

SIMATIC

CP 341 Point-to-Point Communication Installation and Parameter Assignment

Manual

This manual is part of the documentation package
with the order number:

6ES7 341-1AH00-8BA0

04/2000

C79000-G7076-C341

Edition 03

Preface, Contents

Product Description **1**

Basic Principles of Serial Data Transmission **2**

Starting up the CP 341 **3**

Mounting the CP 341 **4**

Configuring and Parameterizing the CP 341 **5**

Communication via Function Blocks **6**

Start-up Characteristics and Operating Mode Transitions of the CP 341 **7**

Diagnostics with the CP 341 **8**

Programming Example for Standard Function Blocks **9**

Appendices

Technical Specifications **A**

Connecting Cables **B**

Communication Matrix of the Protocols **C**

Accessories and Order Numbers **D**

SIMATIC S7 Reference Literature **E**

Glossary, Index



Danger

indicates that death, severe personal injury or substantial property damage **will** result if proper precautions are not taken.



Warning

indicates that death, severe personal injury or substantial property damage **can** result if proper precautions are not taken.



Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:



Warning

This device and its components may only be used for the applications described in the catalog or the technical descriptions, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

SIMATIC®, SIMATIC HMI® and SIMATIC NET® are registered trademarks of SIEMENS AG.

Some of other designations used in these documents are also registered trademarks; the owner's rights may be violated if they are used by third parties for their own purposes.

Copyright © Siemens AG 1998 All rights reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D- 90327 Nuernberg

Siemens Aktiengesellschaft

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

© Siemens AG 1998
Technical data subject to change.

C79000-G7076-C341



Preface

Purpose

This manual explains how to establish and operate a point-to-point connection.

Contents of This Manual

The manual describes the hardware and software of the CP 341 communication processor and its integration in an S7-300 programmable controller. It is divided up into instruction-based chapters and a reference section (appendices).

The following subjects are covered:

- The basics of point-to-point communication with the CP 341
- Starting up the CP 341
- Mounting the CP 341
- Communication via the CP 341
- Debugging
- Application example
- Properties and technical specifications

Scope of This Manual

The manual is relevant for:

Product	Order Number	As of product status
CP 341-RS 232C	6ES7 341-1AH01-0AE0	01
CP 341-20mA TTY	6ES7 341-1BH01-0AE0	01
CP 341-RS 422/485	6ES7 341-1CH01-0AE0	01

Changes Since Edition 01

In contrast with edition 01, this edition describes the additional functions of the CP 341 (MLFB No. 6ES7 341-1_H01-0AE0).

- Baud-rate range extended to 57.6 kbaud
- Diagnostics alarm
- Fast switch-over for RS485 module in half-duplex mode
- Extended modes for reception with end-of-text codes
- Shorter character delay times at low baud rates
- ASCII protocol with fixed message frame length: sending in character delay time matrix can be deactivated

Note

The descriptions of the CP 341 communication processors in this manual were correct at the time of publication. We reserve the right to describe modifications to the functionality of the modules in a separate Product Information.

Conventions

The abbreviation CP 341 is used in the documentation when information applies to all three module variants: CP 341-RS 232C, CP 341-20mA TTY and CP 341-RS 422/485.

Structure of This Manual

To help you to quickly find the information you require, this manual offers the following:

- A comprehensive list of contents.
- In the main body of the text, the information in the left-hand column of each page summarizes the contents of each section.
- Following the appendices, a glossary defines important technical terms used in the manual.
- Finally, a comprehensive index allows quick access to information on specific subjects.

Other Manuals Required

Appendix E contains a list of further publications on the S7-300 and other programmable controllers that you need to commission your system.

Electronic Manuals

The entire set of SIMATIC S7 documentation is available on CD-ROM.

Standards, Certificates and Approvals

The CP 341 communication processor meets the requirements and criteria of IEC 1131, Part 2 and the requirements for CE marking. The CP 341 has CSA certification and UL and FM recognition.

You will find more information on certification/recognition/approval and standards in Appendix A.3.

Recycling and Disposal

The CP 341 is an environment-friendly product. Its characteristic features include:

- Housing plastic with halogen-free flame protection and is highly resistant to fire
- Laser inscriptions (i.e. no labels)
- Plastics identification in accordance with DIN 54840
- Fewer materials used due to size reduction; fewer parts due to integration in ASICs

The CP 341 is suitable for recycling on account of the low level of contaminants in its components.

For further information about environment-friendly recycling and the procedure for disposing of your old equipment, please contact:

Siemens Aktiengesellschaft
Anlagenbau und Technische Dienstleistungen
ATD ERC Recycling/Remarketing
Fronthausenstr. 69
D-45127 Essen

Phone: + 49 201/816 1540 (Hotline)
Fax: + 49 201/816 1504

The people there will adapt their advice to suit your situation and provide a comprehensive and flexible recycling and disposal system at a fixed price. After disposal you will receive information giving you a breakdown of the relevant material fractions and the associated documents as evidence of the materials involved.

Additional Assistance

Please contact your local Siemens representative if you have any queries about the products described in this manual. A list of Siemens representatives worldwide is contained, for example, in the "Siemens Worldwide" Appendix of the manual *S7-300 Programmable Controller, Hardware and Installation*.

If you have any questions or suggestions concerning this manual, please fill out the form at the back and return it to the specified address. Please feel free to enter your personal assessment of the manual in the form provided.

We offer a range of courses to help get you started with the SIMATIC S7 programmable controller. Please contact your local training center or the central training center in Nuremberg, D-90027 Germany, Tel. +49 911 895 3200.

Up to the Minute Information

You can obtain the latest information on SIMATIC products from the following sources:

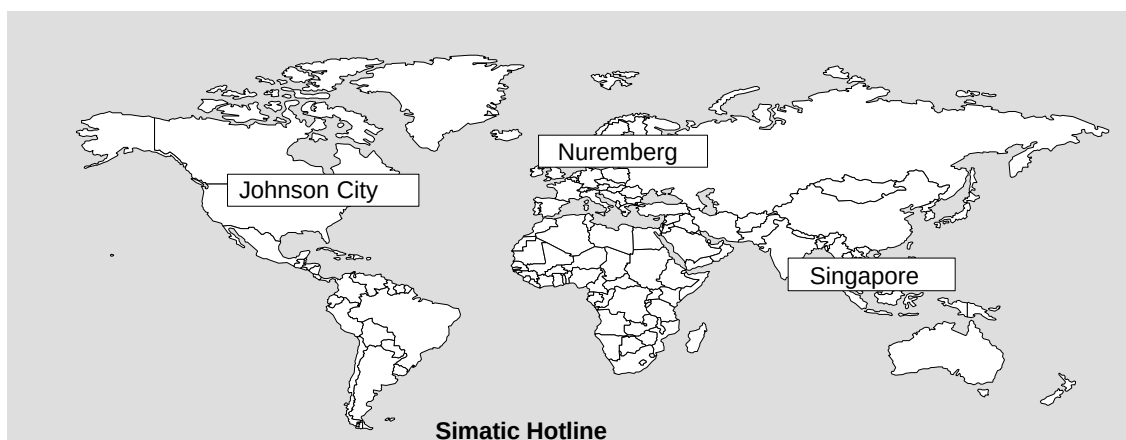
- on the **Internet** at <http://www.ad.siemens.de/>

SIMATIC Customer Support also provides you with the latest information and downloads that will help you use SIMATIC products:

- on the **Internet** at <http://www.ad.siemens.de/simatic-cs>
- From the SIMATIC Customer Support Mailbox at the number +49 (911) 895-7100

To dial the mailbox, use a modem with up to V.34 (28.8 kbauds), with the following parameter settings: 8, N, 1, ANSI, or dial via ISDN (x.75, 64 Kbps).

SIMATIC Customer Support is available at the phone and fax numbers and at the E-mail addresses listed below. You can also contact us via Internet mail or mail at the mailbox mentioned above.



Worldwide (Nuremberg) Technical Support (FreeContact) Local time: Mon.-Fri. 7:00 to 17:00 Phone: +49 (180) 5050-222 Fax: +49 (180) 5050-223 E-Mail: techsupport@ad.siemens.de GMT: +1:00	Worldwide (Nuremberg) Technical Support (fee based, only with SIMATIC Card) Local time: Mon.-Fri. 0:00 to 24:00 Phone: +49 (911) 895-7777 Fax: +49 (911) 895-7001 GMT: +01:00	
Europe / Africa (Nuremberg) Authorization Local time: Mon.-Fri. 7:00 to 17:00 Phone: +49 (911) 895-7200 Fax: +49 (911) 895-7201 E-Mail: authorization@nbgm.siemens.de GMT: +1:00	America (Johnson City) Technical Support and Authorization Local time: Mon.-Fri. 8:00 to 19:00 Phone: +1 423 461-2522 Fax: +1 423 461-2289 E-Mail: simatic.hotline@sea.siemens.com GMT: -5:00	Asia / Australia (Singapore) Technical Support and Authorization Local time: Mon.-Fri. 8:30 to 17:30 Phone: +65 740-7000 Fax: +65 740-7001 E-Mail: simatic.hotline@sae.siemens.com.sg GMT: +8:00
The languages of the SIMATIC Hotlines are generally German and English, in addition, French, Italian and Spanish are spoken on the authorization hotline.		

Contents

1	Product Description	1-1
1.1	Uses of the CP 341	1-2
1.2	Components Required for a Point-to-Point Connection with the CP 341	1-4
1.3	Design of the CP 341	1-6
1.4	Attributes of the Serial Interface	1-8
1.4.1	RS 232C Interface of the CP 341-RS 232C	1-8
1.4.2	20 mA TTY Interface of the CP 341-20mA TTY	1-10
1.4.3	X27 (RS 422/485) Interface of the CP 341-RS 422/485	1-11
1.5	Cables for Connecting the CP 341 to a Communication Partner	1-12
2	Basic Principles of Serial Data Transmission	2-1
2.1	Serial Transmission of a Character	2-2
2.2	Transmission Procedure with a Point-to-Point Connection	2-6
2.2.1	ISO 7-Layer Reference Model for Data Transmission	2-6
2.2.2	Data Transmission with the 3964(R) Procedure	2-11
2.2.3	Data Transmission with the RK 512 Computer Connection	2-23
2.2.4	Data Transmission with the ASCII Driver	2-35
2.3	Parameterization Data	2-49
2.3.1	Parameterization Data of the 3964(R) Procedure	2-49
2.3.2	Parameterization Data of the RK 512 Computer Connection	2-54
2.3.3	Parameterization Data of the ASCII Driver	2-55
3	Starting up the CP 341	3-1
4	Mounting the CP 341	4-1
4.1	CP 341 Slots	4-2
4.2	Mounting and Dismounting the CP 341	4-2
4.3	Installation Guidelines	4-4
5	Configuring and Parameterizing the CP 341	5-1
5.1	Configuring the CP 341	5-2
5.2	Parameterizing the Communication Protocols	5-3
5.3	Managing the Parameter Data	5-4
5.4	Subsequent Loading of Drivers (Transmission Protocols)	5-5
5.5	Subsequent Loading of Firmware Updates	5-6

6	Communication via Function Blocks	6-1
6.1	Communication via Function Blocks	6-2
6.2	Overview of the Function Blocks	6-2
6.3	Using the Function Blocks	6-4
6.3.1	Using the Function Blocks with the 3964(R) Procedure	6-4
6.3.2	Using the Function Blocks with the RK 512 Computer Connection	6-14
6.3.3	Using the System Function Blocks with the ASCII Driver	6-38
6.4	Parameterizing the Function Blocks	6-42
6.4.1	General Information on Data Block Assignment	6-42
6.4.2	Parameterizing the Data Blocks	6-43
6.5	General Information on Program Processing	6-47
6.6	Technical Specifications of the Function Blocks	6-48
7	Start-up Characteristics and Operating Mode Transitions of the CP 341	7-1
7.1	Operating Modes of the CP 341	7-2
7.2	Start-up Characteristics of the CP 341	7-2
7.3	Behavior of the CP 341 on Operating Mode Transitions of the CPU	7-4
8	Diagnostics with the CP 341	8-1
8.1	Diagnostics Functions of the CP 341	8-2
8.2	Diagnosis via the Display Elements of the CP 341	8-3
8.3	Diagnostics Messages of the Function Blocks	8-4
8.4	Error Numbers in the Response Message Frame	8-17
8.5	Diagnosis by Means of the Diagnostic Buffer of the CP 341	8-19
8.6	Diagnostic Alarm	8-21
9	Programming Example for Standard Function Blocks	9-1
9.1	General	9-2
9.2	Device Configuration	9-3
9.3	Settings	9-3
9.4	Blocks Used	9-5
9.5	Installation, Error Messages	9-6
9.6	Activation, Start-Up Program and Cyclic Program	9-7

A	Technical Specifications	A-1
A.1	Technical Specifications of the CP 341	A-2
A.2	Transmission Times	A-7
A.3	Certification and Areas of Application	A-9
B	Connecting Cables	B-1
B.1	RS 232C Interface of the CP 341-RS 232C	B-2
B.2	20 mA TTY Interface of the CP 341-20mA TTY	B-9
B.3	X27 (RS 422/485) Interface of the CP 341-RS 422/485	B-16
C	Communication Matrix of the Protocols	C-1
D	Accessories and Order Numbers	D-1
E	SIMATIC S7 Reference Literature	E-1
	Glossary	Glossary-1
	Index	Index-1

Product Description

1

In Section	You Will Find	on Page
1.1	Uses of the CP 341	1-2
1.2	Components Required for a Point-to-Point Connection with the CP 341	1-4
1.3	Design of the CP 341	1-6
1.4	Attributes of the Serial Interface	1-8
1.5	Cables for Connecting the CP 341 to a Communication Partner	1-12

1.1 Uses of the CP 341

The CP 341 communication processor allows you to exchange data between programmable controllers or computers by means of a point-to-point connection.

Functionality of the CP 341

The CP 341 communication processor provides the following functionality:

- Transmission rate up to 76.8 kbaud, half duplex
- Integration of the most important transmission protocols in the module firmware
 - 3964(R) procedure
 - RK 512 computer connection
 - ASCII driver
- Subsequent loading of other drivers (transmission protocols) with the parameterization interface *CP 341: Point-to-Point Communication, Parameter Assignment*.
- Custom parameterization of the transmission protocols with the parameterization interface *CP341: Point-to-Point Communication, Parameter Assignment*.
- Integrated serial interface:

Three module variants are available, each having a different interface type that is suitable for different communication partners (Table 1-1).

Module Variants

The following variants of the CP 341 communications processor are available:

Table 1-1 CP 341 Module Variants

Module	Order Number	Integrated Interface
CP 341-RS 232C	6ES7 341-1AH01-0AE0	RS 232C interface
CP 341-20mA TTY	6ES7 341-1BH01-0AE0	20 mA TTY interface
CP 341-RS 422/485	6ES7 341-1CH01-0AE0	X27 (RS 422/485) interface

Functions of the Module Variants

The functionality of the drivers depends on the module variant of the CP 341:

Table 1-2 Functions of the CP 341 Module Variants

Function	CP 341- RS 232C	CP 341- 20mA TTY	CP 341-RS 422/485	
			RS 422*	RS 485*
ASCII driver	yes	yes	yes	yes
• Use of RS 232C secondary signals	yes	no	no	no
• Controlling/reading of RS 232C secondary signals with FBs	yes	no	no	no
• RTS/CTS flow control	yes	no	no	no
• XON/XOFF flow control	yes	yes	yes	no
3964(R) procedure	yes	yes	yes	no
RK 512 computer connection	yes	yes	yes	no

* The RS 422 and RS 485 are distinguished by means of parameterization.

Uses of the CP 341

The CP 341 communication processor allows point-to-point communication with a variety of SIEMENS modules and with non-Siemens products.

- SIMATIC S5 via the 3964(R) driver or RK 512 with corresponding submodule on the S5 side.
- Siemens data acquisition terminals from the ES 2 family via the 3964(R) driver.
- MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) and data acquisition terminal ES 030K via the 3964(R) driver.
- PCs via the 3964(R) procedure (supported by development tools for programming on PC: PRODAVE DOS 64R (6ES5 897-2UD11) for MS-DOS, PRODAVE WIN 64R (6ES5 897-2VD01) for Windows or the ASCII driver)
- Barcode readers via the 3964(R) or ASCII driver.
- PLCs from other manufacturers via the 3964(R), ASCII driver or RK 512.
- Other devices with simple protocol structures by means of appropriate protocol adaptation with the ASCII driver.
- Other devices with 3964(R) drivers or RK 512.

Appendix C contains a summary of SIMATIC modules.

The CP 341 can also be operated in a distributed configuration using the ET 200M (IM153) I/O device.

1.2 Components Required for a Point-to-Point Connection with the CP 341

To establish a point-to-point connection between the CP 341 communication processor and a communication partner, you require certain hardware and software components.

Hardware Components

The following table lists the hardware components required for establishing a point-to-point connection with the CP 341.

Table 1-3 Hardware Components for a Point-to-Point Connection with the CP 341

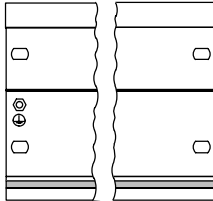
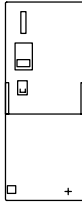
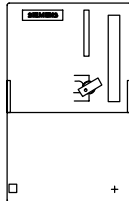
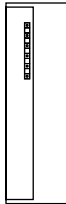
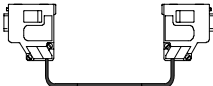
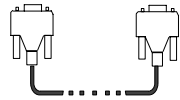
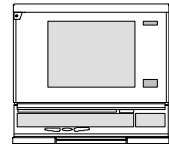
Components	Function	Diagram
Rack	... provides the mechanical and electrical connections of the S7-300.	
Power supply module (PS)	... converts the line voltage (120/230 VAC) into the operating voltage of 24 VDC required to supply the S7-300.	
CPU Accessories: Memory card Back-up battery	... executes the user program; communicates via the MPI interface with other CPUs or with a programming device.	
CP 341 communications processor	... communicates via the interface with a communication partner.	
Standard connecting cable	... connects the CP 341 communication processor to the communication partner.	

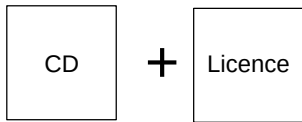
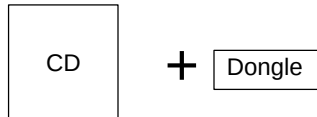
Table 1-3 Hardware Components for a Point-to-Point Connection with the CP 341, continued

Components	Function	Diagram
Programming device cable	... connects a CPU to a programming device/PC.	
Programming device or PC	... communicates with the CPU of the S7-300.	

Software Components

The following table lists the software components required for establishing a point-to-point connection with the CP 341.

Table 1-4 Software Components for a Point-to-Point Connection with the CP 341

Components	Function	Diagramm
STEP 7 software package	... configures, parameterizes, programs and tests the S7-300.	
The parameterization interface <i>CP 341: Point-to-Point Communication, Parameter Assignment</i>	... parameterizes the interface of the CP 341.	
Function blocks (FBs) with programming example	... control communication between the CPU and the CP 341.	
Loadable drivers	... with transmission protocols that can be loaded on the CP 341 in addition to the standard protocols in the module firmware.	

1.3 Design of the CP 341

The CP 341 communications processor is supplied with an integrated serial interface.

Positions of the Module Elements

Fig. 1-1 shows the positions of the module elements on the front panel of the CP 341 communications processor.

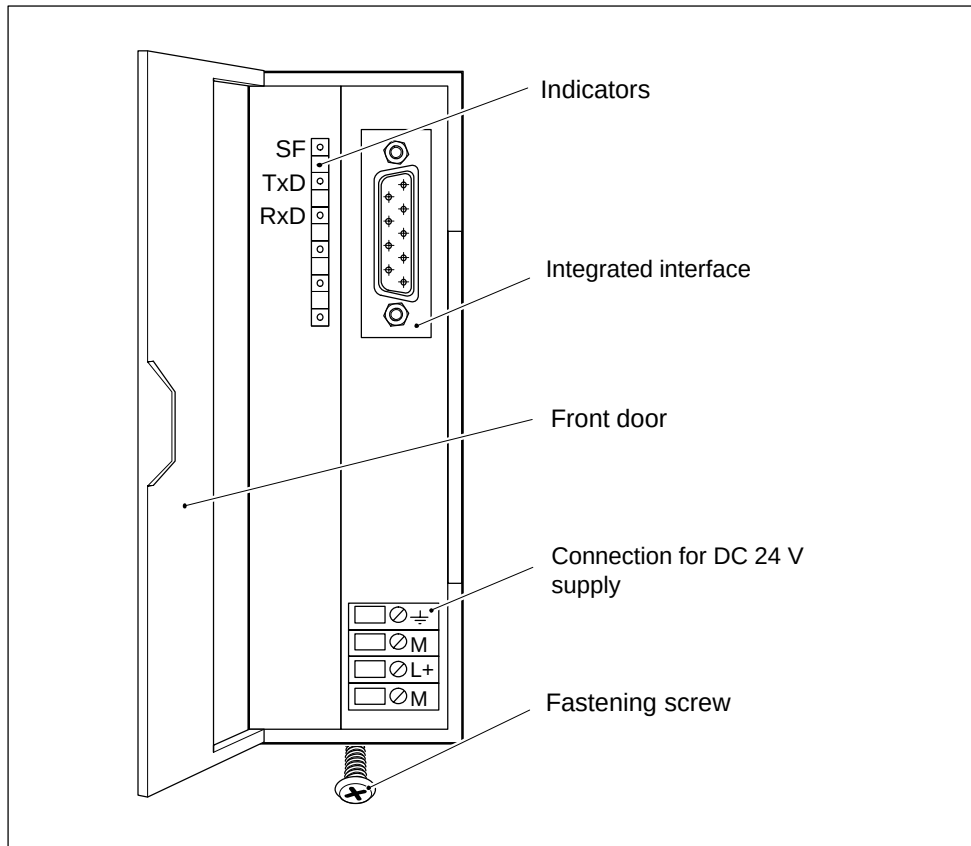


Figure 1-1 Positions of the Module Elements on the CP 341 Communications Processor

LED indicators

The following LED indicators are located on the front panel of the CP 341:

- **SF**(red) Error
- **TxD**(green) interface transmitting
- **RxD**(green) interface receiving

The operating modes and errors indicated by these LEDs are described in Section 8.2. Section 5.5 contains information on LED displays which can occur when you load firmware updates.

Integrated Interface

The CP 341 is available in three variants with different interface types:

- RS 232C
- X27 (RS 422/485)
- 20 mA TTY

The interface type is indicated on the front of the CP 341. You will find a detailed description of the interfaces in Section 1.4.

Expansion Bus for S7 Backplane Bus

The CP 341 is accompanied by an expansion bus. The expansion bus connects to the back panel of the CP 341 (see Section 4.2). The S7-300 backplane bus connects via the expansion bus.

The S7-300 backplane bus is a serial data bus via which the CP 341 communicates with the modules of the programmable controller.

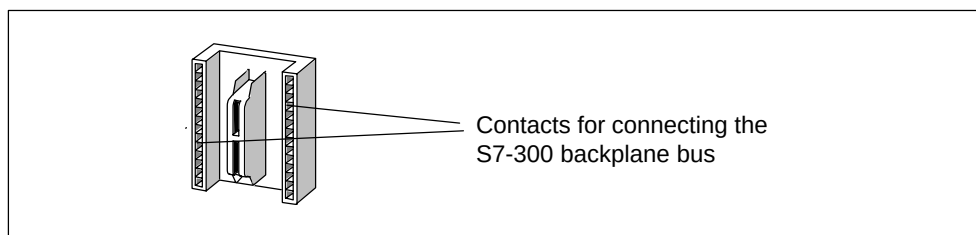


Figure 1-2 Expansion Bus

1.4 Attributes of the Serial Interface

Three module variants of the CP 341 are available, each having a different interface type that is suitable for different communication partners.

The interfaces of the module variants are described in the following sections.

1.4.1 RS 232C Interface of the CP 341-RS 232C

Definition

The RS 232C interface submodule is a voltage interface used for serial data transmission in compliance with the RS 232C standard.

Attributes

The RS 232C interface has the following attributes and is in compliance with the following requirements:

- Type: voltage interface
- Front connector: 9-pin subminiature D male connector with a screw-type fitting (compatible with the 9-pin COM port (PC/PG))
- RS 232C signals: TXD, RXD, RTS, CTS, DTR, DSR, RI, DCD, GND; all isolated against the S7-internal power supply (S7-300 backplane bus) and the external 24 V DC supply
- Max. transmission rate: 76.8 kbaud
- Max. cable length: 15 m, cable type LIYCY 7 × 0.14 (6ES7 902-1Ax00-0AA0)
- Standard: DIN 66020, DIN 66259, EIA-RS 232C, CCITT V.24/V.28

RS 232C Signals

Table 1-5 shows the meanings of the RS232C secondary signals.

Table 1-5 Signals of the RS 232C Interface

Signal	Designation	Meaning
TXD	Transmitted Data	Transmitted data; Transmission line is held by CP 341 on logic "1" in idle state.
RXD	Received Data	Received data; Receive line must be held on logic "1" by communication partner.
RTS	Request To Send	RTS "ON": CP 341 clear to send RTS "OFF": CP 341 not sending
CTS	Clear To Send	Communication partner can receive data from CP 341. The CP 341 expects the signal as response to RTS "ON".
DTR	Data Terminal Ready	DTR "ON": CP 341 is active and ready for operation DTR "OFF": CP 341 is not active and not ready for operation
DSR	Data Set Ready	DSR "ON": Communication partner is active and ready for operation DSR "OFF": Communication partner is not active and not ready for operation
RI	Ring Indicator	Incoming call when connecting a modem
DCD	Data Carrier Detect	Carrier signal when connecting a modem

1.4.2 20 mA TTY Interface of the CP 341-20mA TTY

Definition

The 20 mA TTY interface is a current-loop interface for serial data transmission.

Attributes

The 20 mA TTY interface has the following attributes and is in compliance with the following requirements:

- Type: current-loop interface
- Front connector: 9-pin sub D female with screw fixing
- 20 mA TTY signals: Two isolated 20 mA current sources, receive loop (RX) "-" and "+" transmit loop (TX) "-" and "+"; all isolated against the S7-internal power supply (S7-300 backplane bus) and the external 24 V DC supply
- Max. transmission rate: 19.2 kbaud
- Max. cable length: 1000 m active at 9.6 kbaud* (CP supplies the current loop), 1000 m slave at 9.6 kbaud* (partner supplies the current loop), 500 m active, 500 m passive at 19.2 kbaud; cable type LIYCY 7 × 0.14 (6ES7 902-2Ax00-0AA0)
- Standard: DIN 66258 Part 1

* Active/passive switch by means of wiring at line connector.

1.4.3 X27 (RS 422/485) Interface of the CP 341-RS 422/485

Definition

The X27 (RS 422/485) interface is a differential voltage interface used for serial data transmission in compliance with the X27 standard.

Attributes

The X27 (RS 422/485) interface has the following attributes and is in compliance with the following requirements:

- Type: differential voltage interface
- Front connector: 15-pin sub D female with screw fixing
- RS 422 signals: TXD (A), RXD (A), TXD (B), RXD (B), GND; all isolated against the S7-internal power supply
- RS 485 signals: R/T (A), R/T (B), GND; all isolated against the S7-internal power supply (S7-300 backplane bus) and the external 24 V DC supply
- Max. transmission rate: 76.8 kbaud
- Max. cable length: 250 m at 76.8 kbaud 500 m at 38.4 kbaud
1200 m at 19.2 kbaud; cable type LIYCY
7 × 0.14 (6ES7 902-3Ax00-0AA0)
- Standard: DIN 66259 Parts 1 and 3, EIA-RS 422/485, CCITT V.11

Note

With the RK 512 and 3964(R) protocols, the X27 (RS 422/485) interface can only be used in four-wire mode.

1.5 Cables for Connecting the CP 341 to a Communication Partner

Standard Connecting Cables

For point-to-point connection between the CP 341 and a communication partner, Siemens offers standard connecting cables in various lengths.

The lengths and order numbers of these cables are listed in Appendix D.

Constructing Your Own Connecting Cables

If you construct your own connecting cables, there are some points you must be aware of. These are described in Appendix B, along with wiring plans and the pin allocation for the sub D male connector.

Basic Principles of Serial Data Transmission

2

In Section	You Will Find	on Page
2.1	Serial Transmission of a Character	2-2
2.2	Transmission Procedure with a Point-to-Point Connection	2-6
2.3	Parameterization Data	2-49

2.1 Serial Transmission of a Character

For the exchange of data between two or more communication partners, various networking possibilities are available. The simplest form of data interchange is via a point-to-point connection between two communication partners.

Point-to-Point Connection

In a point-to-point connection the CP 341 communication processor forms the interface between a programmable controller and a communication partner. In a point-to-point connection with the CP 341, the data is transmitted serially.

Serial data transmission

In serial transmission, the individual bits of each byte of information are transmitted one after the other in a fixed order.

Drivers for Bidirectional Data Traffic

The CP 341 itself handles data transmission with communication partners via the serial interface. The CP 341 is equipped with three different drivers for this purpose.

Bidirectional data traffic:

- ASCII driver
- 3964(R) procedure
- RK 512 computer connection

The CP 341 handles data transmission via the serial interface in accordance with the interface type and the selected driver.

Bidirectional Data Traffic – Operating Modes

The CP 341 has two operating modes for bidirectional data traffic:

- Half-duplex operation (3964(R) procedure, ASCII driver, RK 512)

Data is exchanged between the communication partners but only in one direction at a time. In half-duplex operation, therefore, at any one time data is being either sent or received. The exception to this may be individual control characters for data flow control (e.g. XON/XOFF), which can also be sent during a receive operation or received during a send operation.

- Full-duplex operation (ASCII driver)

Data is exchanged between two or more communication partners in both directions simultaneously. In full-duplex operation, therefore, data can be sent and received at the same time. Each communication partner must be able to operate a send and a receive facility simultaneously.

With an RS 485 (2-wire) setting, the X27 (RS 422/485) interface submodule can only be run in half-duplex mode.

Asynchronous Data Transmission

With the CP 341, serial transmission occurs asynchronously. The so-called timebase synchronism (a fixed timing code used in the transmission of a fixed character string) is only upheld during transmission of a character. Each character to be sent is preceded by a synchronization impulse, or start bit. The end of the character transmission is signaled by the stop bit.

Declarations

As well as the start and stop bits, further declarations must be made between the two communication partners before serial transmission can take place. These include:

- Transmission speed (baud rate),
- Character and acknowledgment delay times,
- Parity,
- Number of data bits and
- Number of stop bits.

Sections 2.2 and 2.3 describe the importance of the declarations in the various transmission procedures, and how they are parameterized.

Character Frames

Data is transmitted between the CP 341 and a communication partner via the serial interface in a character frame. Three data formats are available for each character frame. You can parameterize the format for data transmission with the parameterization interface *CP341: Point-to-Point Communication, Parameter Assignment*.

10-Bit Character Frame

The figure below shows examples of the three different data formats for a 10-bit character frame.

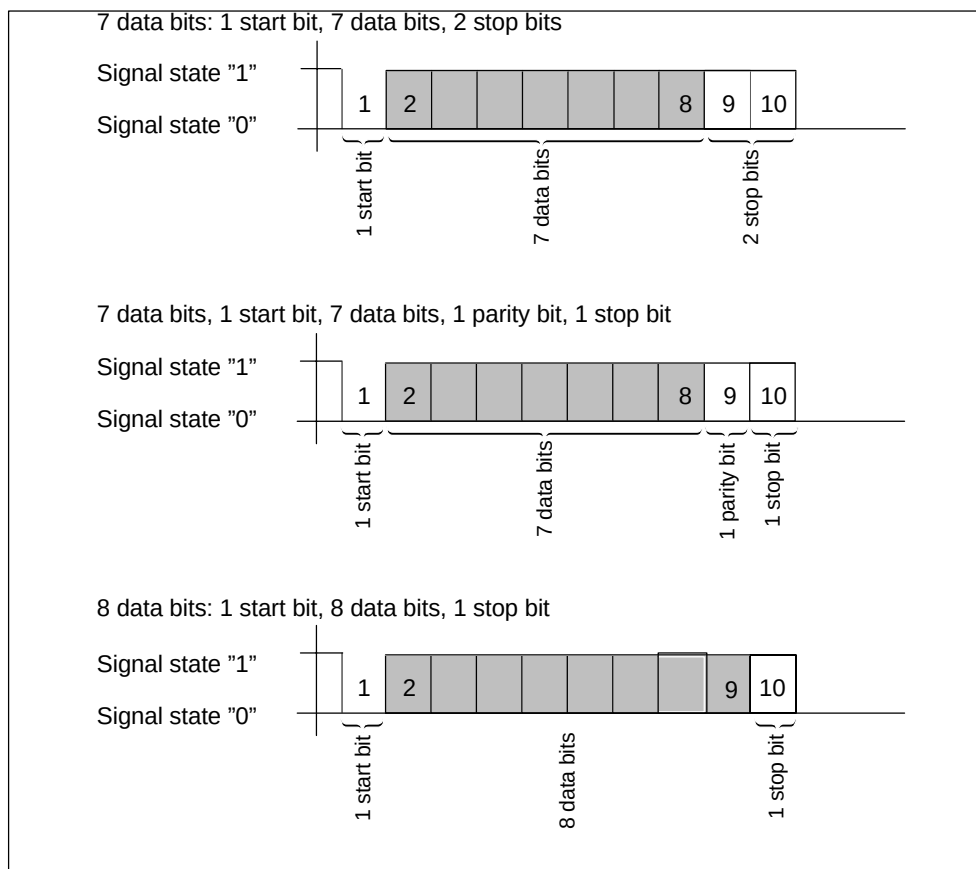


Figure 2-1 10-Bit Character Frame

Character Delay Time

The figure below shows the maximum time permitted between two characters received within a message frame. This is known as the character delay time.

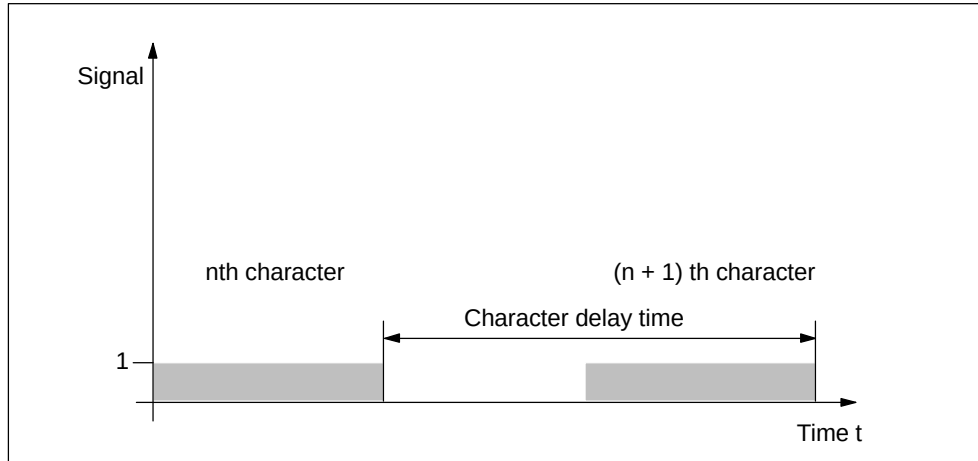


Figure 2-2 Character Delay Time

2.2 Transmission Procedure with a Point-to-Point Connection

When data is transmitted, all communication partners involved must follow fixed rules for handling and implementing the data traffic. The ISO has defined a 7-layer model, which is recognized as the basis for a worldwide standardization of transmission protocols for computer-to-computer communication.

2.2.1 ISO 7-Layer Reference Model for Data Transmission

Protocol

All communication partners involved in data transmission must follow fixed rules for handling and implementing the data traffic. Such rules are called protocols.

A protocol defines the following points:

- **Operating mode**
Half-duplex or full-duplex mode
- **Initiative**
Which communication partners can initiate the transmission and under what conditions
- **Control characters**
Which control characters are to be used for data transmission
- **Character frame**
Which character frames are to be used for data transmission
- **Data backup**
The data backup procedure to be used
- **Character delay time**
The time period within which an incoming character must be received
- **Transmission speed**
The baud rate in bits/s

Procedure

This is the specific process according to which the data is transmitted.

ISO 7-Layer Reference Model

The reference model defines the external behavior of the communication partners. Each protocol layer, except for the lowest one, is embedded in the next one down.

The individual layers are as follows:

1. **Physical layer**
 - Physical conditions for communication, e.g. transmission medium, baud rate
2. **Data-connection layer**
 - Security procedure for the transmission
 - Access modes
3. **Network layer**
 - Network connections
 - Addressing for communication between two partners
4. **Transport layer**
 - Error-recognition procedure
 - Debugging
 - Handshaking
5. **Session layer**
 - Establishing communication
 - Data exchange management
 - Terminating communication
6. **Presentation layer**
 - Conversion of the standard form of data representation of the communication system into a device-specific form (data interpretation rules)
7. **Application layer**
 - Defining the communication task and the functions it requires

Processing the Protocols

The sending communication partner runs through the protocols from the highest layer (no. 7 - application layer) to the lowest (no. 1 - physical layer), while the receiving partner processes the protocols in the reverse order, i.e. starting with layer 1.

Not all protocols have to take all 7 layers into account. If the sending and receiving partners both use the same protocol, layer 6 can be omitted.

Transmission Integrity

Transmission integrity plays an important role in the transmission of data and in selection of the transmission procedure. Generally speaking, the more layers of the reference model are applied, the greater the transmission integrity.

Classifying the Supplied Protocols

The CP 341 supports the following protocols:

- 3964(R) procedure
- RK 512 computer connection
- ASCII driver

The figure below illustrates how these protocols of the CP 341 fit into the ISO reference model:

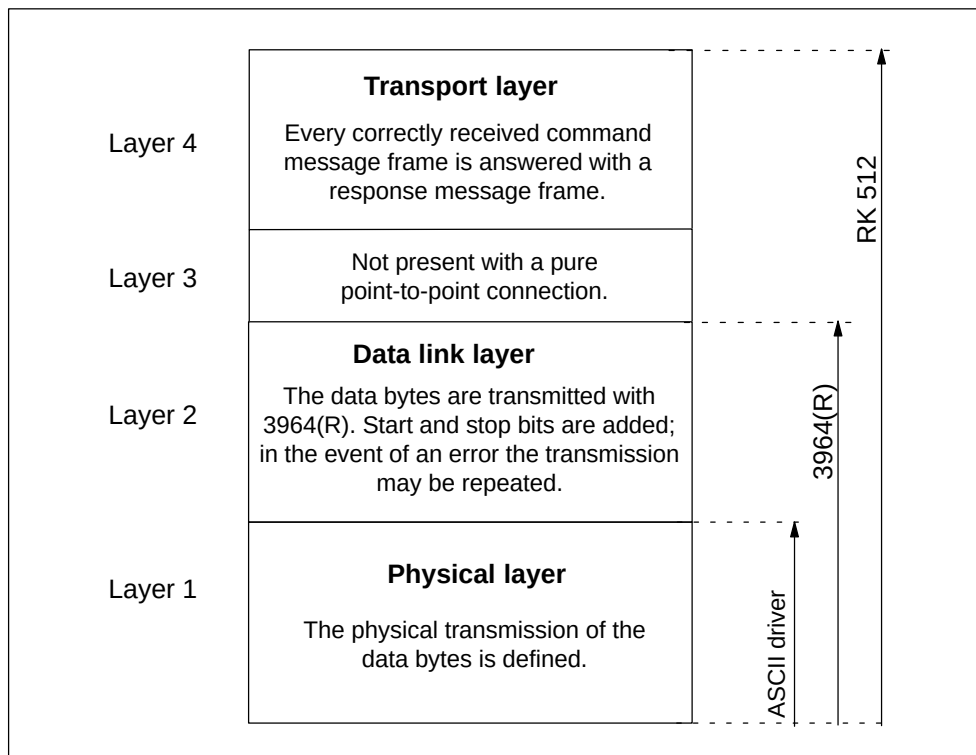


Figure 2-3 Position of the Supplied Protocols of the CP 341 in the ISO Reference Model

Transmission Integrity with the ASCII Driver

Data Integrity When Using the ASCII Driver:

- When data is transmitted via the ASCII driver, there are no data integrity precautions other than the use of a parity bit (can also be canceled, depending on how the character frame is set). This means that, although this type of data transport has a very efficient throughput rate, security is not guaranteed.
- Using the parity bit ensures that the inversion of a bit in a character to be transmitted can be recognized. If two or more bits of a character are inverted, this error can no longer be detected.
- To increase transmission integrity, a checksum and length specification for a message frame can be employed. These measures must be implemented by the user.
- A further increase in data integrity can be achieved by means of acknowledgment message frames in response to send or receive message frames. This is the case with high-level protocols for data communication (see ISO 7-layer reference model).

Transmission Integrity with 3964

Enhanced Data Integrity with the 3964R Procedure:

- The hamming distance with the 3964R is 3. This is a measure of the integrity of data transmission.
- The 3964R procedure ensures high transmission integrity on the data line. This high integrity is achieved by means of a fixed message-frame setup and cleardown as well as the use of a block check character (BCC).

Two different procedures for data transmission can be used, either with or without a block check character:

- Data transmission without a block check character: **3964**
- Data transmission with block check character: **3964R**

In this manual, the designation **3964(R)** is used when descriptions and notes refer to both data transmission procedures.

Performance Limits with 3964R

Performance Limits of the 3964R Procedure:

- Further processing of the send/receive data by the PLC program in the communication partner is not guaranteed. You can only ensure this by using a programmable acknowledgment mechanism.
- The block check of the 3964R procedure (EXOR logic operation) cannot detect missing zeros (as a whole character) because a zero in the EXOR logic operation does not affect the result of the calculation.

Although the loss of an entire character (this character has to be a zero!) is highly unlikely, it could possibly occur under very bad transmission conditions.

You can protect a transmission against such errors by sending the length of the data message along with the data itself, and having the length checked at the other end.

- Such transmission errors are ruled out when the RK 512 computer connection is used for data transmission, because here (unlike with the 3964(R) procedure) further processing is acknowledged via response message frames (e.g. stored in the destination data block) and the send data length is recorded in the message frame header. This enables the RK 512 to achieve a higher hamming distance (of 4) than the 3964R.

Transmission Integrity with RK 512

Very High Data Integrity with the RK 512:

- The hamming distance with the RK 512 and 3964R is 4. This is a measure of the integrity of data transmission.
- Using the RK 512 computer connection guarantees high transmission integrity on the data line (because the RK 512 uses the 3964(R) procedure for data transport).
- Further processing in the communication partner is ensured (because the RK 512 interpreter checks the additional length specification in the header and, after storing the data in the destination data block of the communication partner, generates a message frame acknowledging the success or failure of the data transmission).
- The RK 512 driver guarantees the correct use of the 3964R procedure and the analysis/addition of the length specification as well as the independent generation of the response message frames. There is no user handling! All you need to do is evaluate the positive/negative final acknowledgment.

Performance Limits with RK 512

Performance Limits with RK 512

- The RK 512 computer connection provides a very high degree of data integrity. You can improve this still further by, for example, using other block check mechanisms (e.g. CRC checks).

2.2.2 Data Transmission with the 3964(R) Procedure

The 3964(R) procedure controls data transmission via a point-to-point connection between the CP 341 and a communication partner. As well as the physical layer (layer 1), the procedure 3964(R) also incorporates the data-link layer (layer 2).

Control Characters

During data transmission, the 3964(R) procedure adds control characters to the information data (data-link layer). These control characters allow the communication partner to check whether the data has arrived complete and without errors.

The 3964(R) procedure analyzes the following control codes:

- **STX** Start of text; Start of character string for transfer
- **DLE** Data Link Escape; Data connection escape
- **ETX** End of Text; End of character string for transfer
- **BCC** Block check character (3964R only) Block check character
- **NAK** Negative Acknowledge; Negative acknowledgment

Note

If DLE is transmitted as an information string, it is sent twice so that it can be distinguished from the control code DLE during connection setup and release on the send line (DLE duplication). The receiver then reverses the DLE duplication.

Priority

With the 3964(R) procedure, one communication partner must be assigned a higher priority and the other partner a lower priority. If both partners begin connection setup at the same time, the partner with the lower priority will defer its send request.

Block Checksum

With the 3964R transmission protocol, data integrity is increased by the additional sending of a block check character (BCC).

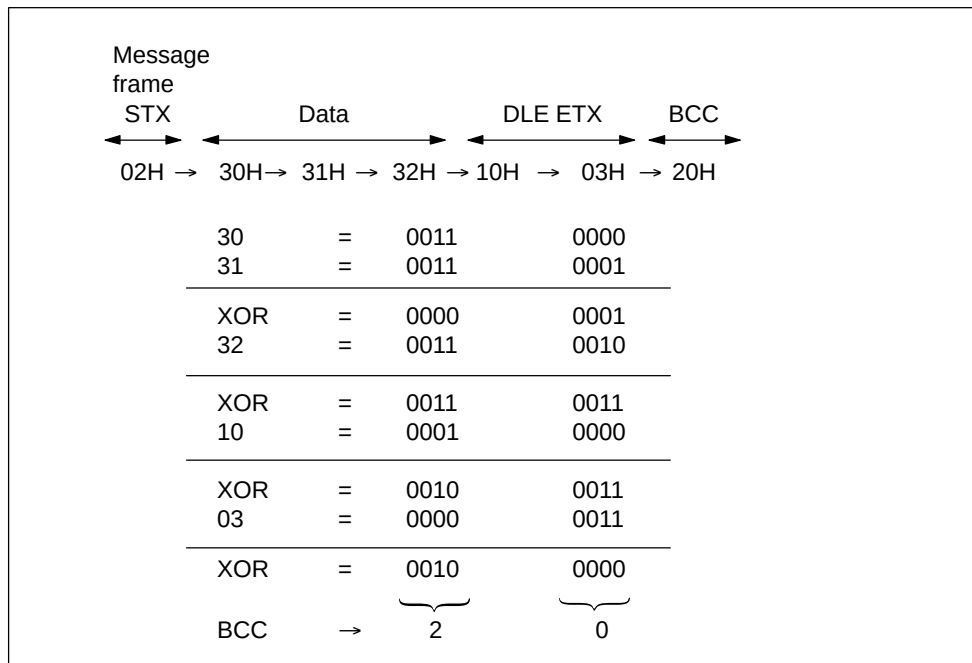


Figure 2-4 Block Checksum

The block checksum is the even longitudinal parity (EXOR logic operation of all data bytes) of a sent or received block. Its calculation begins with the first byte of user data (first byte of the message frame) after the connection setup, and ends after the DLE ETX code on connection release.

