-400 CPUs	
/102/	FAQs for Industrial Ethernet S7-300/400CPs	
/103/	FAQs for PROFIBUS S7-300/400CPs	
/105/	FAQs for IE S7-300/400 CPs	Configuring the connections
/106/	FAQs for PROFIBUS S7-300/400 CPs	
/119/	FAQs on WinAC RTX	

56 ******* PART 4: Other Controllers **********

56.1 Structure and content

Table 56-1

Chapt.	Structure	Content
57	Modbus/TCP	Coupling via PN/IE network: • Characteristics • User interfaces
58	Modbus serial (RTU format)	Coupling via serial interface: • Characteristics • User interfaces
59	Information	Device manuals, FAQs, applications, ...

56.2 Preliminary remarks

Open standards

A SIMATIC controller can communicate with third-party controllers via open standards, if the third-party controllers also have implemented open standards.

This is considered in Part 3 of the documentation in the Properties tables. There the criterion “third-party connection” exists. “Yes” means that the communication type is an open standard. This enables communication with third-party controllers which also have this implemented open standard.

Examples:

- Open communication via T blocks
- Open communication via send/receive blocks

Open protocols

Here, in Part 4 of the documentation the communication via open protocols is described.

Properties of an open protocol:

- The protocol has been opened by the manufacturer.
- The protocol is manufacturer-specific.
- Anybody can use the protocol.
- The protocol is not standardized (no international standard)

57 Modbus/TCP

57.1 Characteristics

General

Modbus is a worldwide distributed protocol which is open to all users. Modbus/TCP enables communication via TCP/IP networks.

A SIMATIC controller can be:

- Modbus server
- Modbus client

Maximal transferrable data with a Modbus TCP job:

Table 57-1

Job	Transfer bit by bit	Transfer word by word
Read job	250 bytes	250 bytes
Write job	100 bytes	200 bytes

Establishing the TCP connection

Two options are available:

- Programming the connection with T-blocks (TCON, TDISCON)
- Configuring the connection with “Modbus TCP Wizard” (/30/)

57.2 Overview of User Interfaces

Table 57-2

Interface		User interface
ET 200 CPU	• integrated PN/IE interface of the CPU	Modbus blocks for CPU (*1)
S7-300	• integrated PN/IE interface of the CPU • CP 343-1	
S7-400	• integrated PN/IE interface of the CPU • CP 443-1	
S7-1200	• integrated PN/IE interface of the CPU	Instructions in STEP 7 (TIA)

(*1): For the Modbus/TCP communication there are separate function blocks (Modbus blocks). The Modbus blocks are not contained in STEP 7, they need to be ordered separately. For further information on the functionality, please refer to /11/.

57.3 S7-1200: Modbus client

57.3.1 Description

The MB_CLIENT instruction communicates as Modbus client via the PN interface of the CPU.

57.3.2 Parameter MB_CLIENT

Table 57-3

INPUT	Type	Remark
REQ	BOOL	Triggering the job
DISCONNECT	BOOL	Establishing and terminating the connection
CONNECT_ID	UINT	Identification of the connection
IP_OCTET_1	USINT	IP address of the Modbus server
IP_OCTET_2	USINT	
IP_OCTET_3	USINT	
IP_OCTET_4	USINT	
IP_PORT	UINT	Port of the Modbus server
MB_MODE	USINT	Mode of the request (read, write, diagnosis)
MB_DATA_ADR	UDINT	Start address of the data accessed by the MB_CLIENT instruction
DATA_LEN	UINT	Data length
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job is processed / Job finished (*1)
BUSY	BOOL	
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
MB_DATA_PTR	VARIANT	Buffer for the data to be received by the Modbus server, or for the data to be send to the Modbus server respectively

57.4 S7-1200: Modbus server

57.4.1 Description

The MB_SERVER instruction communicates as Modbus server via the PN interface of the CPU.

57.4.2 Parameter MB_SERVER

Table 57-4

INPUT	Type	Remark
DISCONNECT	BOOL	Reaction to connection request: establishing and terminating the connection
CONNECT_ID	UINT	Identification of the connection
IP_PORT	UINT	Port of the Modbus client
OUTPUT	**Type**	**Remark**
NDR	BOOL	New data written by the Modbus client
DR	BOOL	New data read by the Modbus client
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
MB_DATA_PTR	VARIANT	Buffer for the data to be received by the Modbus server, or for the data to be send to the Modbus server respectively

58 Modbus Serial (RTU Format)

58.1 Characteristics

General

Modbus is a worldwide distributed protocol which is open to all users. Modbus enables the communication via serial interfaces (RS232C, RS 422/485).

There are two versions for Modbus serial:

- RTU Format: binary coding of the data
- ASCII Format: ASCII coding of the data

The variant with RTU format is described below.

- Modbus serial (RTU format)

Modbus master / Modbus slave

Modbus serial works according to the master / slave principle.

A Modbus master can communicate with one or several Modbus slaves. Only the Modbus slave explicitly addressed by the Modbus master must send data back to the Modbus master.

The Modbus master can send jobs for reading and writing of operands to the Modbus slave:

Table 58-1

Job	Operand	
	Inputs, times, counter	Outputs, flags, data blocks
Read	x	x
Write	---	x

A SIMATIC controller can be:

- Modbus master
- Modbus slave

Modbus slave addressing

The address of a Modbus slave can be in the following range.

- CP341, CP441-2: 1 to 255
- CM PtP, ET200SP/CM PtP: 1 to 247 (*1)

(*1):

1 to 65535, for EXTENDED_ADDRESSING=true

With the Modbus slave address zero the master addresses all slaves at the bus (broadcast).

58.2 Overview of user interfaces

58.2.1 Connection via CP or CM

For communication via Modbus serial (RTU format), a SIMATIC CP or CM and the respective driver is required. The driver is either integrated on the SIMATIC CP or can be downloaded. Downloadable drivers must be ordered separately and installed in STEP 7 (/4 /, chapter "Downloadable drivers").