Note

If DLE duplication occurs, the DLE code is accounted for twice in the BCC calculation.

Sending Data with 3964(R)

The figure below illustrates the transmission sequence when data is sent with the 3964(R) procedure.

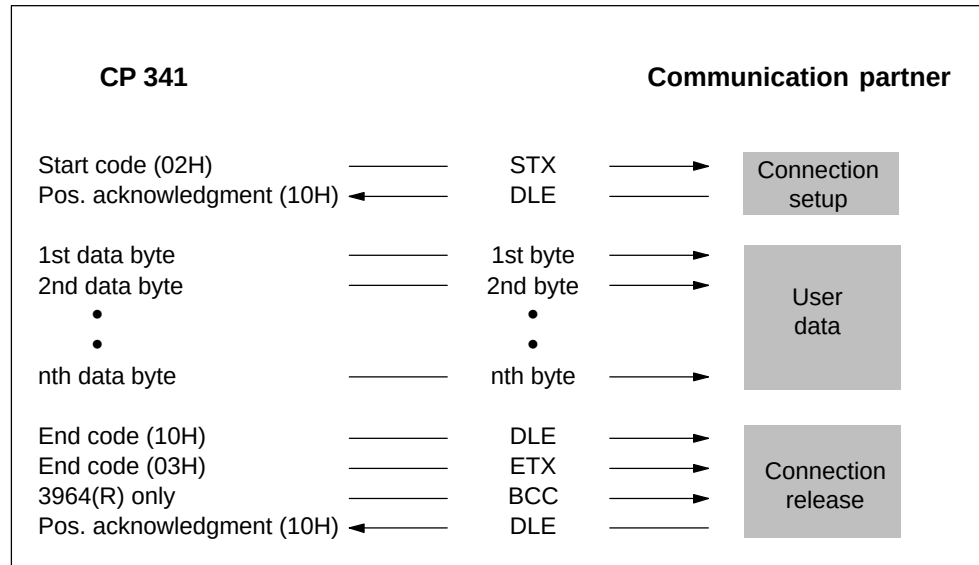


Figure 2-5 Data Traffic when Sending with the 3964(R) Procedure

Establishing a Send Connection

To establish the connection, the 3964(R) procedure sends the control code STX. If the communication partner responds with the DLE code before the acknowledgment delay time (ADT) expires, the procedure switches to send mode.

If the communication partner answers with NAK or with any other control code (except for DLE or STX), or the acknowledgment delay time expires without a response, the procedure repeats the connection setup. After the defined number of unsuccessful setup attempts, the procedure aborts the connection setup and sends the NAK code to the communication partner. The CP 341 reports the error to the function block P_SND_RK (output parameter STATUS).

Send data

If a connection is successfully established, the user data contained in the output buffer of the CP 341 is sent to the communication partner with the chosen transmission parameters. The partner monitors the times between incoming characters. The interval between two characters must not exceed the character delay time.

If the communication partner sends the NAK control code during an active send operation, the procedure aborts its transmission of the block and tries again as described above, beginning with connection setup. If a different code is sent, the procedure first waits for the character delay time to expire and then sends the NAK code to change the mode of the communication partner to idle. Then the procedure starts to send the data again with the connection setup STX.

Releasing a Send Connection

Once the contents of the buffer have been sent, the procedure adds the codes DLE, ETX and **with the 3964R only** the block checksum BCC as the end identifier, and waits for an acknowledgment code. If the communication partner sends the DLE code within the acknowledgment delay time, the data block has been received without errors. If the communication partner responds with NAK, any other code (except DLE), or a damaged code, or if the acknowledgment delay time expires without a response, the procedure starts to send the data again with the connection setup STX.

After the defined number of attempts to send the data block, the procedure stops trying and sends an NAK to the communication partner. The CP 341 reports the error to the function block P_SND_RK (output parameter STATUS).

Receiving Data with 3964(R)

The figure below illustrates the transmission sequence when data is received with the 3964(R) procedure.

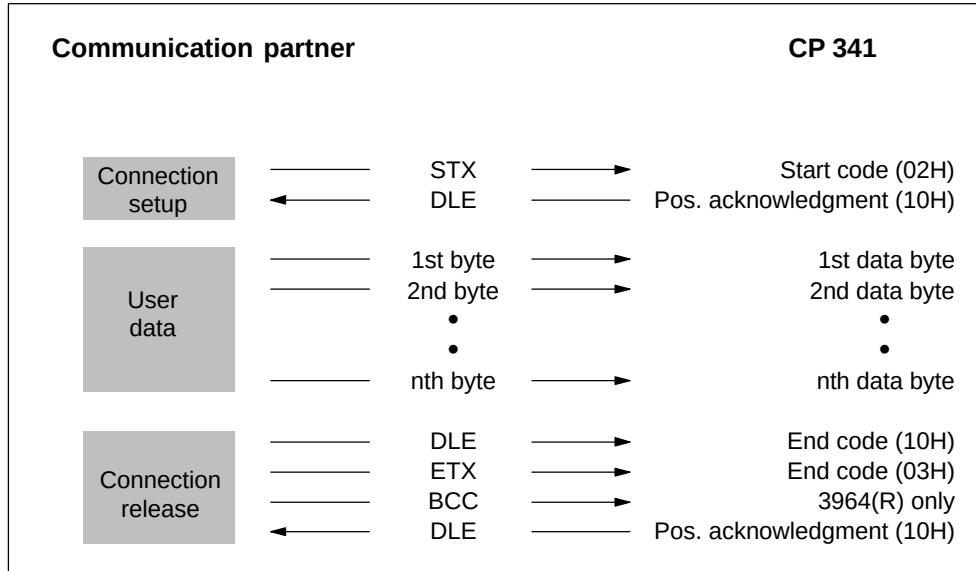


Figure 2-6 Data Traffic when Receiving with the 3964(R) Procedure

Note

As soon as it is ready, the 3964(R) procedure sends a single NAK to the communication partner to set the latter to idle.

Establishing a Receive Connection

In idle mode, when there is no send request to be processed, the procedure waits for the communication partner to establish the connection.

If no empty receive buffer is available during a connection setup with STX, a wait time of 400 ms is started. If there is still no empty receive buffer after this time has expired, the CP 341 reports the error (error message in STATUS output of FB) and the procedure sends an NAK and returns to idle mode. Otherwise, the procedure sends a DLE and receives the data.

If the idle procedure receives any control code except for STX or NAK, it waits for the character delay time to expire, then sends the code NAK. The CP 341 reports the error to the function block P_RCV_RK (output parameter STATUS).

Receive data

After a successful connection setup, the receive characters that arrive are stored in the receive buffer. If two consecutive DLE codes are received, only one of these is stored in the receive buffer.

After each receive character, the procedure waits out the character delay time for the next character. If this period expires before another character is received, a NAK is sent to the communication partner. The system program reports the error to the function block P_RCV_RK (output parameter STATUS). The 3964(R) procedure does not initiate a repetition.

If transmission errors occur during receiving (lost character, frame error, parity error, etc.), the procedure continues to receive until the connection is released, then a NAK is sent to the communication partner. A repetition is then expected. If the undamaged block still cannot be received after the number of transmission attempts defined in the static parameter set, or if the communication partner does not start the repetition within a block wait time of 4 seconds, the procedure aborts the receive operation. The CP 341 reports the first errored transmission and the final abort to the function block P_RCV_RK (output parameter STATUS)

Releasing a Receive Connection

If the **3964** procedure recognizes the string DLE ETX, it stops receiving and sends to the communication partner a DLE if the block was received without errors. If the block is damaged it sends a NAK. A repetition is then expected.

If the **3964R** procedure recognizes the string DLE ETX BCC, it stops receiving and compares the received BCC with the internally calculated longitudinal parity. If the BCC is correct and no other receive errors have occurred, the 3964R procedure sends a DLE and returns to idle mode. If the BCC is errored or a different receive error occurs, a NAK is sent to the communication partner. A repetition is then expected.

Handling Errored Data

The figure below illustrates how errored data is handled with the 3964(R) procedure.

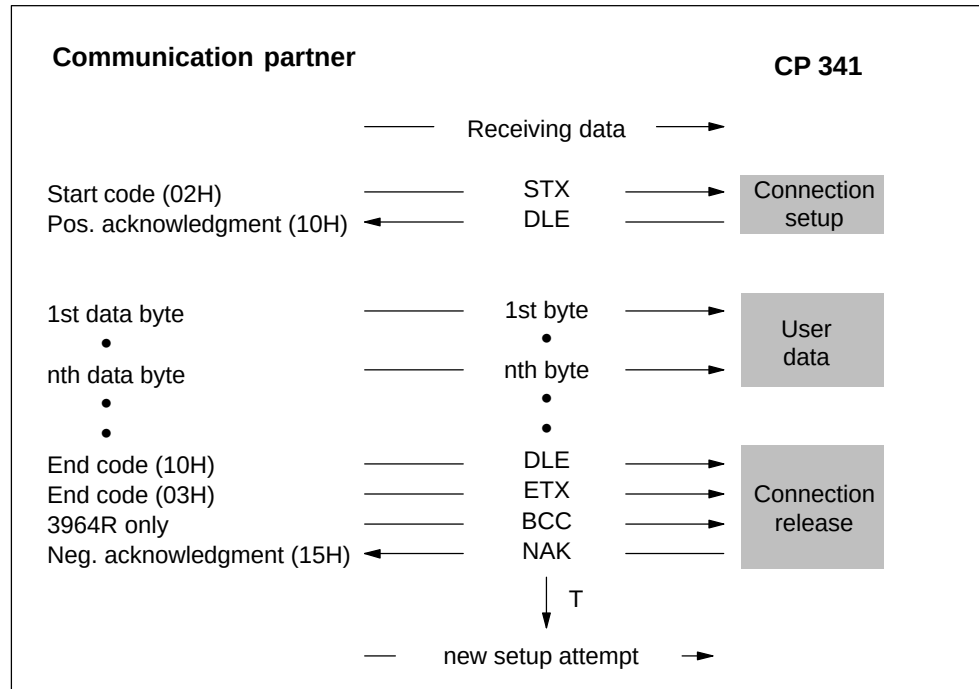


Figure 2-7 Data Traffic when Receiving Errored Data

When DLE, ETX, BCC is received, the CP 341 compares the BCC of the communication partner with its own internally calculated value. If the BCC is correct and no other receive errors occur, the CP 341 responds with DLE.

Otherwise, the CP 341 responds with an NAK and waits the block wait time (T) of 4 seconds for a new attempt. If after the defined number of transmission attempts the block cannot be received, or if no further attempt is made within the block wait time, the CP 341 aborts the receive operation.

Initialization Conflict

The figure below illustrates the transmission sequence during an initialization conflict.

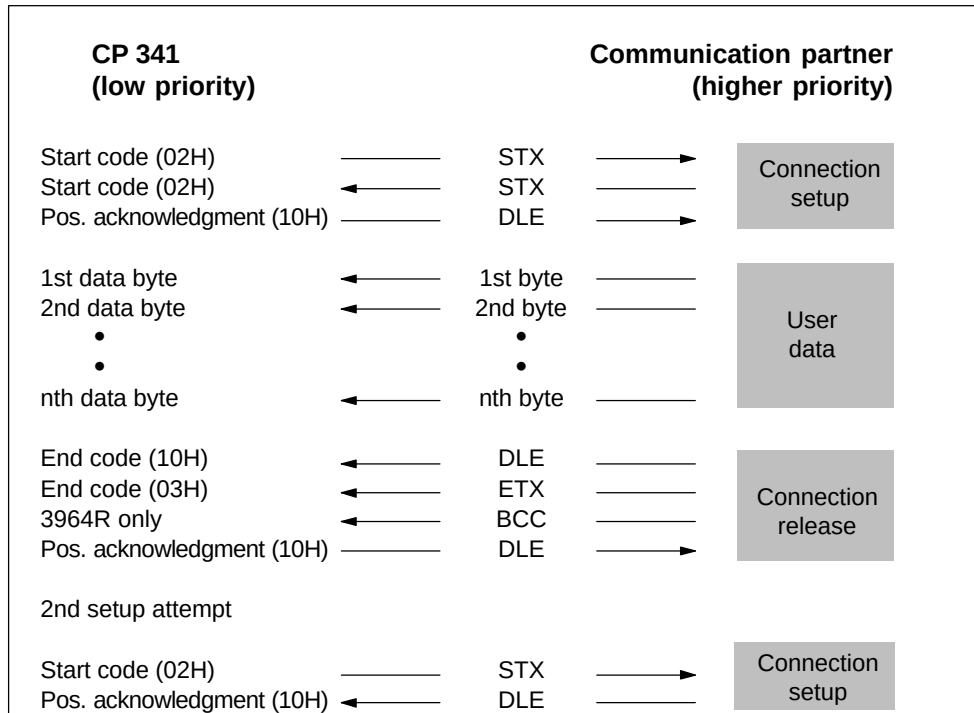


Figure 2-8 Data Traffic during an Initialization Conflict

If a device responds to the communication partner's send request (code STX) within the acknowledgment delay time (ADT) by sending the code STX instead of the acknowledgment DLE or NAK, an initialization conflict occurs. Both devices want to execute a send request. The device with the lower priority withdraws its send request and responds with the code DLE. The device with the higher priority sends its data in the manner described above. Once the connection has been released, the lower-priority device can execute its send request.

To be able to resolve initialization conflicts you must parameterize different priorities for the communication partners.

Procedure Errors

The procedure recognizes both errors which are caused by the communication partner and errors caused by faults on the line.

In both cases, the procedure makes repeated attempts to send/receive the data block correctly. If this is not possible within the maximum number of transmission attempts set (or if a new error status occurs), the procedure aborts the send or receive process. It reports the error number of the first recognized error and returns to idle mode. These error messages are displayed in the STATUS output of the FB.

If the system program frequently reports an error number in the STATUS output of the FB for send and receive repetitions, this implies occasional disturbances in the data traffic. The large number of transmission attempts compensates for this, however. In this case you are advised to check the transmission connection for possible sources of interference, because frequent repetitions reduce the user-data rate and integrity of the transmission. The disturbance could also be caused, however, by a malfunction on the part of the communication partner.

If the receive connection is interrupted, an error message is displayed at the STATUS output of the function block. No repeat is started. The BREAK status in the STATUS output of the FB is automatically reset as soon as the connection is restored on the line.

For every recognized transmission error (lost character, frame or parity error), a standard number is reported, regardless of whether the error was detected during sending or receiving of a data block. The error is only reported, however, following unsuccessful repetitions.

3964(R) Procedure Start-Up

The figure below illustrates the start-up of the 3964(R).

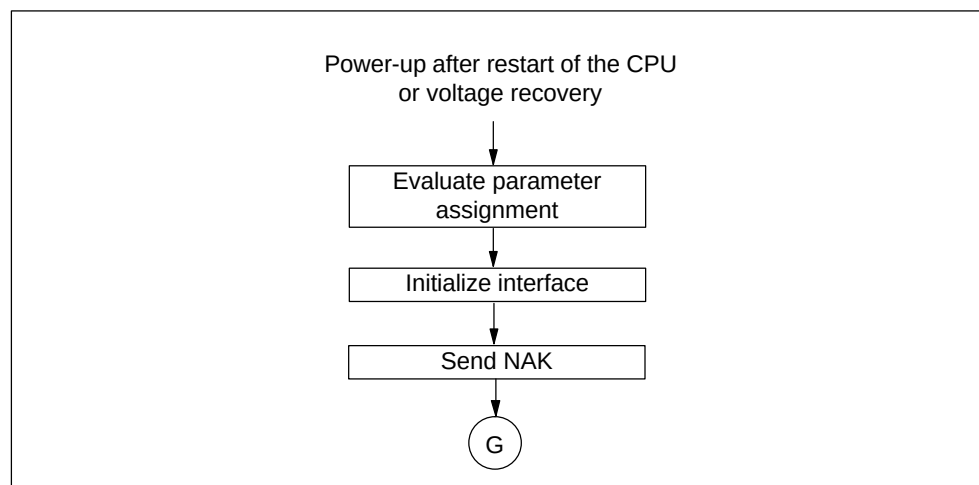


Figure 2-9 Flow Diagram of the Start-Up of the 3964(R) Procedure

Sending with the 3964(R) Procedure

The figure below illustrates sending with the 3964(R) procedure.

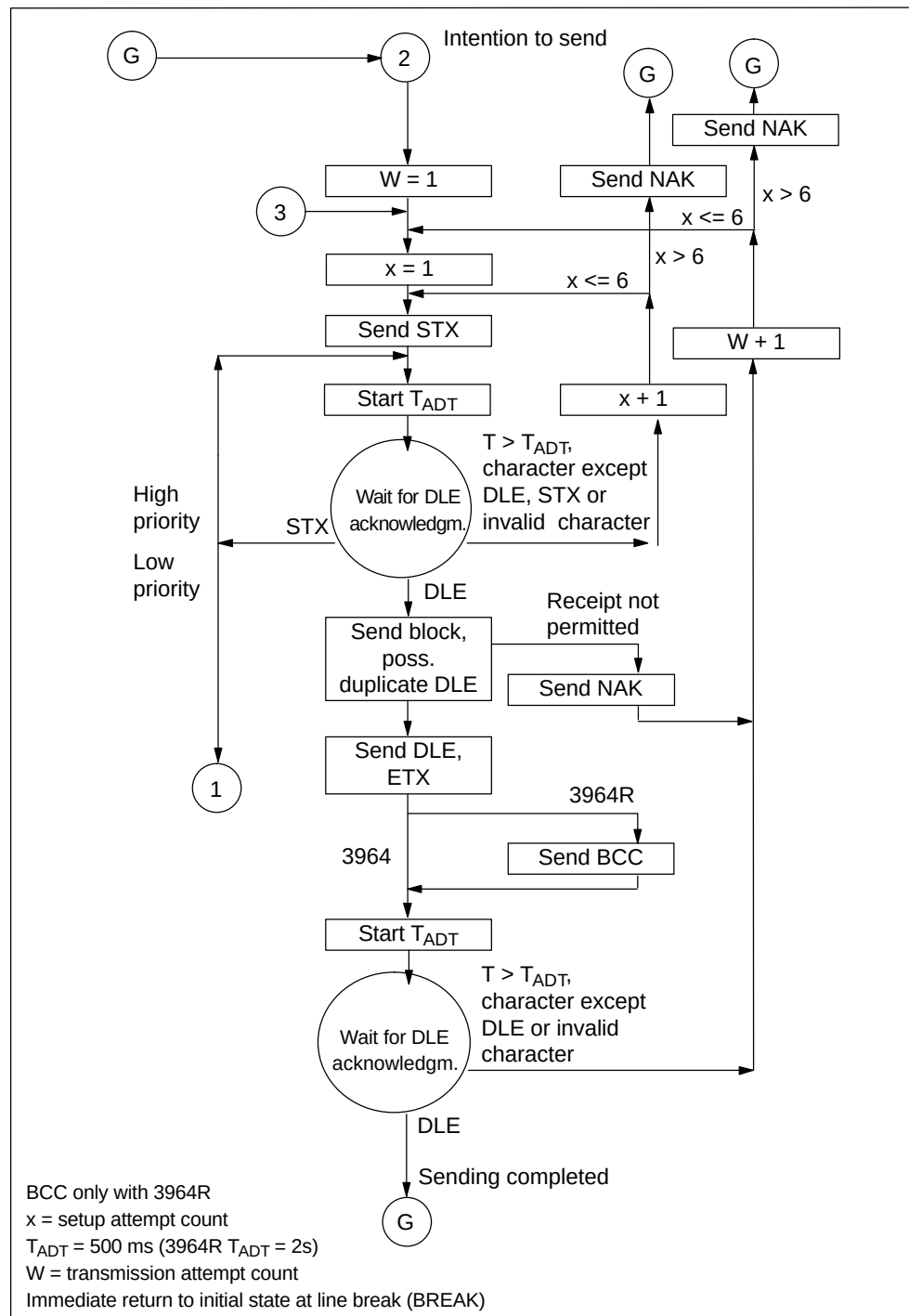


Figure 2-10 Flow Diagram of Sending with the 3964(R) Procedure

Receiving with the 3964(R) Procedure (Part 1)

The figure below illustrates receiving with the 3964(R) procedure.

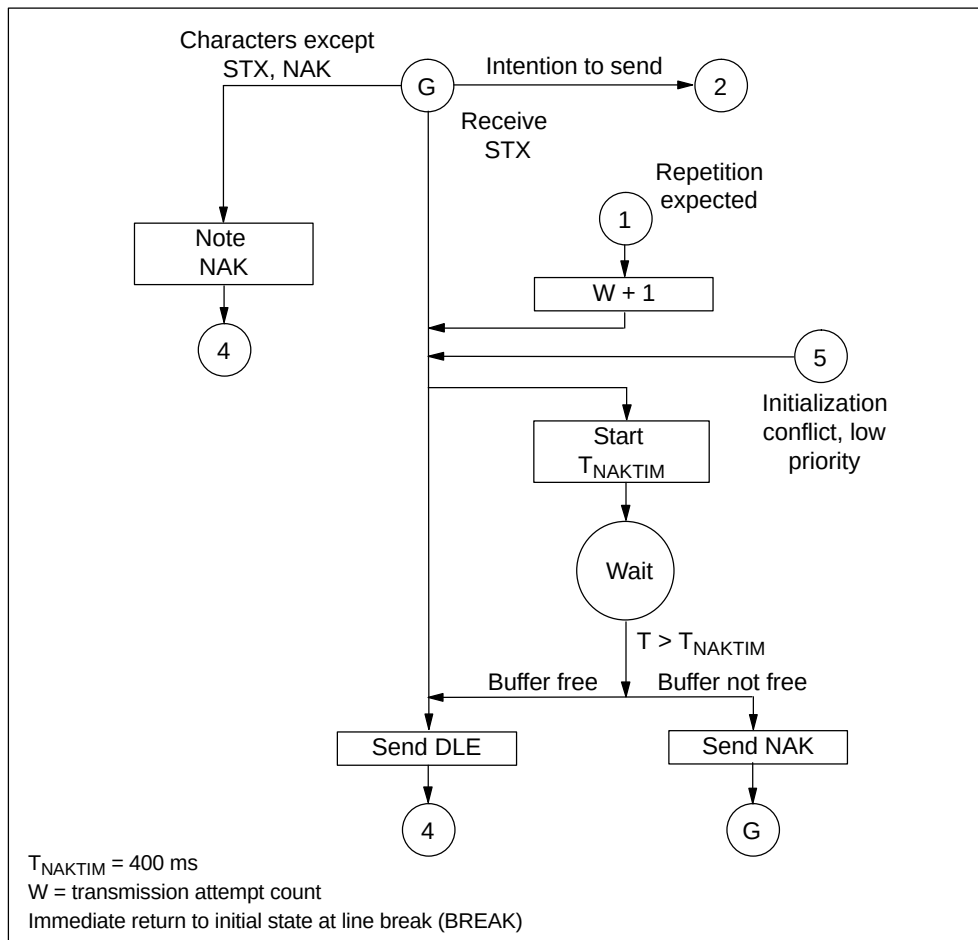


Figure 2-11 Flow Diagram of Receiving with the 3964(R) Procedure (Part 1)

Receiving with the 3964(R) Procedure (Part 2)

The figure below illustrates receiving with the 3964(R) procedure.

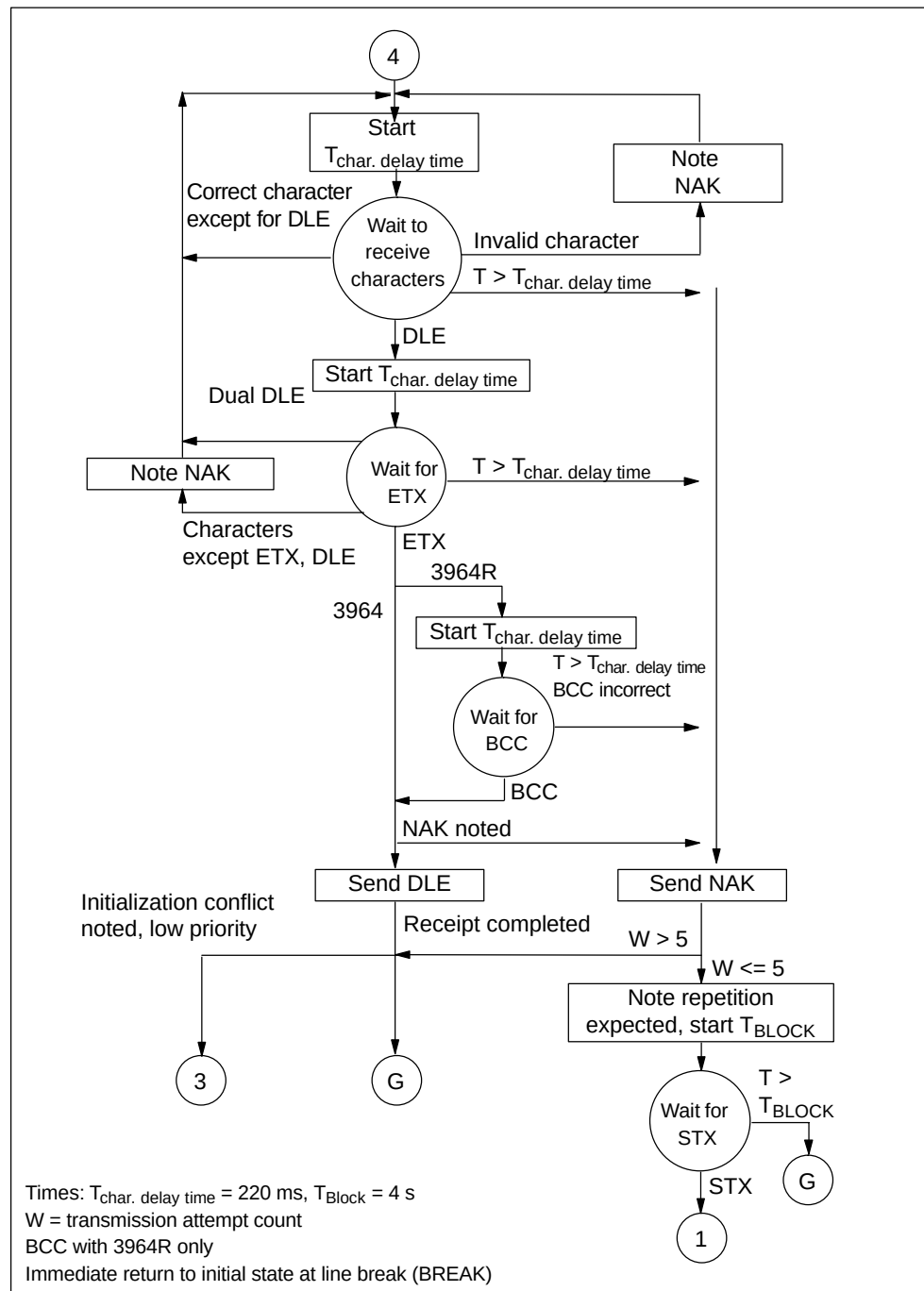


Figure 2-12 Flow Diagram of Receiving with the 3964(R) Procedure (Part 2)

2.2.3 Data Transmission with the RK 512 Computer Connection

The RK 512 computer connection controls data transmission via a point-to-point connection between the CP 341 and a communication partner.

Unlike the 3964(R) procedure, the RK 512 includes not only the physical layer (layer 1), and the data-link layer (layer 2), but also the transport layer (layer 4) of the ISO reference model. The RK 512 computer connection also offers higher data integrity and better addressing.

Response Message Frame

The RK 512 computer connection answers every command message frame it receives correctly with a response message frame to the CPU (transport layer). This allows senders to check whether their data has arrived undamaged at the CPU or whether the data they require is available on the CPU.

Command Message Frame

Command message frames are either SEND or FETCH message frames.

How to initiate a SEND or FETCH message frame is described in Chapter 6.

SEND Message Frame

A SEND message frame is created when the CP 341 sends a command message frame with user data, and the communication partner replies with a response message frame without user data.

FETCH Message Frame

A FETCH message frame is created when the CP 341 sends a command message frame with user data, and the communication partner replies with a response message frame with user data.

Continuation Message Frame

If the volume of data exceeds 128 bytes, SEND and FETCH message frames are automatically accompanied by continuation message frames.

Message frame header

Each message frame with the RK 512 begins with a message frame header. It can contain message frame IDs, information on the data destination and source and an error number.

Structure of the Message Frame Header

The table below indicates the structure of the header of the command message frame.

Table 2-1 Structure of Message Frame Header (RK 512)

Byte	Meaning
1	Message frame ID in command message frames (00H), in continuation command message frames (FFH)
2	Message frame ID (00H)
3	'A' (41H) – for SEND request with destination DB or 'O' (4FH) – for SEND request with destination DX or 'E' (45H) – for FETCH request
4	Data to be transmitted consists of (when sending only 'D' is possible): 'D' (44H) =data block 'X' (58H)= extended data block 'E' (45H) =input bytes 'A' (41H) = output bytes 'M' (4DH) =memory bytes T' (54H) = time cells 'C' (5AH) =counter cells
5	Data destination of SEND request or data source of FETCH request e.g. byte 5 = DB no., byte 6 = DW no. ¹
6	
7	Length of high byte Length of data to be transmitted according to type in bytes or
8	Length of low byte words
9	Byte number of the interprocessor communication flag: FFH is displayed if you have not specified an interprocessor communication flag.
10	Bits 0 to 3: Bit number of the interprocessor communication flag; if you have not specified an interprocessor communication flag, the protocol enters FH here. Bits 4 to 7: CPU number (digit from 1 to 4); 0H is displayed here if you have not specified a CPU No. (0) but you have specified an interprocessor communication flag; FH is displayed here if you have not specified a CPU No. or an interprocessor communication flag.

¹ RK 512 addressing describes the data source and destination with word limits.
Conversion to byte addresses in SIMATIC S7 is automatic.

The letters in bytes 3 and 4 are ASCII characters.

The header of the continuation command message frame consists of bytes 1 to 4 only.

Response Message Frame

Once the command message frame has been transmitted, the RK 512 waits for a response message frame from the communication partner within the monitoring time. The length of the monitoring time depends on the transmission speed (baud rate).

- 300 to 76800 baud 10 seconds

Structure and Contents of the Response Message Frame

The response message frame consists of 4 bytes and contains information on the progress of the request.

Byte	Meaning
1	Message frame ID in response message frames (00H), in continuation response message frames (FFH)
2	Message frame ID (00H)
3	Displays 00H
4	Error number of the communication partner (see Section 8.4) in the response message frame:*
	00H if transmission was error-free > 00H error number

* The error number in the response message frame automatically causes an event number at the STATUS output of the function blocks (see section 8.3).

Sending Data with RK 512

The figure below shows the transmission process when sending data with a response message frame using the RK 512 computer connection.

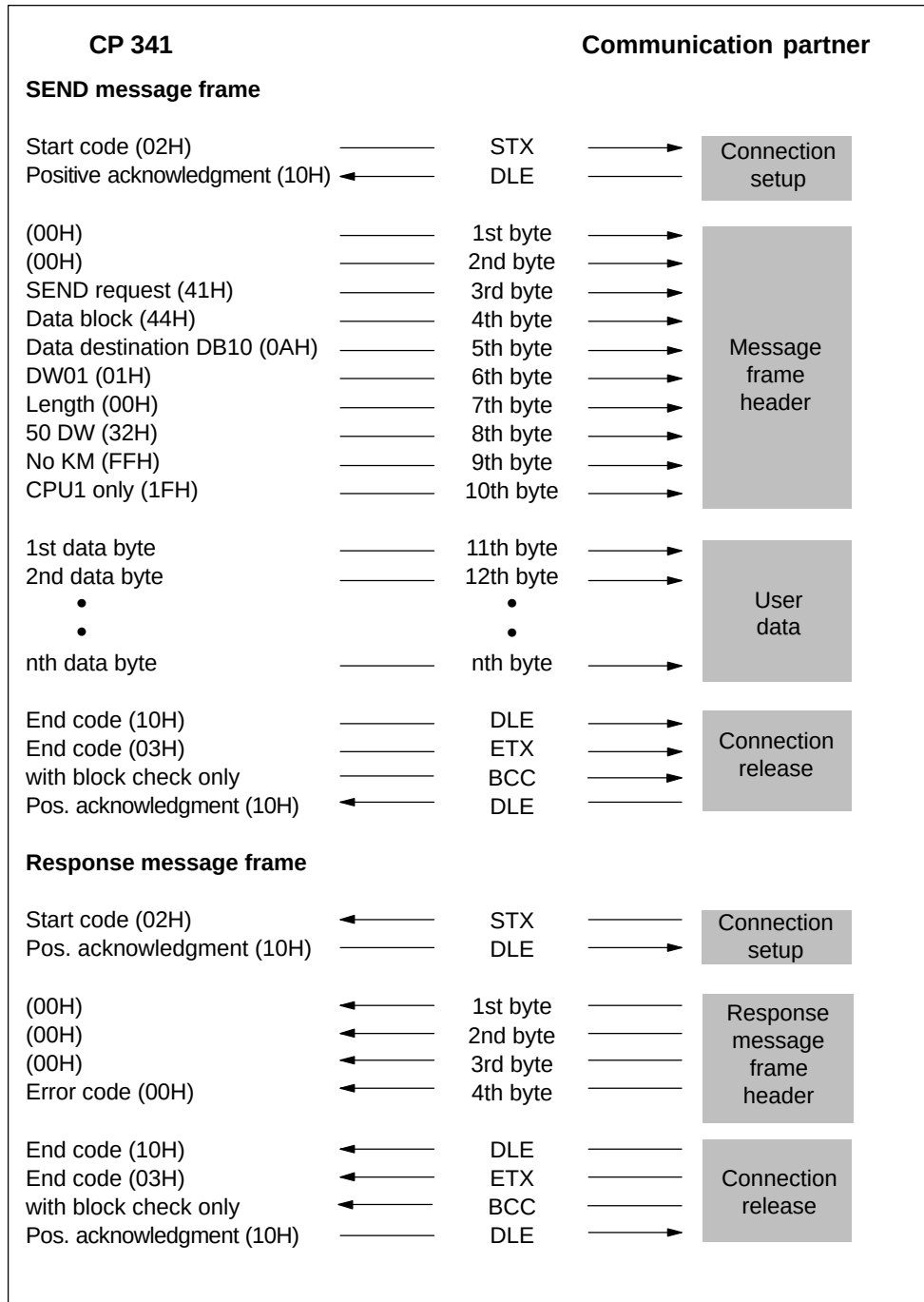


Figure 2-13 Data Traffic when Sending with a Response Message Frame

Send data

The SEND request is executed in the following sequence:

- **Active partner**
Sends a SEND message frame containing the message frame header and data.
- **Passive partner**
Receives the message frame, checks the header and the data, and acknowledges it with a response message frame after passing the data on to the CPU.
- **Active partner**
Receives the response message frame.
Sends user data.
If the volume of user data exceeds **128 bytes**, the active partner sends a continuation SEND message frame.
- **Passive partner**
Receives the continuation SEND message frame, checks the header and the data, and acknowledges it with a continuation response message frame after passing the data on to the CPU.

Note

If the CPU receives an errored SEND message frame or if an error has occurred in the message frame header, the communication partner enters an error number in the 4th byte of the response message frame. This does not apply in the case of protocol errors.

Continuation SEND Message Frame

A continuation SEND message frame is started if the volume of data exceeds **128 bytes**. The process is the same as for SEND message frames.

If more than 128 bytes are sent, the extra bytes are automatically transmitted in one or more continuation message frames.

The figure below shows the data transmission process when sending a continuation SEND message frame with a continuation response message frame.

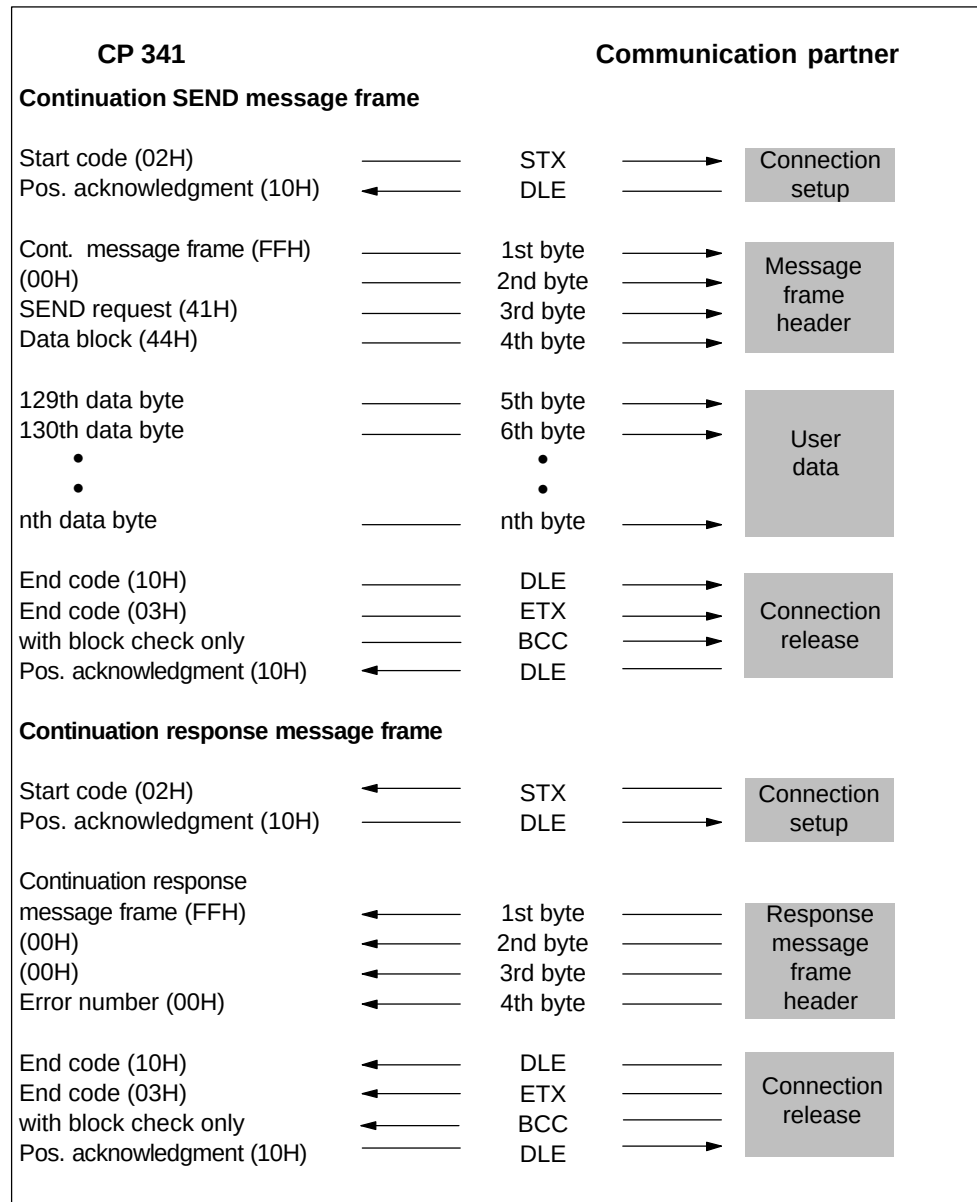


Figure 2-14 Sequence of a Continuation SEND Message Frame with a Continuation Response Message Frame

Fetching Data with RK 512

The figure below shows the transmission process when fetching data with a response message frame using the RK 512 computer connection.

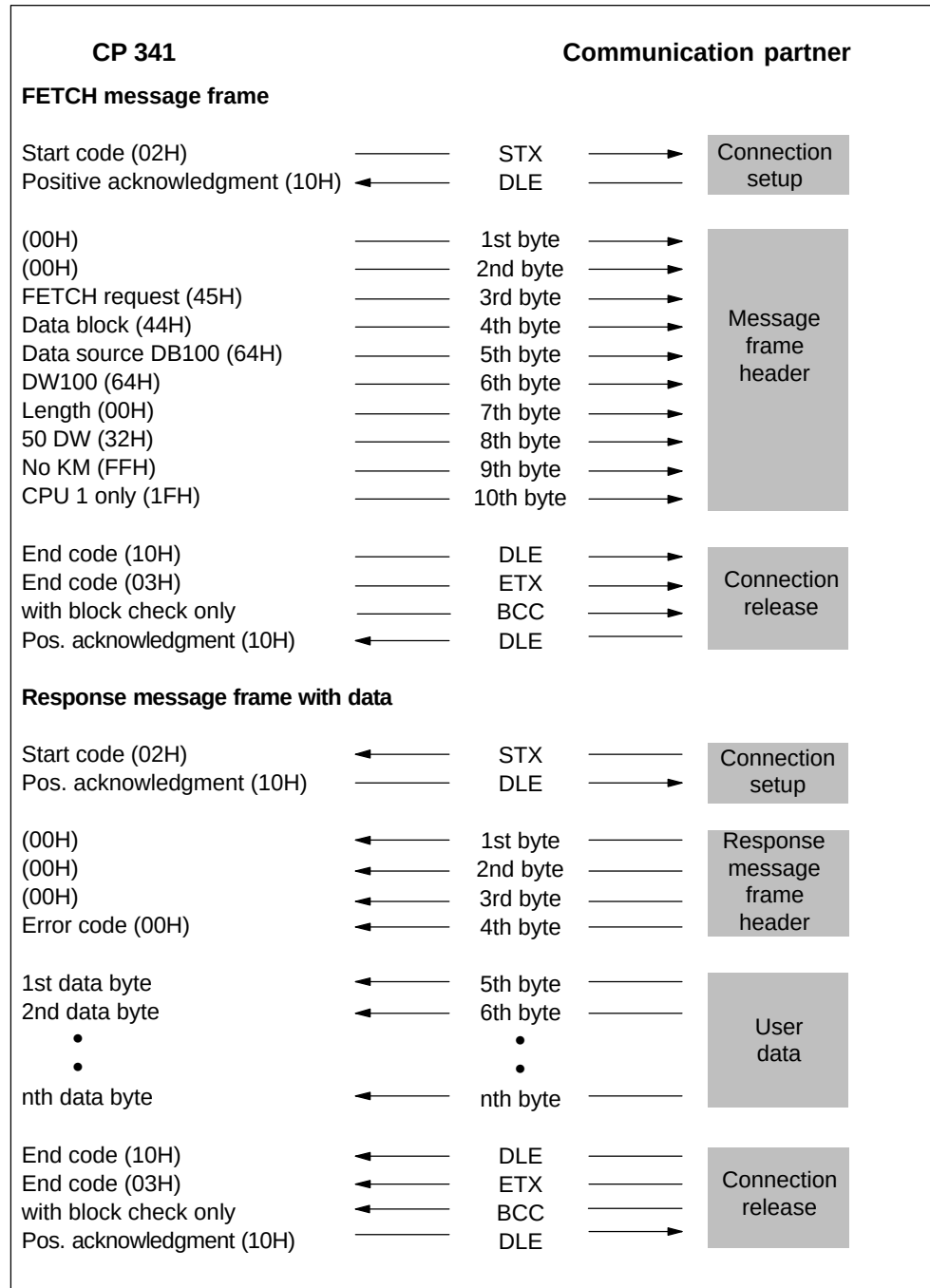


Figure 2-15 Data Traffic when Fetching with a Response Message Frame

Fetching Data

The FETCH request is executed in the following sequence:

- **Active partner**

Sends a FETCH message frame containing the header.

- **Passive partner**

Receives the message frame, checks the header, fetches the data from the CPU, and acknowledges this with a response message frame containing the data.

- **Active partner**

Receives the response message frame.

If the volume of user data exceeds **128 bytes**, the active partner sends a continuation FETCH message frame containing bytes 1 to 4 of the header.

- **Passive partner**

Receives the continuation FETCH message frame, checks the header, fetches the data from the CPU, and acknowledges this with a continuation response message frame containing further data.

If there is an error number (not equal to 0) in the 4th byte, the response message frame does not contain any data.

If more than 128 bytes are requested, the extra bytes are automatically fetched in one or more continuation message frames.

Note

If the CPU receives an errored FETCH message frame or if an error has occurred in the message frame header, the communication partner enters an error number in the 4th byte of the response message frame. This does not apply in the case of protocol errors.

Continuation FETCH Message Frame

The figure below shows the transmission process when fetching data with a continuation response message frame.