SIMATIC CPs with integrated drivers for Modbus serial

Central stations:

- S7-1200: CM 1241, CB 1241
- S7-1500: CM PtP HF

Distributed stations:

- ET 200S: 1SI module
- ET 200SP: CM PtP
- ET 200MP: CM PtP HF

SIMATIC CPs with the option of downloading drivers for Modbus serial

Central stations:

- S7-300: CP 341
- S7-400: CP 441-2

Distributed stations:

- ET 200S: CP 341

58.2.2 Modbus master

Overview of user interfaces:

Table 58-2

Interface		User interface		(*x)
ET 200S	1SI	S_SEND / S_RCV	FB 3 / FB 2	(*1)
S7-300 ET200S	CP 341	P_SND_RK / P_RCV_RK	FB 8 / FB 7	(*2)
S7-400	CP 441-2	BSEND / BRCV	SFB 12 / SFB 13	(*3)
S7-1200	CM 1241	MB_MASTER		(*4)
S7-1500 ET 200MP	CM PtP HF	Modbus_Master	FB 641	(*4)
ET 200SP	CM PtP			

Explanations for the table

The user interface is available in STEP 7:

Table 58-3

(*x)	
(*1)	STEP 7 (not TIA): Library: ET200sSI / ET200S serial interface
(*2)	STEP 7 (not TIA): Library: CP PtP / CP 341
(*3)	STEP 7 (not TIA): Library: Standard library / System function blocks
(*4)	STEP 7 (TIA): Instructions / Communication

58.2.3 Modbus slave

Overview of user interface:

Table 58-4

Interface		User interface		(*x)
ET200S	1SI	S_MODB	FB 81	(*1)
		S_SEND / S_RCV	FB 3 / FB 2	(*1)
S7-300 ET 200S	CP 341	MODB_341	FB 80	(*2)
		P_SND_RK / P_RCV_RK	FB 8 / FB 7	(*3)
S7-400	CP 441-2	MODB_441	FB 180	(*2)
S7-1200	CM 1241	MB_SLAVE		(*4)
S7-1500 ET 200MP	CM PtP HF	Modbus_Slave	FB 642	(*4)
ET 200SP	CM PtP			

Explanations for the table

The user interface is available in STEP 7:

Table 58-5

(*x)	
(*1)	STEP 7 (not TIA): Library: ET200sSI / ET200S Serial Interface
(*2)	STEP 7 (not TIA): After the installation of the Modbus slave CD, the FB is provided in the “Modbus” library.
(*3)	STEP 7 (not TIA): Library: CP PtP / CP 341
(*4)	STEP 7 (TIA): Instructions / Communication

58.3 1SI: Modbus master

The user interface is identical with the user interface for *ASCII* and 3964(R): See chapter 43.

58.4 1SI: Modbus slave

58.4.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves. The Modbus slave performs the job and reacts with a response frame.

S_MODB

Executing a job from the Modbus master.

S_MODB calls internally: S_SEND, S_RCV (see chapter 43).

58.4.2 Parameter S_MODB

Table 58-6

INPUT	Type	Remark
LADDR	INT	Base address ET 200S 1SI
START_TIMER	TIMER	Monitoring time
START_TIME	S5TIME	
DB_NO	INT	Modbus conversion table
OB_MASK	BOOL	Masking I/O access errors, delaying alarms.
CP_START	BOOL	Initialization
CP_START_FM	BOOL	
OUTPUT	**Type**	**Remark**
CP_NDR	BOOL	Modbus write job terminated
CP_START_OK	BOOL	Error information
CP_START_ERROR	BOOL	
ERROR_NR	WORD	Error information
ERROR_INFO	WORD	

58.5 CP 341: Modbus master

58.5.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves.

P_SND_RK Sending the job to the Modbus slave

P_RCV_RK Receiving response frame from the Modbus slave

58.5.2 Parameter P_SND_RK

Table 58-7

INPUT	Type	Remark
SF	CHAR	SF = „S“
REQ	BOOL	Triggering the job
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Send area in local CPU
DBB_NO	INT	
LEN	INT	Length of the data to be sent
R_TYP	CHAR	Number of the remote CPU
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	

(*1): “Job finished”:

For write function codes: after receiving the response frame
For read function codes: after receiving the response frame and transferring the received data in the CPU

58.5.3 Parameter P_RCV_RK

Table 58-8

INPUT	Type	Remark
EN_R	BOOL	Triggering receiving of data
R	BOOL	Cancelling the job
LADDR	INT	Base address of CP 341
DB_NO	INT	Receive area on local CPU
DBB_NO	INT	
OUTPUT	**Type**	**Remark**
NDR	BOOL	Receiving the response frame
LEN	INT	Length of data to be received
ERROR	BOOL	Error information
STATUS	WORD	

58.6 CP 341: Modbus slave

58.6.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves. The Modbus slave performs the job and reacts with a response frame.

FB80

Executing a job from the Modbus master.

Internally, FB80 calls: P_SND_RK, P_RCV_RK (see 48).

58.6.2 Parameter FB80

Table 58-9

INPUT	Type	Remark
LADDR	INT	Base address of the CP 341
START_TIMER	TIMER	Monitoring time
START_TIME	S5TIME	
OB_MASK	BOOL	Masking I/O access errors, delaying alarms.
CP_START	BOOL	Initialization
CP_START_FM	BOOL	
OUTPUT	**Type**	**Remark**
CP_START_NDR	BOOL	Modbus job: terminated
CP_START_OK	BOOL	Error information
CP_START_ERROR	BOOL	
ERROR_NR	WORD	Error information
ERROR_INFO	WORD	

58.7 CP 441-2: Modbus master

58.7.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves.

BSEND Sending the job to Modbus the slave

BRCV Receiving response frame from the Modbus slave

58.7.2 Parameter BSEND

Table 58-10

INPUT	Type	Remark
REQ	BOOL	Triggering send job
R	BOOL	Cancelling the job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DORD	Parameter for addressing within a connection. Both communication partners must use the same value.
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
SD_1	ANY	Send area
LEN	WORD	Length of the data block to be sent

(*1): For write function codes: after receiving a response frame
For reading function codes: after receiving the response frame and transferring the received data in the CPU

58.7.3 Parameter BRCV

Table 58-11

INPUT	Type	Remark
EN_R	BOOL	Triggering receive job
ID	WORD	Reference to the respective connection (from the configured connection in STEP 7)
R_ID	DWORD	Parameter for addressing within a connection. Both communication partners must use the same value.
OUTPUT	**Type**	**Remark**
NDR	BOOL	Receiving the response frame
ERROR	BOOL	Error information
STATUS	WORD	
IN_OUT	**Type**	**Remark**
RD_1	ANY	Receive area on local CPU
LEN	WORD	Length of data to be received

58.8 CP 441-2: Modbus slave

58.8.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves. The Modbus slave performs the job and reacts with a response frame.

FB180

Executing a job from the Modbus master.

58.8.2 Parameter FB180

Table 58-12

INPUT	Type	Remark
ID	INT	Reference to the respective connection (from the configured connection in STEP 7)
START_TIMER	TIMER	Monitoring time
START_TIME	S5TIME	
STATUS_TIMER	TIMER	Monitoring time
STATUS_TIME	S5TIME	
OB_MASK	BOOL	Masking I/O access errors, delaying alarms.
CP_START	BOOL	Initialization
CP_START_FM	BOOL	
OUTPUT	**Type**	**Remark**
CS_START_NDR	BOOL	Modbus job: terminated
CP_START_OK	BOOL	Initialization of error information
CP_START_ERROR	BOOL	
ERROR_NR	WORD	Job error information
ERROR_INFO	WORD	

58.9 CM 1241: Modbus master

58.9.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves.

MB_MASTER

Sending the job to Modbus the slave

Receiving response frame from the Modbus slave

58.9.2 Parameter MB_MASTER

Table 58-13

INPUT	Type	Remark
REQ	BOOL	Triggering the job
MB_ADR	UINT	Modbus station address
MODE	USINT	Selecting the mode
DATA_ADDR	UDINT	Start address in the slave
DATA_LEN	UINT	The length of the read / write data
DATA_PTR	VARIANT	Receive buffer (read) / send buffer (write)
OUTPUT	**Type**	**Remark**
DONE	BOOL	Job finished (*1)
BUSY	BOOL	Job is processed
ERROR	BOOL	Error information
STATUS	WORD	

(*1): Case discrimination:

For write function codes: after receiving the response frame

For read function codes: after receiving the response frame and transferring the received data in the CPU

58.10 CM 1241: Modbus slave

58.10.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves. The Modbus slave performs the job and reacts with a response frame.

MB_SLAVE

Executing a job from the Modbus master.

58.10.2 Parameter MB_SLAVE

Table 58-14

INPUT	Type	Remark
MB_ADDR	UINT	Modbus station address
MB_HOLD_REG	VARIANT	Modbus holding register DB
OUTPUT	**Type**	**Remark**
NDR	BOOL	Modbus write job: terminated
DR	BOOL	Modbus read job: terminated
ERROR	BOOL	Error information
STATUS	WORD	

58.11 CM PtP / CM PtP HF: Modbus master

Note

CM PtP: CM in ET 200SP

CM PtP HF: CM in S7-1500 (central), or CM in ET 200MP (distributed)

58.11.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves.

Modbus_Master

Sending the job to the Modbus slave

Receiving response frame from the Modbus slave

58.11.2 Parameter Modbus_Master

Table 58-15

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
REQ	BOOL	BOOL	Triggering the job
MB_ADR	WORD	UINT	Modbus station address
MODE	BYTE	USINT	Selecting the mode
DATA_ADDR	DWORD	UDINT	Start address in the slave
DATA_LEN	WORD	UINT	The length of the read / write data
OUTPUT			Remark
DONE	BOOL	BOOL	Job finished (*1)
BUSY	BOOL	BOOL	Job is processed
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			Remark
DATA_PTR	ANY	VARIANT	Receive buffer (read) / send buffer (write)
COM_RST	BOOL	---	Initialization

(*1): Case discrimination:

For write function codes: after receiving the response frame

For read function codes: after receiving the response frame and transferring the received data in the CPU

58.12 CM PtP / CM PtP HF: Modbus slave

Note

CM PtP: CM in ET 200SP

CM PtP HF: CM in S7-1500 (central), or CM in ET 200MP (distributed)

58.12.1 Description

The Modbus master can access data in one or several Modbus slaves (write and read). The Modbus master sends jobs to Modbus slaves. The Modbus slave performs the job and reacts with a response frame.

Modbus_Slave

Executing a job from the Modbus master.

58.12.2 Parameter Modbus_Slave

Table 58-16

INPUT	Type		Remark
	S7-300, S7-400	S7-1200, S7-1500	
MB_ADDR	WORD	UINT	Modbus station address
OUTPUT			**Explanation**
NDR	BOOL	BOOL	Modbus write job: terminated
DR	BOOL	BOOL	Modbus read job: terminated
ERROR	BOOL	BOOL	Error information
STATUS	WORD	WORD	
IN_OUT			Remark
MB_HOLD_REG	ANY	VARIANT	Modbus holding register DB
COM_RST	BOOL	---	Initialization

59 Information on Part 4

The tables contain references to information on the topics in Part 4 (third-party controller). All references /x/ are stored centrally in chapter 61. There you also find the respective internet links.

Table 59-1

/x/	Title	Information on
/17/	Communication between SIMATIC S7 and Modicon M340 via Modbus TCP	Modbus TCP
/104/	How can I establish an OPEN Modbus / TCP communication from a SIMATIC S7 and where can I get further information?	
/107/	Which ports are enabled for Modbus/TCP communication and how many Modbus clients can communicate with a SIMATIC S7-CPU as Modbus server?	
/30/	Wizard for the communication via Modbus TCP	
/14/	Function Blocks, Examples and User Manuals of the Serial Interface ET200S 1SI	Modbus RTU
/23/	SIMATIC S7-300/S7-400 Loadable driver for point-to-point CPs: Modbus protocol, RTU format, S7 is slave Operating instructions	
/24/	SIMATIC S7-300/S7-400 Loadable driver for point-to-point CPs: Modbus protocol, RTU format, S7 is master Operating instructions	
/26/	Loadable driver Modbus slave (RTU)	
/27/	Loadable driver Modbus master (RTU)	
/200/	Application for communication Task, solution, STEP 7 project	Applications for Modbus