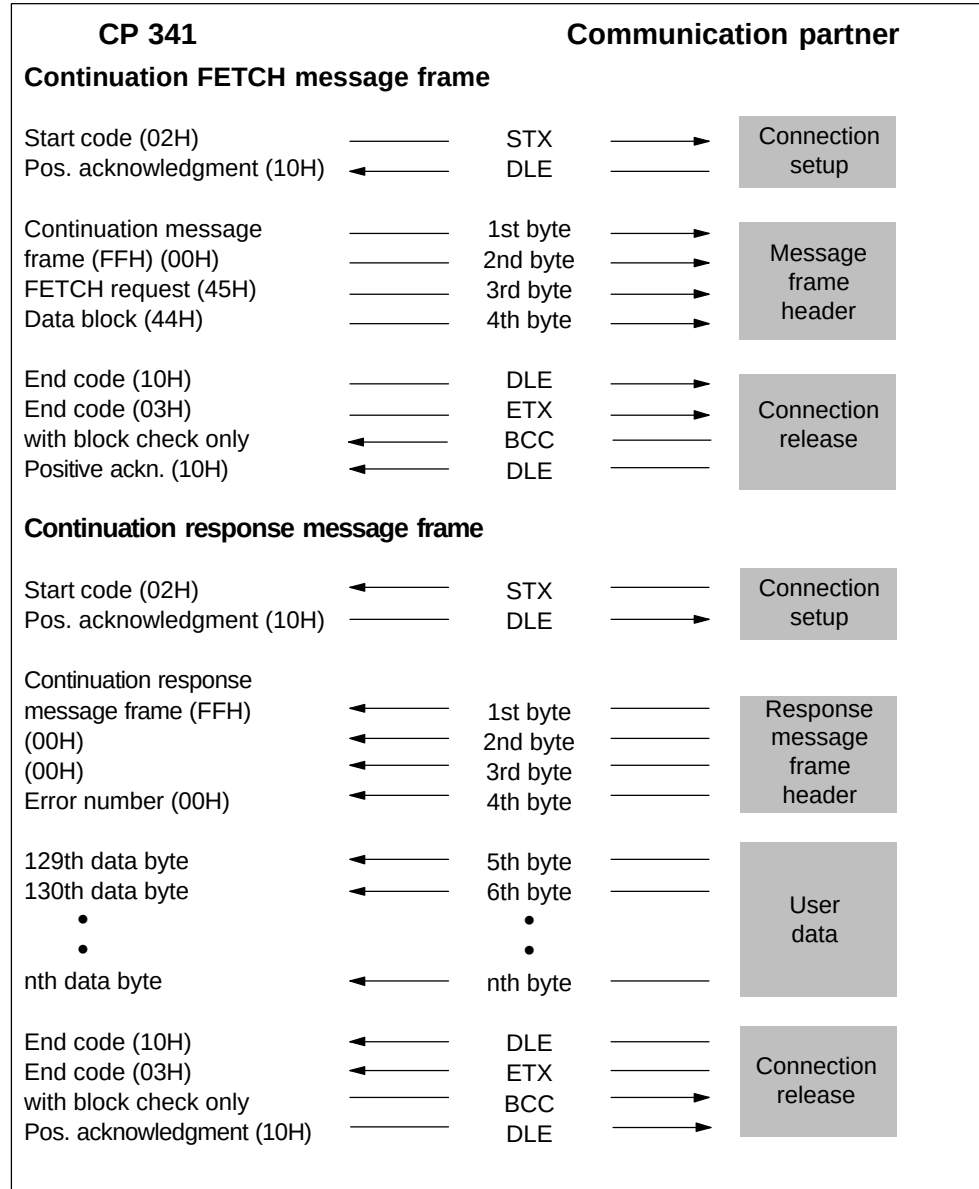


Figure 2-16 Sequence of a Continuation FETCH Message Frame with a Continuation Response Message Frame

Quasi-Full-Duplex Operation

Quasi full-duplex operation means: the partners can send command and response message frames at any time as long as the other partner is not sending. The maximum nesting depth for command and response message frames is “1”. The next command message frame, therefore, cannot be processed until the previous one has been answered with a response message frame.

It is possible under certain circumstances – if both partners want to send – to transmit a SEND message frame from the partner before the response message frame. For example, if a SEND message frame from the partner was entered in the output buffer of the CP 341 before the response message frame.

In the following figure the continuation response message frame to the first SEND message frame is not sent until after the **partner's SEND message frame**.

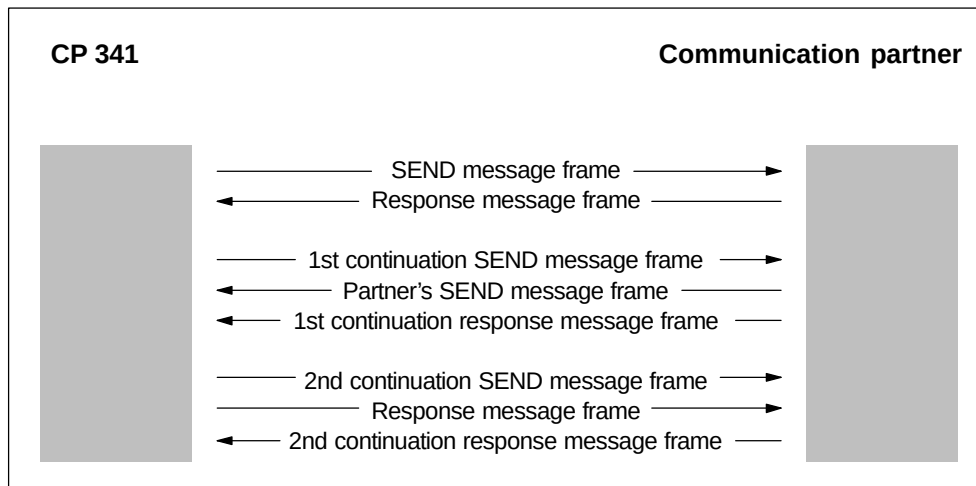


Figure 2-17 Quasi-Full-Duplex Operation

RK 512 CPU Requests

The figure below shows the processes involved in the RK 512 computer connection when CPU requests are made.

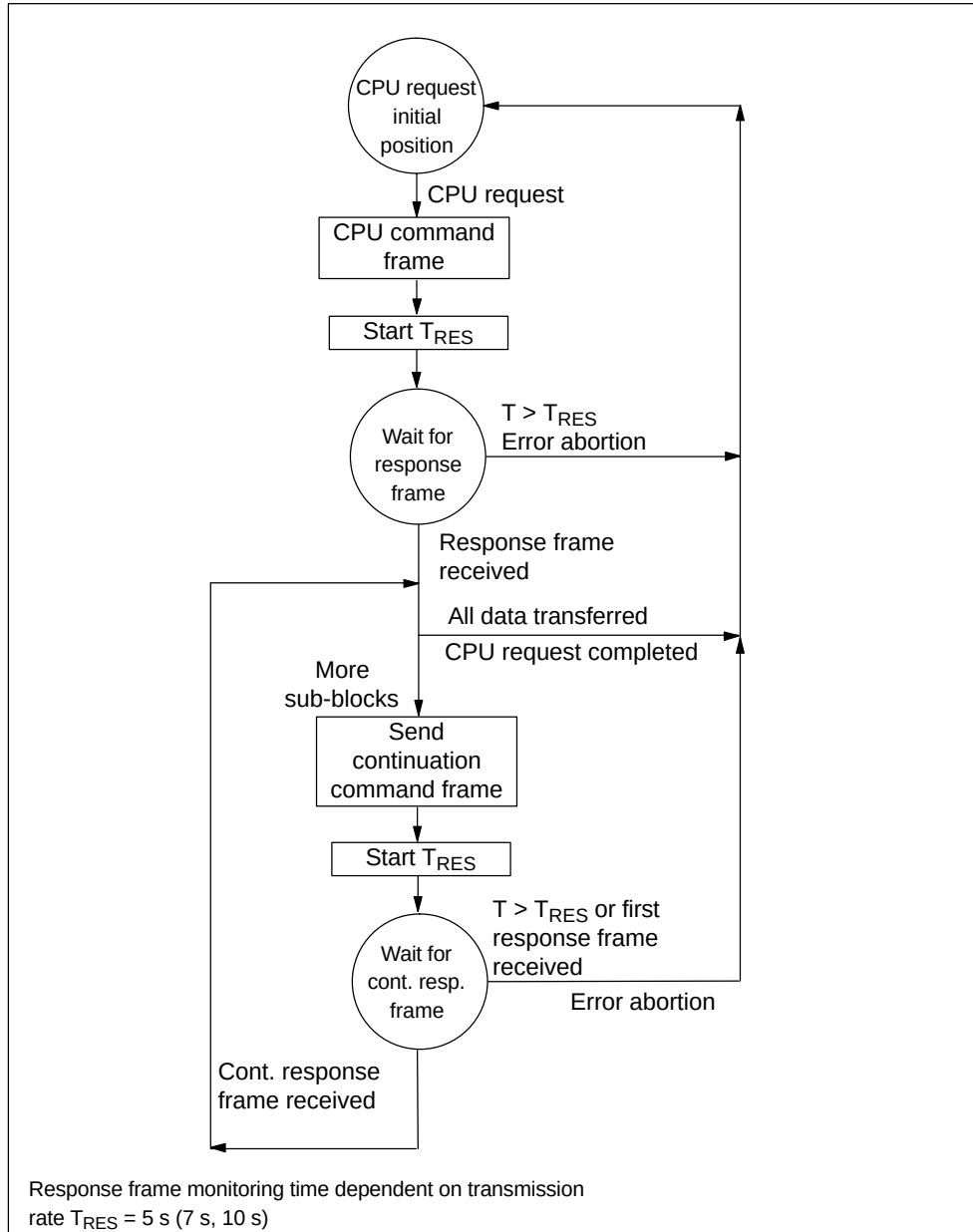


Figure 2-18 Flow Diagram of Data Transmission with the RK 512 When CPU Requests Are Made

RK 512 Partner Requests

The figure below shows the processes involved in the RK 512 computer connection when partner requests are made.

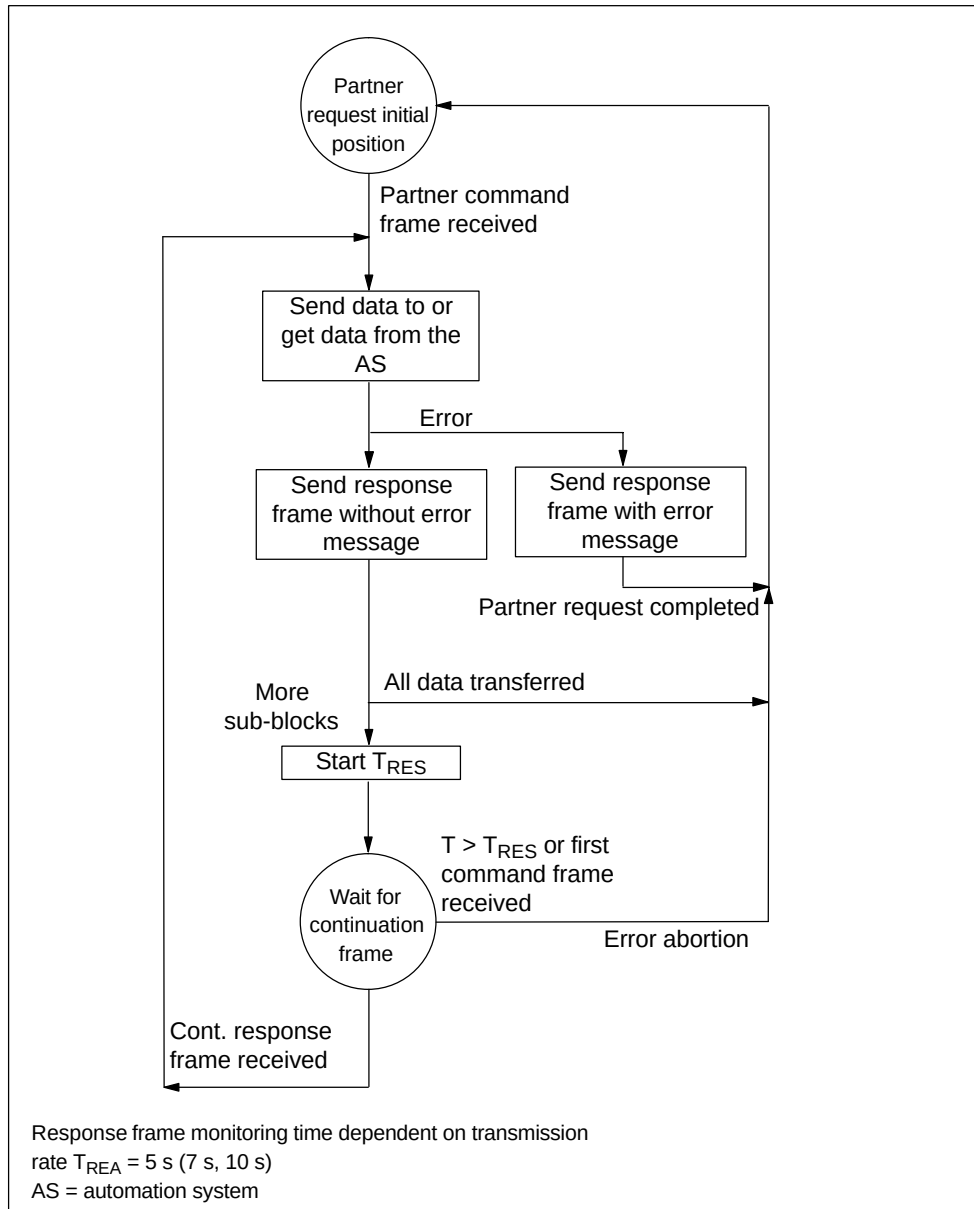


Figure 2-19 Flow Diagram of Data Transmission with the RK 512 When Partner Requests Are Made

2.2.4 Data Transmission with the ASCII Driver

The ASCII driver controls data transmission via a point-to-point connection between the CP 341 and a communication partner. This driver contains the physical layer (layer 1 of the ISO reference model.)

The structure of the message frames is left open through the S7 user passing on the complete send message frame to the CP 341. For the receive direction, the end criterion of a message must be parameterized. The structure of the send message frames may differ from that of the receive message frames.

The ASCII driver allows data of any structure (all printable ASCII characters as well as all other characters from 00 through FFH (with 8 data bit character frames) or from 00 through 7FH (with 7 data bit character frames) to be sent and received.

Sending Data with the ASCII Driver

When you send data, you specify the number of user data bytes to be transferred in the "LEN" parameter of the call for the P_SND_RK function block.

When you work with the end criterion "**Expiry of Character Delay Time**" when receiving data, the ASCII driver pauses between two message frames when sending. You can call the P_SND_FK FB at any time, but the ASCII driver does not begin its output until a period longer than the parameterized character delay time has elapsed since the last message frame was sent.

If you work with the "**End-of-Text Character**" criterion, you have a choice of three options:

- Send up to and including the end-of-text character

The end-of-text character must be included in the data to be sent. Data is sent only up to and including the end-of-text character, even if the data length specified in the FB is longer.

- Send up to length parameterized at the FB

Data is sent up to the length parameterized at the FB. The last character must be the end-of-text character.

- Send up to the length parameterized at the FB and automatically append the end-of-text character or characters

Data is sent up to the length parameterized at the FB. The end-of-text character is automatically appended, in other words the end-of-text characters must not be included in the data to be sent. 1 or 2 characters more than the number specified at the FB are sent to the partner, depending on the number of end-of-text characters.

When you work with the end criterion "**Fixed Message Frame Length**", the number of data bytes transferred in the send direction is as specified for the "LEN" parameter of the P_SND_RK . The number of data bytes transferred in the receive direction, i.e. in the receive DB, is as specified at the receiver using the "fixed message frame length" parameter in the parameterization interface. The two parameter settings must be identical, in order to ensure correct data traffic. A pause equal to the length of the character delay time is inserted between two message frames when sending, to allow the partner to synchronize (recognize start of message frame).

If some other method of synchronization is used, the pause in sending can be deactivated by means of the parameter assignment interface.

Note

When XON/XOFF flow control is parameterized, the user data must not contain any of the parameterized XON or XOFF codes. The default settings are DC1 = 11H for XON and DC3 = 13H for XOFF.

Send data

The figure below illustrates a send operation.

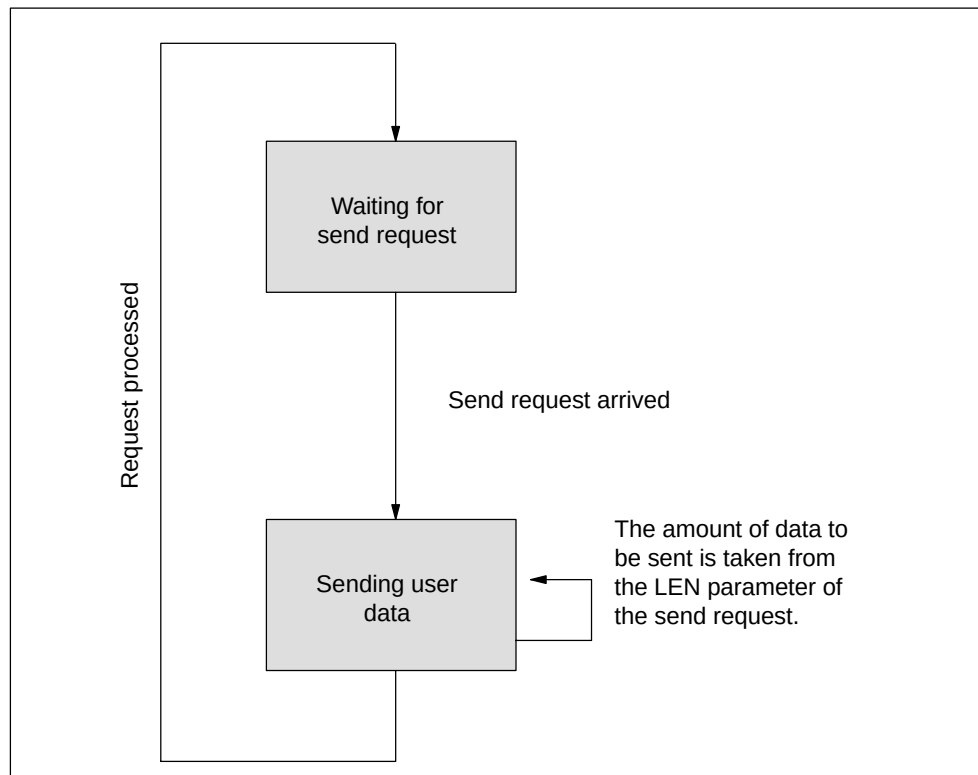


Figure 2-20 Sequence of a Send Operation

Receiving Data with the ASCII Driver

For data transmission using the ASCII driver you can choose between three different end criteria for data reception. The end criterion defines when a complete message frame is received. The possible end criteria are as follows:

- **On Expiry of Character Delay Time**

The message frame has neither a fixed length nor a defined end-of-text character; the end of the message is defined by a pause on the line (expiry of character delay time).

- **On Receipt of End Character(s)**

The end of the message frame is marked by one or two defined end-of-text characters.

- **On Receipt of Fixed Number of Characters**

The length of the receive message frames is always identical.

Code Transparency

The code transparency of the procedure depends on the selection of the parameterized end criterion and the flow control:

- With one or two end-of-text characters
 - not code-transparent
- When end criterion is character delay time or fixed message frame length
 - code-transparent
- Code-transparent operation is not possible when XON/XOFF flow control is used.

Code-transparent means that any character combinations can occur in the user data without the end criterion being recognized.

End Criterion "Expiry of Character Delay Time"

When data is received, the end of the message frame is recognized when the character delay time expires. The received data is accepted from the CPU.

In this case the character delay time must be set such that it easily expires between two consecutive message frames. But it should be long enough so that the end of the message frame is not falsely identified whenever the partner in the connection takes a send pause within a message frame.

The figure below illustrates a receive operation with the end criterion "expiry of character delay time".

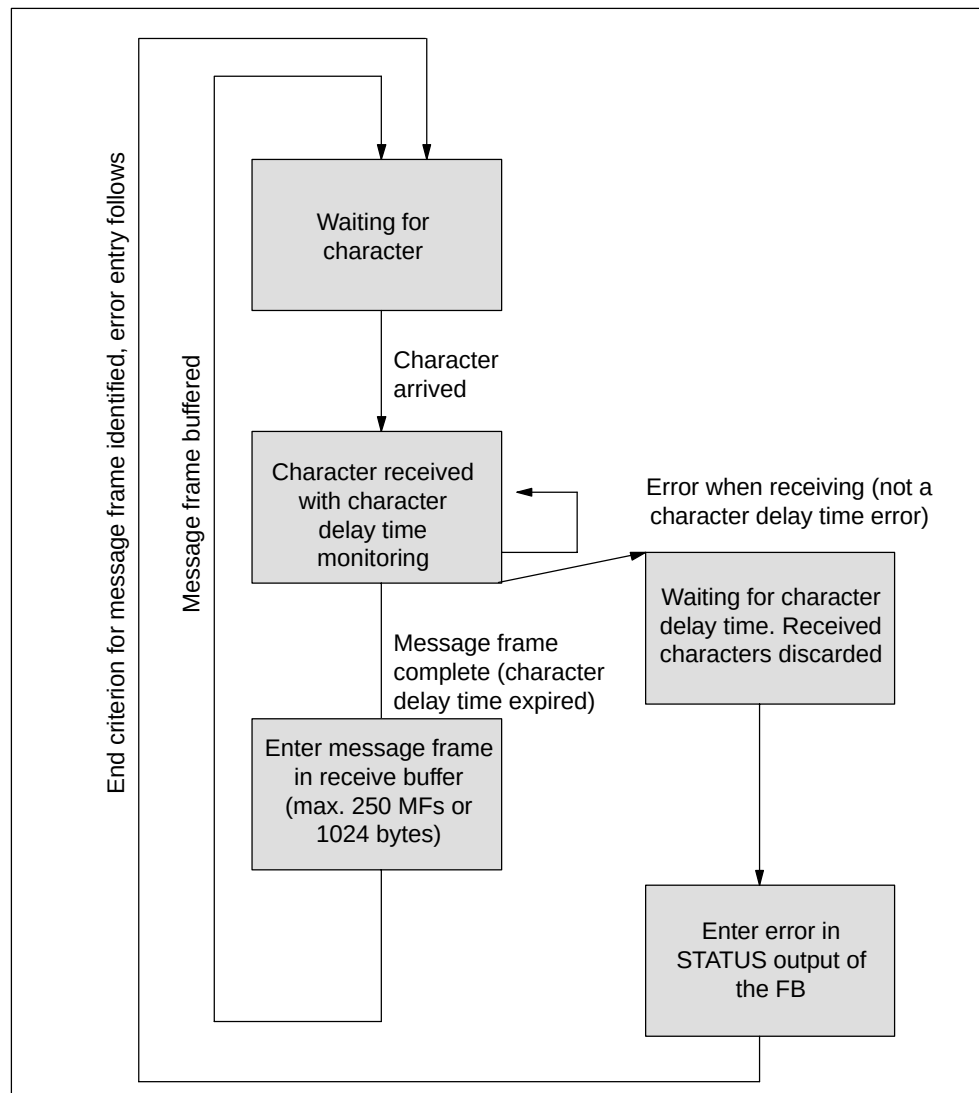


Figure 2-21 Sequence of Receive Operation with End Criterion "Expiry of Character Delay Time"

End Criterion "End-of-Text Character"

When data is received, the end of the message frame is recognized when the parameterized end-of-text character(s) arrive. The received data including the end-of-text character(s) is accepted from the CPU.

If the character delay time expires while the message frame is being received, the receive operation is terminated. An error message is issued and the message frame fragment is discarded.

If you are working with end-of-text characters, transmission is not code-transparent, and you must make sure that the end code(s) are not included in the user data of the user.

Note the following when the last character in the received message frame is not the end-of-text character.

- End-of-text character elsewhere in the message frame:

All characters including the end-of-text character are entered in the receive DB. The characters following the end-of-text character

- are discarded if the character delay time expires at the end of the message frame.
- are blended with the next message frame if a new message frame is received before the character delay time expires.

- End-of-text character **not** included in message frame:

The message frame

- is discarded if the character delay time expires at the end of the message frame.
- is blended with the next message frame if a new message frame is received before the character delay time expires.

The figure below illustrates a receive operation with the end criterion "end-of-text character".

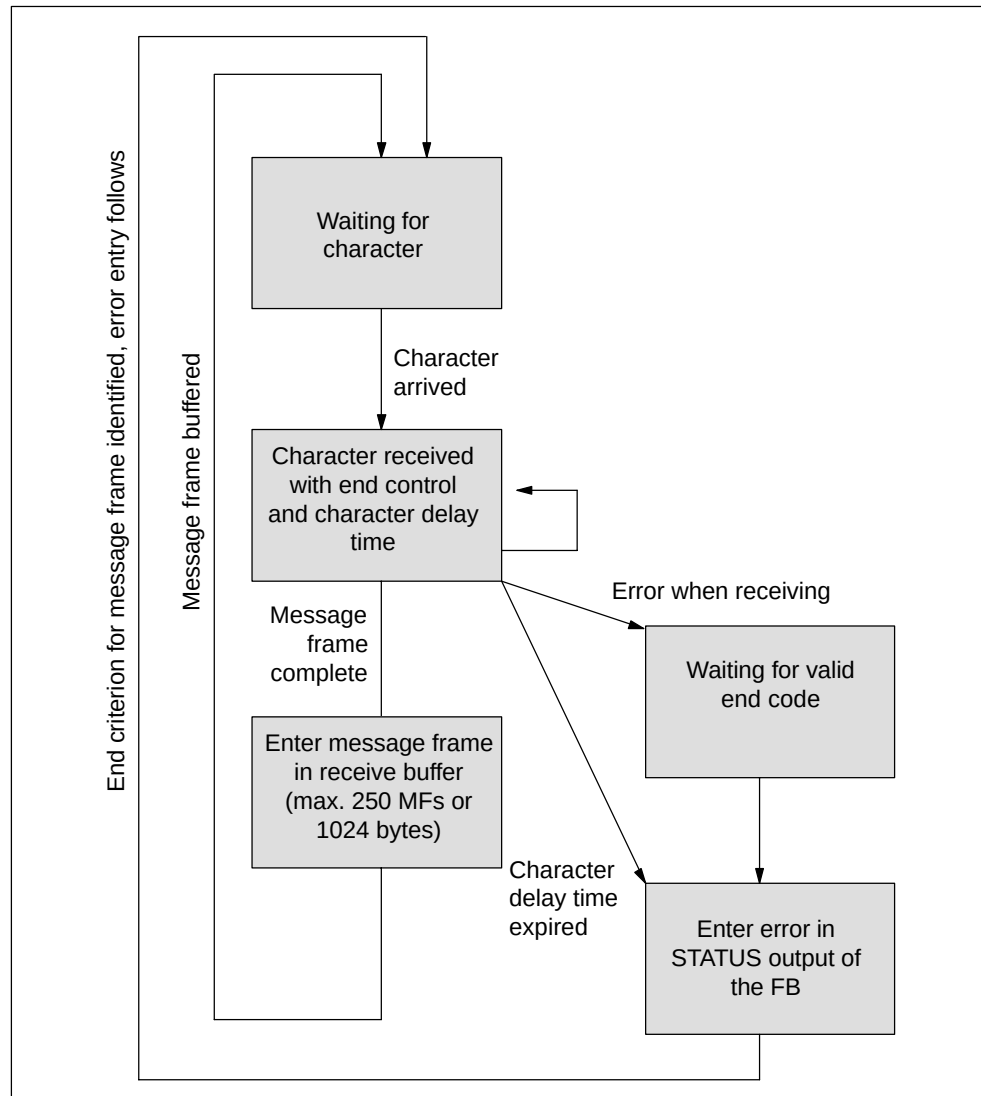


Figure 2-22 Sequence of Receive Operation with End Criterion "End-of-Text Character"

End Criterion "Fixed Message Frame Length"

When data is received, the end of the message frame is recognized when the parameterized number of characters has arrived. The received data is accepted from the CPU.

If the character delay time expires before the parameterized number of characters has been reached, the receive operation is terminated. An error message is issued and the message frame fragment is discarded.

Note the following if the message frame length of the received characters does not match the parameterized fixed message frame length:

- Message frame length of received characters greater than parameterized fixed message frame length:

All characters received after the parameterized fixed message frame length is reached

- are discarded if the character delay time expires at the end of the message frame.
- are blended with the next message frame if a new message frame is received before the character delay time expires.

- Message frame length of received characters less than parameterized fixed message frame length:

The message frame

- is discarded if the character delay time expires at the end of the message frame.
- is blended with the next message frame if a new message frame is received before the character delay time expires.

The figure below illustrates a receive operation with the end criterion "fixed message frame length".

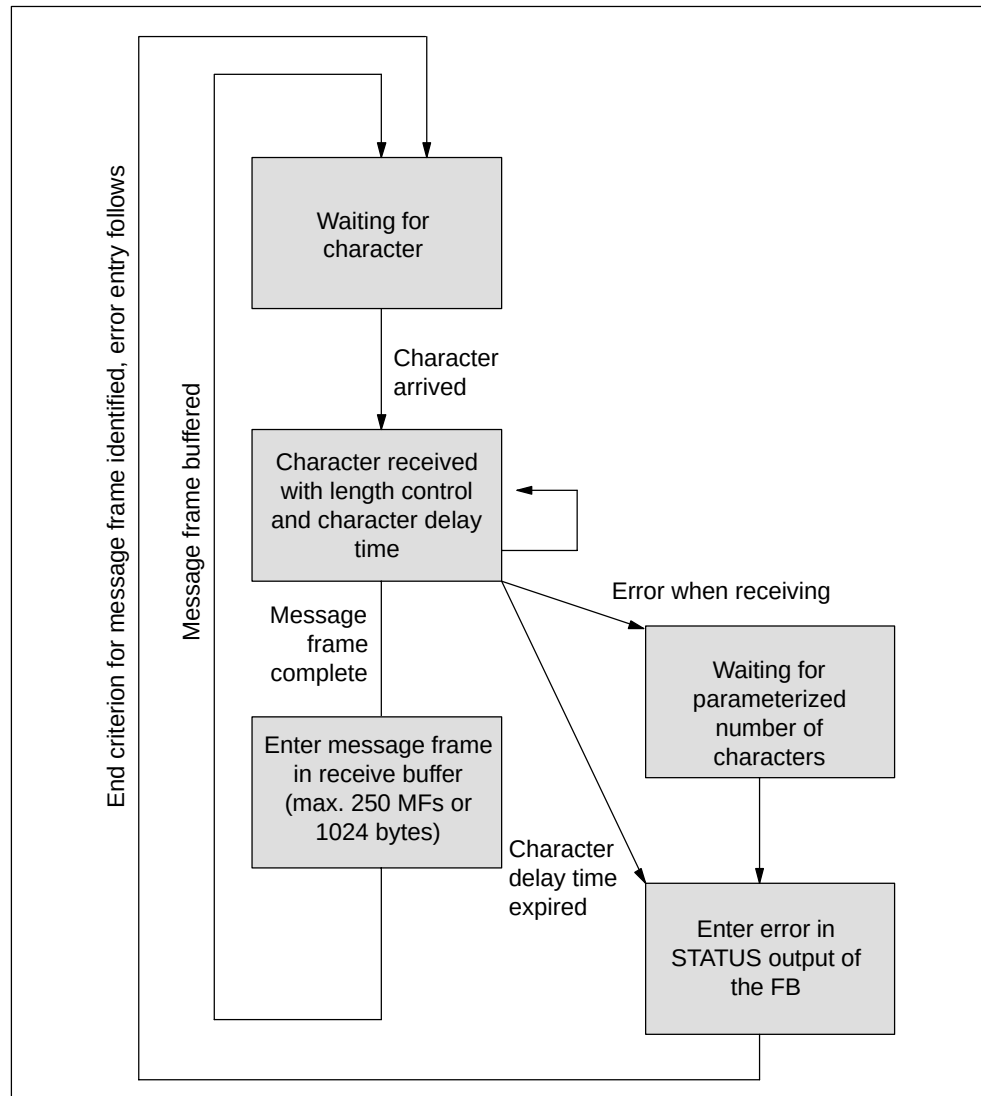


Figure 2-23 Sequence of Receive Operation with End Criterion "Fixed Message Frame Length"

Receive Buffer on CP 341

The CP 341 receive buffer accommodates 1024 bytes. At parameterization you can specify whether overwriting of data in the receive buffer should be prevented. You can also specify the value range (1 to 250) for the number of buffered receive message frames.

The receive buffer on the CP 341 is a ring buffer:

- If two or more message frames are received into the receive buffer of the CP 341, the rule is that the oldest message frame is always transferred by the CP 341 to the CPU.
- If you only ever want to send the most recent message frame to the CPU, you must parameterize the value "1" for the number of buffered message frames **and** deactivate the overwrite protection.

Note

If the constant reading of the receive data in the user program is interrupted for a while, you may find that when the receive data is requested again, the CP 341 transfers old message frames to the CPU before it transfers the latest one.

The old message frames are those on their way when transmission between the CP 341 and the CPU was interrupted, or which had already been received by the FB.

RS 485 Mode

When you run the ASCII driver in RS 485 mode (half-duplex, two-wire mode), you must take steps in the user program to ensure that only one user sends data at any one time. If two users send data simultaneously, the message frame is corrupted.

RS 232C Secondary Signals

The following RS 232C secondary signals exist on the CP 341 (see also Appendix B):

- | | |
|-----------------------|--|
| • DCD (input) | Data carrier detect; Data carrier detected |
| • DTR (output) | Data terminal ready; CP 341 ready for operation |
| • DSR (input) | Data set ready; Communication partner ready for operation |
| • RTS (output) | Request to send; CP 341 ready to send |
| • CTS (input) | Clear to send; Communication partner can receive data from the CP 341 (response to RTS = ON of the CP 341) |
| • RI (input) | Ring Indicator; Indication of an incoming call |

When the CP 341-RS 232C is switched on, the output signals are in the OFF state (inactive).

You can parameterize the way in which the DTR/DSR and RTS/CTS control signals are used with the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface or control them by means of function calls (FCs) in the user program.

Using the RS 232C Secondary Signals

The RS 232C secondary signals can be used as follows:

- When the automatic use of all RS 232C secondary signals is parameterized
- When data flow control (RTS/CTS) is parameterized
- By means of the V24_STAT and V24_SET functions (FCs)

Note

When automatic use of the RS 232C secondary signals is parameterized, neither RTS/CTS data flow control nor RTS and DTR control by means of the V24_SET FC are possible.

When RTS/CTS data flow control is parameterized, RTS control by means of the V24_SET FC is not possible.

On the other hand, it is always possible to read all RS 232C secondary signals by means of the V24_STAT FC.

The sections that follow describe how the control and evaluation of the RS 232C secondary signals is handled.

Automatic Use of the Secondary Signals

The automatic use of the RS 232C secondary signals on the CP 341 is implemented as follows:

- As soon as the CP 341 is switched by means of parameterization to an operating mode with automatic use of the RS 232C secondary signals, it switches the RTS line to OFF and the DTR line to ON (CP 341 ready for use).

Message frames cannot be sent and received until the DTR line is set to ON. As long as DTR remains set to OFF, no data is received via the RS 232C interface. If a send request is made, it is aborted with a corresponding error message.
- When a **send request** is made, RTS is set to ON and the parameterized data output waiting time starts. When the data output time elapses and CTS = ON, the data is sent via the RS 232C interface.
- If the CTS line is not set to ON within the data output time so that data can be sent, or if CTS changes to OFF during transmission, the send request is aborted and an error message generated.
- After the data is sent, the RTS line is set to OFF after the parameterized time to RTS OFF has elapsed. The CP340 does not wait for CTS to change to OFF.
- Data can be **received** via the RS 232C interface as soon as the DSR line is set to ON. If the receive buffer of the CP 341 threatens to overflow, the CP 341 does not respond.
- A send request or data receipt is aborted with an error message if DSR changes from ON to OFF. The message "DSR = OFF (automatic use of V24 signals)" is entered in the diagnostics buffer of the CP 341.

Note

Automatic use of the RS 232C secondary signals is only possible in half-duplex mode.

When automatic use of the RS 232C secondary signals is parameterized, neither RTS/CTS data flow control nor RTS and DTR control by means of the V24_SET FC are possible.

Note

The "time to RTS OFF" must be set in the parameterization interface so that the communication partner can receive the last characters of the message frame in their entirety before RTS, and thus the send request, is taken away. The "data output waiting time" must be set so that the communication partner can be ready to receive before the time elapses.

Time Diagram

Figure 2-24 illustrates the chronological sequence of a send request.

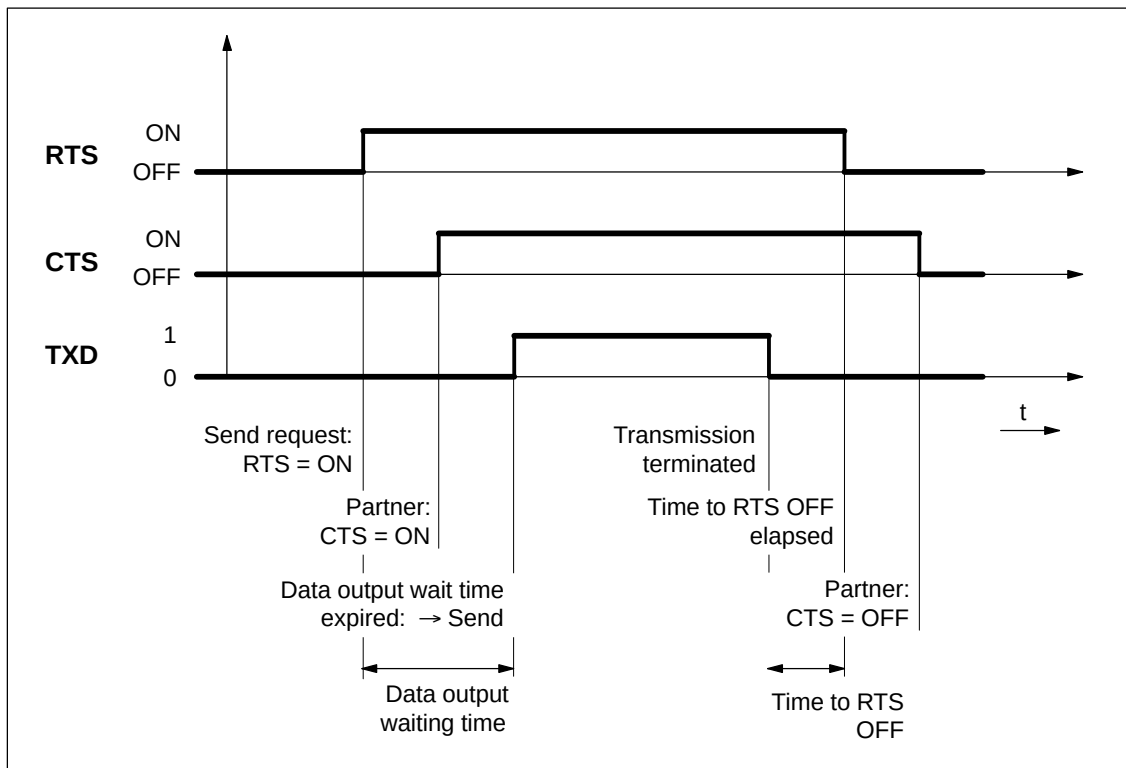


Figure 2-24 Time Diagram for Automatic Use of the RS 232C Secondary Signals

Data Flow Control/Handshaking

Handshaking controls the data flow between two communication partners. Handshaking ensures that data is not lost in transmissions between devices that work at different speeds. There are essentially two types of handshaking:

- Software handshaking (e.g. XON/XOFF)
- Hardware handshaking (e.g. RTS/CTS)

Data flow control is implemented as follows on the CP 341:

- As soon as the CP 341 is switched by parameterization to an operating mode with flow control, it sends the XON character or sets the RTS line to ON.
- When the parameterized number of message frames is reached, or alternatively 50 characters before the receive buffer overflows (size of the receive buffer: 1024 bytes), the CP 341 sends the XOFF character or sets the RTS line to OFF. If the communication partner continues to send data regardless of this, the receive buffer overflows and an error message is generated. The data received in the last message frame is discarded.
- As soon as a message frame is fetched by the S7 CPU and the receive buffer is ready to receive, the CP 341 sends the XON character or sets the RTS line to ON.
- If the CP 341 receives the XOFF character, or the CTS control signal of the communication partner is set to OFF, the CP 341 interrupts the transmission. If neither an XON character is received nor the CTS of the partner is set to ON before a parameterized time has elapsed, the transmission is aborted and an appropriate error message (0708H) is generated at the STATUS output of the function blocks.

Note

When RTS/CTS data flow control is parameterized, you must fully wire the interface signals in the plug connection (see Appendix B).

When RTS/CTS data flow control is parameterized, RTS control by means of the V24_SET FC is not possible.

Reading/Control with the V24_STAT and V24_SET FCs

The V24_STAT function allows the status of each RS 232C secondary signal to be determined. The V24_SET function allows the DTR and RTS output signals to be controlled.

Switch-over Times for RS485 Module in Half-Duplex Mode

The maximum switch-over time between sending and receiving is 1 ms.

This value is applicable to modules as of MLFB number 6ES7 341-1_H01-0AEO.

2.3 Parameterization Data

By selecting different protocols, you can adjust your CP 341 communication processor to suit the communication partner.

The sections that follow describe the parameterization data of the 3964(R) procedure, RK 512 computer connection and ASCII driver.

2.3.1 Parameterization Data of the 3964(R) Procedure

Using the parameterization data of the 3964(R) procedure, you can adjust the CP 341 to suit the properties of the communication partners.

Parameterization Data of the 3964(R) Procedure

Using the *CP 341 Point-to-Point Communication, Parameter Assignment* interface you specify the parameters for the physical layer (layer 1) and the data-connection layer (layer 2) of the 3964(R) procedure. In the following you will find a detailed description of the parameters.

Section 5.2 describes how to enter parameterization data with the parameterization interface *CP 341: Point-to-Point Communication, Parameter Assignment*.

X27 (RS422/485) interface

Please note the following with reference to the X27 (RS 422/485) interface:

Note

In the case of the CP 341-RS 422/485 module variant, the 3964(R) procedure can be used only in four-wire mode.

Protocol

The following table describes the 3964(R) protocol.

Table 2-2 3964(R) Protocol

Parameters	Description	Default Value
3964 with default values and no block check	- The protocol parameters are set to default values. - If the CP 341 recognizes the string DLE ETX, it stops receiving and sends a DLE to the communication partner if the block was received undamaged, or an NAK if it was damaged.	3964R with standard values with block check: char. delay time = 220 ms NAK = 2000 ms Setup attempts = 6 Transmission attempts = 6
3964R with default values and block check	- The protocol parameters are set to default values. - If the CP 341 recognizes the string DLE ETX BCC, it stops receiving. The CP 341 compares the received block check character with the longitudinal parity calculated internally. If the BCC is correct and no other receive errors have occurred, the CP 341 sends the code DLE to the communication partner. (In the event of an error, the NAK code is sent).	
3964 programmable without block check	- The protocol parameters are programmable. - If the CP 341 recognizes the string DLE ETX, it stops receiving and sends a DLE to the communication partner if the block was received undamaged, or an NAK if it was damaged.	
3964R programmable with block check	- The protocol parameters are programmable. - If the CP 341 recognizes the string DLE ETX BCC, it stops receiving. The CP 341 compares the received block check character with the longitudinal parity calculated internally. If the BCC is correct and no other receive errors have occurred, the CP 341 sends the code DLE to the communication partner. (In the event of an error, the NAK code is sent).	

Protocol Parameters

You can only set the protocol parameters if you have not set the default values in the protocol.

Table 2-3 Protocol Parameters (3964(R) Procedure)

Parameters	Description	Value Range	Default Value
Character delay time	The character delay time defines the maximum amount of time permitted between two incoming characters within a message frame.	20 ms to 655350 ms in 10 ms increments The shortest character delay time depends on the baud rate 300 bits/s 60 ms 600 bits/s 40 ms 1200 bits/s 30 ms 2400 to 76800 bits/s 20 ms	220 ms
Acknowledgment delay time (ADT)	The acknowledgment delay time defines the maximum amount of time permitted for the partner's acknowledgment to arrive during connection setup (time between STX and partner's DLE acknowledgment) or release (time between DLE ETX and partner's DLE acknowledgment).	20 ms to 655350 ms in 10 ms increments The shortest acknowledgment delay time (ADT) depends on the baud rate: 300 bits/s 60 ms 600 bits/s 40 ms 1200 bits/s 30 ms 2400 to 76800 bits/s 20 ms	2000 ms (550 ms with 3964 and no block check)
Setup attempts	This parameter defines the maximum number of attempts the CP 341 is allowed in order to establish a connection.	1 to 255	6
Transmission attempts	This parameter defines the maximum number of attempts permitted to transfer a message frame (including the first one) in the event of an error.	1 to 255	6

Baud Rate/Character Frame

The following table describes the baud rate/character frame.

Table 2-4 Baud Rate/Character Frame (3964(R) Procedure)

Parameters	Description	Value Range	Default Value
Baud rate	Speed of data transmission in bits/s Note: The maximum baud rate for the 20 mA TTY interface is 19200.	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800	• 9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	• 1 (fixed value)	• 1
Data bits	Number of bits to which a character is mapped.	• 7 • 8	• 8
Stop bits	During transmission, a stop bit is appended to every character to be sent to signal the end of the character.	• 1 • 2	• 1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value (0 or 1) brings the value of all the bits up to a defined status. Thus the data integrity is enhanced. A parity of "none" means that no parity bit is sent.	• none • odd • even	• even
Priority	A partner has high priority if its send request takes precedence over the send request of the other partner. A partner has low priority if its send request must wait until the send request of the other partner has been dealt with. With the 3964(R) procedure, you must parameterize both communication partners with different priorities, i.e. one partner is assigned high priority, the other low.	• high • low	• low

X27 (RS 422) Interface

The table below contains descriptions of the parameters for the X27 (RS 422) interface. RS 485 operation is not possible in conjunction with the 3964(R) procedure.

Table 2-5 X27 (RS 422) Interface (3964(R) Procedure)

Parameters	Description	Value Range	Default Value
Initial state of the receive line	none: This setting only makes sense with bus-capable special drivers.	none	R(A)5V/R(B)0V
	R(A)5V/R(B)0V: Break detection is possible with this initial state.	R(A)5V/R(B)0V	
	R(A)0V/R(B)5V: Break detection is not possible with this initial state.	R(A)0V/R(B)5V	
	(See also Figure 2-25.)		

Initial State of the Receive Line

Figure 2-25 shows the wiring of the receiver at the X27 (RS 422) interface:

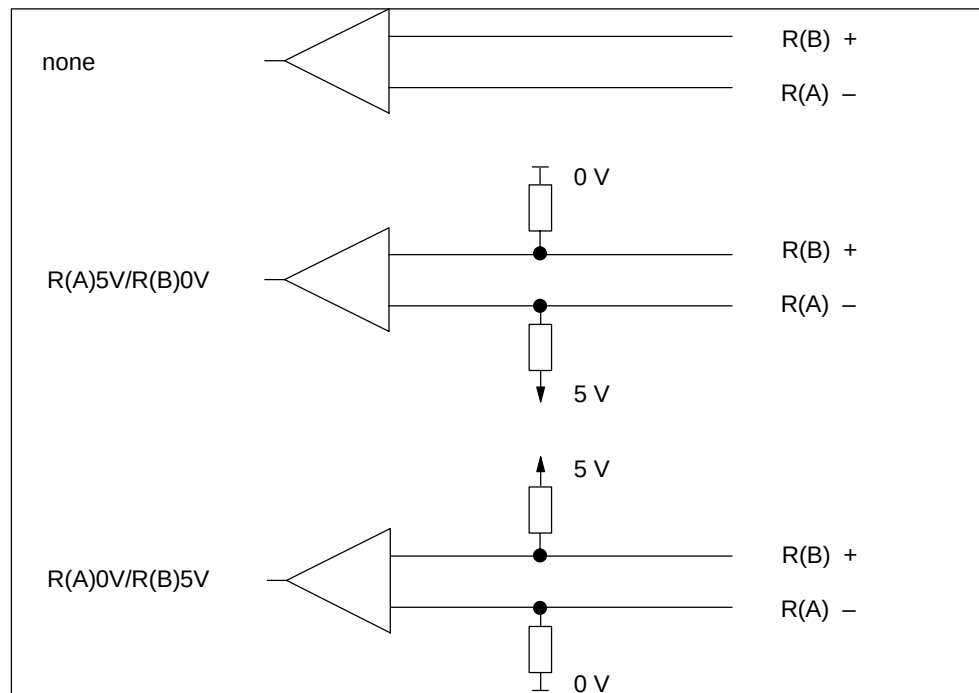


Figure 2-25 Wiring of the Receiver at the X27 (RS 422) Interface (3964(R) Procedure)

2.3.2 Parameterization Data of the RK 512 Computer Connection

You can use the parameterization data of the RK 512 computer connection to adjust the CP 341 to suit the properties of the communication partner.