60 ******* PART 5: Appendix ******************

60.1 Structure and content

Table 60-1

Chapt.	Structure	Content
61	Related literature	References in the text: /x/
62	Central terms	Brief explanation. If a term must be described in greater detail, then see chapter Background Information.
63	Abbreviations	
64	Background information	Description of important correlations
65	Discussed components	Ordering data and versions
66	History	Changes / versions of the documentation

61 Related Literature

Information

The following table contains links to the main topics:
catalogs, brochures, manuals, applications, tools

Table 61-1

/x/	Title	Link
/0/	Siemens Industry Online Support	http://support.automation.siemens.com
/1/	SIMATIC Controller / The innovative solution for all automation tasks. (Overview of SIMATIC controllers)	https://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-controller_overview_en.pdf
/2/	SIMATIC NET / Industrial communication, brochure (Overview of industrial communication)	https://www.automation.siemens.com/mcms/infocenter/dokumentencenter/sc/ic/Documentsu20Brochures/6ZB5530-1AE01-0BB5_K-Schrift_DE.pdf
/3/	SIMATIC / Communication with SIMATIC system manual (Basics on communication with SIMATIC)	http://support.automation.siemens.com/WW/view/en/25074283
/4/	Catalog ST 70 / Products for Totally Integrated Automation and Micro Automation (Overview and ordering data for SIMATIC controllers)	https://www.automation.siemens.com/mcms/infocenter/content/en/Pages/order_form.aspx?HTTPS=REDIR&nodeKey=key_516908&infotype=1
/5/	Catalog IK PI / Industrial communication (Overview and ordering data for devices of industrial communication)	http://www.automation.siemens.com/mcms/infocenter/content/en/Pages/order_form.aspx?nodeKey=key_517518&infotype=1&linkit=null
/6/	SIMATIC system and standard functions for S7-300/400, reference manual	http://support.automation.siemens.com/WW/view/en/44240604
/7/	CPU 31xC and CPU 31 x technical data, device manual	http://support.automation.siemens.com/WW/view/en/12996906
/8/	Automation system S7-400 CPU data, device manual	http://support.automation.siemens.com/WW/view/en/23904550
/11/	S7 OpenModbus/TCP (Product description, technical data, ordering data, contact, downloads)	https://www.industry.siemens.com/services/global/en/IT4Industry/products/simatic_add_ons/s7_open_modbus_tcp/Pages/default_tab.aspx
/12/	SIMATIC NET NCM S7 for PROFIBUS / FMS volume 2	http://support.automation.siemens.com/WW/view/en/1158418
/13/	SIMATIC NET functions (FC) and function blocks (FB) for SIMATIC NET S7-CPs, programming manual	http://support.automation.siemens.com/WW/view/en/30564821
/15/	SIMATIC / Configuring hardware and communication connections STEP 7 V5.5, manual	http://support.automation.siemens.com/WW/view/en/45531110
/16/	From PROFIBUS DP to PROFINET IO, programming manual	http://support.automation.siemens.com/WW/view/en/19289930
/17/	Communication between SIMATIC S7-300/400 and Modicon M340 via Modbus TCP	http://support.automation.siemens.com/WW/view/en/38586568
/18/	Performance data overview (Results of measurements on CPU-CPU communication, in a PROFIBUS, PROFINET/ Industrial Ethernet network, for different configurations)	http://support.automation.siemens.com/WW/view/en/25209605
/21/	S7-300 CPU 31xC technological functions (CPU 312C, CPU 313C, CPU 314C)	http://support.automation.siemens.com/WW/view/en/12429336
/22/	SIMATIC NET program blocks for SIMATIC NET S7-CPs, programming manual	http://support.automation.siemens.com/WW/view/en/62543517

/x/	Title		Link
/29/	Wizard for generating the connection data for the open TCP/IP communication		http://support.automation.siemens.com/WW/view/en/25209116
/30/	Wizard for the communication via Modbus TCP		http://support.automation.siemens.com/WW/view/en/31535566
/32/	SIMATIC STEP 7 Professional V12.0, system manual (Online Help printout)		http://support.automation.siemens.com/WW/view/en/68113685
/33/	SIMATIC programming with STEP 7 V5.5, manual		http://support.automation.siemens.com/WW/view/en/45531107
/34/	S7-1200 automation system, system manual		http://support.automation.siemens.com/WW/view/en/36932465
/35/	S7-1500 automation system, system manual		http://support.automation.siemens.com/WW/view/en/59191792
/36/	SIMATIC WinAC RTX (F) 2010, operating instructions		http://support.automation.siemens.com/WW/view/en/43715176
/37/	PROFINET system connection for SIMATIC S7	S7-300 (manuals)	http://support.automation.siemens.com/WW/view/en/58686942/133300
		S7-400 (manuals)	http://support.automation.siemens.com/WW/view/en/58686811/133300
		PC-based systems (manuals)	http://support.automation.siemens.com/WW/view/en/58691933/133300
/38/	PROFINET system connection for SIMATIC S7	S7-1200 (manuals)	http://support.automation.siemens.com/WW/view/en/44632196/133300
		S7-1500 (manuals)	http://support.automation.siemens.com/WW/view/en/67744877/133300
		S7-300 (manuals)	http://support.automation.siemens.com/WW/view/en/43484958/133300
		S7-400 (manuals)	http://support.automation.siemens.com/WW/view/en/43484515/133300
		S7-mEC (manuals)	http://support.automation.siemens.com/WW/view/en/43482516/133300
/39/	Serial communication	CP340 (manuals)	http://support.automation.siemens.com/WW/view/en/24283637/133300
		CP341 (manuals)	http://support.automation.siemens.com/WW/view/en/24284824/133300
		Loadable drivers for CP 441-2 and CP 341(manuals)	http://support.automation.siemens.com/WW/view/en/10805420/133300
		CP440 (manuals)	http://support.automation.siemens.com/WW/view/en/24254956/133300
		CP441 (manuals)	http://support.automation.siemens.com/WW/view/en/24255094/133300
		S7-1500 CM PtP (manuals)	http://support.automation.siemens.com/WW/view/en/67308868/133300
		ET200SP (manuals)	http://support.automation.siemens.com/WW/view/en/58532616/133300
		ET200S (manuals)	http://support.automation.siemens.com/WW/view/en/9260793
		PtP communication instructions in the application on distributed I/O of an S7-300/400	http://support.automation.siemens.com/WW/view/en/69124220
		S7-1500 / ET 200MP / ET 200SP CM PtP operation with PROFINET	http://support.automation.siemens.com/WW/view/en/59062563

/x/	Title	Link
	controller	
/40/	SIMATIC S7-1500, ET 200MP, ET 200SP communication	http://support.automation.siemens.com/WW/view/en/59192925

FAQ

The following table contains links to FAQs.

Table 61-2

/x/	Title	Link
	Configuring and programming the communication: using communication blocks	
/100/	S7-300 CPU31x	http://support.automation.siemens.com/WW/view/en/22866139
/101/	S7-400 CPU41x	http://support.automation.siemens.com/WW/view/en/23522717
/102/	IE S7-300/400 CPs	http://support.automation.siemens.com/WW/view/en/22548794
/103/	PB S7-300/400 CPs	http://support.automation.siemens.com/WW/view/en/21629966
	Configuring and programming the communication: configuring the connections:	
/105/	IE S7-300/400 CPs	http://support.automation.siemens.com/WW/view/en/22387424
/106/	PB S7-300/400 CPs	http://support.automation.siemens.com/WW/view/en/28526800
/119/	WinAC RTX	http://support.automation.siemens.com/WW/view/en/23337258
	MODBUS	
/104/	How can I establish an OPEN Modbus / TCP communication from a SIMATIC S7 and where can I get further information?	http://support.automation.siemens.com/WW/view/en/22660304

Application examples

The Siemens Industry Online Support contains a number of application examples. These application examples describe real, functional and non-industry specific solutions. They consist of solution approaches, performance data, configuration instruction and tested program code.

Table 61-3

/x/	Content	Link
/200/	Application examples on the communication	http://support.automation.siemens.com/WW/view/en/20229805/136000

62 Terms

This chapter contains an explanation of terms necessary for understanding the document. Some terms are used as equivalents. These terms are marked with “=”. Example: communication partner = partner

Some terms are described in greater detail elsewhere. In this case the following table contains a reference to the respective chapter (“Details” column).

Table 62-1

Term	Explanation	Details
Job = Communication job	A communication block executes a communication job. Example: “send x bytes”	
Client, Server Master, Slave Provider, Consumer	These terms are used for communication models. The terms describe properties of communication partners.	64.2
Controller	A controller is a central or decentralized automation station (station) with the components: CPU, CP and I/O. Central station: • station with centralized I/O. • communicates with distributed stations via PROFINET IO or PROFIBUS DP Decentralized station: • station with distributed I/O • communicates with central stations via PROFINET IO or PROFIBUS DP	5
CPU-CPU communication	CPU-CPU communication occurs between two CPUs: • SIMATIC CPU_1 <-> SIMATIC CPU_2 • SIMATIC CPU <-> CPU of a third-party controller	5
Data	Data refers to: net data, used data, user data, SIMATIC user data areas Examples: data block, flag, inputs, times	---
Hardware configuration of STEP 7	This refers to part of STEP 7 used for handling the following tasks: • device configuration and settings • networking of devices.	
I-Slave I-Device	decentralized station with CPU: I-slave for PROFIBUS I-device for PROFINET	
Communication via an open standard	The following applies for an “open standard”: • protocols are open **and** internationally standardized. • anybody can use the protocols without license. • the protocols do not depend on the manufacturer.	---
Communication blocks	Function blocks (FB, SFB, FC, SFC), for integration into the STEP 7 user program. Implements the data transmission (send, receive).	---
Medium	• Networks: MPI, PB, PN/IE • Backplane bus • Serial interface	---

Term	Explanation	Details
Networks	Here networks refers to industrial networks. These networks are used in the automation technology. A network can consist of one or several subnets.	1.2
Partner = Communication partner	Participants in the communication where data is exchanged	---
Project	When creating an automation solution with STEP 7, the various automation tasks are solved by control programs. STEP 7 combines all control programs and the required data in one project. A project contains the following data (example): • Configuration data via the Hardware setup. • Configuration data for the modules of the controller and for the distributed I/O. • Configuration data for the communication (PROFINET, ...) • Control program (LAD, FBD, ...)	---
Backplane bus	The backplane bus connects the modules (CPU, CP, ...) of a controller.	5
Interface, communication interface	Controllers communicate via media (PN/IE, ...). The controllers are connected to the medium via interfaces. An interface can be an: • integrated interface: CPU • external interface: CP or CM	---
Send/Receive blocks	Collective term for the following communication blocks: AG_SEND, AG_LSEND, AG_SSEND, AG_RECV, AG_LRECV, AG_SRECV	
STEP 7	STEP 7 is the engineering tool for SIMATIC Controller. The following abbreviations are used in the document: "STEP 7": STEP 7 up to V5.5 and/or from V10 "STEP 7 (not TIA)": STEP 7 only up to V5.5 "STEP 7 (TIA)": STEP 7 only from V10	
Subnet	A subnet is located in the area of the LANs (Local Area Networks). It enables communication, for example, between CPUs of controllers, within a spatially restricted area. A subnet is closed in itself, it has its own address space. Several subnets form a network.	---
T-blocks	Collective term for the following communication blocks: TSEND, TUSEND, TRCV, TURCV	
Type "USEND / URCV"	The "USEND / URCV" type designation comprises all variants of the communication blocks (FBx, SFBx) and of the designations in STEP 7 (USEND, USEND_E, ...).	
Type "PUT, GET"	The "PUT, GET" type designation comprises all variants of the communication blocks (FBx, SFBx) and of the designations in STEP 7 (PUT, PUT_E, ...).	
Connection	Generally, CPU-CPU communication with SIMATIC occurs via connections. A connection defines the location of the end points of the communication.	6
Connection blocks	Function blocks (FB, SFB, FC, SFC), for integration into the STEP 7 user program. This realizes and manages the connections.	---

63 Abbreviations

63.1 In the entire document

The table contains abbreviations which are used in the entire document.