Parameterization Data of the RK 512 Computer Connection

The parameters are identical to those of the 3964(R) procedure because the 3964(R) procedure is a subset of the RK 512 computer connection in the ISO 7-layer reference model (see Section 2.3).

Note

Exception: The number of data bits per character is set permanently to **8** with the RK 512 computer connection.

The parameters of the transport layer (layer 4) must be specified in the function blocks (FB) used.

2.3.3 Parameterization Data of the ASCII Driver

Using the parameterization data of the ASCII driver, you can adjust the CP 341 to suit the properties of the communication partner.

Parameterization Data of the ASCII Driver

Using the *CP 341: Point-to-Point Communication, Parameter Assignment* interface you specify the parameters for the physical layer (layer 1) of the ASCII driver. In the following you will find a detailed description of the parameters.

Section 5.2 describes how to enter parameterization data with the parameterization interface *CP 341: Point-to-Point Communication, Parameter Assignment*.

X27 (RS422/485) interface

Please note the following with reference to the X27 (RS 422/485) interface:

Note

In the case of the CP 341-RS 422/485 module variant, the ASCII driver can be used in four-wire mode (RS 422) and two-wire mode (RS 485).

At parameterization, you must specify the type of interface (RS 422 or RS 485).

Protocol Parameters

The table below describes the protocol parameters.

Table 2-6 Protocol Parameters (ASCII Driver)

Parameters	Description	Value Range	Default Value																						
Indicator for end of receive message frame	Defines which criterion signals the end of each message frame.	- On expiry of character delay time - On receipt of end-of-text character - On receipt of fixed number of characters	On expiry of character delay time																						
Character delay time	The character delay time defines the maximum permitted time between 2 consecutively received characters.	2 to 65535 ms The shortest character delay time depends on the baud rate <table><tr><td><u>baud</u></td><td><u>character delay time (ms)</u></td></tr><tr><td>300</td><td>130</td></tr><tr><td>600</td><td>65</td></tr><tr><td>1200</td><td>32</td></tr><tr><td>2400</td><td>16</td></tr><tr><td>4800</td><td>8</td></tr><tr><td>9600</td><td>4</td></tr><tr><td>19200</td><td>2</td></tr><tr><td>38400</td><td>2</td></tr><tr><td>57600</td><td>2</td></tr><tr><td>76800</td><td>2</td></tr></table>	<u>baud</u>	<u>character delay time (ms)</u>	300	130	600	65	1200	32	2400	16	4800	8	9600	4	19200	2	38400	2	57600	2	76800	2	4 ms
<u>baud</u>	<u>character delay time (ms)</u>																								
300	130																								
600	65																								
1200	32																								
2400	16																								
4800	8																								
9600	4																								
19200	2																								
38400	2																								
57600	2																								
76800	2																								
End-of-text character 1 ¹	First end code.	7 data bits²: 0 to 7FH (hex) 8 data bits²: 0 to FFH (hex)	3 (03H = ETX)																						
End-of-text character 2 ¹	Second end code, if specified.	7 data bits²: 0 to 7FH (hex) 8 data bits²: 0 to FFH (hex)	0																						
Message frame length when received ³	When the end criterion is "fixed message frame length", the number of bytes making up a message frame is defined.	1 to 1024 (bytes)	240																						

¹ Can only be set if the end criterion is an end-of-text character.

² Depending on the parameterization of the character frame (7 or 8 data bits) (see Table 2-7).

³ Can only be set if the end criterion is a fixed message frame length.

Baud Rate/Character Frame

The table below describes the parameters for the baud rate and character frame.

Table 2-7 Baud Rate/Character Frame (ASCII Driver)

Parameters	Description	Value Range	Default Value
Baud rate	Speed of data transmission in bits/s Notes: The maximum baud rate for the 20 mA TTY interface is 19200.	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800	• 9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	• 1 (fixed value)	• 1
Data bits	Number of bits to which a character is mapped.	• 7 • 8	• 8
Stop bits	During transmission, a stop bit is appended to every character to be sent to signal the end of the character.	• 1 • 2	• 1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value (0 or 1) brings the value of all the bits up to a defined status. Thus the data integrity is enhanced. A parity of "none" means that no parity bit is sent.	• none • odd • even	• even

Data Flow Control

In the following table the parameters for data flow control are described.

Data flow control is not possible with the RS 485 interface. Data flow control with "RTS/CTS" and "Automatic use of V24 signals" is only possible with the RS 232C interface (see also Table 1-2).

Table 2-8 Data Flow Control (ASCII Driver)

Parameters	Description	Value Range	Default Value
Data flow control	Defines which type of data flow control is used.	- None - XON/XOFF - RTS/CTS - Automatic use of V24 signals	None
XON character ¹	Code for XON character	7 data bits²: 0 to 7FH (hex) 8 data bits²: 0 to FFH (hex)	11 (DC1)
XOFF character ¹	Code for XOFF character	7 data bits²: 0 to 7FH (hex) 8 data bits²: 0 to FFH (hex)	13 (DC3)
Waiting for XON after XOFF (wait time for CTS=ON) ³	Period of time for which the CP 341 shall wait for the XON code or for CTS="ON" of the communication partner when sending.	20 to 655350 ms in 10 ms increments	20000 ms
Time to RTS OFF ⁴	Time to elapse after the transmission before the CP 341 sets the RTS line to OFF.	0 to 655350 ms in 10 ms steps	10 ms
Data output waiting time ⁴	Time that the CP 341 is to wait for the communication partner to set CTS to ON after setting the RTS line to ON and before starting the transmission.	0 to 655350 ms in 10 ms steps	10 ms

¹ Only in the case of XON/XOFF data flow control

² Depending on the parameterization of the character frame (7 or 8 data bits) (see Table 2-7).

³ Only in the case of XON/XOFF or RTS/CTS flow control

⁴ Only in the case of automatic use of the RS 232C secondary signals

Further Information

You will find more information on data flow control with XON/XOFF and RTS/CTS and on automatic use of the RS 232C secondary signals in Section 2.2.4 as of the subsection entitled "RS 232C Secondary Signals".

Receive Buffer on CP

The table below describes the parameters for the CP receive buffer.

Table 2-9 Receive Buffer on CP (ASCII Driver)

Parameters	Description	Value Range	Default Value
Buffered receive message frames	Here you can specify the number of receive message frames to be buffered in the CP receive buffer. If you specify "1" here and deactivate the following parameter "prevent overwrite" and cyclically read the received data from the user program, a current message frame will always be sent to the CPU.	1 to 250	250
Prevent overwrite	You can deactivate this parameter if the parameter "buffered receive message frames" is set to "1". This authorizes the buffered receive message frame to be overwritten.	• yes • no (only if "buffered receive message frames" = "1")	• yes

Further Information

In Section 2.2.4 you can find further information on handling the receive buffer under "Receive Buffer on CP 341".

X27 (RS422/485) interface

The table below contains descriptions of the parameters for the X27 (RS 422/485) interface.

Table 2-10 X27 (RS 422/485) Interface (ASCII Driver)

Parameters	Description	Value Range	Default Value
Operating mode	Specifies whether the X27 (RS 422/485) interface is to be run in full-duplex mode (RS 422) or half-duplex mode (RS 485). (See also Section 2.1).	• Full-duplex (RS 422) four-wire mode • Half-duplex (RS 485) two-wire mode	• Full-duplex (RS 422) four-wire mode
Initial state of the receive line	none: This setting only makes sense with bus-capable special drivers. R(A)5V/R(B)0V: break detection is possible with this initial state in conjunction with "Full Duplex (RS 422) Four-Wire Mode". R(A)0V/R(B)5V: this initial state corresponds to idle (no senders active) in "Half Duplex (RS 485) Two-Wire Mode". Break detection is not possible with this initial state. (See also Figure 2-26)	• None: • R(A)5V/R(B)0V¹ • R(A)0V/R(B)5V	• R(A)5V / R(B)0V²

¹ Only in the case of "Full-Duplex (RS 422) Four-Wire Mode"

² Only in the case of "Full-Duplex (RS 422) Four-Wire Mode" in the case of "Half Duplex (RS 485) Two-Wire Mode", the default setting is R(A)0V/R(B)5V

Initial State of the Receive Line

Figure 2-26 shows the wiring of the recipient at the X27 (RS 422/ 485) interface:

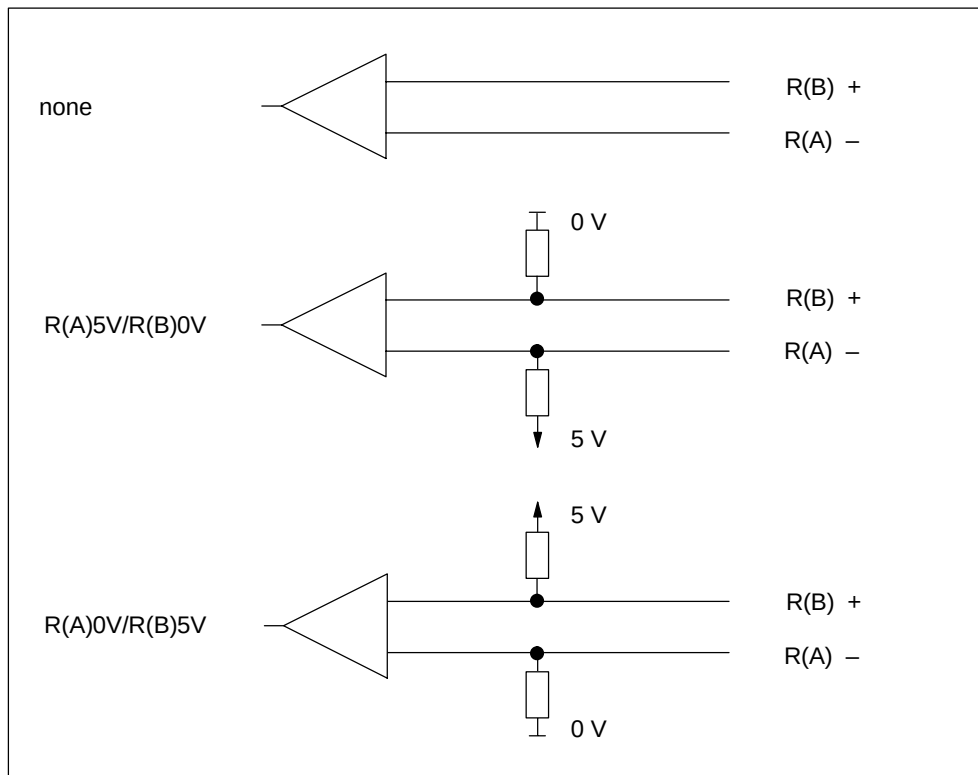


Figure 2-26 Wiring of the Receiver at the X27 (RS 422/485) Interface (ASCII Driver)

Starting up the CP 341

Before starting up the CP 341 you will need to perform the following operations in the order given.

1. Mount the CP 341
2. Configure the CP 341
3. Parameterize the CP 341
4. Store the parameterization data
5. Create a user program for the CP 341

Mounting the CP 341

Mounting the CP 341 involves inserting it into the mounting rack of your programmable controller.

For a detailed description, see Section 5.1.

Configuring the CP 341

Configuring the CP 341 involves entering it in the configuration table. The CP 341 is configured using the STEP 7 software.

For a detailed description, see Section 5.1.

Parameterizing the CP 341

Parameterizing the CP 341 involves creating the specific parameters of the protocols. You parameterize the CP 341 with the parameterization interface *CP 341: Point-to-Point Communication, Parameter Assignment*.

For a detailed description, see Section 5.2.

Storing the Parameterization Data

Storing the parameterization data of the CP 341 involves saving the parameters, loading them in the CPU and transferring them to the CP 341. You use the STEP 7 to store the parameterization data.

For a detailed description, see Section 5.3.

Creating a User Program for the CP 341

Programming the CP 341 involves configuring it for the associated CPU via the STEP 7 user program. The CP 341 is programmed using the language editors of the STEP 7 software.

Chapter 9 contains a detailed programming example. A detailed description of programming with STEP 7 is contained in the STEP 7 manual /1/.

Mounting the CP 341

4

In Section	You Will Find	on Page
4.1	CP 341 Slots	4-2
4.2	Mounting and Dismounting the CP 341	4-2
4.3	Installation Guidelines	4-4

4.1 CP 341 Slots

The following section describes the rules you must observe when positioning the CP 341 in the rack.

Position of the CP 341 in the Rack

The following rules apply when positioning the CP 341 in the rack:

- A maximum of 8 communications modules can be inserted to the right of the CPU.
- The number of communications modules which can be inserted is limited by CPU expandability (e. g. CPU 312 IFM in first row) or by the ET 200M (IM 153) in distributed I/O (single-row configuration only).

For further information on slots, see /2/.

4.2 Mounting and Dismounting the CP 341

When mounting and dismounting the CP 341, you must observe certain rules.

Tool

For mounting and dismounting the CP 341 you require a 4.5 mm cylindrical screwdriver.

Mounting Sequence

To insert the CP 341 in a rack, proceed as follows:

1. Switch the CPU to STOP mode.
2. Switch off the power supply.
3. The CP 341 is accompanied by an expansion bus. Plug this onto the backplane connector of the module to the left of the CP 341.
4. If more modules are to be mounted to the right of the CP 341, plug the expansion bus of the next module onto the right backplane connector of the CP 341.
5. Mount the CP 341 on the rail and lower it into position.
6. Screw the CP 341 tight.
7. Connect the DC 24 V of the load power supply to the CP 341.

DC 24 V Load Power Supply

The CP 341 has an external DC 24 V load power supply. The DC 24 V load power supply must satisfy the following requirements:

The only permissible load power supply is a safe mains-isolated low-voltage supply $\leq$ DC 60 V. The reliable electrical insulation can be implemented in compliance with the requirements of

- VDE 0100 Part 410 / HD 384-4-41 / IEC 364-4-41 (as function low voltage with safe electrical isolation) or
- VDE 0805 / EN 60950 / IEC 950 (as safe electrical low voltage SELV) or VDE 0106 Part 101.

Terminals

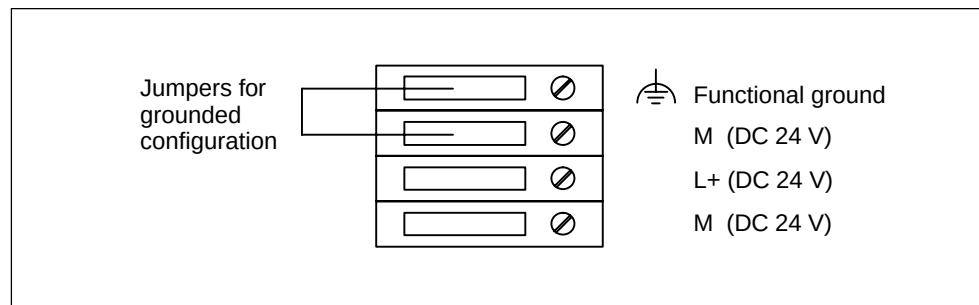


Figure 4-1 Terminals

Connect the positive cable of the 24 V supply to terminal L+.

Connect the negative cable of the 24 V supply to terminal M.

The two M terminals are interconnected internally. The 24-V connection has polarity reversal protection.

If you do not want to ground the 24 V ground cable, remove the jumper between the functional ground and M terminals.

Dismounting Sequence

To dismount the CP 341 from the rack, proceed as follows:

1. Switch the CPU to STOP mode.
2. Switch off the power supply.
3. Open the front doors.
4. Disconnect the connection to the DC 24V supply.
5. Disconnect the sub-D connector from the integral interface.
6. Release the securing screw on the module.
7. Tilt the module and remove it from the rail, and then remove it from the PLC.

Note

Before you mount or dismount the CP 341, you must switch the CPU to STOP mode and switch off the power supply.

You can plug in or unplug the cable to the integrated submodule on the CP 341 at any time. However, you must make sure that there is no data being transmitted via the integrated interface when you do this. Otherwise, data may be lost.

4.3 Installation Guidelines

To be Observed

The general installation guidelines for S7-300 must be observed (see the *S7-300 Programmable Controllers, Hardware and Installation* manual).

To meet the EMC (electromagnetic compatibility) values, the cable shield must be connected to a shield bus.

5

Configuring and Parameterizing the CP 341

In Section	You Will Find	on Page
5.1	Configuring the CP 341	5-2
5.2	Parameterizing the Communication Protocols	5-3
5.3	Managing the Parameter Data	5-4
5.4	Subsequent Loading of Drivers (Transmission Protocols)	5-5
5.5	Subsequent Loading of Firmware Updates	5-6

Parameterization Options

You configure and parameterize the module variants of the CP 341 using STEP 7 or the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface.

Table 5-1 Configuration Options for the CP 341

Product	Order Number	Parameterized Using the Parameterization Interface	Under STEP 7
CP 341-RS 232C	6ES7 341-1AH01-0AE0	As of V5.0	As of V4.02
CP 341-20mA TTY	6ES7 341-1BH01-0AE0		
CP 341-RS 422/485	6ES7 341-1CH01-0AE0		

5.1 Configuring the CP 341

Once you have mounted the CP 341 you must inform the programmable controller that it is there. This process is known as "configuration".

Requirements

The *CP 341 Point-to-Point Communication, Parameter Assignment* parameterization interface must be installed on the programming device/PC under STEP 7 (see also Table 5-1).

Installation

The *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface and the function blocks are supplied together with the programming example on a CD. Proceed as follows to install the parameterization interface:

1. Insert the CD into the CD drive of your programming device/PC.
2. Under *Windows 95/NT*, start the dialog for installing software by double-clicking on the "Software" icon in "Control panel".
3. In the dialog box, select the CD drive and the **setup.exe** file and start installation.
4. Now follow the step-by-step instructions of the installation program.

Configuration

In the sense used here, configuration means entering the CP 341 in the configuration table of the STEP 7 software. In the configuration table you enter the rack, the slot and the order number of the CP 341. STEP 7 then automatically assigns an address to the CP 341.

The CPU is now able to find the CP 341 in its slot in the rack by way of its address.

Requirements

Before you can enter the CP 341 in the configuration table of the STEP 7 software, you must have created a project and a terminal with STEP 7.

Further Information

How to configure S7-300 modules is described in detail in the STEP 7 manual /3/.

In addition, the on-line help for STEP 7 provides sufficient support for configuring an S7-300 module.

5.2 Parameterizing the Communication Protocols

Once you have entered the CP 341 in the configuration table, you have to supply the CP 341 communication processor and its serial interface with parameters.

Parameterization

The term "parameterization" is used in the following to describe the setting of protocol-specific parameters. You set these parameters with the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface.

You start the parameterization interface by double-clicking the order number (CP 341) in the configuration table or by selecting the CP 341 and choosing the **Edit > Object Properties** menu command. The "Properties - CP 341" dialog box appears.

Click on the "Parameters" button to go to protocol selection. Set the protocol and double-click the icon for the transmission protocol (an envelope). This takes you to the dialog box for setting the protocol-specific parameters.

Further Information

The *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface is intuitive and easy to use; the procedure is the same for all communications processors. For this reason, the parameterization interface is not described in detail here.

Also, the on-line help provides sufficient support for working with the parameterization interface.

5.3 Managing the Parameter Data

The configuration and parameterization data of the CP 341 is stored in the current project (on the hard disk of the programming device/PC).

Data Management

When you quit the configuration table (see Section 5.1) via the menu item **Station > Save** or **Station > Save As**, the configuration and parameterization data (including the module parameters) is automatically stored in the project/user file you have created.

Loading Configuration and Parameters

You can load the configuration and parameterization data onto the CPU online from the programming device (by choosing **PLC > Download**). The CPU puts the parameters into effect as soon as they are loaded.

The module parameters are automatically transmitted to the CP 341

- when they are loaded onto the CPU and as soon as the CP 341 can be reached via the S7-300 backplane bus,
or
- when the CPU's operating mode changes from STOP to RUN (CPU start-up).

Unchanged parameters have the default value (see Section 2.3).

Further Information

The Manual for STEP 7 /3/ describes in detail how to

- store the configuration and the parameters
- load the configuration and the parameters into the CPU
- read, modify, copy and print the configuration and parameters.

5.4 Subsequent Loading of Drivers (Transmission Protocols)

To extend the functionality of the CP 341 and adapt it to the communication partner, you can load other transmission protocols on the CP 341 (loadable drivers) in addition to the standard protocols in the module firmware (ASCII, 3964(R), RK 512).

The loadable drivers are not shipped with the CP 341 or the parameterization interface as standard. You have to order them separately (see the chapter entitled "Loadable Drivers" in the ST 70 catalog).

To find out how to install and parameterize a loadable driver and load it onto the CP 341, consult the separate documentation for the loadable driver. Only the requirements and the fundamentals are described below.

Requirements

The prerequisites for loading the drivers are:

- *STEP 7*, V4.02 or higher
- *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface, V5.0 or higher
- The dongle provided with the driver must be connected to the port at the rear of the CP 341.
- The current parameterization is saved beforehand under HW config and uploaded to the CPU.

Entry via Parameterization Interface

You select the loadable driver for parameterization in the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface.

After you have successfully installed the parameterization interface and loadable drivers, you select the driver you want and set the protocol-specific parameters in the same way as you do for the standard protocols. See Section 5.2 for more information on installing the parameterization interface and selecting a transmission protocol.

To find out what is parameterized and how to load the drivers onto the CP 341, consult the separate documentation for the loadable driver.

5.5 Subsequent Loading of Firmware Updates

Firmware updates can be uploaded to the operating-system memory of the CP 341 as patches.

Subsequent loading of firmware updates with the parameterization interface
CP 341: Point-to-Point Communication, Parameter Assignment.

Basic Firmware

The CP 341 is shipped with basic firmware preinstalled.

Requirements

The prerequisites for loading firmware updates are:

- *STEP 7*, V4.02 or higher
- *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface, V5.0 or higher
- You must create a valid project under the hardware configuration and upload it to the CPU before you can update the firmware of the communications processor with the parameterization interface
- The instructions accompanying the firmware update always detail the destination directories for the files.

The `..\CP341.nnn` path always identifies the firmware version.

Loading Firmware

You upload the firmware update to the CP 341 with the aid of the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface.

Proceed as follows:

1. Switch the CPU to STOP mode.
2. Start the parameterization interface (see also section 5.2):

In SIMATIC Manager: **File > Open Object > Project > Open Hardware Config > double-click on CP 341 > select the "Parameters" button.**

3. Select the menu command **Options > Firmware Update.**

Result:

If a connection can be established to the CP 341, the current module firmware status is displayed.

If there is no firmware loaded on the CP 341, the display shows " — — — ". This can occur, for example, if a firmware update was canceled. The original firmware is deleted prior to the cancellation. You have to upload firmware to the module before it can be restarted.

4. Click on the "Find File ..." button to select the firmware to be loaded (*.UPD).

Note:

The basic firmware consists of three files each with a *.UPD extension. Select only the file called **HEADER.UPD** for the basic firmware.

Result:

The version of the firmware you select is displayed under "Status of selected firmware".

5. Click on the "Load Firmware" button to start uploading to the CP 341. You are prompted for confirmation. The upload procedure is canceled immediately if you click on the "Cancel" button.

Note:

Before the basic firmware is deleted from the module, the CP 341 checks the MLFB No. of the firmware to be downloaded in order to ensure that the firmware is suitable for the CP 341.

Result:

The new firmware is loaded into the operating-system memory of the CP 341. "Done" shows progress in bar-graph form and as a percentage. The module is immediately ready for use as soon as the firmware update is completed.

LED Indicators

LED indicators for firmware update:

Table 5-2 LED Indicators for Firmware Update

Status	SF	TXD	RXD	Comment	Remedy
Firmware update in progress	on	on	on	–	–
Firmware update completed	on	off	off	–	–
CP 341 without module firmware	flashing (2Hz)	off	off	Module firmware deleted, firmware update was canceled, firmware update still possible	Reload the firmware
Hardware fault during firmware update	flashing (2Hz)	flashing (2Hz)	flashing (2Hz)	Delete/write failed	Switch power supply to module off and then on again and reload the firmware. Check module for defects.

Viewing the Hardware and Firmware Version

You can view the current hardware and firmware version of the CP 341 in *STEP 7* in the “Module Status” dialog box. To access this dialog box:

In SIMATIC Manager: **File > Open Object > Project > Open HW Config > Station > Open Online >** and double-click on CP 341.

Communication via Function Blocks

6

In Section	You Will Find	on Page
6.1	Communication via Function Blocks	6-2
6.2	Overview of the Function Blocks	6-2
6.3	Using the Function Blocks	6-4
6.4	Parameterizing the Function Blocks	6-42
6.5	General Information on Program Processing	6-47
6.6	Technical Specifications of the Function Blocks	6-48

6.1 Communication via Function Blocks

Communication between the CPU, the CP 341 and a communication partner takes place via the function blocks and the protocols of the CP 341.

Communication CPU and CP 341

The function blocks form the software interface between the CPU and the CP 341. They must be called in cycles from the user program.

Communication CP 341 and Communication Partner

The transmission protocols are translated on the CP 341. By means of the protocol, the interface of the CP 341 is adapted to the interface of the communication partner.

This enables you to link an S7 automation system with any communication partner that can handle the modern, standard protocols available in SIMATIC S5 (ASCII driver, 3964(R) procedure or RK 512 computer connection).

Interrupt Response

Calling the CP 341 function blocks in process (OB 40) or diagnostic (OB 82) interrupts is not permitted.

6.2 Overview of the Function Blocks

The S7-300 programmable controller provides you with a number of function blocks which initiate and control communication between the CPU and the CP 341 communication processor in the user program.

Function Blocks/Functions

The table below lists the function blocks/functions of the CP 341 and describes their purpose.

Table 6-1 Function Blocks / Functions of the CP 341

FB/FC	Meaning	Protocol
FC 5 V24_STAT (version 2.0)	The V24_STAT function allows you to read the signal states at the RS 232C interface of the CP 341-RS 232C.	ASCII driver
FC 6 V24_SET (version 2.0)	The V24_SET function allows you to set/reset the outputs at the RS 232C interface of the CP 341-RS 232C.	ASCII driver

Table 6-1 Function Blocks / Functions of the CP 341, continued

FB/FC	Meaning	Protocol
FB 7 P_RCV_RK	The P_RCV_RK function block allows you to receive data from a communication partner and place it in a data block or ready data for transmission to the communication partner.	3964(R) procedure, ASCII driver, RK 512 computer connection
FB 8 P_SND_RK	The P_SND_RK function block allows you to send an entire area or subarea of a data block to a communication partner or fetch data from the communication partner.	3964(R) procedure, ASCII driver, RK 512 computer connection

Scope of Supply and Installation

The function blocks of the CP 341, together with the parameterization interface and the programming example, are supplied on a CD which comes with this manual.

The function blocks are installed together with the parameterization interface. Installation is described in Section 5.2. After installation, the function blocks are stored in the library:

- **CP341:** FC 5 V24_STAT and FC 6 V24_SET (version 2.0)
FB 7 P_RCV_RK and FB 8 P_SND_RK

You open the library in STEP 7 SIMATIC Manager by choosing **File > Open > Library** under CP PTP\ CP 341\Blocks.

For working with the function blocks, you only need to copy the required function block in your project.

Permissible Versions of the FBs, FCs

Note the following warnings on permissible function blocks and functions:



Warning

For the CP 341, the only permissible FC 5 V24_STAT and FC 6 V24_SET functions are those with **version** $\geq$ **2.0**. Data inconsistencies can occur if you use version 1.0 of these functions.

Use only function blocks FB 7 P_RCV_RK and FB 8 P_SND_RK for data transmission with the CP 341. Use of the function blocks FB 2 P_RCV and FB 3 P_SEND of the CP 340 is **not** permitted, as their use can lead to data inconsistencies.

6.3 Using the Function Blocks

The following sections describe what you must take into account when supplying the function blocks with parameters.

STATUS Indication at the FB

Note the following with regard to STATUS indication at the function blocks:

Note

The parameters DONE, NDR, ERROR and STATUS are valid for only one module run. To display the STATUS, you should therefore copy it to a free data area.

DONE = '1' means that the request was completed without error.

In other words:

- With ASCII driver: Request was sent to the communication partner. This does not necessarily mean that the data was received by the communication partner.
- With 3964(R) procedure: Request was sent to the communication partner and positive acknowledgement was returned. This does not necessarily mean that the data was forwarded to the partner CPU.
- With RK 512 computer connection: Request was sent to the communication partner and transferred to the partner CPU without errors

6.3.1 Using the Function Blocks with the 3964(R) Procedure

The function blocks available for connection to a communication partner with the 3964(R) procedure are as follows:

- FB 8 P_SND_RK for transmitting data
- FB 7 P_RCV_RK for receiving data

Parallel Processing of Requests

Only one FB P_SND_RK and one FB P_RCV_RK can be programmed for each CP 341 in the user program.

Bear in mind, too, that you can have only

- 1 instance data block for FB P_SND_RK und
- 1 instance data block for FB P_RCV_RK,

because the statuses needed for the FB's internal routines are stored in the instance data block.

Data Consistency

Data consistency is limited to 32 bytes by the block size for data transmission between CPU and CP 341.

The following applies to consistent data transmission of more than 32 bytes:

- Sender: Do not access the transmit DB until all data has been transmitted (DONE = 1).
- Receiver: Do not access the receive DB until all data has been received (NDR = 1). After receiving, block the receive DB (EN_R = 0) until you have processed the data.

S7 Sends Data to a Communication Partner, FB P_SND_RK

The FB P_SND_RK transmits data from a data block, specified by the parameters DB_NO, DBB_NO and LEN, to the CP 341. The FB P_SND_RK is called for data transmission either cyclically or, alternatively, statically by a timer-driven program (without conditions).

The data transmission is initiated by a positive edge at the REQ input. A data transmission operation can run over several calls (program cycles), depending on the amount of data involved.

The P_SND_RK function block (FB) can be called in the cycle when the signal state at the R parameter input is "1". This aborts the transmission to the CP 341 and sets the P_SND_RK FB back to its initial state. Data that has already been received by the CP 341 is still sent to the communication partner. If the R input is statically showing the signal state "1", this means that sending is deactivated.

The LADDR parameter specifies the address of the CP 341 to be addressed.

Error Indication at the FB P_SND_RK

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. If there was an error, the corresponding event number is displayed in the STATUS (see Section 8.3). If no error occurs the value of STATUS is 0. DONE and ERROR/STATUS are also output in response to a RESET of FB P_SND_RK (see Figure 6-5). In the event of an error, the binary result BR is reset. If the block is terminated without errors, the binary result has the status "1".

Note

The P_SND_RK function block does not have a parameter check, which means that if there are invalid parameters, the CPU may switch to STOP mode.

Before the CP 341 can process an activated request after the CPU has changed from STOP to RUN mode, the CP CPU start-up mechanism of the P_SND_RK FB must be completed (see Section 6.5). Any requests initiated in the meantime do not get lost. They are transmitted once the start-up coordination with the CP 341 is finished.

What to Do

Block call

STL representation	LAD representation
CALL P_SND_RK, I_P_SND_RK	I_SND_RK
SF: =	P_SND_RK
REQ: =	EN ENO
R: =	SF DONE
LADDR: =	REQ ERROR
DB_NO: =	R STATUS
DBB_NO: =	LADDR
LEN: =	DB_NO
R_CPU_NO: =	DBB_NO
R_TYP: =	LEN
R_NO: =	R_CPU_NO
R_OFFSET: =	R_TYP
R_CF_BYT =	R_NO
R_CF_BIT =	R_OFFSET
DONE: =	R_CF_BYT
ERROR: =	R_CF_BIT
STATUS: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

Assignment in the Data Area

The P_SND_RK FB works with an instance DB I_SND_RK. The DB number is specified in the call. The instance DB has a length of 62 bytes. The data in the instance DB cannot be accessed.

Note

Exception: If the error STATUS == W#16#1E0F occurs, you can consult the SFCERR variable for more details of the error (see Section 8.3). This error variable can only be loaded via a symbolic access to the instance DB.

Parameters, FB P_SND_RK

The following table lists the parameters of the P_SND_RK FB:

Note

The R_CPU_NO, R_TYP, R_NO, R_OFFSET, R_CF_BYT and R_CF_BIT parameters are of no significance for the 3964(R) procedure and can be ignored.

You can also ignore the SF parameter, because it defaults to 'S' for send.

Table 6-2 Parameters of the P_SND_RK FB

Name	Type	Data Type	Explanation	Permitted Values, Comment
REQ	INPUT	BOOL	Initiates request with positive edge	
R	INPUT	BOOL	Aborts request	Current request is aborted. Sending is blocked.
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Send DB No.; CPU-specific, (zero is not allowed)
DBB_NO	INPUT	INT	Data byte number	$0 \leq \text{DBB_NO} \leq 8190$ Transmitted data as of data word
LEN	INPUT	INT	Data length	$1 \leq \text{LEN} \leq 1024$, specified in number of bytes
DONE ¹	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR ¹	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
STATUS ¹	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ Parameter is available until the next time the FB is called.

Time Sequence Chart for P_SND_RK FB

The figure below illustrates the behavior of the DONE and ERROR parameters, depending on how the REQ and R inputs are wired.

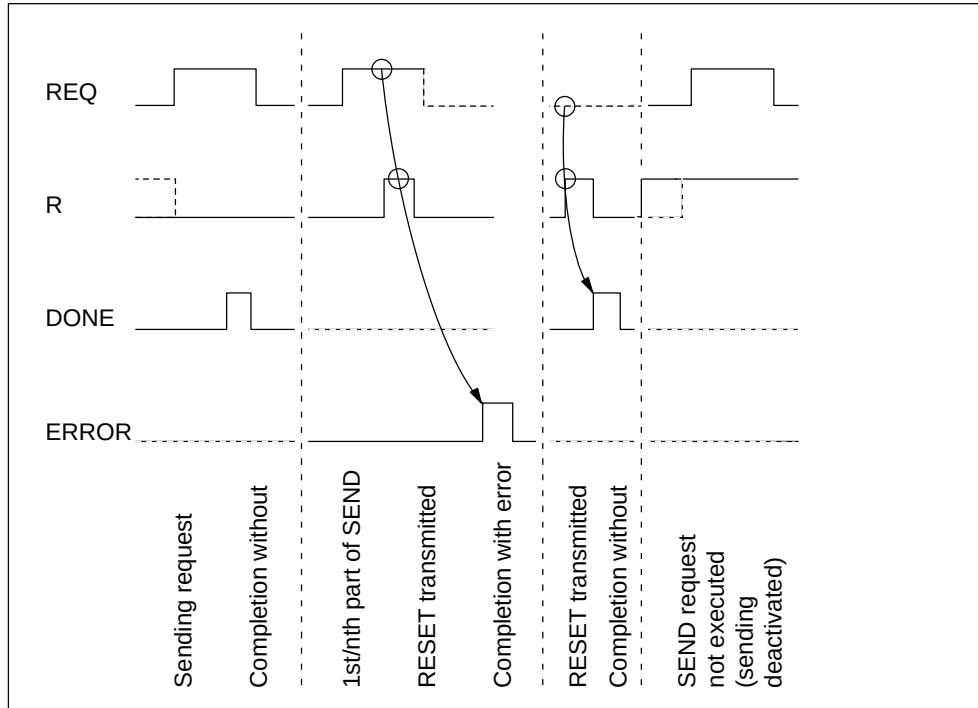


Figure 6-1 Time Sequence Chart, 8 P_SND_RK FB

Note

The REQ input is edge-triggered. A positive edge at the REQ input is adequate. It is not required that the RLO (result of logical operation) is "1" during the whole transmission procedure.

S7 Receives Data from a Communication Partner, FBP_RCV_RK

The P_RCV_RK FB transmits data from the CP 341 to an S7 data area specified by the parameters DB_NO, DBB_NO and LEN. The P_RCV_RK FB is called for data transmission either cyclically or, alternatively, statically by a timer-driven program (without conditions).

With the (static) signal state "1" at parameter EN_R, the software checks whether data can be read by the CP 341. An active transmission can be aborted with signal state "0" at the EN_R parameter. The aborted receive request is terminated with an error message (STATUS output). Receiving is deactivated as long as the EN_R parameter shows the signal state "0". A data transmission operation can run over several calls (program cycles), depending on the amount of data involved.

If the function block recognizes signal state "1" at the R parameter, the current transmission request is aborted and the P_RCV_RK FB is set to the initial state. Receiving is deactivated as long as the R parameter shows the signal state "1".

The LADDR parameter defines the CP 341 to be addressed.

Error Indication at the P_RCV_RK FB

The NDR output shows "request completed without errors/data accepted" (all data read). ERROR indicates whether an error has occurred. If there was an error, the corresponding event number is displayed in the STATUS (see Section 8.3). If no error occurs the value of STATUS is 0. NDR and ERROR/STATUS are also output in response to a RESET of FB P_RCV_RK (LEN parameter == 16#00) (see Figure 6-2). In the event of an error, the binary result BR is reset. If the block is terminated without errors, the binary result has the status "1".

Note

The P_RCV_RK function block does not have a parameter check, which means that if there are invalid parameters, the CPU may switch to STOP mode.

Before the CP 341 can receive a request after the CPU has changed from STOP to RUN mode, the CP-CPU start-up mechanism of the P_RCV_RK FB must be completed (see Section 6.5).

What to Do

Block call

STL representation	LAD representation
CALL P_RCV_RK, I_RCV_RK	I_RCV_RK
EN_R: =	P_RCV_RK
R: =	— EN ENO
LADDR: =	— EN_R NDR
DB_NO: =	— R ERROR
DBB_NO: =	— LADDR LEN
L_TYP: =	— DB_NO STATUS
L_NO: =	— DBB_NO L_TYP
L_OFFSET: =	L_NO
L_CF_BYT =	L_OFFSET
L_CF_BIT =	L_CF_BYT
NDR: =	L_CF_BIT
ERROR: =	
LEN: =	
STATUS: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). The compiler uses the binary result BR to process this parameter. The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

Assignment in the Data Area

The P_RCV_RK FB works with an instance DB I_RCV_RK. The DB number is specified in the call. The instance DB has a length of 60 bytes. The data in the instance DB cannot be accessed.

Note

Exception: If the error STATUS == W#16#1E0E occurs, you can consult the SFCERR variable for more details of the error (see Section 8.3). This error variable can only be loaded via a symbolic access to the instance DB.

Parameters, FB P_RCV_RK

The following table lists the parameters of the P_RCV_RK FB:

Note

The L_TYP, L_NO, L_OFFSET, L_CF_BYT and L_CF_BIT parameters are of no significance for the 3964(R) procedure and can be ignored.

Table 6-3 FB P_RCV_RK Parameters

Name	Type	Data Type	Explanation	Permitted Values, Comment
EN_R	INPUT	BOOL	Enables data read	
R	INPUT	BOOL	Aborts request	Current request is aborted. Receiving is blocked.
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Receive DB No.: CPU-specific, zero is not allowed
DBB_NO	INPUT	INT	Data byte number	$0 \leq \text{DBB_NO} \leq 8190$ Received data as of data word
NDR ¹	OUTPUT	BOOL	Request completed without errors, data accepted	STATUS parameter == 16#00;
ERROR ¹	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
LEN ¹	OUTPUT	INT	Length of message frame received	$1 \leq \text{LEN} \leq 1024$, specified in number of bytes
STATUS ¹	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ Parameter is available until the next time the FB is called.

Time Sequence Chart for P_RCV_RK FB

The figure below illustrates the behavior of the NDR, LEN and ERROR parameters, depending on how the EN_R and R inputs are wired.

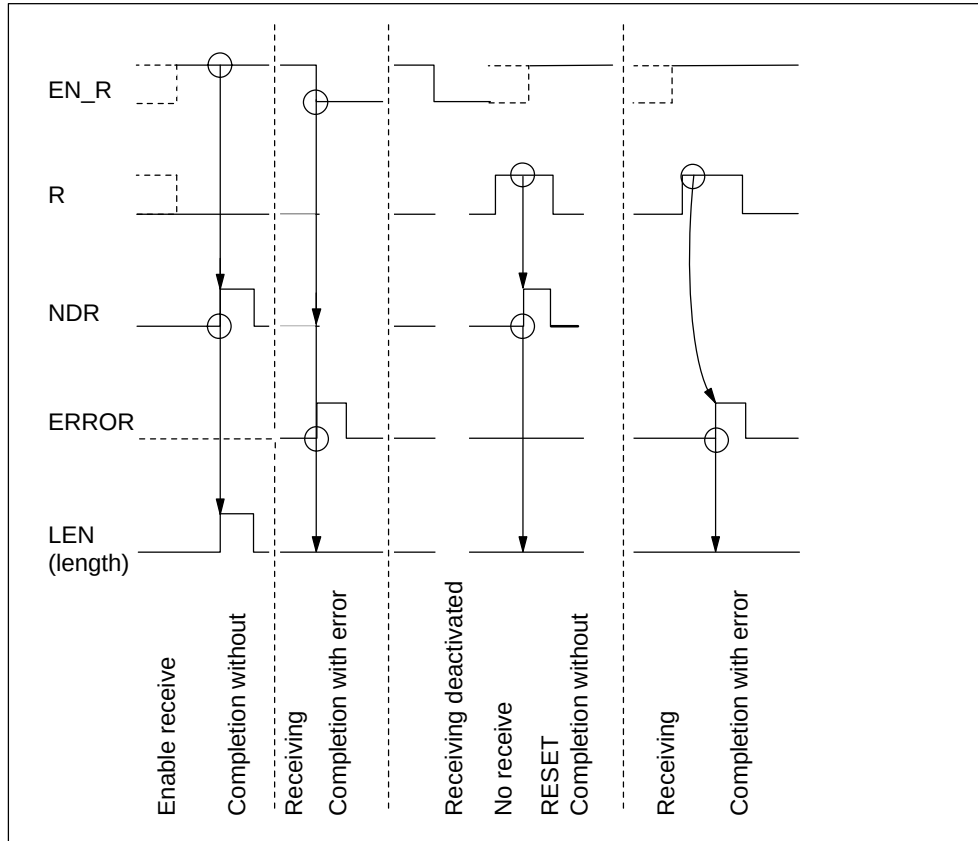


Figure 6-2 Time Sequence Chart FB 7 P_RCV_RK

Note

The EN_R must be set to static "1". During the receive request, the EN_R parameter must be supplied with RLO "1" (result of logic operation).

6.3.2 Using the Function Blocks with the RK 512 Computer Connection

The function blocks available for connection to a communication partner with the RK 512 procedure are as follows:

- FB 8 P_SND_RK for sending or fetching data
- FB 7 P_RCV_RK for receiving or readying data

Transmission Options

Active requests:

Use the 8 P_SND_RK function block (FB) to send active requests to the CP 341 in the user program of the CPU. You can

- send data from your programmable controller to a remote communication partner (see section "Sending Data with FB P_SND_RK")
- fetch data from a remote communication partner and place it in an S7 data area of your programmable controller (see "Fetching Data with FB P_SND_RK")

Note: If you fetch data from a CP 341, you must always program a P_RCV_RK function block on the CP 341.

Passive requests:

Use the 7 P_RCV_RK function block (FB) to coordinate the reading and readying of data on the CP 341 by means of passive requests. The communication partner is active. You can

- read data sent by the communication partner in the S7 data area of your programmable controller (see section "Receiving data with FB P_RCV_RK")
- Ready data in your programmable controller for a remote communication partner (see section "Readying Data with FB P_RCV_RK")

Parallel Processing of Requests

Only one active and one passive request can be programmed for each CP 341 in the user program. The CP 341 can process a passive request while handling an active request.

Bear in mind, too, that you can have only

- 1 instance data block for FB P_SND_RK und
- 1 instance data block for FB P_RCV_RK,

because the statuses needed for the FB's internal routines are stored in the instance data block.

Interprocessor Communication Flags

The interprocessor communication flag functionality familiar from SIMATIC S5 is supported as a means of coordinating asynchronous overwriting on receipt or readying of data (FB 7 P_RCV_RK) by the CP 341 and the processing of data on the CPU. Interprocessor communication flags can be used only in conjunction with the RK 512 computer connection.

Data Consistency

Data consistency is limited to 32 bytes by the block size for data transmission between CPU and CP 341.

The following applies to consistent data transmission of more than 32 bytes:

- Sender: Do not access the transmit DB until all data has been transmitted (DONE = 1).
- Fetcher: Do not access the transmit DB until all data has been transmitted (DONE = 1).
- Receiver: Use the interprocessor communication flag functionality. Do not access the receive DB until all data has been received (analysis of the interprocessor communication flag for this request: Interprocessor communication flag is set for one cycle at the FB if NDR = 1). Do not reset the interprocessor communication flag to "0" until you have processed the received data.
- Reader: Use the interprocessor communication flag functionality. Do not access the readied data until all data has been fetched (analysis of the interprocessor communication flag for this request: Interprocessor communication flag is set for one cycle at the FB if NDR = 1). Do not reset the interprocessor communication flag to "0" until you have processed the data to be fetched.

If your partner fetches the data from the areas I (inputs), O (outputs), F (flags), T (times) or C (counters), data consistency is limited to 32 bytes unless you use the interprocessor communication flag to prevent access to these areas by other instances in the user program while data transfer is in progress.

Sending Data with FB P_SND_RK (Active Request)

The P_SND_RK function block can be used with parameter setting **SF = 'S'** to transmit data from an S7 data area to a CP 341.

The data transmission is initiated by a positive edge at the REQ input. A data transmission operation can run over several calls (program cycles), depending on the amount of data involved (LEN).