Table 63-1

Abbreviation	Explanation
ASCII	In the document on hand, the abbreviation *ASCII* is used for a communication type: In the document, *ASCII* stands for a serial data transmission for which the transferred characters are ASCII coded.
ASCII	American Standard Code for Information Interchange
Box PC	Embedded Box PC bundles
CBA	PROFINET CBA (Component Based Automation)
CP	Communications Processor: module which deals with communication tasks, and connects a controller to a medium.
CPU	Central Processing Unit: module on which a user program runs. In this user program, data is sent or received.
DP	Distributed Periphery
E, A, M, D, T, Z	SIMATIC S7 storage areas: process image inputs (E), process image output (A), flag (M), data block (D), times (T), counter (Z)
FMS	Fieldbus Message Specification
GD	Global Data
IOC	PROFINET IO Controller
IOD	PROFINET IO Device
IoT	ISO-on-TCP (In the document, the term IoT was selected in order to save space)
IPC	Industrial PC
MPI	Multi Point Interface
OP	Operator Panel
Panel PC	Embedded Panel PC-Bundles
PB	PROFIBUS
PG	Programming unit
PN/IE	PROFINET / Industrial Ethernet
PNIO	PROFINET IO
S7-CP	CP of SIMATIC S7
S7-CPU	CPU of SIMATIC S7.
S7-mEC	S7-modular Embedded Controller
TIA	Totally Integrated Automation

63.2 Only in tables

The following table contains abbreviations which are only used in the Interfaces and Combinations tables: In order to save space, some abbreviations needed to be introduced in these tables.

Table 63-2

Abbreviation	Meaning
C1 C2	Controller 1 Controller 2
C1 Server C2 Server	Controller 1 is the server Controller 2 is the server
DP	DP communication
GD	Global data communication
Interface: IE	PROFINET interface without PN functionality
Interface: PN	PROFINET interface with PN functionality
IOD	Here, the abbreviation has the following meaning: IO controller as I-device
OC	Open Communication: The term stands for the communication types: • open communication with T blocks • open communication with Send/Receive blocks
PN	PN communication: Communication between PROFINET IO controller and PROFINET IO device.
S7	S7 communication
S7 B S7 Basis	S7 basic communication
X: IE	Abbreviation for: "Interface: IE"
X: PN	Abbreviation for: "Interface: PN"

64 Background Information

This chapter provides background information.

64.1 ISO/OSI reference model

The ISO/OSI reference model is a standardized model for describing open (manufacturer-independent) communication systems. The model describes the requirements for a communication system. The concrete implementation is not described. Most of the free to use protocols are based on this reference model (for example: TCP/IP).

The model consists of 7 layers with the following properties:

- Each layer has to fulfill fixed defined tasks
- The layers are independent of one another

Explanation of the layers

Table 64-1

Layer	Name	Task (examples)	Classification
Layer 7	Application layer	Interface with STEP 7 user program (confirmation on user level, ...).	Application-oriented
Layer 6	Presentation layer	Interpretation of the data (converting the standardized representation of the communication system into a device-specific form.)	
Layer 5	Session layer	Organization of data exchange. Behavior during failures and interrupted connection.	
Layer 4	Transport layer	Creating a transport connection between two devices: • establishing, canceling, maintaining the connection Transferring data packages: • dividing the data into packages (segmentation) • flow control • confirmation on transport level	Transport-oriented
Layer 3	Network layer	Transfer and delivery of data: • defining the communication paths (routing) • addressing the communication partners in the network	
Layer 2 (mac / layer 2)	Data link layer	Monitoring and organizing the access to the transfer medium Correct transfer of data (checksum, ...)	
Layer 1	Physical layer	Defining the physical connection between two devices (transfer medium, baud rate, ...)	

64.2 Communication models

Communication models (short: models) describe the principle of a communication relation. They specify the role both communication partners play during data exchange.

This chapter describes which communication models and which terms are used in the document on hand.

64.2.1 Client and server

The terms client and server are used in the document as follows:

Field of application

Networks: PN/IE, PB, MPI

Client

<u>Properties</u>

A client can exchange data with a client or a server.

<u>Providing the communication in the client</u>

Provisions must be made in the STEP 7 user program:

- programming the communication blocks, and/or
- configuring/programming the connections

Server

<u>Properties</u>

A server can exchange data with a client.

The trigger for data exchange always comes from a client. I.e. a server cannot take initiative for a data exchange.

<u>Providing the communication in the server</u>

Two different cases must be distinguished here:

Case 1: the communication is provided by the operating system only. I.e. the communication is a system functionality.

Case 2: provisions must be made in the STEP 7 user program:

- programming the communication blocks, and/or
- configuring/programming the connections

Client / Client communication

Both communication partners are clients.

One of both clients takes the initiative for the communication.

Client / Server communication

One communication partner is client, one communication partner is server.

Only the client can take the initiative for the communication.

64.2.2 Master and slave

The terms master and slave are used in the document as follows:

Field of application

Networks: PB (communication type DP communication)

Serial interface: Modbus serial, ...

Master

A master has the initiative during data exchange (behaves active):

- sends data to slave
- receives data from slave which he has requested from the slave beforehand

Slave

A slave has no initiative during data exchange (behaves passive):

- sends data to the master only if prompted by the master
- receives data from the master

Master / Slave communication

One communication partner is master, the other communication partners are slaves. The master takes the initiative.

Master / Master communication

Both communication partners are master. Both communication partners can take the initiative to send

DP communication

Master / Master communication is possible, however, this is not discussed in the document. This would require additional hardware (DP/DP coupler).

Serial interface

Master / Master communication is not possible.

64.2.3 Consumer and provider

The terms consumer and provider used in the document as follows:

Field of application

Network: PN/IE (communication type PNIO)

Consumer

Receives data from the provider without request.

Provider

Sends data to the consumer without request.

Consumer / provider communication

One communication partner is the consumer, one communication partner is the provider.

Consumer and provider are equal nodes in the network.

64.3 Confirmation

If data is transferred there are different feedback messages (confirmations) to the STEP 7 user program.

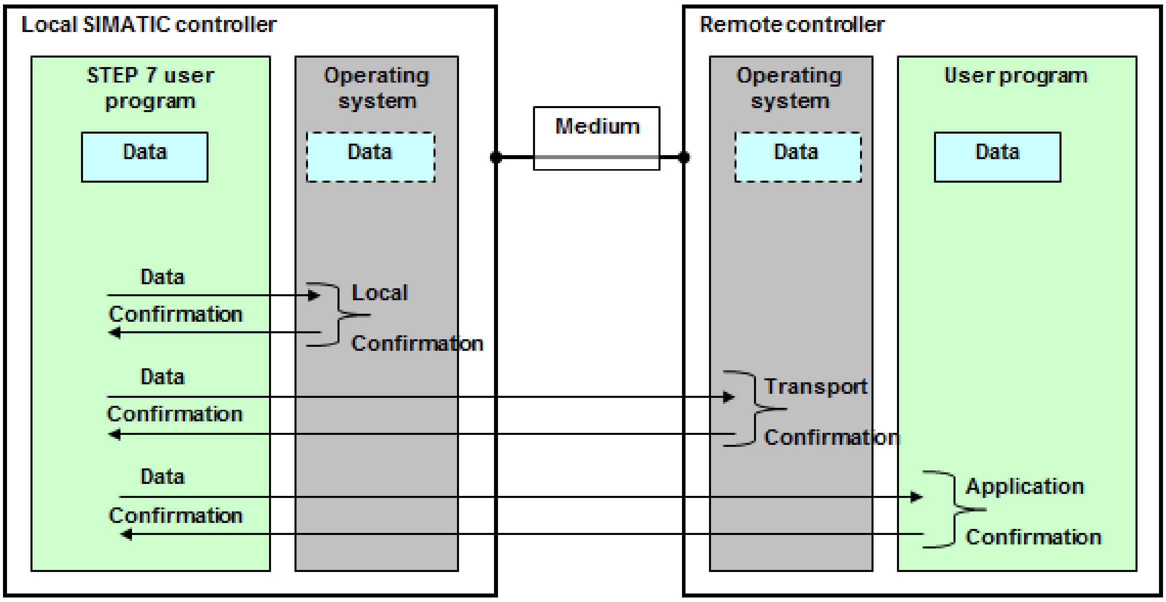
In the document the following confirmations are distinguished:

Table 64-2

Confirmation	Meaning	Comments
local	The data is located in the operating system (CPU, CP, CM) of the local controller	No statement on whether the data was transferred via the medium (PROFINET/IE, …).
Transport	The data is located in the operating system (CPU, CP, CM) of the remote controller	The data was transferred via the medium (PROFINET/IE, …).
Application	The data is located in the application of the remote controller.	

The following figure illustrates the relationships:

Figure 64-1



65 Discussed Components

Here, the components are listed which were considered in the document (as of March 2013).

65.1 SIMATIC CPU

Table 65-1

Family	CPU		MLFB	Version
ET 200 CPU	ET 200S	IM151-8(F) PN/DP CPU	6ES7 151-8AB01-0AB0 (6ES7 151-8FB01-0AB0)	FW V3.2
		IM151-7(F) CPU	6ES7 151-7AA21-0AB0 (6ES7 151-7FA21-0AB0)	FW V3.3
	ET 200Pro	IM154-8(F) PN/DP CPU	6ES7 154-8AB01-0AB0 (6ES7 154-8FB01-0AB0)	FW V3.2
S7-300	CPU 312		6ES7 312-1AE14-0AB0	FW V3.3
	CPU 314		6ES7 314-1AG14-0AB0	FW V3.3
	CPU 312C		6ES7 312-5BF04-0AB0	FW V3.3
	CPU 313C		6ES7 313-5BG04-0AB0	FW V3.3
	CPU 313C-2 DP		6ES7 313-6CG04-0AB0	FW V3.3
	CPU 314C-2 DP		6ES7 314-6CH04-0AB0	FW V3.3
	CPU 313C-2 PtP		6ES7 313-6BG04-0AB0	FW V3.3
	CPU 314C-2 PtP		6ES7 314-6BH04-0AB0	FW V3.3
	CPU 314C-2 PN/DP		6ES7 314-6EH04-0AB0	FW V3.3
	CPU 315(F)-2 DP		6ES7 315-2AH14-0AB0 (6ES7 315-6FF04-0AB0)	FW V3.3
	CPU 317(F)-2 DP		6ES7 317-2AK14-0AB0 (6ES7 317-6FF04-0AB0)	FW V3.3
	CPU 315(F)-2 PN/DP		6ES7 315-2EH14-0AB0 (6ES7 315-2FJ14-0AB0)	FW V3.2
	CPU 317(F)-2 PN/DP		6ES7 317-2EK14-0AB0 (6ES7 317-2FK14-0AB0)	FW V3.2
	CPU 319(F)-3 PN/DP		6ES7 318-3EL01-0AB0 (6ES7 318-3FL01-0AB0)	FW V3.2
S7-400	CPU 412-1 (MPI/DP)		6ES7 412-1XJ05-0AB0	FW V5.3
	CPU 412-2 (MPI/DP, DP)		6ES7 412-2XJ05-0AB0	FW V5.3
	CPU 412-2 PN (MPI/DP, PN)		6ES7412-2EK06-0AB0	FW V6.0
	CPU 414-2 (MPI/DP, DP)		6ES7 414-2XK05-0AB0	FW V5.3
	CPU 414-3 (MPI/DP, DP, IFM)		6ES7 414-3XM05-0AB0	FW V5.3
	CPU 414(F)-3 PN/DP (MPI/DP, PN, IFM)		6ES7 414-3EM06-0AB0 (6ES7 414-3FM06-0AB0)	FW V6.0
	CPU 416(F)-2 (MPI/DP, DP)		6ES7 416-2XN05-0AB0 (6ES7 416-2FN05-0AB0)	FW V5.3
	CPU 416-3 MPI/DP, DP, IFM)		6ES7 416-3XR05-0AB0	FW V5.3
	CPU 416(F)-3 PN/DP (MPI/DP, PN, IFM)		6ES7 416-3ES06-0AB0 (6ES7 416-3FS06-0AB0)	FW V6.0
	CPU 417-4 (MPI/DP, DP, IFM, IFM)		6ES7 417-4XT05-0AB0	FW V5.3