The LADDR parameter specifies the address of the CP 341 to be addressed.

The area of the data blocks is the only permissible source for data to be sent. The source is fully specified by the data block number (DB_NO) and the offset (DBB_NO) of the first data byte to be sent in this data block.

Permissible destination areas are data types (R_TYP), data blocks (DB) and extended data blocks (DX). The destination is fully specified by the CPU number (R_CPU_NO, relevant only for multiprocessor communication), the data type (R_TYP: DB or DX), the data block number (R_NO) and the offset (R_OFFSET) at which the first byte is to be written.

R_CF_BYT and R_CF_BIT define the interprocessor communication flag byte and bit on the partner CPU.

The P_SND_RK function block (FB) can be called in the cycle when the signal state at the R parameter input is "1". This aborts the transmission to the CP 341 and sets the P_SND_RK FB back to its initial state. Data that has already been received by the CP 341 is still sent to the communication partner. If the R input is statically showing the signal state "1", this means that sending is deactivated.

Error Indication at the FB P_SND_RK

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. If there was an error, the corresponding event number is displayed in the STATUS (see Section 8.3). If no error occurs the value of STATUS is 0. DONE and ERROR/STATUS are also output in response to a RESET of FB P_SND_RK (see Figure 6-3). In the event of an error, the binary result BR is reset. If the block is terminated without errors, the binary result has the status "1".

Note

The P_SND_RK function block (FB) does not have a parameter check, which means that if there are invalid parameters, the CPU may switch to STOP mode.

Points to Note on Sending Data

Note the following points with regard to sending data:

- RK 512 allows only an even number of data to be sent. If you specify an odd number of data as length (LEN), an additional filler byte with a value of "0" is sent at the end.
- RK 512 allows only an even offset. If you specify an odd offset the data is stored as of the next lowest offset in the partner's data area.

Example: Offset is 7, data is stored as of byte 6.

What to Do

Block call

STL representation	LAD representation
CALL P_SND_RK, I_SND_RK	I_SND_RK
SF: =	P_SND_RK
REQ: =	EN ENO
R: =	SF DONE
LADDR: =	REQ ERROR
DB_NO: =	R STATUS
DBB_NO: =	LADDR
LEN: =	DB_NO
R_CPU_NO: =	DBB_NO
R_TYP: =	LEN
R_NO: =	R_CPU_NO
R_OFFSET: =	R_TYP
R_CF_BYT =	R_NO
R_CF_BIT =	R_OFFSET
DONE: =	R_CF_BYT
ERROR: =	R_CF_BIT
STATUS: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

Assignment in the Data Area

The P_SND_RK FB works with an instance DB I_SND_RK. The DB number is specified in the call. The instance DB has a length of 62 bytes. The data in the instance DB cannot be accessed.

Parameters, FB P_SND_RK

The table below describes the parameters of the 8 P_SND_RK function block for the “send data” request.

Table 6-4 Parameters of the 8 P_SND_RK Function Block for the “Send Data” Request

Name	Type	Data Type	Explanation	Permitted Values, Comment
SF	INPUT	CHAR	Selector for send data or fetch data	SF = 'S' (Send) default: 'S'
REQ	INPUT	BOOL	Initiates request with positive edge	
R	INPUT	BOOL	Aborts request	Current request is aborted. Sending is blocked. Default: 0
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number of source	Send DB No.; CPU-specific, (zero is not allowed)
DBB_NO	INPUT	INT	Data byte number of source	$0 \leq \text{DBB_NO} \leq 8190$ Transmitted data as of data word
LEN	INPUT	INT	Data length of frame to be sent	$1 \leq \text{LEN} \leq 1024$, specify in bytes, must be an even number
R_CPU_NO	INPUT	INT	CPU No. of the partner CPU	$0 \leq \text{R_CPU_NO} \leq 4$, only for multiprocessor operation, default: 1
R_TYP	INPUT	CHAR	Address type on partner CPU	'D': data block 'X': extended data block
R_NO	INPUT	INT	Data block number on partner CPU	$0 \leq \text{R_NO} \leq 255$
R_OFFSET	INPUT	INT	Data byte number on partner CPU	$0 \leq \text{R_OFFSET} \leq 510$, even numbers only
R_CF_BYT	INPUT	INT	Interprocessor communication flag byte on partner CPU	$0 \leq \text{R_CF_BYTE} \leq 255$ default: 255 (means: without interprocessor communication flag)
R_CF_BIT	INPUT	INT	Interprocessor communication flag bit on partner CPU	$0 \leq \text{R_CF_BIT} \leq 7$
DONE ¹	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR ¹	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
STATUS ¹	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ Parameter is available until the next time the FB is called.

Specifications in the Message Frame Header

The following table shows the specifications in the RK 512 message frame header.

Table 6-5 Specifications in the RK 512 Message Frame Header for the "Send Data" Request

Source on your S7 programmable controller (local CPU)	To destination, partner CPU	Message frame header, bytes		
		3/4 command type	5/6 D-DBNO/D offset	7/8 number in
Data block	Data block	AD	DB/DW	words
Data block	Extended data block	AD	DB/DW	words

Abbreviations:

D-DBNO	Destination data block number
D-Offset	Destination start address
DW	Offset in words

Time Sequence Chart for P_SND_RK FB

The figure below illustrates the behavior of the DONE and ERROR parameters, depending on how the REQ and R inputs are wired.

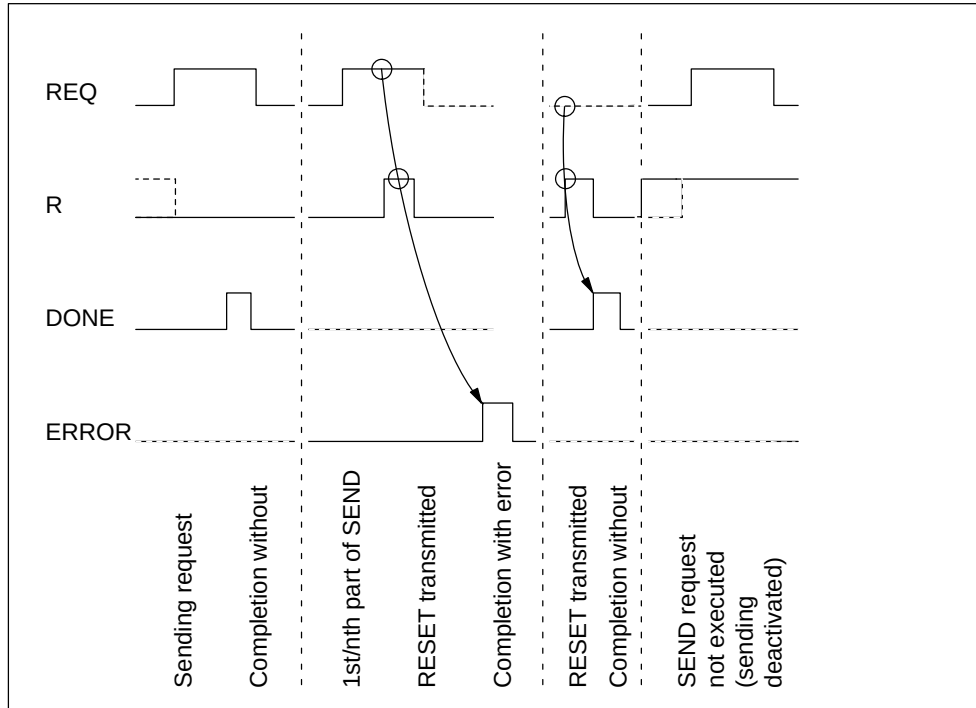


Figure 6-3 Time Sequence Chart of the 8 P_SND_RK Function Block for the "Send Data" Request

Note

The REQ input is edge-triggered. A positive edge at the REQ input is adequate. It is not required that the RLO (result of logical operation) is "1" during the whole transmission procedure.

Fetching Data with FB P_SND_RK (Active Request)

The P_SND_RK function block can be used with parameter setting **SF = F** fetch data from a remote communication partner and place it in an S7 data area of your programmable controller.

Note

If you fetch data from a CP 341, you must always program a P_RCV_RK function block on the CP 341.

The data transmission is initiated by a positive edge at the REQ input. A data transmission operation can run over several calls (program cycles), depending on the amount of data involved (LEN).

The LADDR parameter specifies the address of the CP 341 to be addressed.

The communication partner from which you want to fetch data is specified by the CPU number (R_CPU_NO, relevant only for multiprocessor communication).: The following data types (R_TYP) are permissible sources for the data to be fetched: data blocks, extended data blocks, flags, inputs, outputs, counters and times. The source is fully specified by the data type (R_TYP), the data block number (R_NO, relevant only for data blocks and extended data blocks) and the offset (R_OFFSET) of the first data byte to be sent in this area.

R_CF_BYT and R_CF_BIT define the interprocessor communication flag byte and bit on the partner CPU.

The only permissible destinations are data blocks (DB). The destination is fully specified by the data block number (DB_NO) and the offset (DBB_NO) of the first data byte to be written in this data block.

The P_SND_RK function block (FB) can be called in the cycle when the signal state at the R parameter input is "1". This aborts the transmission from the CP 341 and sets the P_SND_RK FB back to its initial state. If the R input is statically showing the signal state "1", this means that fetching is deactivated.

Error Indication at the FB P_SND_RK

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. If there was an error, the corresponding event number is displayed in the STATUS (see Section 8.3). If no error occurs the value of STATUS is 0. DONE and ERROR/STATUS are also output in response to a RESET of FB P_SND_RK (see Figure 6-4). In the event of an error, the binary result BR is reset. If the block is terminated without errors, the binary result has the status "1".

Note

The P_SND_RK function block (FB) does not have a parameter check, which means that if there are invalid parameters, the CPU may switch to STOP mode.

Points to Note on (Extended) Data Blocks

Note the points with regard to fetching data from data blocks and extended data blocks:

- RK 512 allows only an even number of data to be fetched. If you specify an odd number of data as length (LEN), an extra byte is always transmitted. In the destination DB, however, the correct number of data is entered.
- RK 512 allows only an even offset. If you specify an odd offset the data is fetched as of the next lowest even offset in the partner's data area.

Example: Offset is 7, data is fetched as of byte 6.

Points to Note on Times and Counters

If you fetch times or counters from the communication partner, remember that 2 bytes are fetched for each time or counter. For example, if you want to fetch 10 counters, you must enter 20 as the length.

What to Do

Block call

STL representation	LAD representation
CALL P_SND_RK, I_SND_RK	I_SND_RK
SF: =	P_SND_RK
REQ: =	EN ENO
R: =	SF DONE
LADDR: =	REQ ERROR
DB_NO: =	R STATUS
DBB_NO: =	LADDR
LEN: =	DB_NO
R_CPU_NO: =	DBB_NO
R_TYP: =	LEN
R_NO: =	R_CPU_NO
R_OFFSET: =	R_TYP
R_CF_BYT =	R_NO
R_CF_BIT =	R_OFFSET
DONE: =	R_CF_BYT
ERROR: =	R_CF_BIT
STATUS: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

Assignment in the Data Area

The P_SND_RK FB works with an instance DB I_SND_RK. The DB number is specified in the call. The instance DB has a length of 62 bytes. The data in the instance DB cannot be accessed.

Parameters, FB P_SND_RK

The table below describes the parameters of the 8 P_SND_RK function block for the “fetch data” request.

Table 6-6 Parameters of the 8 P_SND_RK Function Block for the “Fetch Data” Request

Name	Type	Data Type	Explanation	Permitted Values, Comment
SF	INPUT	CHAR	Selector for send data or fetch data	SF = 'F' (Fetch) default: 'S' (send)
REQ	INPUT	BOOL	Initiates request with positive edge	
R	INPUT	BOOL	Aborts request	Current request is aborted. Fetching is blocked. Default: 0
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	Destination data block number	Send DB No.; CPU-specific, (zero is not allowed)
DBB_NO	INPUT	INT	Destination data byte number	$0 \leq \text{DBB_NO} \leq 8190$ Transmitted data as of data word
LEN	INPUT	INT	Data length of frame to be fetched	$1 \leq \text{LEN} \leq 1024$, specified in bytes ²
R_CPU_NO	INPUT	INT	CPU No. of the partner CPU	$0 \leq \text{R_CPU_NO} \leq 4$, only for multiprocessor operation, default: 1
R_TYP	INPUT	CHAR	Address type on partner CPU	'D': data block 'X': extended data block 'F': flag 'I': inputs 'O': outputs 'C': counters 'T': times
R_NO	INPUT	INT	Data block number on partner CPU	$0 \leq \text{R_NO} \leq 255$
R_OFFSET	INPUT	INT	Data byte number on partner CPU	see Table 6-7
R_CF_BYT	INPUT	INT	Interprocessor communication flag byte on partner CPU	$0 \leq \text{CF_BYTE} \leq 255$ default: 255 (means: without interprocessor communication flag)
R_CF_BIT	INPUT	INT	Interprocessor communication flag bit on partner CPU	$0 \leq \text{CF_BIT} \leq 7$
DONE ¹	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR ¹	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
STATUS ¹	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ Parameter is available until the next time the FB is called.

² Always specify two bytes as the length for each time and each counter.

Parameters on FB for Data Source (Partner CPU)

The following table lists the data types which can be transmitted.

Table 6-7 Transferable Data Types for the "Fetch Data" Request

Source on partner CPU	R_TYP	R_NO	R_OFFSET** (in bytes)
Data block	'D'	0-255	0 - 510*
Extended data block	'X'	0-255	0 - 510*
Memory Marker	'M'	irrelevant	0-255
Inputs	'E'	irrelevant	0-255
Outputs	'A'	irrelevant	0-255
Counters	'Z'	irrelevant	0-255
Times	'T'	irrelevant	0-255

* Must be an even number!

** The value is determined by the partner CPU.

Specifications in the Message Frame Header

The following table shows the specifications in the RK 512 message frame header.

Table 6-8 Specifications in the RK 512 Message Frame Header for the "Fetch Data" Request

Source on partner CPU	Destination on your S7 programmable controller (local CPU)	Message frame header, bytes		
		3/4 command type	5/6 S-DBNO/S offset	7/8 number in
Data block	Data block	ED	DB/DW	words
Extended data block	Data block	EX	DB/DW	words
Memory Marker	Data block	EM	Byte address	bytes
Inputs	Data block	EI	Byte address	bytes
Outputs	Data block	EQ	Byte address	bytes
Counters	Data block	EC	Counter number	words
Times	Data block	ET	Time number	words

Abbreviations:

S-DBNO Source data block number

S-Offset Source start address

Time Sequence Chart for FB P_SND_RK FB

The figure below illustrates the behavior of the DONE and ERROR parameters, depending on how the REQ and R inputs are wired.

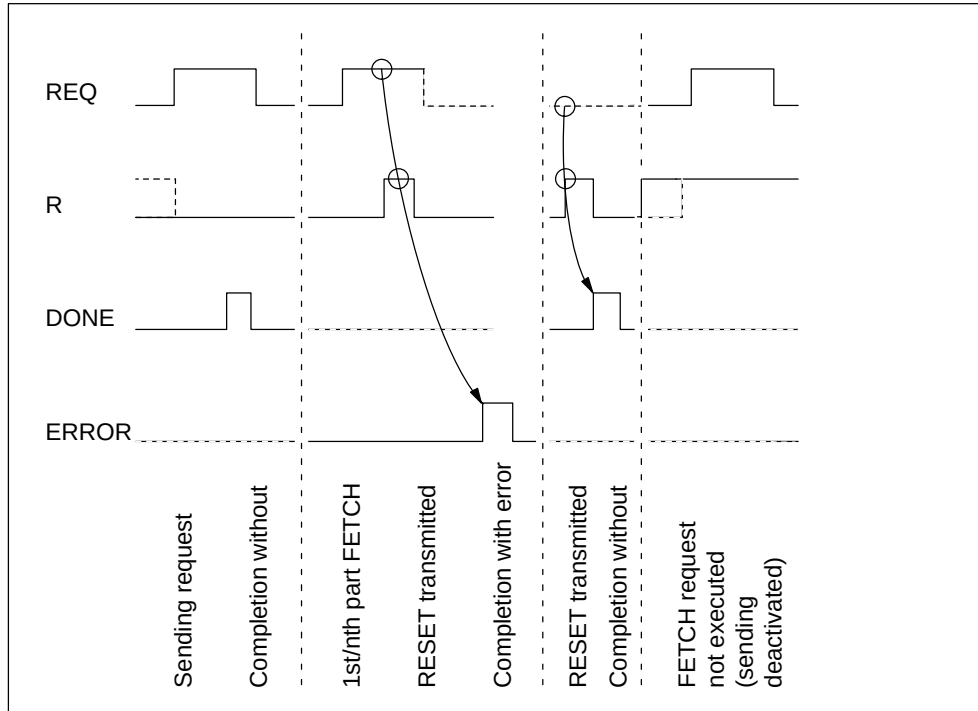


Figure 6-4 Time Sequence Chart of the 8 P_SND_RK Function Block for the "Fetch Data" Request

Note

The REQ input is edge-triggered. A positive edge at the REQ input is adequate. It is not required that the RLO (result of logical operation) is "1" during the whole transmission procedure.

Receiving Data with FB P_RCV_RK (Passive Request)

The P_RCV_RK FB transmits data from the CP 341 to an S7 data area.

The P_RCV_RK FB is called for data transmission either cyclically or, alternatively, statically by a timer-driven program (without conditions).

With the (static) signal state "1" at parameter EN_R, the software checks whether data can be read from the CP 341. An active transmission can be aborted with signal state "0" at the EN_R parameter. The aborted receive request is terminated with an error message (STATUS output). Receiving is deactivated as long as the EN_R parameter shows the signal state "0". A data transmission operation can run over several calls (program cycles), depending on the amount of data involved.

The LADDR parameter specifies the address of the CP 341 to be addressed.

If the communication partner specifies the **destination "DB"**, the data is placed in the data area specified in the RK 512 message frame header. The (L_...) parameter notifies the user of the type of the destination area (L_TYP), the destination data block number (L_NO, relevant only in conjunction with L_TYP = DB), the offset in the destination area (L_OFFSET) and the length (LEN) of the data transmitted. If the partner specifies the **destination "DX"**, the data is placed in the data block (DB) specified by the parameters DB_NO and DBB_NO.

If the function block recognizes signal state "1" at the R parameter, the current transmission request is aborted and the P_RCV_RK FB is set to the initial state. Receiving is deactivated as long as the R parameter shows the signal state "1".

The NDR output shows "request completed without errors/data accepted" (all data read). The parameters L_TYP, L_NO and L_OFFSET subsequently show for one cycle where the data is to be placed. The parameters L_CF_BYT and L_CF_BIT are also shown for one cycle, along with the LEN of the corresponding request.

Error Indication at the FB P_RCV_RK FB

ERROR indicates whether an error has occurred. If there was an error, the corresponding event number is displayed in the STATUS (see Section 8.3). If no error occurs the value of STATUS is 0. NDR and ERROR/STATUS are also output in response to a RESET of FB P_RCV_RK (LEN parameter == 16#00) (see Figure 6-5). In the event of an error, the binary result BR is reset. If the block is terminated without errors, the binary result has the status "1".

Note

The P_RCV_RK function block does not have a parameter check, which means that if there are invalid parameters, the CPU may switch to STOP mode.

Use of Interprocessor Communication Flags

Prior to data receipt, the interprocessor communication flags in the RK 512 message frame header are checked. The data is not transmitted unless the value of the interprocessor communication flag is "0". When transmission completes the function block sets the interprocessor communication flag to "1" and the interprocessor communication flag (NDR) is output for one cycle at the function block.

In the user program, analysis of the interprocessor communication flag indicates whether the data transmitted can be processed. As soon as the data have been processed, the user must reset the interprocessor communication flag to "0". The next SEND request can be started by the communication partner.

What to Do

Block call

STL representation	LAD representation
CALL P_RCV_RK, I_RCV_RK	I_RCV_RK
EN_R: =	P_RCV_RK
R: =	— EN — ENO —
LADDR: =	— EN_R — NDR —
DB_NO: =	— R — ERROR —
DBB_NO: =	— LADDR — LEN —
L_TYP: =	— DB_NO — STATUS —
L_NO: =	— DBB_NO — L_TYP —
L_OFFSET: =	— — L_NO —
L_CF_BYT =	— — L_OFFSET —
L_CF_BIT =	— — L_CF_BYT —
NDR: =	— — L_CF_BIT —
ERROR: =	
LEN: =	
STATUS: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

Assignment in the Data Area

The P_RCV_RK FB works with an instance DB I_RCV_RK. The DB number is specified in the call. The instance DB has a length of 60 bytes. The data in the instance DB cannot be accessed.

Note

Exception: If the error STATUS == W#16#1E0E occurs, you can consult the SFCERR variable for more details of the error (see Section 8.3). This error variable can only be loaded via a symbolic access to the instance DB.

Parameters, FB P_RCV_RK

The table below describes the parameters of the 7 P_RCV_RK function block for the “receive data” request.

Table 6-9 Parameters of the 7 P_RCV_RK Function Block for the “Receive Data” Request

Name	Type	Data Type	Explanation	Permitted Values, Comment
EN_R	INPUT	BOOL	Enables data receive	
R	INPUT	BOOL	Aborts request	Current request is aborted. Receiving is blocked. Default: 0
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number of the receive data (destination)	Receive DB No.: CPU-specific, zero is not allowed (relevant only if destination is DX)
DBB_NO	INPUT	INT	Data byte number of the receive data (destination)	$0 \leq \text{DBB_NO} \leq 8190$ Received data as of data word (relevant only if destination is DX)
L_TYP ¹	OUTPUT	CHAR	Type of area on local CPU (destination)	'D': data block
L_NO ¹	OUTPUT	INT	Data block number on local CPU (destination)	$0 \leq \text{L_NO} \leq 255$
L_OFFSET ¹	OUTPUT	INT	Data byte number on local CPU (destination)	$0 \leq \text{L_OFFSET} \leq 510$
L_CF_BYT ¹	OUTPUT	INT	Interprocessor communication flag byte on local CPU	$0 \leq \text{L_CF_BYTE} \leq 255$ 255 means: without interprocessor communication flag
L_CF_BIT ¹	OUTPUT	INT	Interprocessor communication flag bit on local CPU	$0 \leq \text{L_CF_BIT} \leq 7$
NDR ¹	OUTPUT	BOOL	Request completed without errors, data accepted	STATUS parameter == 16#00;
ERROR ¹	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
LEN ¹	OUTPUT	INT	Length of message frame received	$0 \leq \text{LEN} \leq 1024$, specified in number of bytes
STATUS ¹	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ Parameter is available until the next time the FB is called.

Time Sequence Chart for FB P_RCV_RK FB

The figure below illustrates the behavior of the NDR, LEN and ERROR parameters, depending on how the EN_R and R inputs are wired.

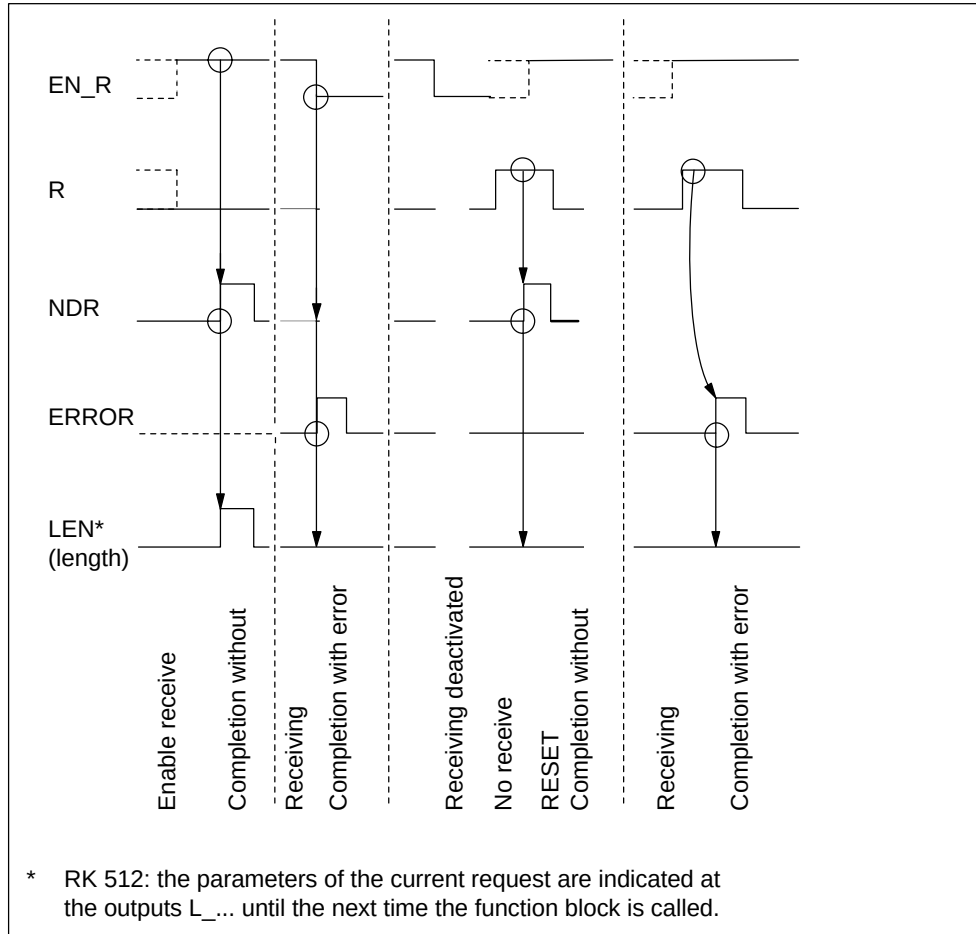


Figure 6-5 Time Sequence Chart of the 7 P_RCV_RK Function Block for the "Receive Data" Request

Note

The EN_R must be set to static "1". During the receive request, the EN_R parameter must be supplied with RLO "1" (result of logic operation).

Readying Data with FB P_RCV_RK (Passive Request)

It is not necessary to call the P_RCV_RK function block if the communication partner implements a FETCH request.

The P_RCV_RK FB readies data from an S7 data area for transmission to the CP 341. The P_RCV_RK FB is called for data transmission either cyclically or, alternatively, statically by a timer-driven program (without conditions).

With the (static) signal state "1" at parameter EN_R, the software checks whether data can be readied for the CP 341. An active transmission can be aborted with signal state "0" at the EN_R parameter. The aborted request is terminated with an error message (STATUS output). The request is deactivated as long as the EN_R parameter shows the signal state "0". A data transmission operation can run over several calls (program cycles), depending on the amount of data involved.

The type of source area (L_TYP), the source data block number (L_NO, relevant only in conjunction with L_TYP = DB), the offset in the source area (L_OFFSET) and the length (LEN) of the data to be readied are determined from the first RK 512 message frame. The function block analyzes the information from this message frame and transfers the requested data to the CP 341. The DB_NO and DBB_NO parameters are of no significance for the P_RCV_RK function block.

The LADDR parameter specifies the address of the CP 341 to be addressed.

If the function block recognizes signal state "1" at the R parameter, the current transmission request is aborted and the P_RCV_RK FB is set to the initial state. The request is deactivated as long as the R parameter shows the signal state "1".

The NDR output shows "request completed without errors/data accepted" (all data read). The parameters L_TYP, L_NO and L_OFFSET subsequently show for one cycle the source from which the data was fetched (possible data types: data blocks, input bytes, output bytes, times and counters). The parameters L_CF_BYT and L_CF_BIT are also shown for one cycle, along with the LEN of the corresponding request.

Note

When the communication partner fetches times or counters from the CP 341, the length is limited to a maximum of 32 bytes (16 times or counters, each consisting of 2 bytes).

Error Indication at the P_RCV_RK FB

ERROR indicates whether an error has occurred. If there was an error, the corresponding event number is displayed in the STATUS (see Section 8.3). If no error occurs the value of STATUS is 0. NDR and ERROR/STATUS are also output in response to a RESET of FB P_RCV_RK (LEN parameter == 16#00) (see Figure 6-5). In the event of an error, the binary result BR is reset. If the block is terminated without errors, the binary result has the status "1".

Note

The P_RCV_RK function block does not have a parameter check, which means that if there are invalid parameters, the CPU may switch to STOP mode.

Use of Interprocessor Communication Flags

Subsequent to receipt of the message frame, the interprocessor communication flags in the RK 512 message frame header are checked. The data is not readied unless the value of the interprocessor communication flag is "0". When transmission completes the function block sets the interprocessor communication flag to "1" and the interprocessor communication flag (NDR) is output for one cycle at the function block.

In the user program, analysis of the interprocessor communication flag indicates whether the data readied for transmission can be accessed. As soon as the data have been processed, the user must reset the interprocessor communication flag to "0". The next FETCH request can be started by the communication partner.

What to Do

Block call

STL representation	LAD representation
CALL P_RCV_RK, I_RCV_RK EN_R: = R: = LADDR: = DB_NO: = DBB_NO: = L_TYP: = L_NO: = L_OFFSET: = L_CF_BYT = L_CF_BIT = NDR: = ERROR: = LEN: = STATUS: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

Assignment in the Data Area

The P_RCV_RK FB works with an instance DB I_RCV_RK. The DB number is specified in the call. The instance DB has a length of 60 bytes. The data in the instance DB cannot be accessed.

Note

Exception: If the error STATUS == W#16#1E0E occurs, you can consult the SFCERR variable for more details of the error (see Section 8.3). This error variable can only be loaded via a symbolic access to the instance DB.

Parameters, FB P_RCV_RK

The table below describes the parameters of the 7 P_RCV_RK function block for the “ready data” request.

Table 6-10 Parameters of the 7 P_RCV_RK Function Block for the “Ready Data” Request

Name	Type	Data Type	Explanation	Permitted Values, Comment
EN_R	INPUT	BOOL	Enables data ready	
R	INPUT	BOOL	Aborts request	Current request is aborted. Ready is blocked. Default: 0
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	irrelevant	
DBB_NO	INPUT	INT	irrelevant	
L_TYP ¹	OUTPUT	CHAR	Type of area on local CPU (source)	'D': data block 'F': flag 'I': inputs 'O': outputs 'C': counters 'T': times
L_NO ¹	OUTPUT	INT	Data block number on local CPU (source)	$0 \leq L_NO \leq 255$ (relevant only if $L_TYP = D$)
L_OFFSET ¹	OUTPUT	INT	Data byte number on local CPU (source)	$0 \leq L_OFFSET \leq 510$ (depends on area type)
L_CF_BYT ¹	OUTPUT	INT	Interprocessor communication flag byte on local CPU	$0 \leq L_CF_BYTE \leq 255$ 255 means: without interprocessor communication flag
L_CF_BIT ¹	OUTPUT	INT	Interprocessor communication flag bit on local CPU	$0 \leq CF_BIT \leq 7$
NDR ¹	OUTPUT	BOOL	Request completed without errors, data accepted	STATUS parameter == 16#00;
ERROR ¹	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
LEN ¹	OUTPUT	INT	Length of message frame received	$0 \leq LEN \leq 1024$, specified in number of bytes
STATUS ¹	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ Parameter is available until the next time the FB is called.

Time Sequence Chart for P_RCV_RK FB

The figure below illustrates the behavior of the NDR, LEN and ERROR parameters, depending on how the EN_R and R inputs are wired.

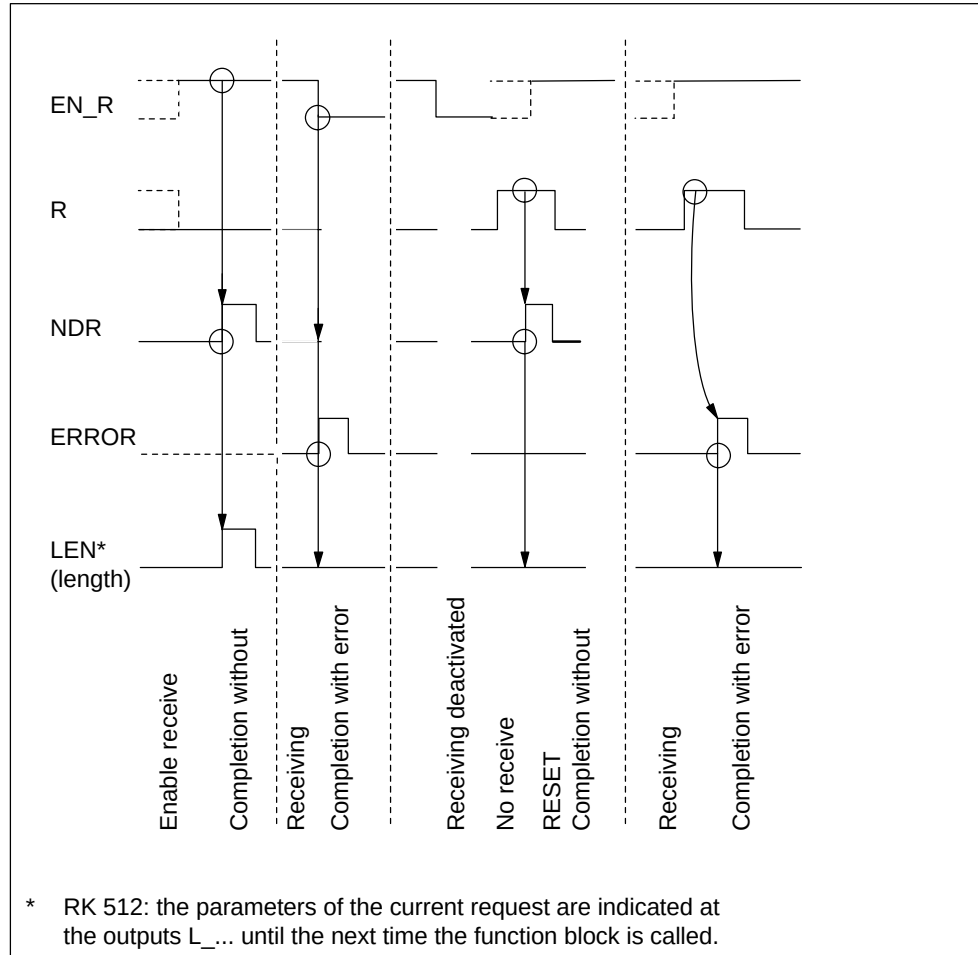


Figure 6-6 Time Sequence Chart of the 7 P_RCV_RK Function Block for the "Ready Data" Request

Note

The EN_R must be set to static "1". During the receive request, the EN_R parameter must be supplied with RLO "1" (result of logic operation).

6.3.3 Using the System Function Blocks with the ASCII Driver

The same functions can be used for data transfer with the ASCII driver as with the 3964(R) procedure. In other words, all the information on the function blocks FB P_SND_RK and FB P_RCV_RK given in Section 6.3.1 for the 3964(R) procedure also applies to the ASCII driver.

In addition, when the ASCII driver is used with the RS 232C interface submodule, you can read and control the RS 232C secondary signals. The following describes only what you have to do to use these additional functions.

Function blocks are available to you for reading and controlling the RS 232C secondary signals as follows:

- FC 5 V24_STAT for checking the interface status
- FC 6 V24_SET for setting/resetting the interface outputs

Note

For the CP 341, the only permissible FC 5 V24_STAT and FC 6 V24_SET functions are those with **version** $\geq$ **2.0**. Data inconsistencies can occur if you use version 1.0 of these functions.

Checking Interface Status of the CP 341

The V24_STAT FC reads the RS 232C secondary signals from the CP 341 and makes them available to the user at the module parameters. The V24_STAT FC is called statically (without conditions) for data transmission in the cycle or alternatively in a time-controlled program.

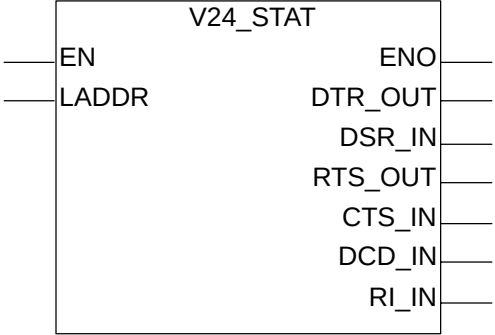
The RS 232C secondary signals are updated each time the function is called (cyclic polling). The CP 341 updates the status of the inputs/outputs in a timebase of 20 ms. The inputs/outputs are constantly updated independently of this.

The binary result BR is not affected. The function does not issue error messages.

The LADDR parameter defines the CP 341 to be addressed.

What to Do

Block call

STL representation	LAD representation
CALL V24_STAT LADDR: = DTR_OUT: = DSR_IN: = RTS_OUT: = CTS_IN: = DCD_IN: = RI_IN: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

Assignment in the Data Area

The V24_STAT function does not occupy any data areas.

Note

A minimum pulse time is necessary for a signal change to be identified. Significant time periods are the CPU cycle time, the updating time on the CP 341 and the response time of the communication partner.

Parameters FC 5 V24_STAT

The following table lists the parameters of the 5 V24_STAT FC:

Table 6-11 FC 5 V24_STAT Parameters

Name	Type	Data Type	Explanation	Permitted Values, Comment
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
DTR_OUT	OUTPUT	BOOL	D ata t erminal ready, CP 341 ready for operation	(CP 341 output)
DSR_IN	OUTPUT	BOOL	D ata s et ready, communication partner ready for operation	(CP 341 input)
RTS_OUT	OUTPUT	BOOL	R equ ~~e~~ st to s end, CP 341 ready to send ¹	(CP 341 output)
CTS_IN	OUTPUT	BOOL	C lea ~~r~~ to s end, communication partner can receive data from the CP 341 (response to RTS = ON of the CP 341) ¹	(CP 341 input)
DCD_IN	OUTPUT	BOOL	D ata C arrier D etect, receive signal level	(CP 341 input)
RI_IN	OUTPUT	BOOL	R ing Indicator, call signal	(CP 341 input)

¹ For further information on this RS 232C secondary signal, see Section 2.2.4.

Setting/Resetting Interface Outputs of the CP 341

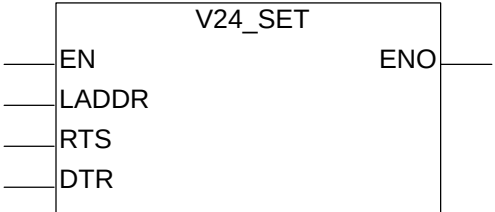
The user can use the parameter inputs of the V24_SET FC function to set or reset the corresponding interface outputs. The V24_SET FC is called in the cycle or alternatively in a time-controlled program statically (without conditions).

The binary result BR is not affected. The function does not issue error messages.

The LADDR parameter defines the CP 341 to be addressed.

What to Do

Block call

STL representation	LAD representation
CALL V24_SET LADDR: = RTS: = DTR: =	

Note

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result BR.

Assignment in the Data Area

The V24_STAT function does not occupy any data areas.

Parameters FC 6 V24_SET

The following table lists the parameters of the 6 V24_SET FC:

Table 6-12 FC 6 V24_SET Parameters

Name	Type	Data Type	Explanation	Permitted Values, Comment
LADDR	INPUT	INT	Basic address of CP 341	The basic address is taken from STEP 7.
RTS	INPUT	BOOL	Request to send, CP 341 ready to send ¹	(Control CP 341 output)
DTR	INPUT	BOOL	Data terminal ready, CP 341 ready for operation ¹	(Control CP 341 output)

¹ For further information on the RS 232C secondary signals, see Section 2.2.4.

6.4 Parameterizing the Function Blocks

This section is intended for users upgrading from SIMATIC S5 to SIMATIC S7. The various subsections contain important information on programming function blocks in STEP 7.

6.4.1 General Information on Data Block Assignment

Addressing

The data operands in data blocks are addressed bitwise in STEP 7 (in contrast to STEP 5, where addressing is word-based). You therefore have to convert the addresses of the data operands.

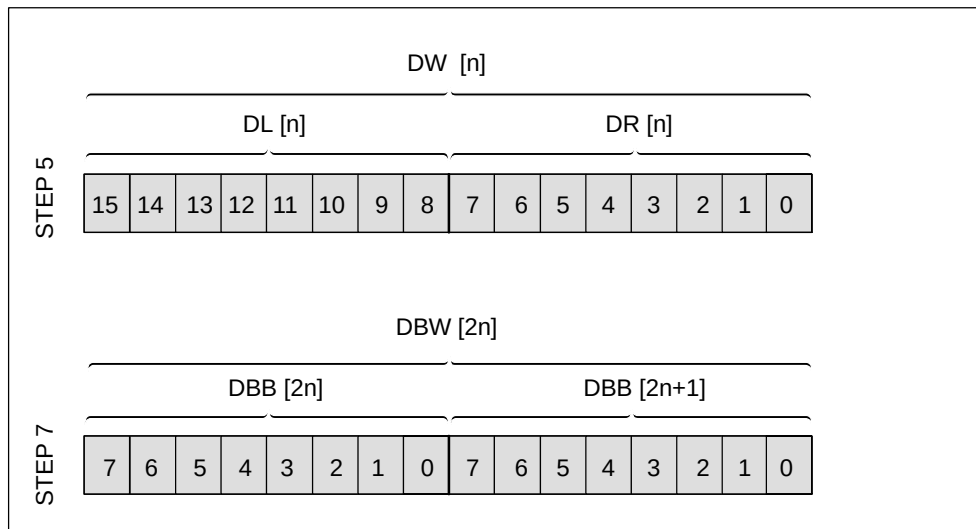


Figure 6-7 Contrasting Data Addressing in STEP 5 and STEP 7

The address of a data word in STEP 7 is doubled compared to STEP 5. It is no longer divided into a right and a left data byte. Bit numbering is always from 0 to 7.

Examples

The STEP 5 data operands in the left column of the table become the STEP 7 data operands in the right column.

STEP 5	STEP 7
DW 10	DBW 20
DL 10	DBB 20
DR 10	DBB 21
D 10.0	DBX 21.0
D 10.8	DBX 20.0
D 255.7	DBX 511.7

6.4.2 Parameterizing the Data Blocks

Direct/Indirect Parameterization

With STEP 7 the data blocks cannot be indirectly parameterized (parameters transferred in the currently selected data block) as they can with STEP 5.

All block parameters accept both constants and variables, so the distinction between direct and indirect parameterization is no longer necessary with STEP 7.

Example of "Direct Parameterization"

Call for FB 8 in accordance with the rules for "direct parameterization":

STL	Explanation
Network 1:	
CALL FB 8, DB8	
SF := S	//Sending request
REQ := M 0.6	//Activates SEND
R := M 5.0	//Activates RESET
LADDR := +336	//Basic address, PB336
DB_NO := +11	//Data block DB11
DBB_NO := +0	//As of data word DBB 0
LEN := +15	//Length 15 bytes
R_CPU_NO :=	//
R_TYP :=	//
R_NO :=	//
R_OFFSET :=	//
R_CF_BYT :=	//
R_CF_BIT :=	//
DONE := M 26.0	//Terminated without error
ERROR := M 26.1	//Terminated with error
STATUS := MW 27	//Status word

Example of "Indirect Parameterization"

Call for FB 8 in accordance with the rules for "indirect parameterization":

STL	Explanation
Network 1:	
CALL FB 8, DB8	
SF := S	//Sending request
REQ := M 0.6	//Activates SEND
R := M 5.0	//Activates RESET
LADDR := MW 21	//Basic address in MW21
DB_NO := MW 40	//DB no. in MW40
DBB_NO := MW 42	//DBB No. in MW42
LEN := MW 44	//Length in MW44
R_CPU_NO :=	//
R_TYP :=	//
R_NO :=	//
R_OFFSET :=	//
R_CF_BYT :=	//
R_CF_BIT :=	//
DONE := M 26.0	//Terminated without error
ERROR := M 26.1	//Terminated with error
STATUS := MW 27	//Status word

Parameterization of Data Words

The specification of data words (partially qualified specification) is not allowed because (depending on the actual operands) the currently selected data block can no longer be determined in the standard function. If a data operand is specified as an actual parameter, the fully qualified specification must always be used.

A fully qualified specification can be either absolute or symbolic. Mixed addressing with fully qualified data operands is rejected by the compiler.

Example 1

The symbol name for the data block is entered in the symbol table, while the symbol name for the data operand is declared in the corresponding data block.

STL	Explanation
DB 10.DBW 0	Absolute fully qualified addressing
CP_DB.SEND_DW_NO	Symbolic fully qualified addressing

Example 2

The symbol name of the data block used, DB 10, is "CP_DB"; the symbol name for the send DB number is "SEND_DBNO" and is located in data block DB 10 in the data word DBW 0.

The start address of the send message frame is "SEND_DWNO" and is located in the data block DB 10 in DBW 2, and the message frame length is "SEND_LEN" and is located in the data block DB 10 in DBW 4.

The variable used for the module address is the memory word "BGADR" (MW21), for the DONE parameter the flag "SEND_DONE" (M26.0), for the ERROR parameter the memory bit "SEND_ERROR" (M26.1), and for the STATUS parameter the memory word "SEND_STATUS" (MW27).

The STL listings for the example are shown on the following page.

Example of an Absolutely Addressed Actual Operand

Calling FB 8 with absolutely addressed actual operands:

STL	Explanation
Network 1:	
CALL FB 8, DB8	
SF := S	//Sending request
REQ := M 0.6	//Activates SEND
R := M 5.0	//Activates RESET
LADDR := MW 21	//Basic address in MW21
DB_NO := DB10.DBW0	//DB no. in DBW0 of DB10
DBB_NO := DB10.DBW2	//From DBB no., located in DBW2 of DB10
LEN := DB10.DBW4	//Length located in DBW4 of DB10
R_CPU_NO :=	//
R_TYP :=	//
R_NO :=	//
R_OFFSET :=	//
R_CF_BYT :=	//
R_CF_BIT :=	//
DONE := M 26.0	//Terminated without error
ERROR := M 26.1	//Terminated with error
STATUS := MW 27	//Status word

Example of a "Symbolically Addressed Actual Operand"

Calling FB 8 with symbolically addressed actual operands:

STL	Explanation
Network 1:	
CALL FB 8, DB8	
SF := S	//Sending request
REQ := M 0.6	//Activates SEND
R := M 5.0	//Activates RESET
LADDR := BGADR	//Basic address
DB_NO := CP_DB.SEND_DBNO	//Send DB no.
DBB_NO := CP_DB.SEND_DWNO	//Message frame as of data word
LEN := CP_DB.SEND_LEN	//Message frame length
R_CPU_NO :=	//
R_TYP :=	//
R_NO :=	//
R_OFFSET :=	//
R_CF_BYT :=	//
R_CF_BIT :=	//
DONE := SEND_DONE	//Terminated without error
ERROR := SEND_ERROR	//Terminated with error
STATUS := SEND_STATUS	//Status word

EN/ENO Mechanism

The parameters EN and ENO are only present in the graphical representation (LAD or FBD). To process these parameters, the compiler uses the binary result (BR).