Family	CPU	MLFB	Version
S7-1200	CPU 1211C	6ES7211-xx31-0XB0	FW V3.0
	CPU 1212C	6ES7212-xx31-0XB0	FW V3.0
	CPU 1214C	6ES7214-xx31-0XB0	FW V3.0
	CPU1215C	6ES7215-xx31-0XB0	FW V3.0
S7-1500	CPU 1511-1 PN	6ES7511-1AK00-0AB0	FW V1.0
	CPU 1513-1 PN	6ES7513-1AL00-0AB0	FW V1.0
	CPU 1516-3 PN/DP	6ES7516-1AN00-0AB0	FW V1.0
S7-mEC (*1)	EC31-RTX (F)	6ES7677-1DD10-0BB0 (6ES7677-1FD10-0FB0)	08/2010
Box PC (*1)	SIMATIC embedded bundles (with RTX):	IPC2x7D, IPC4x7C	
Panel PC (*1)	SIMATIC embedded bundles (with RTX):	IPC277D, IPC477C	
WinAC RTX	WinAC RTX (F) 2010 (software)	6ES7 671-0RC08-0YA0 (6ES7 671-1RC08-0YA0)	V4.6

(*1): with WinAC RTX (F) 2010 as software controller

65.2 SIMATIC CP or CM

65.2.1 Application in stations with CPU

The table contains all CPs and CMs, discussed in part 2 (selection aid) of the document.

Table 65-2

Family	CP or CM			MLFB	Version
ET 200 CPU	PB	ET 200S	DP master module	6ES7138-4HA00-0AB0	FW V1.0
	PtP **(*1)**	ET 200S	1SI 3964/*ASCII*	6ES7138-4DF01-0AB0	FW V1.4
	PtP **(*1)**	ET 200S	1SI Modbus/USS	6ES7138-4DF11-0AB0	FW V1.4
S7-300	PB	CP 342-5		6GK7 342-5DA03-0XE0	FW V6.0
	PB	CP 342-5 FO		6GK7342-5DF00-0XE0	FW V5.7
	PB	CP 343-5		6GK7 343-5FA01-0XE0	FW V4.2
	PN/IE	CP 343-1 Lean		6GK7 343-1CX10-0XE0	FW V3.0
	PN/IE	CP 343-1		6GK7 343-1EX30-0XE0	FW V3.0
	PN/IE	CP 343-1 Advanced		6GK7 343-1GX31-0XE0	FW V3.0
	PN/IE	CP 343-1 ERPC		6GK7343-1FX00-0XE0	FW V1.0
	PtP **(*2)**	CP 340		6ES7340-1xH02-0AE0	FW V1.0
	PtP **(*2)**	CP 341		6ES7341-1xH02-0AE0	FW V2.1
S7-400	PB	CP 443-5 Basic		6GK7 443-5FX02-0XE0	FW V4.0
	PB	CP 443-5 Extended		6GK7 443-5DX05-0XE0	FW V7.0
	PN/IE	CP 443-1		6GK7 443-1EX30-0XE0	FW V3.1
	PN/IE	CP 443-1 Advanced		6GK7 443-1GX30-0XE0	FW V3.1
	PtP	CP 440		6ES7440-1CS00-0YE0	FW V1.0
	PtP	CP 441-1		6ES7441-1AA05-0AE0	FW V2.0
	PtP	CP 441-2		6ES7441-2AA05-0AE0	FW V2.0
S7-1200	PB	CM 1242-5 PB Slave		6GK7242-5DX30-0XE0	FW V1.0
	PB	CM 1243-5 PB Master		6GK7243-5DX30-0XE0	FW V1.2
	PtP	CM 1241 RS422/485		6ES7241-1CH31-0XB0	FW V1.0
	PtP	CM 1241 RS232		6ES7241-1AH30-0XB0	FW V1.0
	PtP	CB 1241 RS485		6ES7241-1CH30-1XB0	FW V1.0
S7-1500	PtP **(*3)**	CM PtP RS232 BA		6ES7 540-1AD00-0AA0	FWV1.0
	PtP **(*3)**	CM PtP RS232 HF		6ES7 541-1AD00-0AB0	FWV1.0
	PtP **(*3)**	CM PtP RS422/485 BA		6ES7 540-1AB00-0AA0	FWV1.0
	PtP **(*3)**	CM PtP RS422/485 HF		6ES7 541-1AB00-0AB0	FWV1.0
	IE	CP 1543-1		6GK7 543-1AX00-0XE0	FWV1.0
	PB	CM 1542-5		6GK7 542-5DX00-0XE0	FWV1.0
S7-mEC	PB	EM PCI-104		6ES7677-1DD60-1AA0	FWV1.0
	PN/IE	EM PC		6ES7677-1DD50-2AA0	FWv1.0
	PtP	CP 340		6ES7340-1xH02-0AE0	FWV1.0

Explanations on **(*x):** see chapter 65.2.2.

65.2.2 Application in distributed stations without CPU

CPs or CMs with serial interface can also be operated in distributed stations to which no CPU is plugged.

CPs or CMs from the table in chapter 65.2.1

(*1): also in distributed ET 200S station

(*2): also in distributed ET 200M station

(*3): also in distributed ET 200P station

Additional options

Apart from the CPs or CMs listed above, there are also the following options:

Table 65-3

Decentralized station	CM	MLFB	Version
ET 200SP	CM PtP	6ES7137-6AA00-0AB0	FW V1.0

Note:

CM PtP controls the protocols: 3964(R), Freeport, USS, Modbus RTU.

CM PtP for ET 200SP has the same functionality as the CM PtP for S7-1500 (central) or ET 200MP (distributed).

66 History

66.1 Versions

Table 66-1

Version	Date	
V2.1	04 / 2013	Update by new SIMATIC CPUs
V2.01	01 / 2011	Error elimination
V2.0	11 / 2010	Complete revision
V1.0	04 / 2004	First version

66.2 Main changes

Table 66-2

Version	Modifications
V2.01 -> V2.1	Updating all of the data
	S7-1200: new modules
	S7-1500: new system
	Summary of the SIMATIC families: WinAC = WinAC + S7-mEC +Panel PC + Box PC
	Multipanel with WinAC MP 2008: deleted, since fade out product
	Differentiation where necessary: STEP 7 (not TIA) / STEP 7 (TIA)
V2.0 -> V2.01	Page 368: Paragraph deleted: SIMATIC S7-1200
	Page 398: SIMATIC CPU supplemented: IM151-7 F CPU
V1.0 -> V2.0	Update with new components
	New structure of the document