The binary result is set to signal state "1" if the block was terminated without errors. If there was an error, the BR is set to "0".

6.5 General Information on Program Processing

Start-up Characteristics of the CP 341 Programmable Controller

The parameterization data is created with the aid of the *CP341: Point-to-Point Communication, Parameter Assignment* parameterization interface and transferred to the CPU with the STEP7 software. Each time the CPU is started up, the current parameters are transferred to the CP 341 by the system service of the CPU.

Start-up Characteristics, FB-CP 341

Once the connection between the CPU and the CP 341 has been established, the CP 341 must be initialized.

For each P_SND_RK, P_RCV_RK function block, there is a separate start-up coordination. Before requests can be actively processed, the accompanying start-up procedure must be completed.

Disabling Alarms

In the function blocks the interrupts are not disabled.

Addressing the Module

The logical basic address is defined via STEP 7 and must be specified by the user under the block parameter LADDR.

6.6 Technical Specifications of the Function Blocks

Memory Requirements

The table below shows the memory requirements of the CP 341 function blocks/functions.

Table 6-13 Memory Requirements of the Function Blocks/Functions (in bytes)

Block	Name	Version	Load Memory	Work Memory	Local Data
FC 5	V24_STAT	2.0	188	72	2
FC 6	V24_SET	2.0	156	48	2
FB 7	P_RCV_RK	2.1	3584	2982	106
FB 8	P_SND_RK	2.3	3036	2490	32

Run Times

The table below shows the runtimes of the CP 341 function blocks/functions.

Table 6-14 Runtimes of the Function Blocks/Functions in μ s

Block	Name	Version	Function	CPU 313/ CPU 314	CPU 315/ CPU 315DP	CPU 318-2 DP	CPU 614
FC 5	V24_STAT	2.0	Reading RS 232C secondary signals	140	120	29	120
FC 6	V24_SET	2.0	Setting RS 232C secondary signals	160	130	37	130
FB 7	P_RCV_RK	2.1	Idling	510	450	65	400
			Receiving*	1800	1800	140	1500
			Readying*	1800	1800	140	1500
FB 8	P_SND_RK	2.3	Idling	410	360	65	350
			Sending*	2300	2200	140	1800
			Fetching*	2300	2200	140	1800

* The run times are referenced to a data block of 1 to 32 bytes; in the case of RK 512 the listed run time must be added once again per request as the basic allowance for transmission of the parameters.

Minimum Number of CPU Cycles

The table below describes the minimum number of CPU cycles (FB/FC calls) required to process a "minimum request" (32 bytes SEND/RECEIVE for the transported user data set per program cycle). This applies only in centralized operation.

Table 6-15 Minimum Number of CPU Cycles

	Number of CPU Cycles for Processing ...		
	Completion without	Completion with error	RESET/RESTART
P_RCV_RK	≥ 3	≥ 3	≥ 4
P_SND_RK	≥ 3	≥ 3	≥ 4
V24_STAT	1	–	–
V24_SET	2	$\gg 2$	–

Before the CP 341 can process an activated request after the CPU has changed from STOP to RUN mode, the CP-CPU start-up mechanism P_SND_RK must be completed. Any requests initiated in the meantime do not get lost. They are transmitted once the start-up coordination with the CP 341 is finished.

Before the CP 341 can receive a message frame in the user program after a change in the CPU mode from STOP to RUN, the CP-CPU start-up mechanism P_RCV_RK must be completed.

System Functions Used

The following system functions are used in the blocks:

- SFC 58 WR_REC Write data record
- SFC 59 RD_REC Read data record

Start-up Characteristics and Operating Mode Transitions of the CP 341

7

In Section	You Will Find	on Page
7.1	Operating Modes of the CP 341	7-2
7.2	Start-up Characteristics of the CP 341	7-2
7.3	Behavior of the CP 341 on Operating Mode Transitions of the CPU	7-4

7.1 Operating Modes of the CP 341

The CP 341 has the operating modes STOP, reparameterization and RUN.

STOP

When the CP 341 is in STOP mode, no protocol driver is active and all send and receive requests from the CPU are given a negative acknowledgment.

The CP 341 remains in STOP mode until the cause of the stop is removed (e.g. break, invalid parameter).

Reparameterization

For reparameterization, the protocol driver is initialized. The SF LED is on during reparameterization.

Sending and receiving are not possible, and send and receive message frames stored in the CP 341 are lost when the driver is restarted. Communication between the CP and the CPU is started afresh (active message frames are aborted).

At the end of the reparameterization, the CP 341 is ready to send and receive.

RUN

The CP 341 processes the requests from the CPU. It provides the message frames received by the communications partner to be fetched by the CPU.

7.2 Start-up Characteristics of the CP 341

The CP 341 start-up is divided into two phases:

- Initialization (CP 341 in POWER ON mode)
- Parameterization

Initialization

As soon as the CP 341 is energized, all module components are initialized.

Parameterization

Parameterization means that the CP 341 receives the module parameters assigned to the current slot as created with the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface.

Reparameterization is performed. The CP 341 is now ready for operation.

Start-up Characteristics when Module Power is Switched On

If the CP 341 has its own DC 24 V supply independent of the CPU, communication between the CPU and the CP 341 is interrupted if the DC 24 V supply of the CP 341 temporarily fails.

To resume communication between CPU and CP 341, proceed as follows for the CPUs and devices listed below:

CPU/device	Order Number	Procedure
313	6ES7 313-1AD00-0AB0	Switch power supply to CPU off then on again.
314	6ES7 314-1AE00-0AB0 6ES7 314-1AE01-0AB0	
314 IFM	6ES7 314-5AE00-0AB0	
315	6ES7 315-1AF00-0AB0	
315-2 DP	6ES7 315-2AF00-0AB0	
614	6ES7 614-1AH00-0AB0	

The procedure for the CPUs listed below is as follows:

CPU	Order Number	Procedure
313	6ES7 313-1AD01-0AB0	Switch the CPU to STOP and then back to RUN.
314	6ES7 314-1AE02-0AB0	
314 IFM	6ES7 314-5AE01-0AB0	
315	6ES7 315-1AF01-0AB0	
315-2 DP	6ES7 315-2AF01-0AB0	
318-2 DP	6ES7 318-2AJ00-0AB0	
614	6ES7 614-1AH01-0AB0	

7.3 Behavior of the CP 341 on Operating Mode Transitions of the CPU

Once the CP 341 has been started up, all data is exchanged between the CPU and the CP 341 by means of the function blocks.

CPU STOP

In CPU-STOP mode, communication via the S7 backplane bus is not possible. Any active CP-CPU data transmission, including both send and receive message frames, is aborted and the connection is reestablished.

Data traffic at the interface of the CP 341 is continued with the ASCII driver in the case of parameterization without flow control. In other words, the current send request is completed. In the case of the ASCII driver, receive message frames continue to be received until the receive buffer is full.

CPU START-UP

On start-up, the CP sends the parameters created with the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface. The CP 341 only reparameterizes if the parameters have changed.

CPU RUN

When the CPU is in RUN mode, sending and receiving are unrestricted. In the first FB cycles following the CPU restart, the CP 341 and the corresponding FBs are synchronized. Only then is a new FB or FC executed.

Points to Note when Sending Message Frames

Message frames can be sent only when the CPU status is RUN.

If the CPU changes to STOP mode during transmission CPU > CP, the P_SND_RK FB reports the error "current program interrupted, request aborted due to BREAK/restart/reset" after restart.

Note

The CP 341 does not send data to the communications partner until it has received all data from the CPU.

Diagnostics with the CP 341

8

In Section	You Will Find	on Page
8.1	Diagnostics Functions of the CP 341	8-2
8.2	Diagnosis via the Display Elements of the CP 341	8-3
8.3	Diagnostics Messages of the Function Blocks	8-4
8.4	Error Numbers in the Response Message Frame	8-17
8.5	Diagnosis by Means of the Diagnostic Buffer of the CP 341	8-19
8.6	Diagnostic Alarm	8-21

8.1 Diagnostics Functions of the CP 341

The diagnostics functions of the CP 341 enable you to quickly localize any errors which occur. The following diagnostics options are available:

- Diagnosis via the display elements of the CP 341
- Diagnosis via the STATUS output of the function blocks
- Diagnosis via the diagnostic buffer of the CP 341
- Diagnosis via the diagnostic alarm

Display Elements (LED)

The display elements show the operating mode or possible error states of the CP 341. The display elements give you an initial overview of any internal or external errors as well as interface-specific errors (see Chapter 8.2).

Section 5.5 contains information on LED displays which can occur when you load firmware updates.

STATUS Outputs of the FBs

The P_SND_RK and P_RCV_RK function blocks have a STATUS output for error diagnostics. Reading the STATUS output of the function blocks gives you information on errors which have occurred during communication. You can interpret the STATUS parameter in the user program (see Section 8.3).

The CP 341 also enters the diagnostic events at the STATUS output in its diagnostic buffer.

Error Numbers in the Response Message Frame

If you are working with the RK 512 computer connection and there is a SEND or FETCH message frame error at the communication partner, the communication partner sends a response message frame with an error number in the 4th byte (see Section 8.4).

Diagnostic Buffer of the CP 341

All the CP 341 errors are entered in its diagnostic buffer.

In the same way as with the diagnostic buffer of the CPU, you can also use the STEP 7 information functions on the programming device to display the information in the CP diagnostic buffer (see Section 8.5).

Diagnostics Alarm

The CP 341 can trigger a diagnostics alarm on the CPU assigned to it. The CP 341 provides 4 bytes of diagnostics information on the S7-300 backplane bus. This information is analyzed via the user program (OB 82) or using a programming device to read from the CPU diagnostics buffer.

The CP 341 also enters diagnostic events that trigger a diagnostics alarm in its diagnostic buffer.

When a diagnostics alarm event occurs, the SF LED (red) lights up.

8.2 Diagnosis via the Display Elements of the CP 341

The display elements of the CP 341 provide information on the CP 341. The following display functions are distinguished:

- **Group alarm LED**
 - SF (red) Fault has occurred or reparameterization in progress
- **Special LEDs**
 - TXD (green) Sending active; lights up when the CP 341 is sending user data via the interface
 - RXD (green) Receiving active; lights up when the CP 341 is receiving user data via the interface

Note

Section 5.5 contains information on LED displays which can occur when you load firmware updates.

Group Alarm LED SF

The group alarm LED SF always lights up after POWER ON and goes out after initialization. If parameterization data has been generated for the CP 341, the SF LED again lights up briefly during reparameterization.

The group alarm LED SF lights up whenever the following occur:

- Hardware faults
- Firmware errors
- Parameterization error or no parameterization
- BREAKs (receive cable between CP 341 and communications partner becomes disconnected)

8.3 Diagnostics Messages of the Function Blocks

Each function block has a STATUS parameter for error diagnostics. The STATUS message numbers always have the same meaning, irrespective of which function block is used.

Numbering Scheme for Event Class/Event Number

The figure below illustrates the structure of the STATUS parameter.

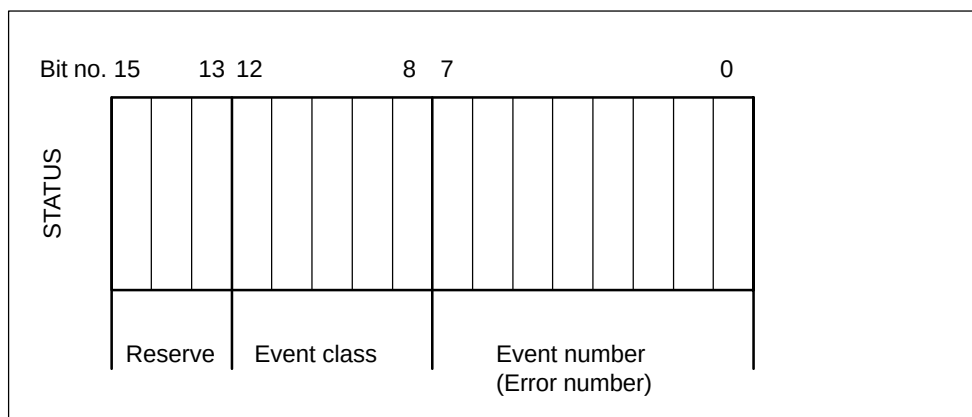


Figure 8-1 Structure of the STATUS Parameter

Example

The figure below illustrates the contents of the STATUS parameter for the event "Request aborted due to complete restart, restart or reset" (event class: 1EH, event number 0DH).

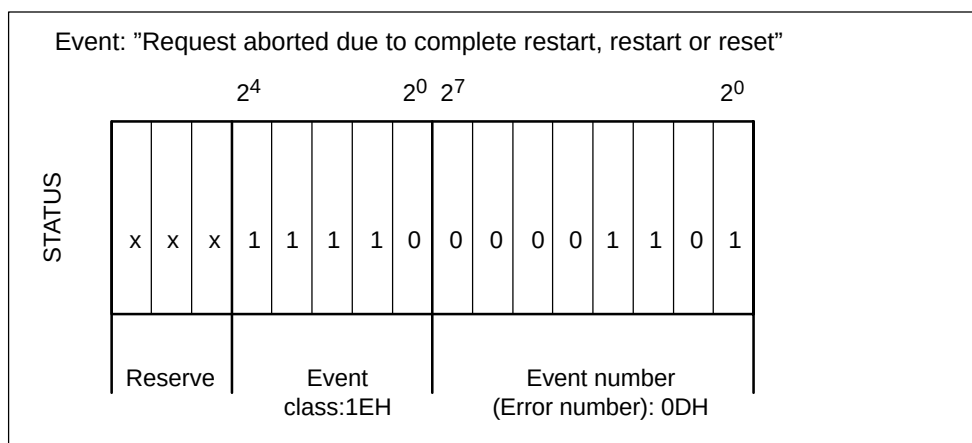


Figure 8-2 Example: Structure of the STATUS Parameter for the Event "Request Aborted Due to Complete Restart, Restart or Reset"

Event Classes

The table below describes the various event classes and numbers.

Table 8-1 Event Classes and Event Numbers

Event Class 0 (00H): "CP start-up"		
Event No.	Event Text	Remedy
(00)03H	PtP parameters accepted	–
(00)04H	Parameter already on CP (timers match)	–
(00)07H	Status transition: CPU to STOP	–
(00)08H	Status transition: CPU to RUN/STARTUP	–
Event Class 1 (01H): "Hardware fault on CP"		
Event No.	Event Text	Remedy
(01)01H	Fault while testing operating system EPROM of CP	CP defective; replace CP.
(01)02H	RAM test of CP errored	
(01)03H	Request interface of CP defective	
(01)10H	Fault in CP firmware	Switch module off and on again. If necessary, replace module.
Event Class 2 (02H): "Initialization error"		
Event No.	Event Text	Remedy
(02)0FH	Invalid parameterization detected at start of parameterized communication. Interface could not be parameterized	Correct invalid parameterization and restart.
Event Class 3(03H): "Error parameterizing FBs" (not displayed in diagnostic buffer)		
Event No.	Event Text	Remedy
(03)01H	Invalid or no source/destination data type Invalid area (start address, length) Invalid or no DB (e.g. DB 0), or Other data type invalid or missing Interprocessor communication flag byte number invalid or interprocessor communication flag bit number invalid or neither 'S' nor 'F' selected (for FB P_SND_RK)	Check parameterization on CPU and CP, and correct if necessary. RK 512 only: Partner returns invalid parameters in message frame header. Check parameterization on CPU and CP; possibly create block. See request tables for valid data types. RK 512 only: Partner returns incorrect parameters in message frame header.

Table 8-1 Event Classes and Event Numbers, continued

Event Class 4 (04H): "CP detected error in data traffic CP – CPU"		
Event No.	Event Text	Remedy
(04)03H	Incorrect, unknown or illegal data type	Check program, e.g. for incorrect parameterization of the FB.
(04)07H	Error during data transmission between CPU and CP	If fault indication persists, check whether function blocks you have called in user program are parameterized correctly. If error is indicated immediately after POWER ON, no connection has yet been set up to the CPU. In the case of the ASCII driver and the 3964(R) procedure, the receiving CP 341 re-attempts data transfer until the data is transmitted to the CPU. In the case of RK 512 the request receives a negative acknowledgment and must be repeated in the user program. If fault indication is sporadic in the course of data transfer, the CPU is temporarily unable to accept data. In the case of the ASCII driver and the 3964(R) procedure, the receiving CP 341 re-attempts data transfer until the data is transmitted to the CPU. In the case of RK 512 the request receives a negative acknowledgment and must be repeated in the user program. You can rectify the situation by calling the FB P_RCV_RK more frequently in your user program.
(04)08H	Error during data transmission between CPU and CP (reception) • CPU is temporarily overloaded, request queued for repetition • CPU data area temporarily unavailable for access, for example because receive block is called too infrequently. • CPU data area temporarily unavailable for access, for example because receive block is temporarily locked (EN=false).	• Reduce number of communication calls • Call the receive block more frequently • Check whether the receive block is disabled for too long

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event Text	Remedy
(04)09H	Data cannot be received. Error during data transmission between CPU and CP (reception) Data cannot be received. Request is canceled in 10 seconds following multiple attempts, because - Receive block is not called - Receive block is disabled - Access to CPU data area denied - CPU data area too short	- Check whether your application runs the receive block. - Check whether the receive block is disabled. - Check that the data area to which the data is to be transferred is available. - Check the length of the data area.
(04)0AH	Error during data transmission between CPU and CP Data transfer canceled by RESET because: - Destination DB is not available - Destination DB is too short - RESET bit set at FB	Create destination DB in the user program or increase the length of the existing destination DB, as applicable.
Event Class 5 (05H): "Error while processing CPU request"		
Event No.	Event	Remedy
(05)01H	Current request aborted as a result of CP restart.	No remedy is possible at POWER ON. When reparameterizing the CP from the programming device, before writing an interface you should ensure there are no more requests running from the CPU.
(05)02H	Request not permitted in this operating mode of CP (e.g. device interface not parameterized).	Parameterize the device interface.
(05)14H	Specified start addresses too high for desired data type, or start address or DB/DX number too low.	Obtain from the request tables the permissible start addresses and DB/DX numbers that can be specified in the program.
(05)15H	RK 512 only: Wrong bit number specified for interprocessor communication flag.	Permissible bit numbers: 0 to 7
(05)16H	RK 512 only: Specified CPU too high.	Permissible CPU numbers: none, 0, 1, 2, 3 or 4
(05)17H	Transmission length > 1 KB too great for CP or too short for interface parameters.	Split the request up into several shorter requests.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(05)1AH	RK 512 only: Error sending a command message frame An associated procedure number has just been entered in STATUS.	See the remedy for the previous error number.
Event Class 6 (06H): "Error processing a partner request" only with RK512		
Event No.	Event Text	Remedy
(06)01H	Error in 1st command byte (not 00 or FFH)	Header layout error at partner. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(06)02H	Error in 3rd command byte (not A, 0 or FFH)	Header layout error at partner. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(06)03H	Error in 3rd command byte in the case of continuation message frames (command not as for 1st message frame)	Header layout error at partner. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(06)04H	Error in 4th command byte (command letter incorrect)	Header layout error at partner or a command combination has been requested that is not permitted at the CP. Check the permissible commands. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(06)05H	Error in 4th command byte in the case of continuation message frames (command not as for 1st message frame)	Header layout error at partner. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(06)06H	Error in 5th command byte (DB number not permissible)	Obtain from the request tables the permissible DB numbers, start addresses or lengths.
(06)07H	Error in 5th or 6th command byte (start address too high)	Obtain from the request tables the permissible DB numbers, start addresses or lengths.
(06)08H	Error in 7th or 8th command byte (impermissible length)	Obtain from the request tables the permissible DB/DX numbers, start addresses or lengths.
(06)09H	Error in 9th and 10th command byte (coordination flag for this data type impermissible or bit number too high)	Header layout error at partner. Find out from the request tables when a coordination flag is permitted.
(06)0AH	Error in 10th command byte (CPU number not permitted)	Header layout error at partner.
(06)0BH	SEND message frame was longer/shorter than expected (more/less data received than announced in message frame header).	Correction required at the partner

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event Text	Remedy
(06)0CH	FETCH command message frame received with user data.	Correction required at the partner
(06)0DH	CP received message frame in impermissible mode. - Receive connection between CPU and CP not set up or not correctly set up - CP startup not fully completed - Status of receiving CPU is STOP mode - Interface is currently being reparameterized	- Check whether the addressed connection is (correctly) parameterized. - This error message can occur only during CP startup. Repeat the request. - Switch the CPU to RUN and repeat the request. - This is a temporary error. Repeat the request.
(06)0EH	Synchronous fault of partner - New (continuation) command message frame received before response message frame sent. 1st command message frame expected and continuation message frame came. - Continuation command message frame expected and 1st message frame came	This error may be reported after your own programmable controller is restarted in the case of long message frames or when the partner is restarted. These cases represent normal system start-up behavior. The error can also occur during operation as a consequence of error statuses only recognized by the partner. Otherwise, you have to assume an error on the part of the partner device. The error may not occur in the case of requests < 128 bytes.
(06)0FH	DB locked by coordination function	In own program: After processing of the last transmission data, reset the interprocessor communication flag. In partner program: Repeat the request
(06)10H	Message frame received too short (length < 4 bytes in the case of continuation or response message frames or < 10 bytes in the case of command message frames)	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(06)11H	Message frame length and length specified in message frame header are not the same.	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event Text	Remedy
(06)12H	Error sending the (continuation) response message frame. An associated procedure error number has been entered in STATUS immediately beforehand.	See remedy for the error number entered immediately beforehand in STATUS.
Event Class 7 (07H): "Send error"		
Event No.	Event	Remedy
(07)01H	Transmission of the first repetition: - An error was detected during transmission of the message frame, or - The partner requested a repetition by means of a negative acknowledgment code (NAK).	A repetition is not an error, but it can be an indication that there are disturbances on the transmission line or that the partner device is behaving incorrectly. If the message frame still has not been transmitted after the maximum number of repetitions, an error number describing the first error that occurred is output.
(07)02H	With 3964(R) only: Error during connection setup: After STX was sent, NAK or any other code (except for DLE or STX) was received.	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(07)03H	With 3964(R) only: Acknowledgment delay time exceeded: After STX was sent, no response came from partner within acknowledgment delay time.	Partner device is too slow or not ready to receive, or there is a break on the send line, for example. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(07)04H	With 3964(R) only: Termination by partner: During current send operation, one or more characters were received by partner.	Check whether the partner is also showing an error, possibly because not all transmission data has arrived (e.g. due to break on line) or due to serious faults or because the partner device has malfunctioned. If necessary, use an interface test device switched into the transmission line for this purpose.
(07)06H	With 3964(R) only: Error at end of connection: - Partner rejected message frame at end of connection with NAK or a random string (except for DLE), or - Acknowledgment code (DLE) received too early.	Check whether the partner is also showing an error, possibly because not all transmission data has arrived (e.g. due to break on line) or due to serious faults or because the partner device has malfunctioned. If necessary, use an interface test device switched into the transmission line for this purpose.
(07)07H	With 3964(R) only: Acknowledgment delay time exceeded at end of connection or response monitoring time exceeded after send message frame: After connection release with DLE ETX, no response received from partner within acknowledgment delay time.	Partner device faulty or too slow. If necessary, use an interface test device switched into the transmission line to check.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(07)08H	With ASCII driver only: The waiting time for XON or CTS = ON has elapsed.	The communication partner has a fault, is too slow or is switched off-line. Check the communication partner or, if necessary, change the parameterization.
(07)09H	Connection setup not possible. Number of permitted setup attempts exceeded.	Check the interface cable or the transmission parameters. Also check that receive function between CPU and CP is correctly parameterized at the partner device.
(07)0AH	The data could not be transmitted. The permitted number of transfer attempts was exceeded.	Check the interface cable or the transmission parameters.
Event Class 8 (08H): "Receive error"		
Event No.	Event	Remedy
(08)01H	Expectation of the first repetition: An error was detected on receipt of a message frame, and the CP requests a repetition by means of negative acknowledgment (NAK) at the partner.	A repetition is not an error, but it can be an indication that there are disturbances on the transmission line or that the partner device is behaving incorrectly. If the message frame still has not been transmitted after the maximum number of repetitions, an error number describing the first error that occurred is output.
(08)02H	With 3964(R) only: Error during connection setup: • In idle mode, one or more random codes (other than NAK or STX) were received, or • after an STX was received, partner sent more codes without waiting for response DLE. After the partner has signaled POWER ON: • While partner is being activated, CP receives an undefined code.	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(08)05H	With 3964(R) only: Logical error during receiving: After DLE was received, a further random code (other than DLE or ETX) was received.	Check whether partner DLE in message frame header and in data string is always in duplicate or the connection is released with DLE ETX. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(08)06H	Character delay time exceeded: - Two successive characters were not received within character delay time, or With 3964(R) only: 1st character after sending of DLE during connection setup was not received within character delay time.	Partner device faulty or too slow. Use an interface test device switched into the transmission line to check.
(08)08H	With 3964(R) only: Error in block check character (BCC) Internally calculated value of BCC does not match BCC received by partner at end of connection.	Check whether connection is badly damaged; in this case you may also occasionally see error codes. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(08)0AH	There is no free receive buffer available: No empty receive buffer was available to receive data.	The P_RCV_RK FB must be called more frequently.
(08)0CH	Transmission error: - Transmission error (parity error, stop bit error, overflow error) detected. With 3964(R) only: - If faulty character is received in idle mode, the error is reported immediately so that disturbances on the transmission line can be detected early. Only in the case of RK 512 and 3964(R): - If this occurs during send or receive operation, repetitions are initiated.	Disturbances on the transmission line cause message frame repetitions, thus lowering user data throughput. Danger of an undetected error increases. Correct fault by changing system setup or line installation. Check connecting cable of communications partner or check whether both devices have same setting for baud rate, parity and number of stop bits.
(08)0DH	BREAK: Receive line to partner is interrupted.	Reconnect or switch partner on again.
(08)15H	Discrepancy between settings for transfer attempts at CP and communication partner.	Parameterize same number of transfer attempts at communications partner as at CP. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(08)16H	- The length of a received message frame was longer than the length agreed upon or - the length of the parameterized receive buffer (with CP 441 only) is too short.	- A correction is necessary at the partner or - the length of the receive buffer (with CP 441 only) must be enlarged
(08)18H	With ASCII driver only: DSR = OFF or CTS = OFF	The partner has switched the DSR or CTS signal to "OFF" before or during a transmission. Check the partner's control of the RS 232C secondary signals.

Table 8-1 Event Classes and Event Numbers, continued

Event Class 9 (09H): "Response message frame received from interconnection partner with error or error message frame"		
Event No.	Event Text	Remedy
(09)02H	RK 512 only: Memory access error at partner (memory does not exist) When SIMATIC S5 is the partner: • Incorrect area at status word, or • Data area does not exist (except DB/DX), or • Data area too short (except DB/DX)	Check that the partner has the desired data area and that it is big enough, or check the parameters of the called system function block. Check the specified length at the system function block.
(09)03H	RK 512 only DB/DX access error at the partner (DB/DX does not exist or is too short) When SIMATIC S5 is the partner: • DB/DX does not exist, or • DB/DX too short, or • DB/DX number impermissible Permissible source area for FETCH request exceeded	Check that the partner has the desired data area and that it is big enough, or check the parameters of the called system function block. Check the specified length at the system function block.
(09)04H	RK 512 only: Partner returns "Request type not permitted".	Partner malfunction, because a system command is never issued from the CP.
(09)05H	RK 512 only: Error at partner or at SIMATIC S5 as partner: • Source/destination type not permissible, or • Memory error in partner programmable controller, or • Error notifying CP/CPU at the partner, or • Partner programmable controller is in STOP state	Check whether the partner can transmit the desired data type. Check the structure of the hardware at the partner. Set the partner programmable controller to RUN.
(09)08H	RK 512 only: Partner detecting synchronization error: Message frame sequence error.	This error occurs at restart of your own programmable controller or of the partner. This represents normal system start-up behavior. You do not need to correct anything. The error is also conceivable during operation as a consequence of previous errors. Otherwise, you can assume an error on the part of the partner device.
(09)09H	RK 512 only: DB/DX disabled at partner by coordination flag	In partner program: After processing of the last transmission data, reset the coordination flag. In own program: Repeat the request.
(09)0AH	RK 512 only: Error detected by partner in message frame header: 3rd command byte in header is incorrect	Check whether the error is the result of disturbances or of a malfunction at the partner. Use an interface test device switched into the transmission line to check.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event Text	Remedy
(09)0BH	RK 512 only: Error in message frame header: 1st or 4th command byte in header is incorrect	Check whether the error is the result of disturbances or of a malfunction at the partner. Use an interface test device switched into the transmission line to check.
(09)0CH	RK 512 only: Partner detects incorrect message frame length (total length).	Check whether the error is the result of disturbances or of a malfunction at the partner. Use an interface test device switched into the transmission line to check.
(09)0DH	RK 512 only: Partner has not yet restarted.	Restart the partner programmable controller or set the mode selector on the CP to RUN.
(09)0EH	RK 512 only: Unknown error number received in response message frame.	Check whether the error is the result of disturbances or of a malfunction at the partner. Use an interface test device switched into the transmission line to check.
Event class 10 (0AH): "Errors in response message frame of the partner detected by the CP"		
Event No.	Event Text	Remedy
(0A)01H	RK 512 only: Synchronization error of partner, because: • Response message frame without request • Response message frame received before continuation message frame sent • Continuation response message frame received after an initial message frame was sent • A first response message frame was received after a continuation message frame was sent	This error is reported after your own programming device is restarted in the case of long message frames or when the partner is restarted. This represents normal system start-up behavior. You do not have to correct anything. The error can also occur during operation as a consequence of error statuses only recognized by the partner. Otherwise, you can assume an error on the part of the partner device. The error may not occur in the case of requests < 128 bytes.
(0A)02H	RK 512 only: Error in the structure of the received response message frame (1st byte not 00 or FF)	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(0A)03H	RK 512 only: Received response message frame has too many data or not enough data.	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event Text	Remedy
(0A)04H	RK 512 only: Response message frame for SEND request arrived with data.	Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(0A)05H	RK 512 only: No response message frame from partner within monitoring time.	Is the partner a slow device? This error is also often displayed as a consequence of a previous error. For example, procedure receive errors (event class 8) can be displayed after a FETCH message frame was sent. Reason: As a result of disturbances, the response message frame could not be received, and the monitoring time elapsed. This error can also occur if a restart was carried out at the partner before it could respond to the most recently received FETCH message frame.

Displaying and Evaluating the STATUS Output

You can display and interpret the actual operands in the STATUS output of the function blocks.

Note

An error message is only output if the ERROR bit (request completed with error) is set. In all other cases the STATUS word is zero.

Event Class 30

Event class 30 contains error messages which might occur during communication between the CP 341 and the CPU via the S7 backplane bus.

The table below describes event class 30.

Table 8-2 Event Class 30

Event Class 30 (1EH): "Error during communication between CP and CPU"		
Event No.	Event	Further Information/Remedy
(1E)0DH	Request aborted due to complete restart, restart or reset	
(1E)0EH	Static error when the RD_RED SFC was called. Return value RET_VAL of SFC is available for evaluation in SFCERR variable in instance DB.	Load SFCERR variable from instance DB.
(1E)0FH	Static error when the WR_REC SFC was called. Return value RET_VAL of SFC is available for evaluation in SFCERR variable in instance DB.	Load SFCERR variable from instance DB.
(1E)41H	Number of bytes set in LEN parameter of FBs illegal.	Keep to the value range of 1 to 1024 bytes.

Calling the SFCERR Variable

You can obtain more information on errors 14 (1E0EH) and 15 (1E0FH) in event class 30 by means of the SFCERR variable.

You can load the SFCERR variable from the instance DB belonging to the corresponding function block. The programming example in Chapter 9 shows how you can load the variable SFCERR.

The error messages entered in the SFCERR variable are described in the section on the system functions SFC 58 "WR_REC" and SFC 59 "RD_REC" in the *System Software for S7-300/400, System and Standard Functions* reference manual.

8.4 Error Numbers in the Response Message Frame

If you are working with the RK 512 computer connection and an error occurs at the communication partner in a SEND or FETCH message frame, the communication partner sends a response message frame with an error number in the 4th byte.

Error Numbers in the Response Message Frame

The table below shows how the error numbers in the response message frame (REATEL) are assigned to the event classes/numbers in the STATUS output of the communication partner. The error numbers in the response message frame are output as hexadecimal values.

Table 8-3 Error Messages in the Response Message Frame with RK 512

REATEL	Error Messages
	Event Class/Event Number
0AH	0905H
0CH	0301H 0607H 0609H 060AH 0902H
10H	0301H 0601H 0604H 0605H 090BH
12H	0904H
14H	0606H 0903H
16H	0602H 0603H 090AH
2AH	060DH 090DH
32H	060FH 0909H

Table 8-3 Error Messages in the Response Message Frame with RK 512, continued

REATEL	Error Messages
	Event Class/Event Number
34H	0608H
	060BH
	060CH
	0611H
	090CH
36H	060EH
	0908H

8.5 Diagnosis by Means of the Diagnostic Buffer of the CP 341

Diagnostic Buffer on the CP 341

The CP 341 has its own diagnostic buffer in which all the diagnostic events of the CP 341 are entered in the order in which they occur.

The following are displayed in the diagnostic buffer of the CP 341:

- Hardware/firmware errors on the CP 341
- Initialization and parameterization errors
- Errors during execution of a CPU request
- Data transmission errors (send and receive errors)

The diagnostic buffer allows the causes of errors in point-to-point communication to be evaluated subsequently in order, for example, to determine the causes of a STOP of the CP 341 or to trace the occurrence of individual diagnostic events.

Note

The diagnostic buffer is a ring buffer for a maximum of 9 diagnostic entries. When the diagnostic buffer is full, the oldest entry is deleted when a new entry is made in it.

This means that the most recent entry is always first. The contents of the diagnostics buffer are lost in the event of a POWER OFF or when the CP 341 is reparameterized.

Note

If you want to see the times of the individual diagnostic entries, you must select the CPU in "HW Config" and synchronize the clocks in the "Diagnosis/Clock" folder (synchronization mode "Master", time interval e. g. 10 seconds). The time cannot be displayed if the CP 341 is operated in a distributed configuration (ET 200M).

Reading the Diagnostic Buffer at the Programming Device

The contents of the diagnostic buffer of the CP 341 can be read by means of the STEP 7 information functions.

Note

Diagnostic events in the diagnostic buffer of the CP 341 can be read using STEP 7 **as of Version 3.2**.

All the user-relevant information in the CP diagnostic buffer is displayed to you on the "Diagnostic Buffer" in the "Module Information" dialog box. You can call the "Module Information" dialog box under STEP 7 from SIMATIC Manager.

Precondition: In order to obtain the status of the module, there must be an on-line connection from the programming device to the programmable controller (on-line view in the project window).

Proceed as follows:

1. Open the relevant SIMATIC 300 station (by double-clicking it or by choosing the **Edit > Open Object**) menu command.
2. Open the "Hardware" object contained in it (again by double-clicking it or by choosing the **Edit > Open**) menu command.

Result: The window containing the configuration table appears.

3. Select the CP 341 in the configuration table.
4. Choose the **PLC > Module** menu command.

Result: The "Module Information" dialog box appears for the CP 341. The "General" tab is displayed by default the first time you call it.

5. Select the "Diagnostic Buffer" tab.

Result: The "Diagnostic Buffer" tab displays the most recent diagnostic events of the CP 341. Any additional information on the cause of the problem appears in the "Details of the event" part of the tab.

The event's numeric code is displayed in the "Event ID" field. The initial 16#F1C8 is always the same. The rest of the ID code corresponds to event class and event number of the events described in Section 8.3.

By clicking the "Help on Event" button you can display the help text on the event text as described in Section 8.4.

If you click the "Update" button, the current data is read from the CP 341. By clicking the "Help on Event" button you can display a help text on the selected diagnostic event with information on error correction.

8.6 Diagnostic Alarm

The CP 341 can trigger a diagnostics alarm on the assigned CPU, thus indicating a malfunction of the CP 341. You can specify at parameterization whether the CP 341 is to trigger a diagnostics alarm or not in the event of serious errors.

"Diagnostics alarm = NO" is the default.

Diagnostics Alarm

In the event of an error, the CP 341 provides diagnostics data on the S7-300 backplane bus. In response to a diagnostics alarm, the CPU reads the system-specific diagnostics data and enters it in its diagnostics buffer. You can read the contents of the diagnostics buffer on the CPU by means of an attached programming device.

When a diagnostics event occurs, the SF LED (red) lights up. In addition, the OB 82 is called with this diagnostics data as start information.

Organization Block OB 82

You have the option of programming error responses in the user program in OB 82.

If no OB 82 is programmed, the CPU automatically enters STOP mode in the event of a diagnostics alarm.

Diagnostics Information (as Bit Pattern)

The CP 341 provides 4 bytes of diagnostics information. To display the error that has occurred, these 4 bytes are occupied as follows:

2nd byte:

The 2nd byte of diagnostics data contains the class ID of the CP 341 in bits 0 to 3.

2nd byte							
7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0

1st, 3rd and 4th byte:

The 1st, 3rd and 4th bytes of the diagnostics data represent the error which has occurred.

Bit 0 in the 1st byte is the group error display (SF). Bit 0 is always set to "1" if at least one bit from bits 1 to 7 is set to "1", i.e. if at least one error is entered in the diagnostics data.

Event	1st byte								3rd byte								4th byte							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Wire break	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Incorrect parameter	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Diagnostics Information (hexadecimal)

The table below shows the 4th byte in diagnostics data of the CP 341 in hexadecimal notation.

Event	1st byte	2nd byte	3rd byte	4th byte
Wire break	25H	0CH	02H	00H
Incorrect parameter	83H	0CH	00H	00H

Dependency of Diagnostics Alarm on CPU Operating Mode

A diagnostics alarm is generated via the I/O bus when fault events (rising edge) and back-to-normal events (falling edge) occur.

When the CPU switches from STOP mode to RUN mode, the following happens:

- Events (both fault and back-to-normal) which occurred when the CPU was in STOP mode are not stored,
- Events that are still present when the CPU is back to RUN mode are signaled via the diagnostics alarm.

Programming Example for Standard Function Blocks

9

In Section	You Will Find	on Page
9.1	General	9-2
9.2	Device Configuration	9-3
9.3	Settings	9-3
9.4	Blocks Used	9-5
9.5	Installation, Error Messages	9-6
9.6	Activation, Start-Up Program and Cyclic Program	9-7

9.1 General

The programming example given here and included in the **cp340_41** project describes standard functions for operating the CP 341 communications processor.

Objectives

The programming example

- aims to show examples of the most important functions
- enables the correct functioning of the connected hardware to be checked (and is therefore simple and easy to follow)
- can be extended for your own purposes without difficulty.

The example shows how an 3964(R)/ASCII connection or an RK 512 computer connection to a communications partner can be configured using the standard function blocks P_SND_RK and P_RCV_RK (to send and receive data respectively).

The example also shows how the inputs and outputs of the CP 341 can be controlled and monitored using the V24_STAT and V24_SET standard functions.

There are three SIMATIC stations in the example because the CP 341 has to be parameterized differently for data transfer:

- CP341 protocol 3964: Connection with FB P_SND_RK and FB P_RCV_RK
- CP341 protocol RK512: Connection with FB P_SND_RK and FB P_RCV_RK
- CP341 V24: Read and control RS 232C secondary signals with FC V24_STAT and FC V24_SET

Note that the “CP340 PTP Connection” (point-to-point) and “CP340 Printing and V24” stations contain the examples for the CP 340.

The CP 341 is parameterized by the CPU when the latter is started up (system service).

Requirements

The example can be executed with the minimum hardware equipment. The STEP 7 function **Monitor/Modify Variables** is also used (e.g. to modify transmitted data).

Program Example

The programming example of the CP 341, together with the parameterization interface and the function blocks, is supplied on the installation CD which come with this manual.

It is available both compiled and as an ASCII source file. A list of all the symbols used in the example is also included.

9.2 Device Configuration

Using the Example

The following are some of the devices that could be used to try out the program example:

- One S7-300 PLC (mounting rack, power supply, CPU)
- One CP 341 module with a communications partner (e.g. a second CP), or you could plug in a "short-circuit connector", i.e. the send line is bridged to the receive line
- One programming device (e.g. PG 740).

9.3 Settings

Settings in the CPU by means of STEP 7

Use STEP 7 to configure your controller as follows.

- Slot 1: power supply
- Slot 2: CPU
- Slot 4: CP 341, start address 256
- Slot 5: CP 341, start address 272

Settings on the CP 341

You cannot make any hardware settings on the CP 341.

Use STEP 7 to configure all relevant data, including the parameters for the CP 341 with the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface and upload them to the CPU.

You can run the "CP341 protocol 3964" program example without making changes in the application program with:

- 3964(R) procedure
- ASCII driver with "on expiry of character delay time" end criterion
- ASCII driver with "on receipt of fixed message frame length" end criterion.

For the ASCII driver with the "on receipt of the end character(s)" end criterion, you must also program the end codes.

The functions for reading and controlling RS 232C secondary signals can only be executed with the ASCII driver. The prerequisite for control is that the "Automatic Use of V24 Signals" parameter in the "Transmission" tab is not selected.

9.4 Blocks Used

Blocks Used

The table below lists the blocks used for the example program.

Block	Symbol	Explanation
OB 1	CYCLE	Cyclic program processing
OB 100	RESTART	Cold restart processing
DB 21	SEND IDB	Instance DB for FB P_SND_RK
DB 22	RECV IDB	Instance DB for FB P_RCV_RK
DB 40	SEND WORD DB	Work DB for the standard FB 8
DB 41	RECV WORK DB	Work DB for the standard FB 7
DB 42	SEND SRC DB	Send data block
DB 43	RECV DST DB	Receive data block
FB 7	P_RCV_RK	Standard FB for receiving data (RK 512)
FB 8	P_SND_RK	Standard FB for sending data (RK 512)
FC 5	V24_STAT	Standard FC for reading CP outputs
FC 6	V24_SET	Standard FC for writing to CP outputs
FC 14	V24_CYC	Control RS 232C secondary signals
FC 21	SEND	Send data
FC 22	RECEIVE	Receive data

9.5 Installation, Error Messages

Scope of Supply and Installation

The programming example of the CP 341, together with the *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface and the function blocks, is supplied on a 3.5" CD which comes with this manual.

The program example is installed together with the parameterization interface. Installation is described in Section 5.2. After installation, the program example is stored in the following project:

CP340_41

Open the project using the STEP 7 SIMATIC Manager by calling the menu command **File > Open > Project**.

The program example is available both compiled and as an ASCII source file. A list of all the symbols used in the example is also included.

If you do not have access to a second CP 341 as communication partner, open the HW Config and remove the CP 341 with the **Edit > Delete >** menu command. In OB 1, moreover, you must bracket the call for FC 22 (FC for Receive) as a comment.

Loading into the CPU

The hardware for the example is fully set up and the programming device is connected.

After the overall reset of the CPU (operating mode STOP), transfer the complete example to the user memory. Then use the operating mode switch to change from STOP to RUN.

Malfunction

If an error occurs during start-up, the cyclically processed module calls are not executed and the error display is activated.

If there is an error message, the parameter output ERROR of the modules is set. A more precise error description is then stored in the STATUS parameter of the blocks. If STATUS contains one of the error messages 16#1E0E or 16#1E0F, the exact error description is stored in the SFCERR variable in the instance DB.

9.6 Activation, Start-Up Program and Cyclic Program

Activation, Start-Up Program

The start-up program is located in the OB 100.

The control bits and the counters are reset in the start-up procedure

Cyclic Program

The cyclic program is defined in the organization block OB 1.

In the example, the function blocks FB 7 P_RCV_RK and FB 8 P_SND_RK work with the functions FC 21 and FC 22 and with data blocks DB 21 and DB 22 as instance DBs and DB 42 and DB 43 as send and receive DBs.

The functions FC 5 V24_STAT and FC 6 V24_SET work with the function FC 14.

In the example the function blocks are parameterized partly via constants and partly via symbolically addressed actual operands.

Description, "CP341 Protocol 3964", "CP341 Protocol RK512"

Data is transferred from the CP 341 in slot 4 to the CP 341 in slot 5. If you are using some other communication partner, the FC 22 call (RECEIVE) is omitted.

Description of FC 21 (SEND)

The "Generate edge P_SND_REQ" program section:

The P_SND_RK is run through once at the start with P_SND_RK REQ=0. P_SND_RK REQ is then set to 1. The P_SND_RK request is started when a signal state change from "0" to "1" is detected at the P_SND_RK REQ control parameter.

When P_SND_RK DONE=1 or P_SND_RK ERROR=1, P_SND_RK REQ is reset to 0.

The "P_SND_RK DONE=1" program section:

If the transfer is successful, P_SND_RK DONE at the parameter output of the P_SND_RK is set to 1.

To distinguish between consecutive transfers, a P_SND_RK COUNTER_OK send counter is included in data word 0 of the source block DB 42.

The "P_SND_RK ERROR=1" program section:

If P_SND_RK runs through with P_SND_RK ERROR=1, the P_SND_RK COUNTER_ERR error counter in data word 2 increments. The P_SND_RK STATUS is copied, because it will be overwritten with 0 in the next run and could not subsequently be read.

Description of FC 22 (RECEIVE)

The "Enable Receive Data" program section:

For data to be received, the receive enable P_RCV_RK EN_R in block P_RCV_RK must be set to 1.

The "P_RCV_RK NDR=1" program section:

When P_RCV_RK NDR is set, new data has been received and the P_RCV_RK COUNTER_OK receive counter increments.

The "P_RCV_RK ERROR=1" program section:

If the outcome is unsuccessful, i.e. if the error bit is set at the parameter output of P_RCV_RK, the P_RCV_RK COUNTER_ERR error counter increments. The P_RCV_RK STATUS is copied, because it will be overwritten with 0 in the next run and could not subsequently be read.

All relevant values can be observed for test purposes in the variable table.

Description of "CP341 V24"

The functions for reading and controlling RS 232C secondary signals can only be executed with the ASCII driver. The prerequisite for control is that the "Automatic Use of V24 Signals" parameter in the "Transmission" tab is not selected.

The V24 signals can be read and controlled with the aid of the variable table. The signal states SET_DTR and SET_RTS can be preselected by means of flags F 1.6 and F 1.7. When the signal at F 0.7 changes from "0" → "1", this state is transferred to the CP by the V24_SET function.

The V24_STAT function is called cyclically. The status of the CP 341 V24 signals can be read by means of flags 3.0 to 3.5.

Technical Specifications

A

In Section	You Will Find	on Page
A.1	Technical Specifications of the CP 341	A-2
A.2	Transmission Times	A-7
A.3	Certification and Areas of Application	A-9

A.1 Technical Specifications of the CP 341

General Technical Specifications

The following table contains the general technical specifications of the CP 341.

The technical specifications for the function blocks are in Section 6.6.

You will find more general technical specifications for the SIMATIC S7-300 in the reference manual entitled *S7-300 and M7-300 Programmable Controllers, Module Specifications*, Section 1 “General Technical Specifications”.

Table A-1 General Technical Specifications

Technical Specifications	
Dimensions W × H × D	40 × 125 × 120 mm
Weight	0.3 kg
Current consumption at 24 V (DC 24 V supply via front-panel connector)	CP 341-RS 232C: typ. 200 mA CP 341-20mA TTY typ. 200 mA CP 341-RS 422/485: typ. 240 mA
Range, static	20.4...28.8 V
Range, dynamic	18.5...30.2 V
Polarity reversal protection	yes
Isolation	yes, against all other voltages
Current consumption from backplane bus	approx. 70 mA
Power loss	CP 341-RS 232C: typ. 4.8 W CP 341-20mA TTY: typ. 4.8 W CP 341-RS 422/485: typ. 5.8 W
Indicators	LEDs for transmit (TXD), receive (RXD) and group alarm (SF)
Alarms	
Diagnostics alarm	parameterizable
Diagnostic functions	
• Diagnostics information dump	yes
Protocol drivers	ASCII driver 3964(R) procedure RK 512 computer connection
Transmission rates with 3964(R) protocol	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 baud (half-duplex)
Transmission rates with the RK 512 computer connection	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 76800 baud (half-duplex)
Transmission rates with ASCII driver	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 baud

Table A-1 General Technical Specifications

Technical Specifications	
Character frame	number of bits per character (7 or 8) number of start/stop bits (1 or 2) parity (none, even, odd)
User data transported per program cycle	send/receive: 32 bytes
Memory requirements, FB P_SND_RK and FB P_RCV_RK (main memory)	together approx. 5500 bytes

Technical Specifications of the RS 232C Interface

The following table contains the technical specifications of the RS 232C interface of the CP 341-RS 232C:

Table A-2 Technical Specifications of the RS 232C Interface

Technical Specifications	
Interface	RS 232C, 9-pin sub D male
RS 232C signals	TXD, RXD, RTS, CTS, DTR, DSR, RI, DCD, GND all isolated against the S7-internal power supply (backplane bus) and the external 24 V DC supply
Max. distance	15 m
Max. baud rate	76800 baud

Technical Specifications of the 20 mA TTY Interface

The following table contains the technical specifications of the 20 mA TTY interface of the CP 341-20mA TTY:

Table A-3 Technical Specifications of the 20 mA TTY Interface

Technical Specifications	
Interface	20 mA current-loop TTY, 9-pin sub D female
TTY signals	two isolated 20 mA sources, receive (RX) "-" and "+" transmit (TX) "-" and "+" all isolated against the S7-internal power supply (backplane bus) and the external 24 V DC supply
Max. distance	1000 m active, 1000 m passive
Max. baud rate	19200 baud

Technical Specifications of the X27 (RS 422/485) Interface

The following table contains the technical specifications of the X27 (RS 422/ 485) interface of the CP 341-RS 422/485.

Table A-4 Technical Specifications of the X27 (RS 422/485) Interface

Technical Specifications	
Interface	RS 422 or RS 485, 15-pin sub D female
RS 422 signals	TXD (A), RXD (A), TXD (B), RXD (B), GND
RS 485 signals	R/T (A), R/T (B), GND all isolated against the S7-internal power supply (backplane bus) and the external 24 V DC supply
Max. distance	1200 m
Max. baud rate	76800 baud

Technical Specifications of the 3964(R) Procedure

The following table contains the technical specifications of the 3964(R) procedure.

Table A-5 Technical Specifications of the 3964(R) Procedure

3964(R) procedure with default values	
Max. message frame length	1024 bytes
Parameters	parameterizable: • with/without block check character • priority: low/high • transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 baud • character frame: 9, 10, 11 or 12 bits • initial state of receive line: none, R(A)5V/R(B)0V, R(A)0V/R(B)5V

Table A-5 Technical Specifications of the 3964(R) Procedure

3964(R) procedure, parameterizable	
Max. message frame length	1024 bytes
Parameters	parameterizable: • with/without block check character • priority: low/high • transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 baud • character frame: 9, 10, 11 or 12 bits • character delay time: 20 ms to 655350 ms in 10 ms increments • acknowledgment delay time: 20 ms to 655350 ms in 10 ms increments • number of setup attempts: 1 to 255 • number of transmission attempts: 1 to 255 • initial state of receive line: none, R(A)5V/R(B)0V, R(A)0V/R(B)5V

Technical Specifications of the RK 512 Computer Connection

The following table contains the technical specifications of the RK 512 computer connection.

Table A-6 Technical Specifications of the RK 512 Computer Connection

RK 512 computer connection	
Max. message frame length	1024 bytes
Parameters	parameterizable: • transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 baud • character frame: 10, 11 or 12 bits • character delay time: 20 ms to 655350 ms in 10 ms increments • acknowledgment delay time: 20 ms to 655350 ms in 10 ms increments • number of setup attempts: 1 to 255 • number of transmission attempts: 1 to 255 • initial state of receive line: none, R(A)5V/R(B)0V, R(A)0V/R(B)5V

Technical Specifications of the ASCII Driver

The following table contains the technical specifications of the ASCII driver.

Table A-5 Technical Specifications of the ASCII Driver

ASCII driver	
Max. message frame length	1024 bytes
Parameters	parameterizable: • transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 baud • character frame: 9, 10, 11 or 12 bits • character delay time: 2 ms to 65535 ms in 1 ms increments • flow control: none, XON/XOFF, RTS/CTS; Automatic use of the V.24 signals • XON/XOFF characters (only with "flow control" = "XON/XOFF") • wait for XON after XOFF (waiting time for CTS=ON): 20 ms to 655350 ms in 10 ms increments • time to RTS off: 0 ms to 655350 ms in 10 ms increments (only with "automatic use of the V.24 signals") • data output wait time: 0 ms to 655350 ms in 10 ms increments (only with "automatic use of the V.24 signals") • number of buffered receive message frames: 1 to 250 • disable overwrite: yes/no (only with "buffered receive message frames" = 1") • indicator for end of receive message frame: – on expiry of character delay time – on receipt of end-of-text character(s) – on receipt of fixed number of characters
ASCII driver with end of message frame recognition by expiry of character delay time	
Parameters	No further parameterization necessary. End of message frame is recognized when the parameterized character delay time expires.
ASCII driver with end of message frame recognition by parameterizable end character	
Parameters	also parameterizable: • number of end-of-text characters: 1 or 2 • hex code for first/second end-of-text character
ASCII driver with end of message frame recognition by configured message-frame length	
Parameters	also parameterizable: • message frame length: 1 to 1024 bytes

A.2 Transmission Times

Transmission Times

The tables below indicate the transmission times required depending on the transmission protocol selected.

Two S7-300s each with a CPU 315-2DP (6ES7 315-2AF01-0AB0) and a CP 341 were used to measure the times. An FB P_SND_RK function block was programmed in the user program of the active CPU, and an FB P_RCV_RK function block was programmed in the user program of the passive CPU. The time that elapsed between the initiation and completion of the request was measured.

ASCII driver

Transmission times with the ASCII driver:

Table A-7 Transmission Times with the ASCII Driver

Baud rate (bd) User	76800	57600	38400	19200	9600	4800	2400	1200	600	300
1 byte	0.010 s	0.010 s	0.010 s	0.010 s	0.010 s	0.010 s	0.020 s	0.039 s	0.077 s	0.154 s
10 bytes	0.010 s	0.010 s	0.010 s	0.020 s	0.020 s	0.031 s	0.062 s	0.131 s	0.251 s	0.492 s
20 bytes	0.011 s	0.013 s	0.020 s	0.020 s	0.030 s	0.055 s	0.110 s	0.220 s	0.431 s	0.853 s
50 bytes	0.020 s	0.026 s	0.030 s	0.040 s	0.080 s	0.140 s	0.251 s	0.492 s	1.000 s	2.000 s
100 bytes	0.035 s	0.043 s	0.050 s	0.090 s	0.150 s	0.258 s	0.491 s	0.952 s	2.000 s	4.000 s
200 bytes	0.060 s	0.082 s	0.100 s	0.160 s	0.271 s	0.501 s	1.000 s	2.000 s	4.000 s	8.000 s
500 bytes	0.145 s	0.191 s	0.206 s	0.357 s	0.651 s	1.213 s	2.400 s	4.800 s	9.600 s	18.800 s
1000 bytes	0.261 s	0.335 s	0.402 s	0.692 s	1.263 s	2.400 s	4.800 s	10.600 s	21.200 s	37.600 s

3964(R) procedure

Transmission times with the 3964(R) procedure:

Table A-8 Transmission Times with the 3964(R) Procedure

Baud rate (bd) User data	76800	57600	38400	19200	9600	4800	2400	1200	600	300
1 byte	0.010 s	0.010 s	0.011 s	0.020 s	0.020 s	0.027 s	0.042 s	0.076 s	0.139 s	0.271 s
10 bytes	0.011 s	0.016 s	0.020 s	0.020 s	0.030 s	0.050 s	0.083 s	0.158 s	0.305 s	0.600 s
20 bytes	0.021 s	0.021 s	0.021 s	0.031 s	0.040 s	0.071 s	0.129 s	0.251 s	0.490 s	0.969 s
50 bytes	0.027 s	0.029 s	0.030 s	0.050 s	0.080 s	0.142 s	0.272 s	0.528 s	1.000 s	2.000 s
100 bytes	0.041 s	0.046 s	0.051 s	0.081 s	0.145 s	0.262 s	0.506 s	0.993 s	2.100 s	4.000 s
200 bytes	0.060 s	0.077 s	0.090 s	0.151 s	0.272 s	0.500 s	1.000 s	2.000 s	4.200 s	8.000 s
500 bytes	0.129 s	0.175 s	0.202 s	0.351 s	0.642 s	1.220 s	2.400 s	4.800 s	9.600 s	18.800 s
1000 bytes	0.251 s	0.297 s	0.342 s	0.681 s	1.260 s	2.415 s	4.800 s	10.600 s	21.200 s	37.600 s

RK 512 computer connection

Transmission times with the RK 512 computer connection:

Table A-9 Transmission Times with the RK 512 Computer Connection

Baud rate (bd) User data	76800	57600	38400	19200	9600	4800	2400	1200	600	300
1 byte	0.027 s	0.029 s	0.030 s	0.038 s	0.054 s	0.083 s	0.144 s	0.266 s	0.522 s	1.046 s
10 bytes	0.029 s	0.032 s	0.034 s	0.043 s	0.063 s	0.098 s	0.181 s	0.338 s	0.666 s	1.334 s
20 bytes	0.030 s	0.034 s	0.037 s	0.049 s	0.075 s	0.125 s	0.227 s	0.431 s	0.855 s	1.701 s
50 bytes	0.041 s	0.047 s	0.051 s	0.072 s	0.114 s	0.199 s	0.372 s	0.712 s	1.407 s	2.804 s
100 bytes	0.061 s	0.069 s	0.076 s	0.114 s	0.184 s	0.326 s	0.612 s	1.183 s	2.326 s	4.645 s
200 bytes	0.105 s	0.125 s	0.141 s	0.211 s	0.350 s	0.635 s	1.200 s	2.400 s	4.800 s	9.067 s
500 bytes	0.221 s	0.265 s	0.301 s	0.471 s	0.812 s	1.666 s	3.000 s	4.800 s	11.000 s	22.000 s
1000 bytes	0.441 s	0.517 s	0.592 s	0.912 s	1.700 s	3.000 s	6.000 s	11.000 s	22.000 s	44.000 s

A.3 Certification and Areas of Application

This section contains information on:

- The most important standards with which the CP 341 complies
- Certificates and approvals of the CP 341

Safety

The CP 341 communications processor meets the requirements and criteria for the safety of electrical equipment of IEC 61131, Part 2.

CE Marking

Our products fulfill the requirements and safety objectives of the following EC Directive and comply with the harmonized European standards (EN) published for programmable logic controllers in the official journals of the European Communities:

- 89/336/EEC Electromagnetic Compatibility Directive (EMC Directive)

The EC Declarations of Conformity are available to the relevant authorities at the following address:

Siemens Aktiengesellschaft
Bereich Automatisierungstechnik
A&D AS E 48
Postfach 1963
D-92209 Amberg

EMC Directive

SIMATIC products are designed for use in industrial environments.

Area of Application	Requirements	
	Emitted Interference	Noise Immunity
Industry	EN 50081-2 : 1993	EN 50082-2 : 1995

UL Recognition

UL Recognition Mark
Underwriters Laboratories (UL) to
UL 508, File No. 116536

CSA Certification

CSA Certification Mark
Canadian Standard Association (CSA) to
Standard C22.2 No. 142, File No. LR 63533

FM Recognition

Factory Mutual Approval Standard Class Number 3611, Class I, Division 2, Groups
A, B, C, D.



Warning

Risk of injury and property damage.

Injury and property damage can occur if you pull an S7-300 plug-in connection in a hazardous environment with the system in operation.

Always de-energize the S7-300 before pulling a plug-in connection in a hazardous environment.



Warning

WARNING - DO NOT DISCONNECT WHILE CIRCUIT IS LIVE
UNLESS LOCATION IS KNOWN TO BE NONHAZARDOUS

Connecting Cables

B

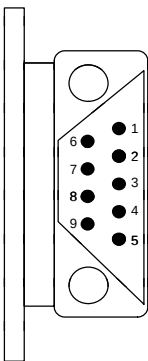
In Section	You Will Find	on Page
B.1	RS 232C Interface of the CP 341-RS 232C	B-2
B.2	20 mA TTY Interface of the CP 341-20mA TTY	B-9
B.3	X27 (RS 422/485) Interface of the CP 341-RS 422/485	B-16

B.1 RS 232C Interface of the CP 341-RS 232C

Pin Allocation

The table below shows the pin allocation for the 9-pin sub D male connector in the front panel of the CP 341-RS 232C (compatible with the 9-pin COM port (PC/programming device)).

Table B-1 Pin Allocation for the 9-Pin Sub D Male Connector of the Integrated Interface of the CP 341-RS 232C

Male Connector on CP 341-RS 232C*	Pin	Designation	Input/Output	Meaning
	1	DCD1 Received Detector	Input	Received signal level
	2	RXD Received Data	Input	Received data
	3	TXD Transmitted Data	Output	Transmitted data
	4	DTR Data Terminal Ready	Output	Data terminal is ready
	5	GND Ground	–	Signal ground (GND _{int})
	6	DSR Data Set Ready	Input	Data set ready
	7	RTS Request To Send	Output	Request to send
	8	CTS Clear To Send	Input	Clear to send
	9	RI Ring Indicator	Input	Ring indicator

* Front view

Connecting Cables

If you construct your own connecting cables you must remember that unconnected inputs at the communication partner may have to be connected to open-circuit potential.

Note that you must only use shielded connector casings. A large surface area of both sides of the cable shield must be in contact with the connector casing. You are advised to use Siemens V42 254 shielded connector casings.



Caution

Never connect the cable shield with the GND, as this could destroy the submodules.

GND must always be connected on both sides (pin 5), otherwise the submodules could again be destroyed.

In the Following

On the following pages you will find examples of connecting cables for a point-to-point connection between the CP 341-RS 232C and S7 modules or SIMATIC S5.

Connecting Cables, RS 232C (S7/M7 (CP 341) – S7/M7 (CP 340/ CP 341/CP 441))

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441.

For the connecting cables you require the following female connectors:

- at the CP 341: 9-pin sub D female with screw fixing
- at the communication partner: 9-pin sub D female with screw fixing

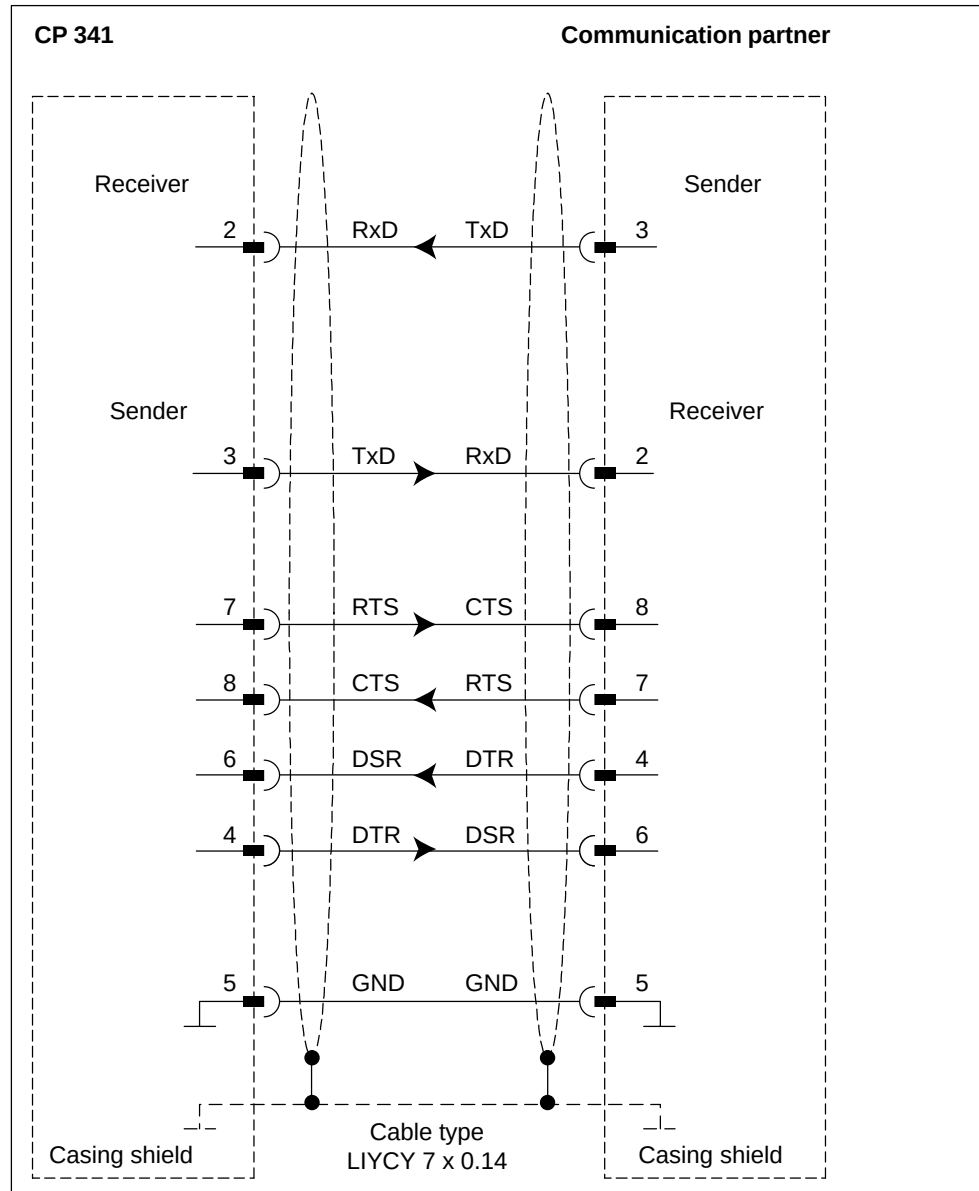


Figure B-1 RS 232C Connecting Cable, CP 341 – CP 340/CP 341/CP 441

The connecting cable (max. 15 m) is available using the order number (6ES7 902-1...) specified in Appendix D.

RS 232C Connecting Cables (S7/M7 (CP 341) – CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 544, CP 524, CPU 928B, CPU 945 or CPU 948.

For the connecting cables you require the following female/male connectors:

- at the CP 341: 9-pin sub D female with screw fixing
- at the communication partner: 25-pin sub D male with screw fixing

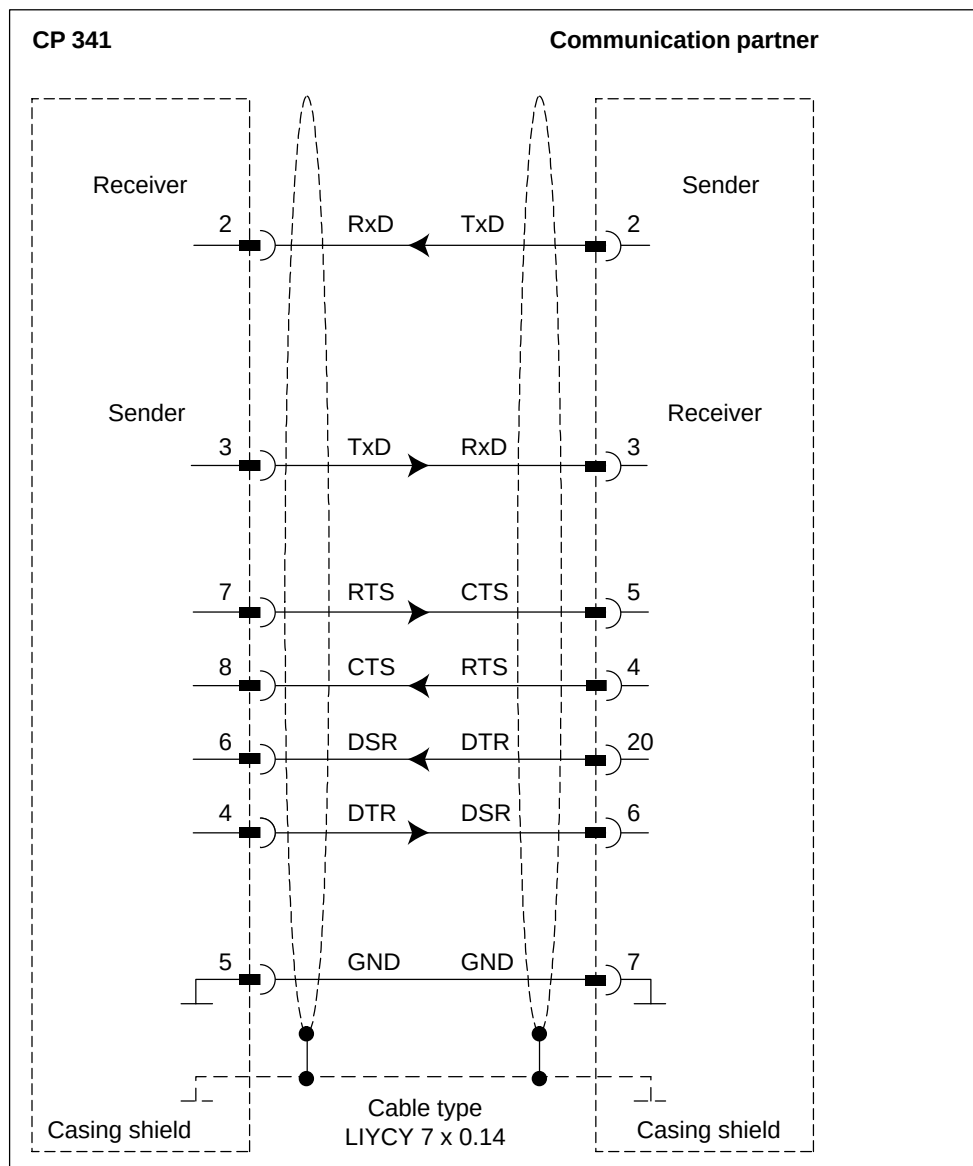


Figure B-2 RS 232C Connecting Cable CP 341 – CP 544, CP 524, CPU 928B, CPU 945, CPU 948

Connecting Cables, RS 232C (S7/M7 (CP 341) – CP 521 SI/CP 521 BASIC)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 521 SI/CP 521 BASIC.

For the connecting cables you require the following female/male connectors:

- at the CP 341: 9-pin sub D female with screw fixing
- at the communication partner: 25-pin sub D male with screw fixing

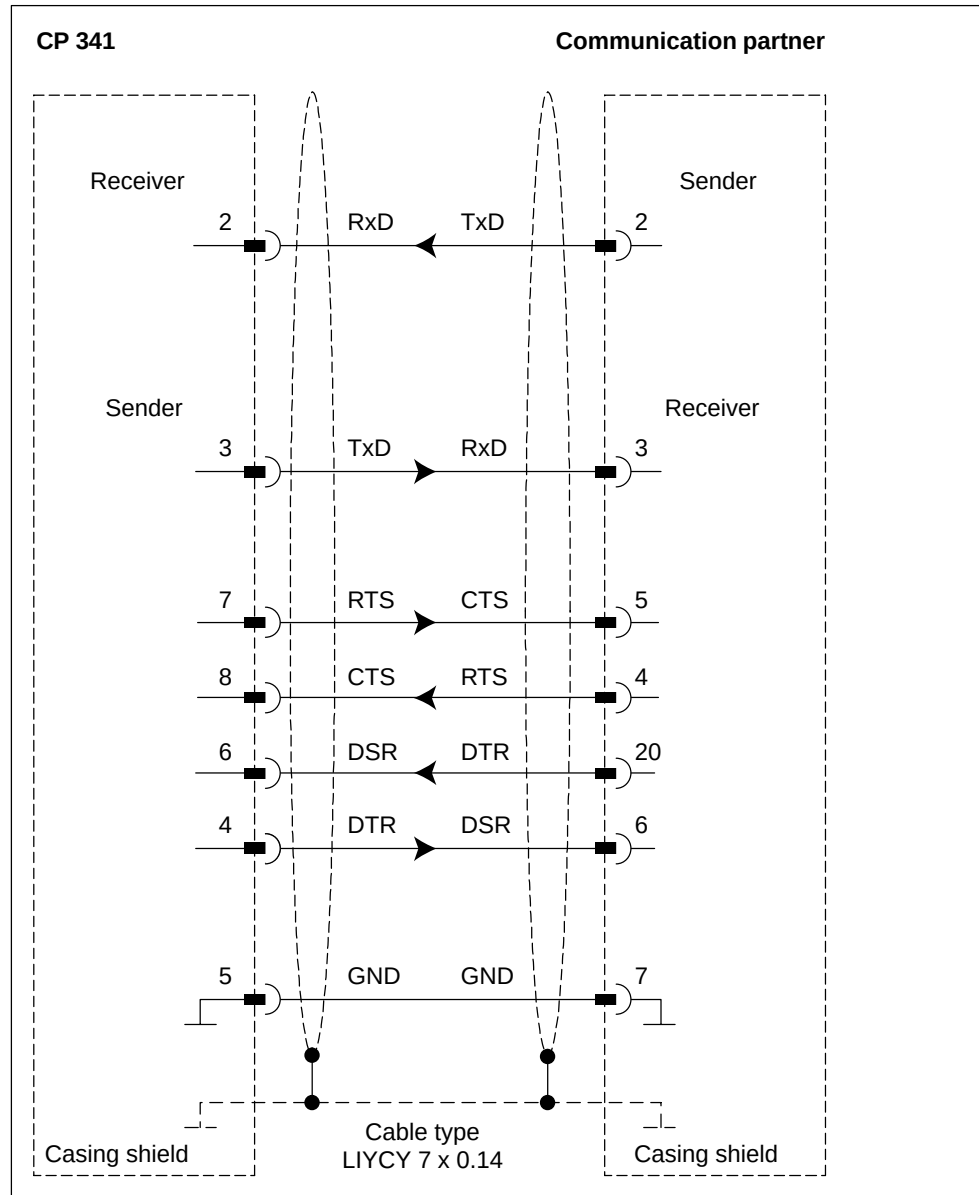


Figure B-3 RS 232C Connecting Cable CP 341 – CP 521 SI/CP 521 BASIC

RS 232C Connecting Cables (S7/M7 (CP 341) – CP 523)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 523.

For the connecting cables you require the following female/male connectors:

- at the CP 341: 9-pin sub D female with screw fixing
- at the communication partner: 25-pin sub D male with screw fixing

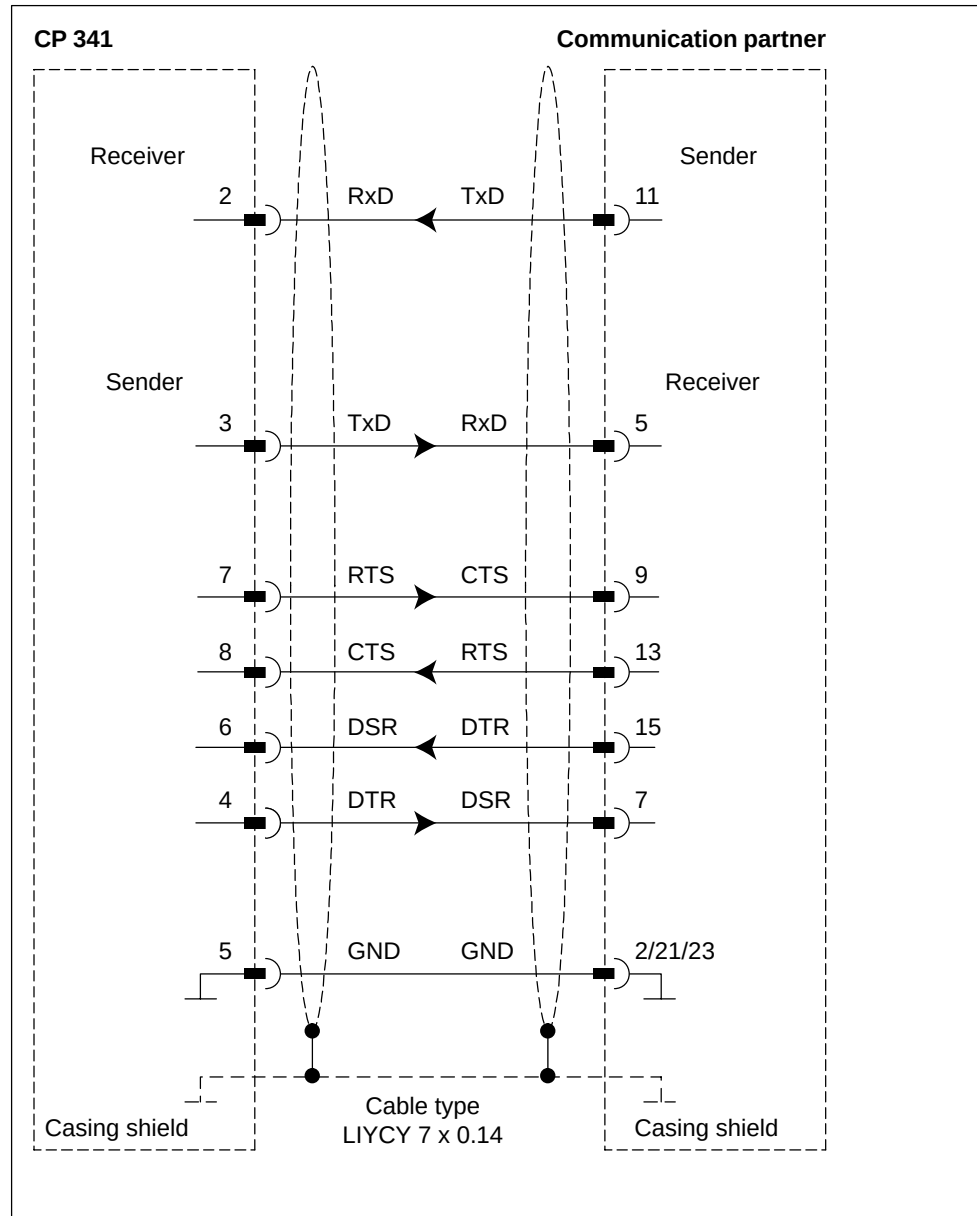


Figure B-4 RS 232C Connecting Cable CP 341 – CP 523

RS 232C Connecting Cable (S7/M7 (CP 341) – IBM Proprinter (PT 88), DR 230)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and an IBM Proprinter with a serial interface (PT 88 or IBM-compatible printer).

For the connecting cable you require the following female/male connectors:

- at the CP 341: 9-pin sub D female
- at the IBM Proprinter: 25-pin sub D male

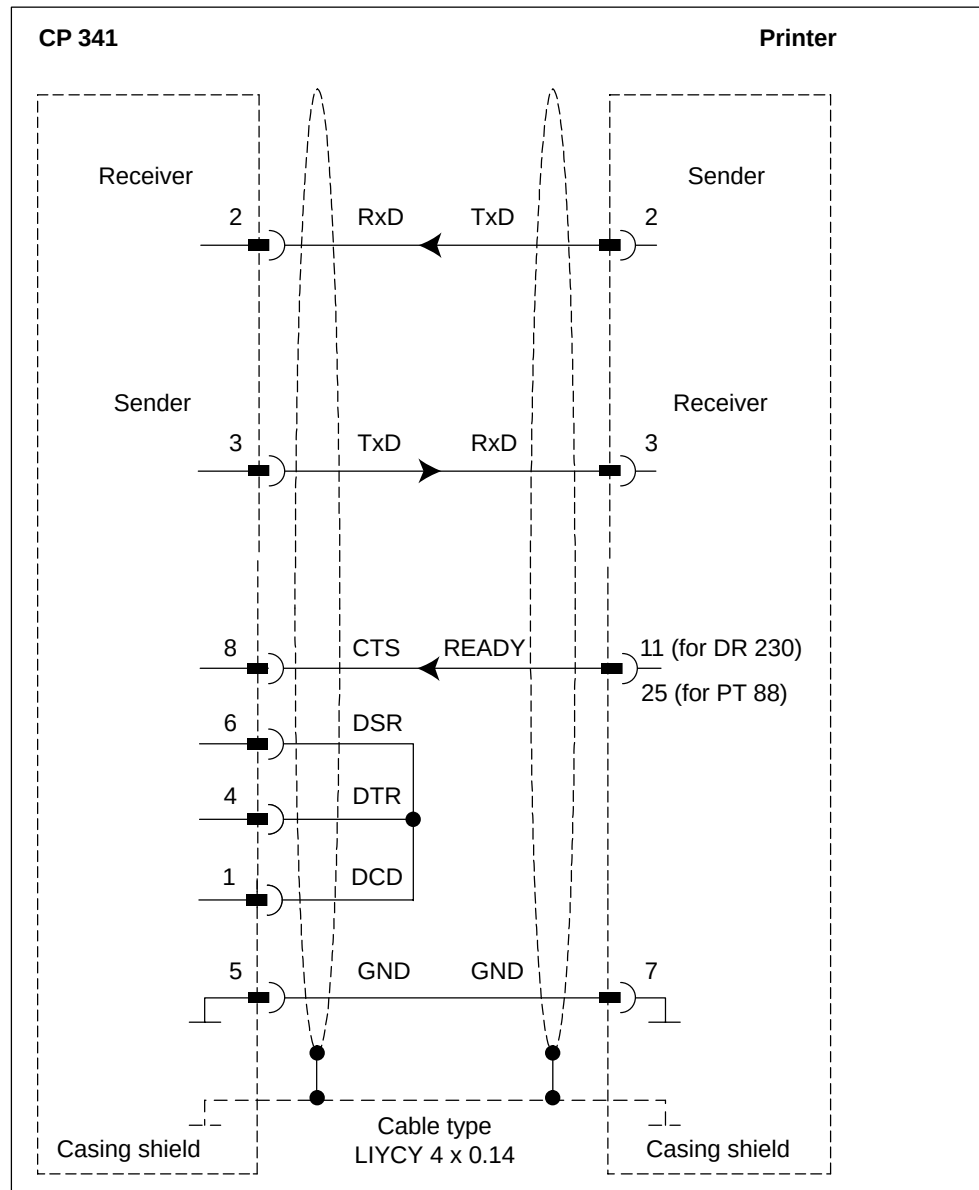


Figure B-5 RS 232C Connecting Cable CP 341 – IBM Proprinter

RS 232C Connecting Cable (S7/M7 (CP 341) – Laser Printer)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a laser printer with a serial interface (PT 10 or LaserJet Series II).

For the connecting cable you require the following female/male connectors:

- at the CP 341: 9-pin sub D female
- at the IBM Proprinter: 25-pin sub D male

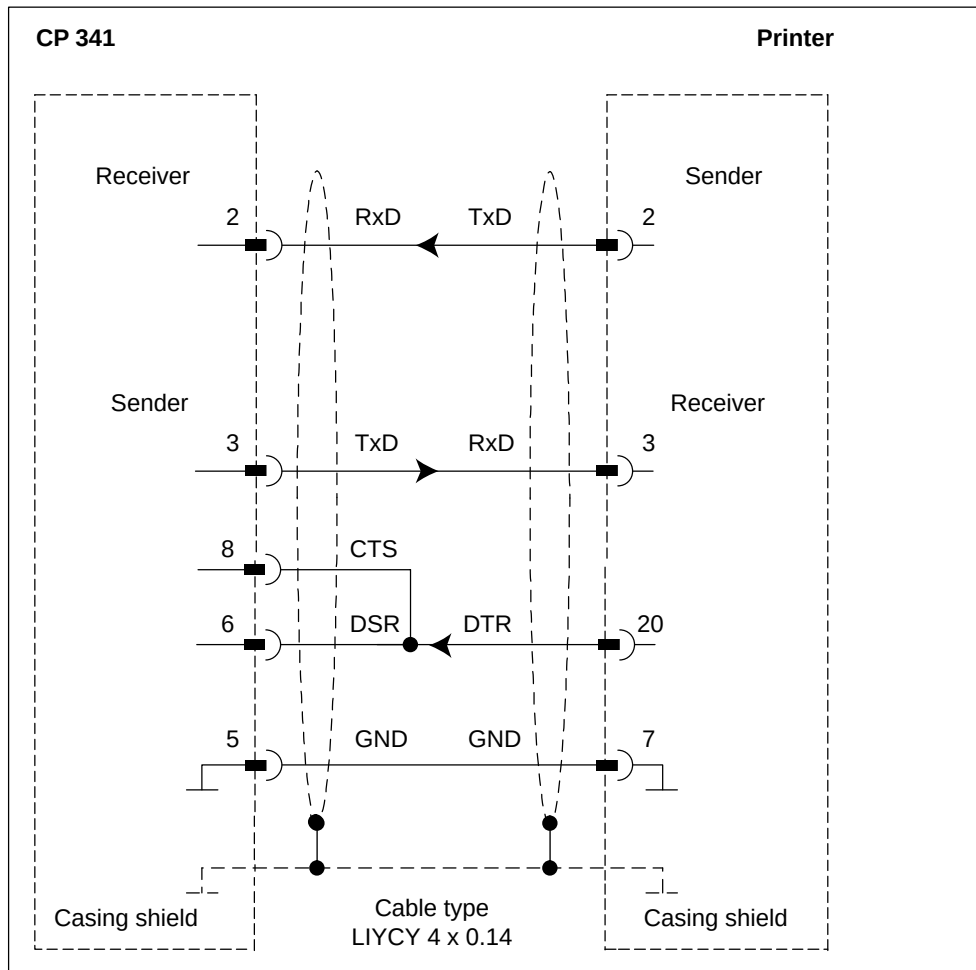


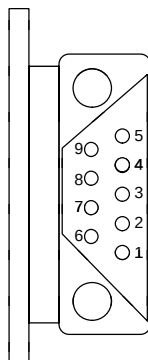
Figure B-6 RS 232C Connecting Cable CP 341 – Laser Printer

B.2 20 mA TTY Interface of the CP 341-20mA TTY

Pin Allocation

The table below shows the pin allocation for the 9-pin sub D female connector in the front panel of the CP 341-20mA TTY.

Table B-2 Pin Allocation for the 9-Pin Sub D Female Connector of the Integrated Interface of the CP 341-20mA TTY

Female Connector on CP 341-20mATTY*	Pin	Designation	Input/Output	Meaning
	1	TxD –	Output	Transmitted data
	2	20 mA –	Input	Ground 24 V
	3	20 mA + (I ₁)	Output	20 mA current generator 1
	4	20 mA + (I ₂)	Output	20 mA current generator 2
	5	RxD +	Input	Received data +
	6	–		
	7	–		
	8	RxD	Output	Received data –
	9	TxD +	Input	Transmitted data +

* Front view

Block Diagram

The figure below shows the block diagram of a 20 mA TTY interface.

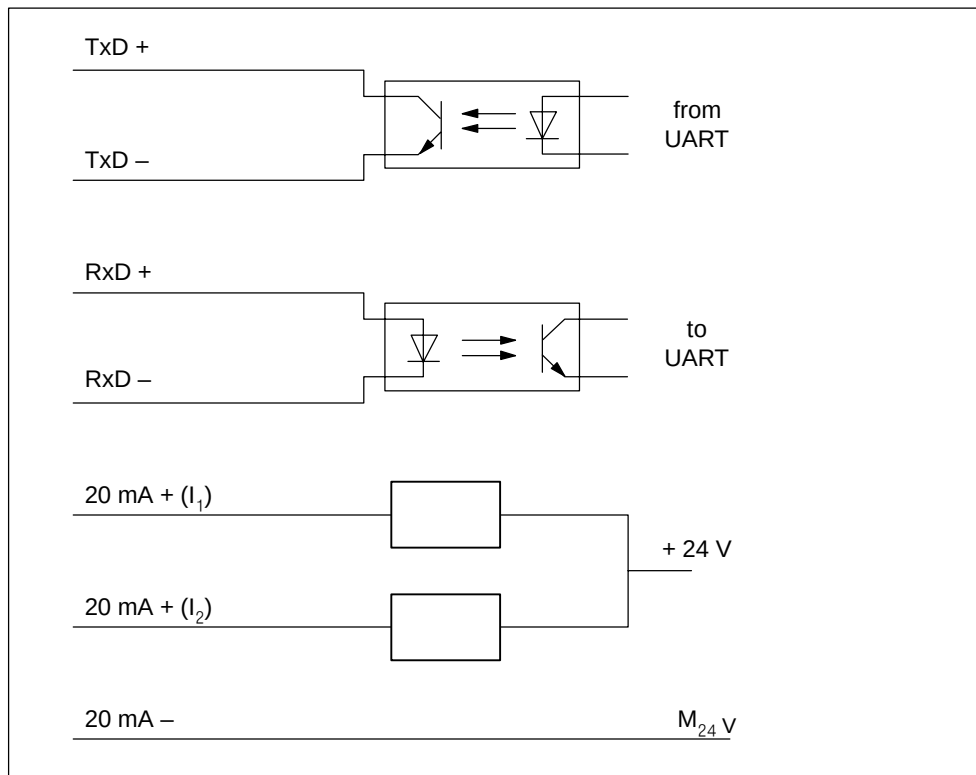


Figure B-7 Block Diagram of the 20 mA TTY Interface

Connecting Cables

If you construct your own connecting cables you must remember that unconnected inputs at the communication partner may have to be connected to open-circuit potential.

Note that you must only use shielded connector casings. A large surface area of both sides of the cable shield must be in contact with the connector casing and the shield contact. You are advised to use Siemens V42 254 shielded connector casings.



Caution

Never connect the cable shield with the GND, as this could destroy the interface submodules.

Note

This cable type (LIYCY 4 x 0.14) can be used in the following lengths for the CP 341 as communication partner:

- Max. 1000 m at 9600 baud
- Max. 500 m at 19.2 kbaud

Connecting Cable 20 mA TTY (S7/M7 (CP 341) – CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 544, CP 524, CPU 928B, CPU 945 or CPU 948.

For the connecting cables you require the following male connectors:

- at the CP 341: 9-pin sub D male with screw fixing
- at the communication partner: 25-pin sub D male with clip fixing

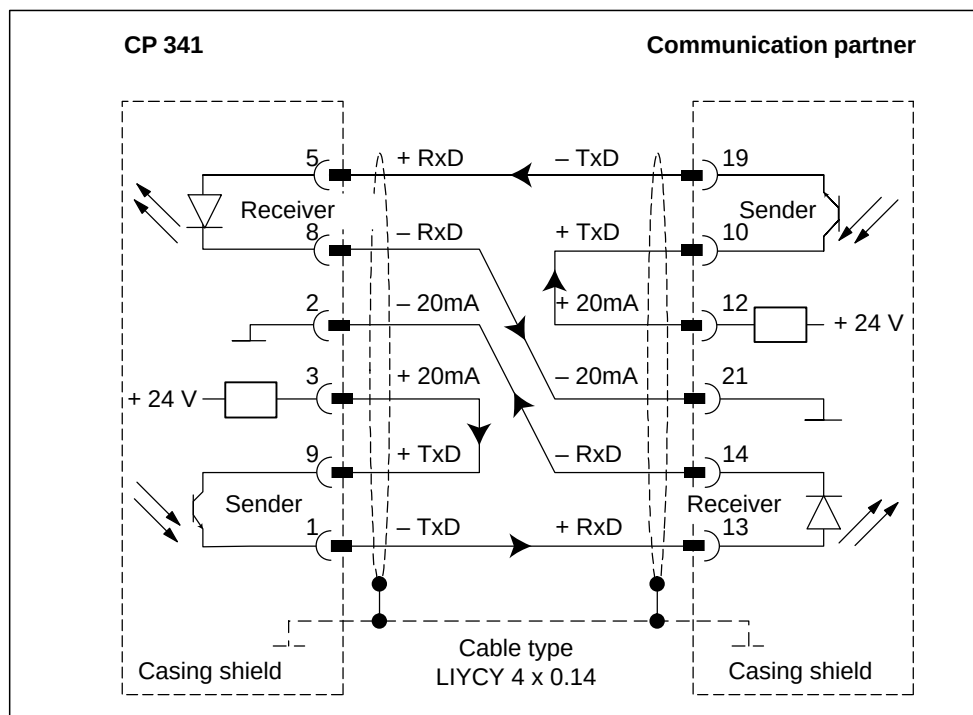


Figure B-9 20 mA TTY Connecting Cable CP 341 – CP 544, CP 524, CPU 928B, CPU 945, CPU 948

20 mA TTY Connecting Cables (S7/M7 (CP 341) – CP 523)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 523.

For the connecting cables you require the following male connectors:

- at the CP 341: 9-pin sub D male with screw fixing
- at the communication partner: 25-pin sub D male with screw fixing

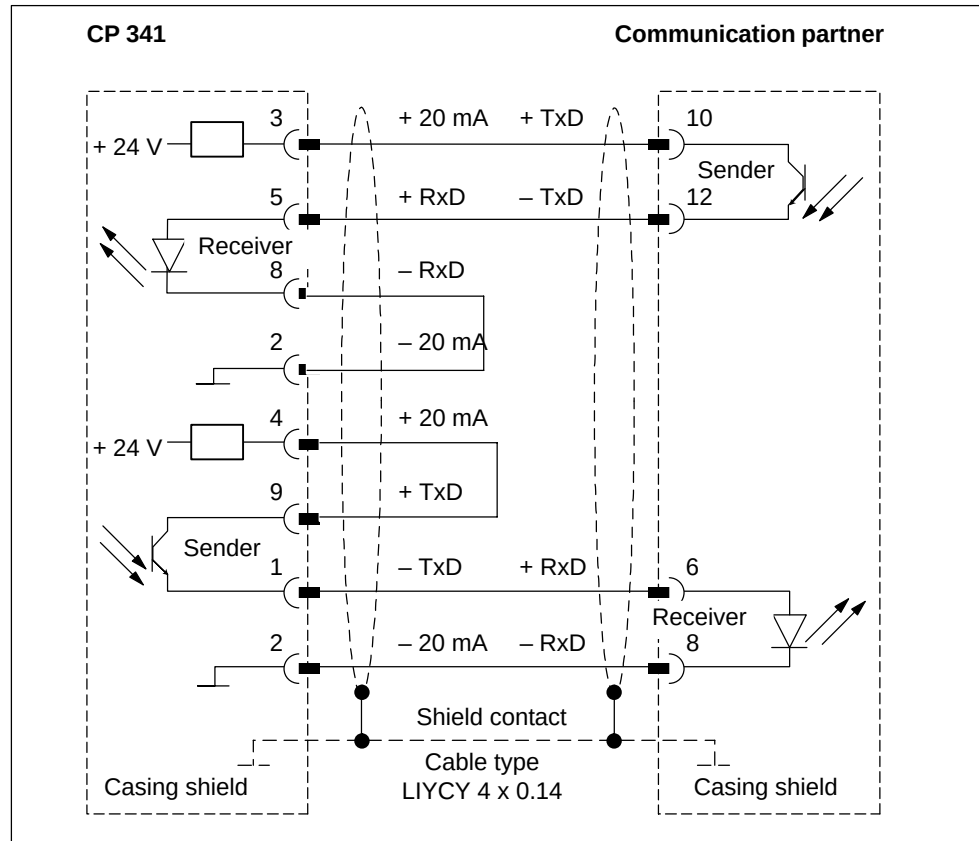


Figure B-10 20 mA TTY Connecting Cable CP 341 – CP 523

20 mA TTY Connecting Cable (S7/M7 (CP 341) – CP 521 SI/CP 521 BASIC/ IBM-Compatible Printer)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 521 SI/CP 521 BASIC.

For the connecting cables you require the following male connectors:

- at the CP 341: 9-pin sub D male with screw fixing
- at the communication partner: 25-pin sub D male with screw fixing

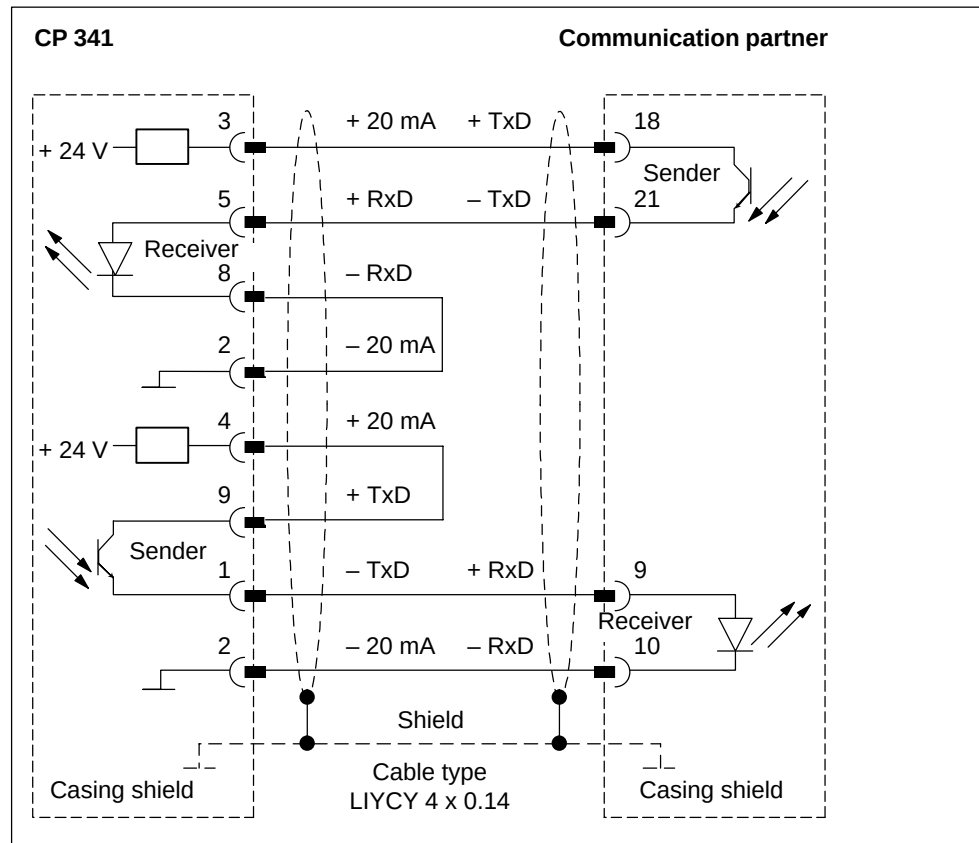


Figure B-11 20 mA TTY Connecting Cable CP 341 – CP 521SI/CP 521BASIC

20 mA TTY Connecting Cables (S7/M7 (CP 341) – CPU 944/AG 95)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CPU 944/AG 95.

For the connecting cables you require the following male connectors:

- at the CP 341: 9-pin sub D male with screw fixing
- at the communication partner: 15-pin sub D male with clip fixing

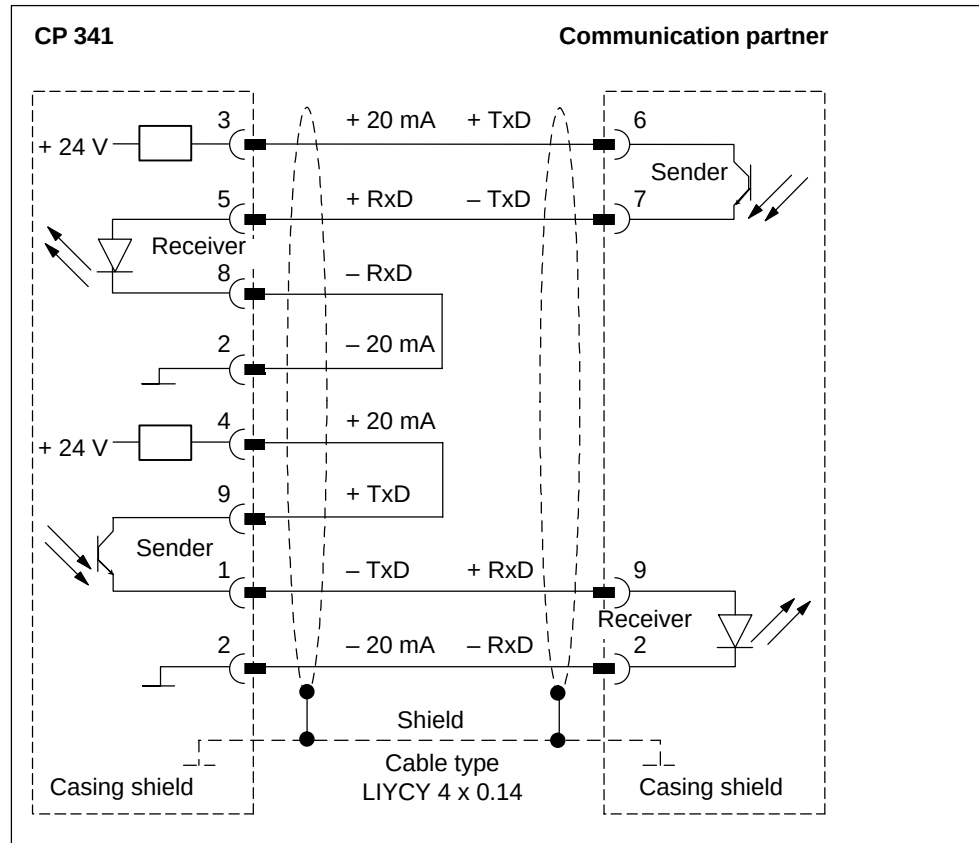


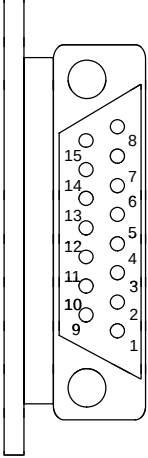
Figure B-12 20 mA TTY Connecting Cable CP 341 – CPU 944/AG 95

B.3 X27 (RS 422/485) Interface of the CP 341-RS 422/485

Pin Allocation

The table below shows the pin allocation for the 15-pin sub D females connector in the front panel of the CP 341-RS 422/485.

Table B-3 Pin Allocation for the 15-Pin Female Connector of the Integrated Interface of the CP 341-RS 422/485

Female Connector on CP341-RS422/485*	Pin	Designation	Input/Output	Meaning
	1	–	–	–
	2	T (A) –	Output	Transmitted data (four-wire operation)
	3	–	–	–
	4	R (A)/ T (A) –	Input Input/Output	Received data (four-wire operation) Received/transmitted data (two-wire operation)
	5	–	–	–
	6	–	–	–
	7	–	–	–
	8	GND	–	Functional ground (floating)
	9	T (B) +	Output	Transmitted data (four-wire operation)
	10	–	–	–
	11	R (B)/ T (B) +	Input Input/Output	Received data (four-wire operation) Received/transmitted data (two-wire operation)
	12	–	–	–
	13	–	–	–
	14	–	–	–
	15	–	–	–

* Front view

Connecting Cables

If you construct your own connecting cables you must remember that unconnected inputs at the communication partner may have to be connected to open-circuit potential.

Note that you must only use shielded connector casings. A large surface area of both sides of the cable shield must be in contact with the connector casing and the shield contact. You are advised to use Siemens V42 254 shielded connector casings.



Caution

Never connect the cable shield with the GND, as this could destroy the interface submodules.

GND (pin 8) must always be connected at both ends, otherwise the interface submodules could be destroyed.

In the Following

On the following pages you will find examples of connecting cables for a point-to-point connection between the CP 341-RS 422/485 and S7 modules or SIMATIC S5.

Connecting Cable X 27 (S7/M7 (CP 341) – CP 340/CP 341/CP 441)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441 for RS 422 operation.

For the connecting cables you require the following male connectors:

- at the CP 341: 15-pin sub D male with screw fixing
- at the communication partner: 15-pin sub D male with screw fixing

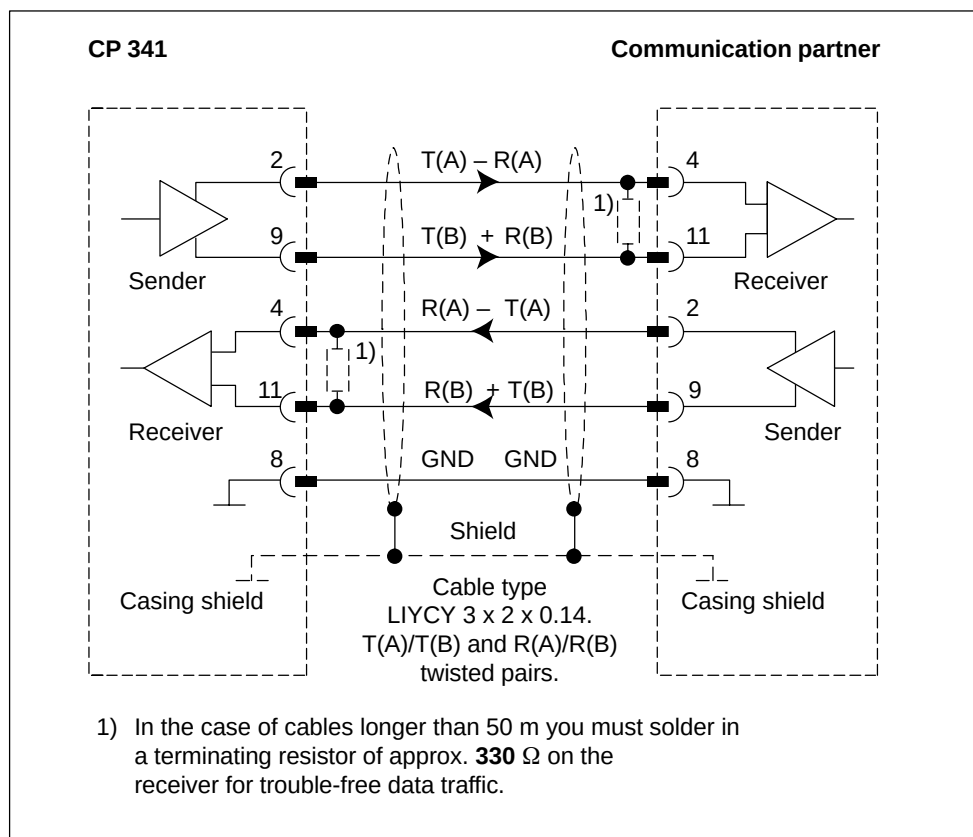


Figure B-13 X27 Connecting Cable CP 341 – CP 340/CP 341/CP 441 for RS 422 Operation (Four-Wire)

The connecting cable is available using the order number (6ES7 902-3...) specified in Appendix D.

Note

This cable type can be used in the following lengths for the CP 341 as communication partner:

- Max. 1200 m at 19 200 baud
- Max. 500 m at 38 400 baud
- Max. 250 m at 76 800 baud

Connecting Cable X 27 (S7/M7 (CP 341) – CP 340/CP 341/CP 441)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441 for RS 485 operation.

For the connecting cables you require the following male connectors:

- at the CP 341: 15-pin sub D male with screw fixing
- at the communication partner: 15-pin sub D male with screw fixing

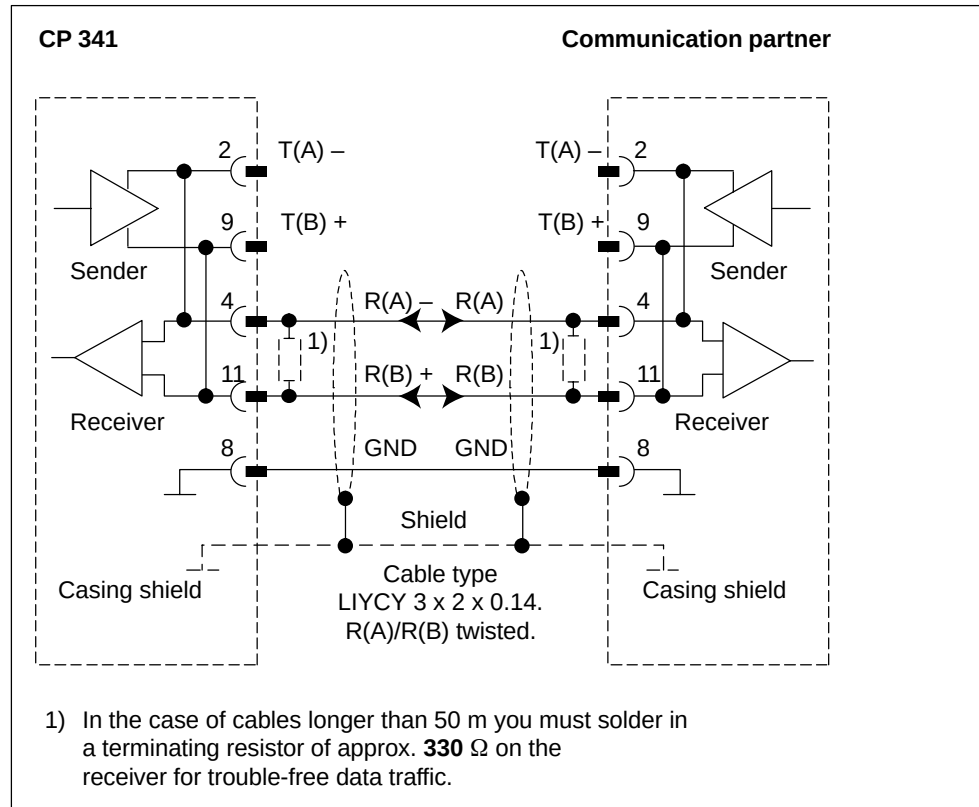


Figure B-14 X27 Connecting Cable CP 341 – CP 340/CP 341/CP 441 for RS 485 Operation (Two-Wire)

X 27 Connecting Cable (S7/M7 (CP 341) – CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 544, CP 524, CPU 928B, CPU 945, CPU 948 for RS 422 operation.

For the connecting cables you require the following male connectors:

- at the CP 341: 15-pin sub D male with screw fixing
- at the communication partner: 15-pin sub D male with clip fixing

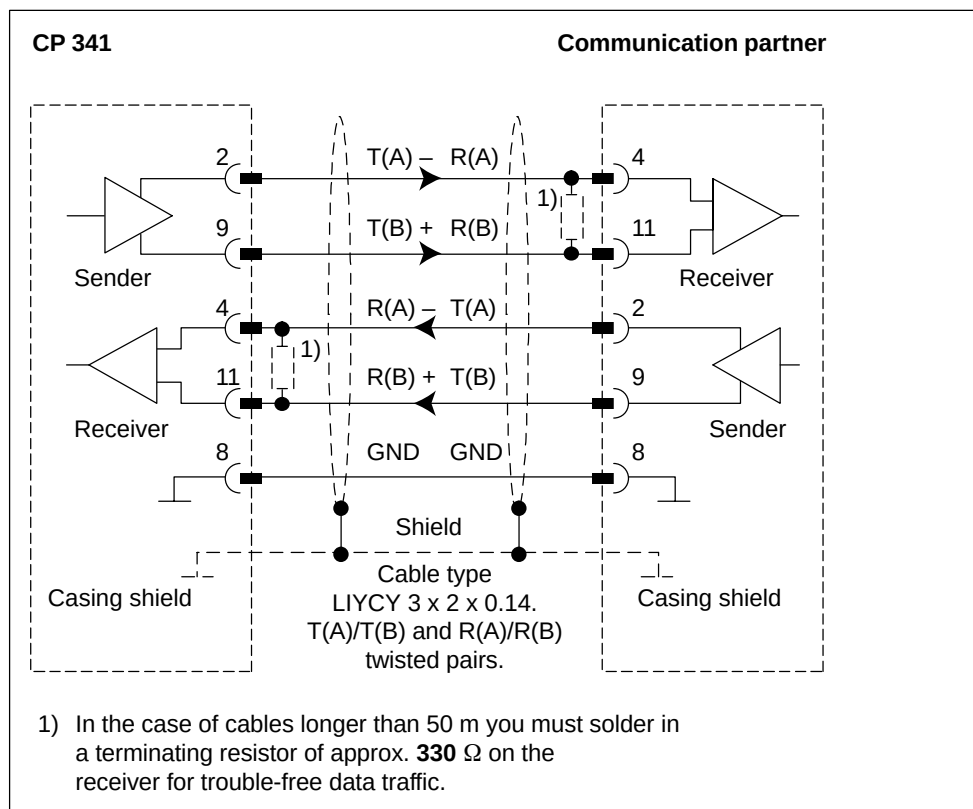


Figure B-15 X27 Connecting Cable CP 341 – CP 544, CP 524, CPU 928B, CPU 945, CPU 948 for RS 422 Operation (Four-Wire Mode)

Communication Matrix of the Protocols

C

The CP 341 communication processor can communicate with the following CPs and CPUs of the SIMATIC S5 and SIMATIC S7 programmable controllers.

Communication Matrix 3964(R)

The diagram below shows the communication matrix of the 3964 (R) procedure.

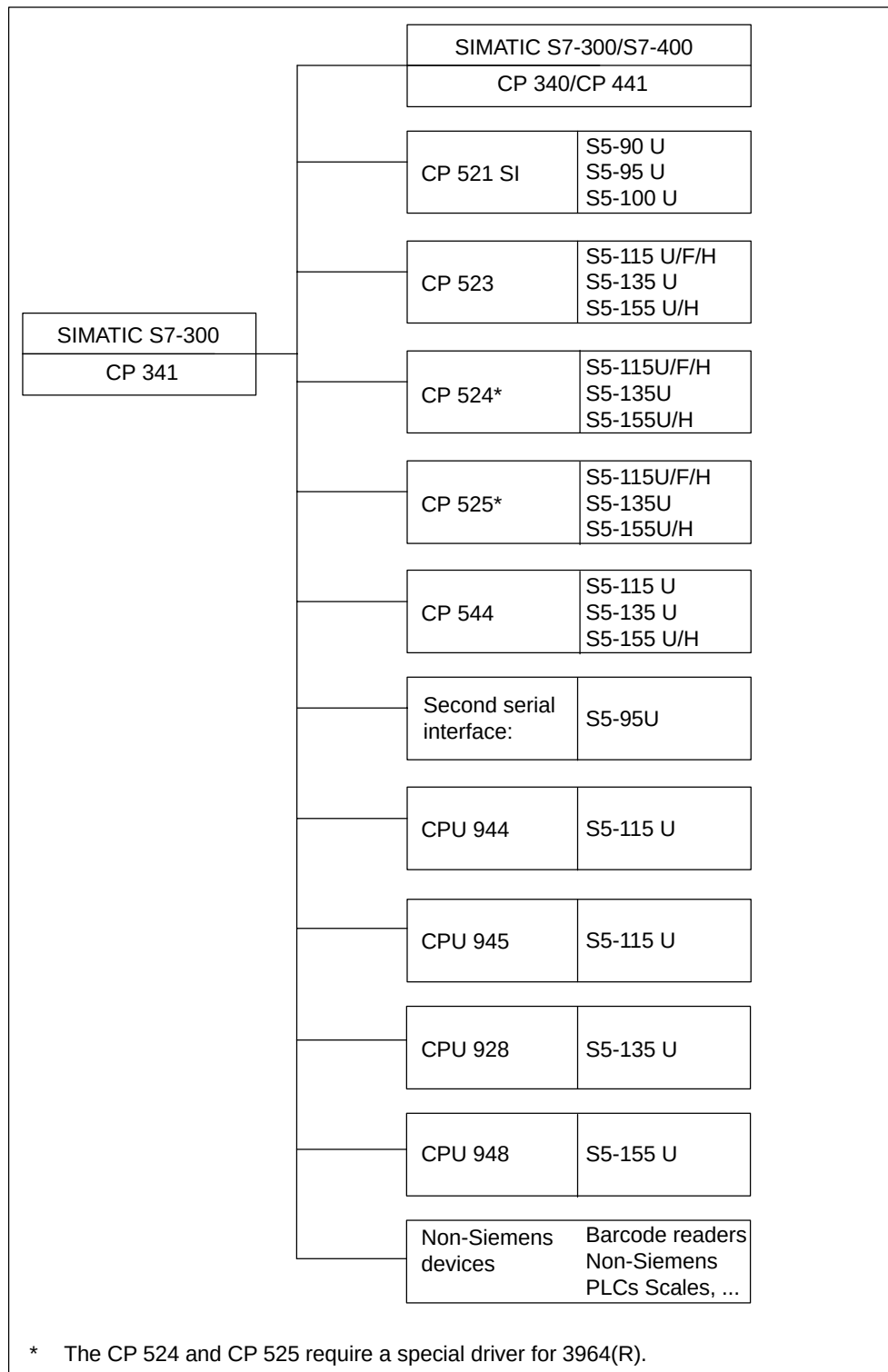


Figure C-1 Communication Matrix of the 3964(R) Procedure

Communication Matrix RK 512

The diagram below shows the communication matrix of the RK 512 computer connection.

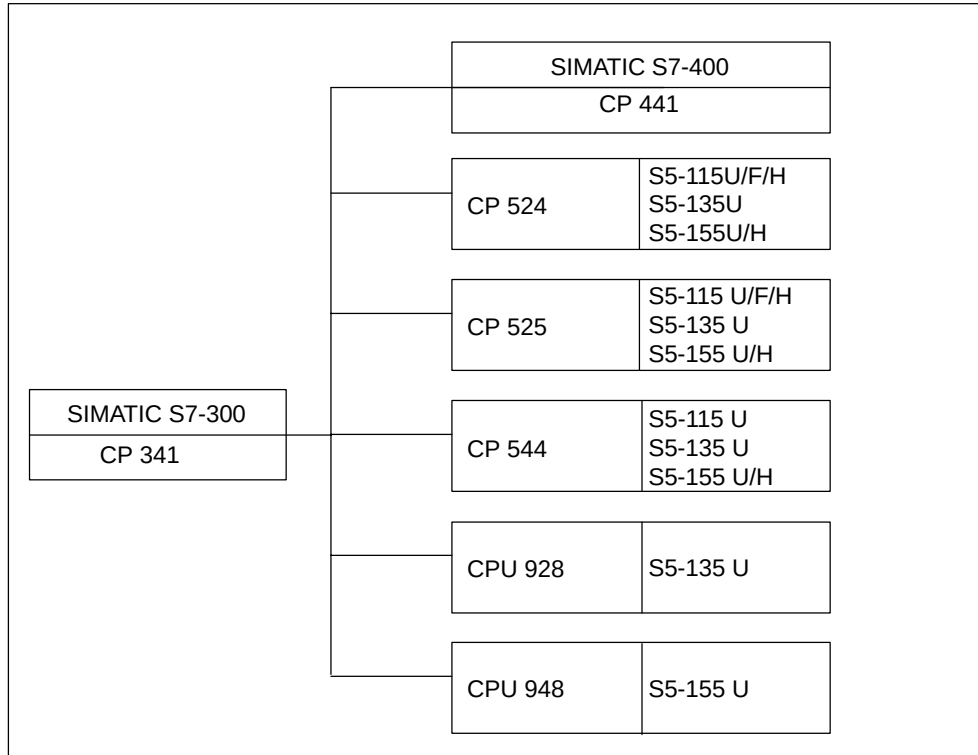


Figure C-2 Communication Matrix of the RK 512 Computer Connection

Communication Matrix ASCII Driver

The figure below shows the communication matrix of the ASCII driver.

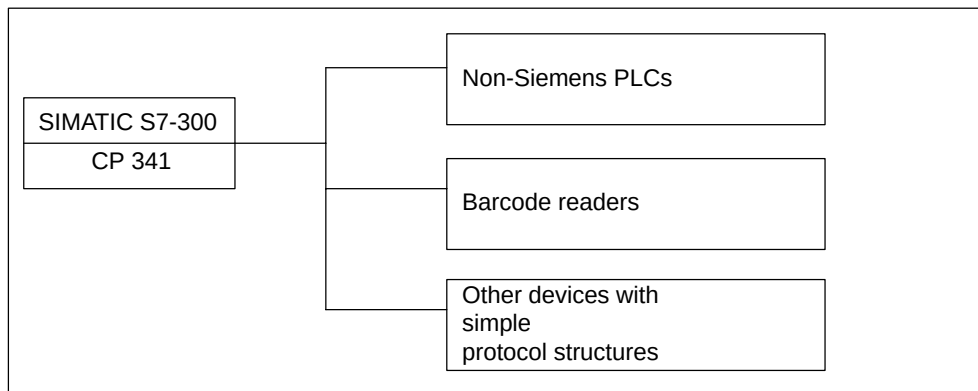


Figure C-3 Communication Matrix of the ASCII Driver

Accessories and Order Numbers

Module Variants

The table below contains the different variants of the CP 341.

Table D-1 Order Numbers of the Module Variants of the CP 341

Product	Order Number
CP 341-RS 232C	6ES7 341-1AH01-0AE0
CP 341-20mA TTY	6ES7 341-1BH01-0AE0
CP 341-RS 422/485	6ES7 341-1CH01-0AE0

Connecting Cables

Connecting cables are available in the following standard lengths 5 m, 10 m and 50 m.

Table D-2 Order Numbers of the Connecting Cables

Connecting cables for CP 341 – CP 340 CP 341 – CP 341 CP 341 – CP 441	Variant	Order Number
RS 232C interface (corresponding to Figure B-1)	RS 232C, 5 m	6ES7 902-1AB00-0AA0
	RS 232C, 10 m	6ES7 902-1AC00-0AA0
	RS 232C, 15 m	6ES7 902-1AD00-0AA0
20-mA TTY interface (corresponding to Figure B-8)	20 mA TTY, 5 m	6ES7 902-2AB00-0AA0
	20 mA TTY, 10 m	6ES7 902-2AC00-0AA0
	20 mA TTY, 50 m	6ES7 902-2AG00-0AA0
X27 (RS 422) interface (corresponding to Figure B-13)	X27 (RS 422), 5 m	6ES7 902-3AB00-0AA0
	X27 (RS 422), 10 m	6ES7 902-3AC00-0AA0
	X27 (RS 422), 50 m	6ES7 902-3AG00-0AA0

SIMATIC S7 Reference Literature



Literature Referenced in This Manual

- | | |
|------------|--|
| I1/ | <i>Programming with STEP 7 V5.1</i>
Manual |
| I2/ | <i>S7-400/M7-400 Programmable Controllers, Hardware and Installation</i> Manual |
| I3/ | <i>Configuring Hardware and Communication Connections</i>
<i>STEP 7 V5.1</i> Manual |
| I4/ | <i>System Software for S7-300 and S7-400, System and Standard Functions</i> Reference Manual |

Literature on SIMATIC S7

On the following pages, you will find a comprehensive overview of:

- manuals that you require for configuring and programming the S7-300,
- manuals which describe the components of a PROFIBUS DP network,
- technical overviews which provide you with an overview of the SIMATIC S7 and *STEP 7*.

Manuals for Configuring and Starting Up

An extensive user documentation is available to assist you in configuring and programming the S7-300. You can select and use this documentation as required. Table E-1 lists also documentation for *STEP 7*.

Table E-1 Manuals for Configuring and Programming the S7-300

Title	Contents
<i>Programming with STEP 7 V5.1 Manual</i>	The programming manual offers basic information on the design of the operating system and a user program of an S7 CPU. For novice users of an S7-300/400 it provides an overview of the programming principles on which the design of user programs is based.
<i>Configuring Hardware and Communication Connections STEP 7 V5.1 Manual</i>	The STEP 7 user manual explains the principles for using the STEP 7 automation software and its functions. Novice users of STEP 7 as well as experienced users of STEP 5 are provided with an overview of the configuring, programming and start-up procedures for an S7-300/400. When working with the software, an on-line help assists you if you require detailed information on the software.
<i>Statement List (STL) for S7-300 and S7-400 Reference Manual</i>	The manuals for the STL, LAD, FBD and SCL packages each comprise the user manual and the language description. For programming an S7-300/400 you need only one of the languages, but, if required, you can switch between the language to be used in a project. If it is the first time that you use one of the languages, the manuals will help you in getting familiar with the programming principles. When working with the software, you can use the on-line help, which provides you with detailed information on editors and compilers.
<i>Ladder Logic (LAD) for S7-300 and S7-400 Reference Manual</i>	
<i>Function Block Diagram (FBD) for S7-300 and S7-400 Reference Manual</i>	
<i>Structured Control Language (SCL)¹ for S7-300 and S7-400 Reference Manual</i>	
<i>S7-GRAPH¹ for S7-300 and S7-400 Programming Sequential Control Systems Manual</i>	With the GRAPH, HiGraph, CFC languages, you can implement sequential function charts, state diagrams or graphic interconnections of blocks. Each of the manuals comprises a user manual and a language description. If it is the first time that you use one of these languages, the manual will help you in getting familiar with the programming principles. When working with the software, you can also use the on-line help (not for HiGraph), which provides you with detailed information on editors and compilers.
<i>S7-HiGraph¹ for S7-300 and S7-400 Programming State Graphs Manual</i>	
<i>Continuous Function Charts¹ for S7 and M7 Manual</i>	
<i>System Software for S7-300 and S7-400 Systems and Standard Functions Reference Manual</i>	The S7 CPU's offer systems and standard functions which are integrated in the operating system. You can use these functions when writing programs in one of the languages, that is STL, LAD and SCL. The manual provides an overview of the functions available with S7 and, for reference purposes, detailed interface descriptions which you require in your user program.

¹ Optional packages for S7-300/400 system software

Manuals for PROFIBUS-DP

For the configuration and startup of a PROFIBUS-DP network, you will need the descriptions of the other nodes and network components integrated in the network. For this purpose, you can order the manuals listed in Table E-2.

Table E-2 Manuals for PROFIBUS-DP

Manual
<i>ET 200M Distributed I/O Station</i>
<i>SINEC L2-DP Interface of the S5-95U Programmable Controller</i>
<i>ET 200B Distributed I/O Station</i>
<i>ET 200C Distributed I/O Station</i>
<i>ET 200U Distributed I/O Station</i>
<i>ET 200 Handheld Unit</i>
<i>SINEC L2/L2FO-Network Components</i>

Glossary

A

Address

The address identifies a physical storage location. If the address is known, the operand stored there can be directly accessed.

B

Block

Blocks are elements of the user program which are defined by their function, structure, or purpose. With STEP 7 there are

- Code blocks (FB, FC, OB, SFB, SFC)
- Data blocks (DB, SDB)
- User-defined data types (UDT)

Block Call

A block call occurs when program processing branches to the called block.

Block Parameter

Block parameters are wildcards within multiple-use blocks, which are replaced with current values when the relevant block is called.

C

Communications Processor

Communications processors are modules for point-to-point connections and bus connections.

Configuration

The configuration is the setup of individual modules of the PLC in the configuration table.

CPU

Central processing unit of the S7 programmable controller with control and arithmetic unit, memory, operating system, and interfaces to I/O modules.

Cycle Time

The cycle time is the time the CPU needs to scan the user program once.

Cyclic Program Processing

In cyclic program processing, the user program is executed in a constantly repeating program loop, called a cycle.

D

Data Block (DB)

These are blocks containing data and parameters with which the user program works. Unlike all other blocks, data blocks do not contain instructions. They are subdivided into global data blocks and instance data blocks. The data held in the data blocks can be accessed absolutely or symbolically. Complex data can be stored in structured form.

Data Type

Data types allow users to define how the value of a variable or constant is to be used in the user program. They are subdivided into elementary and structured data types.

Default Setting

The default setting is a practical basic setting which is always used if no other value is specified.

Diagnostic Events

Diagnostic events are, for example, errors on a module or system errors in the CPU, which are caused by, say, a program error or by operating mode transitions.

Diagnostic Buffer

Every CPU has a diagnostic buffer, in which detailed information on diagnostic events is stored in the order in which they occur.

The CP 341 has its own diagnostic buffer, in which all the diagnostic events of the CP 341 are entered (hardware/firmware errors, initialization/parameterization errors, send and receive errors).

Diagnostics Functions

The diagnostics functions cover the entire system diagnosis and include detection, analysis and reporting of errors within the PLC.

Download

Downloading means loading load objects (e.g. code blocks) from the programming device into the load memory of the CPU.

F

Function Block (FB)

Function blocks are components of the user program and, in accordance with the IEC standard, are "blocks with memory". The memory for the function block is an assigned data block of the "instance data block". Function blocks can be parameterized but can also be used without parameters.

H

Hardware

Hardware is the term given to all the physical and technical equipment of a PLC.

I

Instance Data Block

An instance data block is a block assigned to a function block and contains data for this special function block.

Interrupt

An interrupt occurs when program processing in the processor of a PLC is interrupted by an external alarm.

M

Module

Modules are pluggable printed circuit boards for programmable controllers.

Module Parameter

Module parameters are used to set the module reactions. A distinction is made between static and dynamic module parameters.

O

On-line/Off-line

On-line means that a data circuit exists between PLC and programming device.
Off-line means that no such data circuit exists.

On-line Help

STEP 7 allows you to display contextual help texts on the screen while working with the programming software.

Operand

An operand is part of a STEP 7 instruction and states with what the processor is to do something. It can be both absolutely and symbolically addressed.

Operating Mode

The SIMATIC S7 programmable controllers have three different operating modes: STOP, RESTART and RUN. The functionality of the CPUs varies in the individual operating modes.

Operating System of the CPU

The operating system of the CPU organizes all functions and operations of the CPU which are not connected to a specific control task.

P

Parameter

Parameters are values that can be assigned. A distinction is made between block parameters and module parameters.

Parameterization

Parameterization means setting the behavior of a module.

Parameterization Interface *CP 341: Point-to-Point Communication, Parameter Assignment*

The *CP 341: Point-to-Point Communication, Parameter Assignment* parameterization interface is used to parameterize the submodules of the communications processor.

Point-to-Point Connection

In a point-to-point connection the communications processor forms the interface between a PLC and a communications partner.

Procedure

The execution of a data interchange operation according to a specific protocol is called a procedure.

Process Image

This is a special memory area in the PLC. At the beginning of the cyclic program, the signal states of the input modules are transferred to the process image input table. At the end of the cyclic program, the process image output table is transferred to the output modules as signal state.

Programmable Controller

Programmable controllers (PLCs) are electronic control devices consisting of at least one central processing unit, various input/output modules, and operator control and monitoring devices.

Protocol

The communications partners involved in a data interchange must abide by fixed rules for handling and implementing the data traffic. These rules are called protocols.

R

Rack

A rack is the rail containing slots for mounting modules.

RESTART

On transition from the STOP to the RUN mode, the PLC goes through the RESTART mode.

S

S7-300 Backplane Bus

The S7-300 backplane bus is a serial data bus via which the modules communicate with each other and are supplied with the necessary voltage.

Software

Software is the term given to all programs used on a computer system. These include the operating system and the user programs.

STEP 7

This is the programming software for SIMATIC S7 programmable controllers.

System Block

System blocks differ from the other blocks in that they are already integrated into the S7-300 system and are available for already defined system functions. They are subdivided into system data blocks, system functions, and system function blocks.

System Function (SFC)

System functions are modules without memory which are already integrated into the operating system of the CPU and can be called up by the user as required.

System Function Block (SFB)

System function blocks are modules with memory which are already integrated into the operating system of the CPU and can be called up by the user as required.

U

Upload

Uploading means loading load objects (e.g. code blocks) from the load memory of the CPU into the programming device.

The user program contains all instructions and declarations for signal processing, by means of which a system or a process can be controlled. The user program for SIMATIC S7 is structured and is divided into smaller units called blocks.

V

Variable

A variable is an operand (e.g. E 1.0) which can have a symbolic name and can therefore also be addressed symbolically.

W

Work Memory

The work memory is a RAM on the CPU which the processor accesses while processing the user program.

Index

Numbers

20 mA TTY interface, B-9
3964(R) Procedure, Sending Data, 2-13

A

Acknowledgment delay time (ADT), 2-51
Actual operand, 6-45
 absolutely addressed, 6-45
 symbolically addressed, 6-46
Addressing the module, 6-47
ASCII driver, 2-35
 data flow control, 2-48
 parameters, 2-55
 receive buffer, 2-44
 receiving data, 2-37
 RS 232C-secondary signals, 2-45
 sending data, 2-36

B

Baud rate, 2-52, 2-57
Block call
 P_RCV_RK, 6-11, 6-35
 P_SND_RK, 6-6, 6-18, 6-24, 6-29
 V24_SET, 6-41
 V24_STAT, 6-39
Block checksum, 2-12

C

Calling the SFCERR variable, 8-16
CE, marking, A-9
Character delay time, 2-56
Character delay time (CDT), 2-5, 2-51
Character frame, 2-4, 2-52, 2-57
 10-bit character frame, 2-4
Command message frame, 2-23

Communication matrix, C-1
 3964(R), C-1
 ASCII driver, C-3
 RK 512, C-3
Communication via function blocks, 6-2
Computer connection RK 512, 2-23
 command message frame, 2-23
 fetching data, 2-29
 message frame header, 2-24
 parameters, 2-54
 response message frame, 2-23, 2-25
 sending data, 2-26
Configuration, 5-2
Connecting cables, B-2
Continuation FETCH message frame, 2-31
Continuation message frame, 2-23
Continuation SEND message frame, 2-28
Control characters, 2-11
CP 341 slots, 4-2
CPU RUN, 7-4
CPU STOP, 7-4
CSA, A-10

D

Data bits, 2-52, 2-57
Data block assignment, 6-42
Data flow control, 2-48, 2-58
Diagnosis, diagnostic buffer, 8-19
Diagnostic, error numbers in the response
 message frame, 8-17
Diagnostics, 8-2
 display elements, 8-3
 messages at the STATUS output of the
 FBs, 8-4
Diagnostics functions, 8-2
Disabling alarms, 6-47
Dismounting the CP 341, 4-2
Display elements (LED), 8-2

Disposal, v

E

EMC Directive, A-9
EN/ENO mechanism, 6-46
End criterion, 2-39

- end-of-text character, 2-40
- expiry of character delay time, 2-39
- fixed message frame length, 2-42

End-of-text character, 2-56
Event class, 8-4
Event number, 8-4
Expansion bus for S7-300 backplane bus, 1-7

F

FETCH message frame, 2-23
Fetching data, RK 512, 2-29
Firmware update, 5-6
FM, recognition, A-10
Full-duplex operation, 2-3
Function blocks, 1-5, 6-2

- FB 7 P_RCV_RK, 6-12, 6-31, 6-36
- FB 8 P_SND_RK, 6-8, 6-19, 6-25

Functions, 6-2

- FC 5 V24_STAT, 6-40
- FC 6 V24_SET, 6-41

G

Group alarm LED, 8-3

H

Half-duplex operation, 2-3

I

IEC 1131, A-9
Indicator for end of receive message frame, 2-56
Indicators (LED), 1-7
Initial state of the receive line, 2-53, 2-60
Initialization, 7-2
Initialization conflict, 2-18
Installation guidelines, 4-4

Interface, 1-7

- 20 mA TTY, B-9
- 20-mA-TTY, 1-10
- RS 232C, B-2
- technical specifications, A-3
- uses of, 1-3
- X27 (RS 422/485), 1-11, B-16

Interface outputs of the CP 341,
 setting/resetting, 6-40
Interface status of the CP 341, checking, 6-38
Interface submodules, RS 232C, 1-8
Interprocessor communication flag, 2-24, 6-29,
 6-34
ISO 7-layer reference model, 2-7

L

LED indicators, 1-7
Loadable drivers, 5-5
Loading the configuration and parameters, 5-4

M

Managing the parameter data, 5-4
Memory requirements, 6-48
Message frame header, structure of the RK
 512 message frame header, 2-24
Message frame length when received, 2-56
Minimum number of CPU cycles, 6-49
Mounting the CP 341, 4-2

O

Operating mode transitions, 7-4
Operating modes, 7-2
Order numbers, D-1

P

Parameterization, 5-3, 6-43, 7-3

- direct, 6-43
- indirect, 6-43
- of data words, 6-44

Parameterization data, 2-49

- ASCII driver, 2-55
- procedure 3964(R), 2-49
- RK 512, 2-54

Parameterization interface, 1-5

Parameters

FB 7 P_RCV_RK, 6-12, 6-31, 6-36

FB 8 P_SND_RK, 6-8, 6-19, 6-25

FC 5 V24_STAT, 6-40

FC 6 V24_SET, 6-41

Parity, 2-52, 2-57

Points to note, when sending message frames,
7-4

Priority, 2-52

Procedure, 2-6

Procedure 3964(R), 2-11

block checksum, 2-12

control characters, 2-11

handling errored data, 2-17

initialization conflict, 2-18

parameter, 2-49

priority, 2-11

procedure errors, 2-19

receiving data, 2-15

Programming device, 1-5

Programming device cable, 1-5

Protocol, 2-6, 2-50

integrated in module, 1-2

Protocol parameters, 2-51, 2-56

R

Receive buffer, 2-44, 2-59

Receiving data

ASCII driver, 2-37

with 3964(R), 2-15

Recycling, v

Reparameterization, 7-2

Response message frame, 2-23, 2-25

Error Numbers, 8-2

error numbers, 8-17

structure and contents, 2-25

RS 232C interface, B-2

RS 232C secondary signals, 2-45, 6-38

automatic use, 2-46

controlling, 6-40

reading, 6-38

RUN, 7-2

Run times, 6-48

S

Scope of This Manual, iii

SEND message frame, 2-23

Sending data

3964(R) procedure, 2-13

ASCII driver, 2-36

RK 512, 2-26

Setup attempts, 2-51

Standard connecting cables, 1-4, 1-12

Start bit, 2-52, 2-57

Start-up characteristics, 6-47, 7-2

STATUS output of the FBs, 8-2, 8-4

STOP, 7-2

Stop bits, 2-52, 2-57

System functions used, 6-49

T

Transmission attempts, 2-51

Transmission integrity, 2-8

with 3964, 2-9

with RK 512, 2-10

with the ASCII driver, 2-9

U

UL, A-9

X

X27 (RS 422/485) interface, B-16

Siemens AG
A&D AS E 81

Oestliche Rheinbrueckenstr. 50
D-76181 Karlsruhe
Federal Republic of Germany

From:

Your Name: _ _ _ _ _

Your Title: _ _ _ _ _

Company Name: _ _ _ _ _

Street: _ _ _ _ _

City, Zip Code _ _ _ _ _

Country: _ _ _ _ _

Phone: _ _ _ _ _

Please check any industry that applies to you:

☐ Automotive

☐ Chemical

☐ Electrical Machinery

☐ Food

☐ Instrument and Control

☐ Nonelectrical Machinery

☐ Petrochemical

☐ Pharmaceutical

☐ Plastic

☐ Pulp and Paper

☐ Textiles

☐ Transportation

☐ Other _ _ _ _ _



Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

- | | | |
|----|--|--------------------------|
| 1. | Do the contents meet your requirements? | ☐ |
| 2. | Is the information you need easy to find? | ☐ |
| 3. | Is the text easy to understand? | ☐ |
| 4. | Does the level of technical detail meet your requirements? | ☐ |
| 5. | Please rate the quality of the graphics/tables: | ☐ |

Additional comments:
