

SIMATIC

CP 340 Point-to-Point Communication Installation and Parameter Assignment

Manual

This manual has the order number:

6ES7340-1AH00-8BA0

Edition 06/2002

A5E00053578-05

Preface, Contents	
Product Description	1
Basic Principles of Serial Data Transmission	2
Starting Up the CP 340	3
Mounting the CP 340	4
Configuring and Parameterizing the CP 340	5
Communication via Function Blocks	6
Start-up Characteristics and Operating Mode Transitions of the CP 340	7
Diagnostics with the CP 340	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	

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:



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

The device/system may only be set up and operated in conjunction with this manual.

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

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Copyright © Siemens AG 1997-2002 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 Automation and Drives
Geschäftsgebiet Industrial Automation Systems
Postfach 4848, D-90327 Nuernberg

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.

Technical data subject to change.
© Siemens AG 1997-2002

Preface

Purpose

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

Contents

The manual, which is divided into a main section and a reference section (appendices), describes the hardware and software of the CP 340 communications processor and how it is integrated into the S7-300 programmable controller.

The following subjects are covered:

- The basics of point-to-point connections with the CP 340
- Starting up the CP 340
- Mounting the CP 340
- Communication via the CP 340
- Debugging
- Application example
- Attributes and technical specifications

Scope of This Manual

The manual is relevant for:

Product	Order Number	From Edition
CP 340-RS 232C	6ES7 340-1AH01-0AE0	01
CP 340-20mA TTY	6ES7 340-1BH00-0AE0	01
CP 340-RS 422/485	6ES7 340-1CH00-0AE0	01

Note

The description of the CP 340 communications processor contained in this manual is correct at the date of publication. We reserve the right to describe changes to module functionality in a product information sheet.

Changes Since the Previous Edition

Since edition 01 of this manual, *CP 340 Point-to-Point Communication, Installation and Parameter Assignment*, descriptions of the following have been added:

- The CP 340-RS 422/485 and CP 340-20mA TTY module variants
- The output of message texts to a printer
- Diagnostics by means of the diagnostic buffer of the CP 340

Further Documentation

Appendix E contains a list of further publications on the S7-300 and other programmable controllers.

Electronic Manuals

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

Assistance in Using This Manual

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

- A comprehensive list of contents followed by lists of all figures and tables which appear in the manual.
- 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.

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.

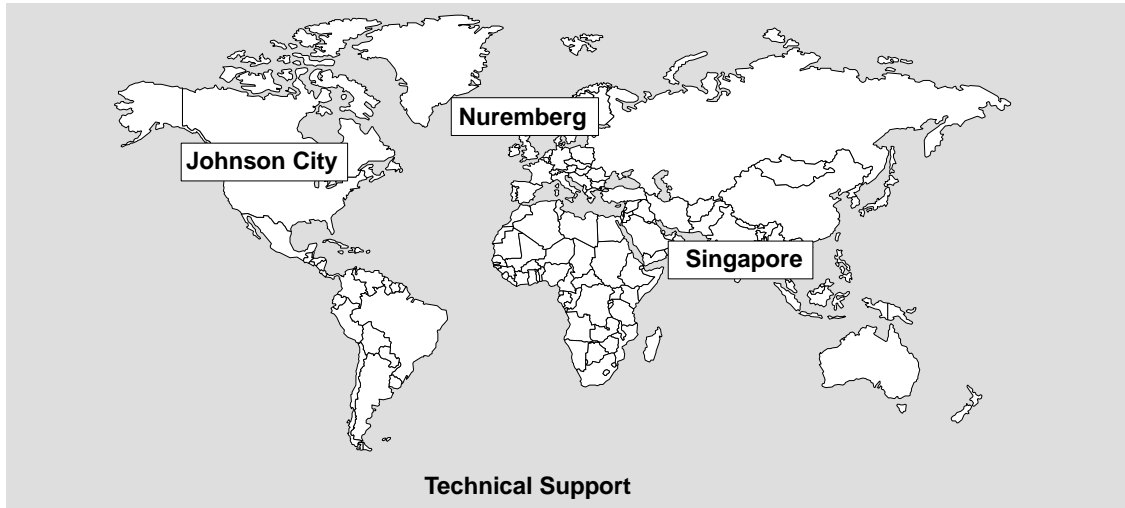
We offer a range of courses to help get you started with the SIMATIC S7 programmable controller. Please contact your local training center or the central training center in Nuremberg, D-90027 Germany, Tel. +49 911 895 3200.

Conventions

The abbreviation CP 340 is used in the documentation when information applies to all three module variants: CP 340-RS 232C, CP 340-20mA TTY and CP 340-RS 422/485.

A&D Technical Support

Available worldwide, around the clock:



<p>Worldwide (Nuremberg) Technical Support</p> <p>Local time: 0:00 to 24:00 / 365 days Phone: +49 (0) 180 5050-222 Fax: +49 (0) 180 5050-223 E-Mail: adsupport@siemens.com GMT: +1:00</p>		
<p>Europe / Africa (Nuremberg) Authorization</p> <p>Local time: Mon.-Fri. 7:00 to 17:00 Phone: +49 (0) 180 5050-222 Fax: +49 (0) 180 5050-223 E-Mail: adsupport@siemens.com GMT: +1:00</p>	<p>America (Johnson City) Technical Support and Authorization</p> <p>Local time: Mon.-Fri. 8:00 to 17:00 Phone: +1 (0) 770 740 3505 Fax: +1 (0) 770 740 3699 E-Mail: isd-callcenter@sea.siemens.com GMT: -5:00</p>	<p>Asia / Australia (Singapore) Technical Support and Authorization</p> <p>Local time: Mon.-Fri. 8:30 to 17:30 Phone: +65 (0) 740-7000 Fax: +65 (0) 740-7001 E-Mail: simatic.hotline@sea.siemens.com.sg GMT: +8:00</p>
<p>The languages of the SIMATIC Hotlines and the authorization hotline are generally German and English.</p>		

Service & Support on the Internet

In addition to our documentation, we offer our Know-how online on the internet at:

<http://www.siemens.com/automation/service&support>

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.
- The right documents via our Search function in Service & Support.
- A forum, where users and experts from all over the world exchange their experiences.
- Your local representative for Automation & Drives via our representatives database.
- Information on field service, repairs, spare parts and more under “Services”.

Contents

1	Product Description	1-1
1.1	Uses of the CP 340	1-2
1.2	Components Required for a Point-to-Point Connection with the CP 340	1-5
1.3	Design of the CP 340	1-7
1.4	Attributes of the Serial Interface	1-9
1.4.1	RS 232C Interface of the CP 340-RS 232C	1-9
1.4.2	20 mA TTY Interface of the CP 340-20mA TTY	1-10
1.4.3	X27 (RS 422/485) Interface of the CP 340-RS 422/485	1-11
1.5	Cables for Connecting the CP 340 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	2-7
2.2.1	ISO 7-Layer Reference Model for Data Transmission	2-7
2.2.2	Data Transmission with the 3964(R) Procedure	2-12
2.2.3	Data Transmission with the ASCII Driver	2-21
2.2.4	Data Transmission with the Printer Driver	2-32
2.3	Parameterization Data	2-36
2.3.1	Basic Parameter of the CP 340	2-36
2.3.2	Parameterization Data of the 3964(R) Procedure	2-37
2.3.3	Parameterization Data of the ASCII Driver	2-43
2.3.4	Parameterization Data of the Printer Driver	2-50
2.3.5	Conversion and Control Statements for Printer Output	2-56
3	Starting Up the CP 340	3-1
4	Mounting the CP 340	4-1
4.1	CP 340 Slots	4-2
4.2	Mounting and Dismounting the CP 340	4-3
5	Configuring and Parameterizing the CP 340	5-1
5.1	Configuring the CP 340	5-2
5.2	Parameterizing the Communications Protocols	5-3
5.3	Managing the Parameter Data	5-4

6	Communication via Function Blocks	6-1
6.1	Communication via Function Blocks	6-2
6.2	Overview of the Function Blocks	6-3
6.3	Using the Function Blocks	6-4
6.3.1	Connection to a Communication Partner	6-4
6.3.2	Outputting Message Texts to a Printer	6-12
6.3.3	Reading and Controlling the RS 232C Secondary Signals	6-17
6.4	Parameterizing the Function Blocks	6-21
6.4.1	General Information on Data Block Assignment	6-21
6.4.2	Parameterizing the Data Blocks	6-22
6.5	General Information on Program Processing	6-25
6.6	Technical Specifications of the Function Blocks	6-26
7	Start-up Characteristics and Operating Mode Transitions of the CP 340	7-1
7.1	Operating Modes of the CP 340	7-2
7.2	Start-up Characteristics of the CP 340	7-3
7.3	Behavior of the CP 340 on Operating Mode Transitions of the CPU	7-4
8	Diagnostics with the CP 340	8-1
8.1	Diagnostics Functions of the CP 340	8-2
8.2	Diagnosis via the Display Elements of the CP 340	8-3
8.3	Diagnostics Messages of the Function Blocks P_SEND, P_RCV and P_PRINT	8-4
8.4	Diagnosis via the S7-300 Backplane Bus	8-12
8.5	Diagnosis by Means of the Diagnostic Buffer of the CP 340	8-15
9	Programming Example for Standard Function Blocks	9-1
9.1	General Information	9-2
9.2	Device Configuration	9-3
9.3	Settings	9-4
9.4	Blocks Used	9-5
9.5	Example "Point-to-Point Connection"	9-6
9.6	Example "Printing" and "Reading and Controlling the CP 340 Inputs/Outputs"	9-8
9.7	Installation, Error Messages	9-10
9.8	Activation, Start-Up Program and Cyclic Program	9-11

A	Technical Specifications	A-1
A.1	Technical Specifications of the CP 340	A-2
A.2	Certification and Application Areas	A-7
A.3	Recycling and Disposal	A-9
B	Connecting Cables	B-1
B.1	RS 232C Interface of the CP 340-RS 232C	B-2
B.2	20 mA TTY Interface of the CP 340-20mA TTY	B-10
B.3	X27 (RS 422/485) Interface of the CP 340-RS 422/485	B-17
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

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
1.1	Uses of the CP 340	1-2
1.2	Components Required for a Point-to-Point Connection with the CP 340	1-5
1.3	Design of the CP 340	1-7
1.4	Attributes of the Serial Interface	1-9
1.5	Cables for Connecting the CP 340 to a Communication Partner	1-12

1.1 Uses of the CP 340

Introduction The CP 340 communications processor allows you to exchange data between programmable controllers or computers by means of a point-to-point connection.

Functionality of the CP 340 The CP 340 communications processor provides the following functionality:

- Integrated serial interface according to RS 232C
- Transmission rate up to 19.2 Kbaud, half duplex
- Integration of the most important transmission protocols in the module firmware
 - 3964(R) procedure
 - ASCII driver
 - Printer driver
- Adaptation of the transmission protocols by parameterization using the *CP 340: Point-to-Point Communication, Parameter Assignment* interface.
- Integrated serial interface:

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

Module Variants The following variants of the CP 340 communications processor are available:

Table 1-1 CP 340 Module Variants

Module	Order Number	Integrated Interface
CP 340-RS 232C	6ES7 340-1AH01-0AE0	RS 232C interface
CP 340-20mA TTY	6ES7 340-1BH00-0AE0	20 mA TTY interface
CP 340-RS 422/485	6ES7 340-1CH00-0AE0	X27 (RS 422/485) interface

Functions of the Module Variants

The functionality of the drivers depends on the module variant of the CP 340:

Table 1-2 Functions of the CP 340 Module Variants

Function	CP 340- RS 232C	CP 340- 20mA TTY	CP 340-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
Printer driver	Yes	Yes	Yes	Yes
RTS/CTS flow control	Yes	No	No	No
XON/XOFF flow control	Yes	Yes	Yes	No

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

Uses of the CP 340

The CP 340 communications processor allows point-to-point connections with various Siemens modules and non-Siemens products, including:

- SIMATIC S5 via the 3964(R) driver 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.
- SIMOVERT and SIMOREG (USS protocol) via the ASCII driver (CP 340-RS 422/485) with appropriate protocol adaptation using a STEP 7 program.
- PCs via the 3964(R) procedure (the following development tools exist for programming on PCs: 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) or ASCII driver.
- Other devices with simple protocol structures by means of appropriate protocol adaptation with the ASCII driver.
- Other devices with 3964(R) drivers.
- Printers (HP Deskjet, HP Laserjet, Postscript, Epson, IBM).

Appendix C contains a summary of SIMATIC modules.

The CP 340 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 340

Introduction To establish a point-to-point connection between the CP 340 communications 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 340.

Table 1-3 Hardware Components for a Point-to-Point Connection with the CP 340

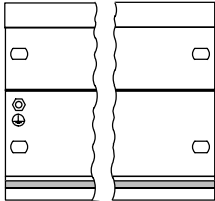
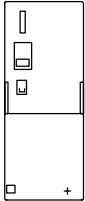
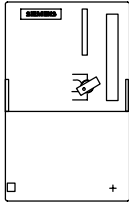
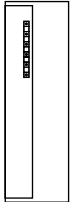

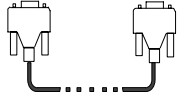
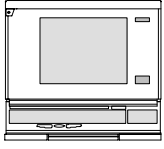
Components	Function	Diagram
Mounting rack	... provides the mechanical and electrical connections of the S7-300.	
Power supply module (PS)	... converts the line voltage (120/230 VAC) into the operating voltage of 24 VDC required to supply the S7-300.	
CPU Accessories: memory card backup battery	... executes the application program; communicates via the MPI interface with other CPUs or with a programming device.	
CP 340 communications processor	... communicates via the interface with a communication partner.	
Standard connecting cable	... connects the CP 340 communications processor to the communication partner.	

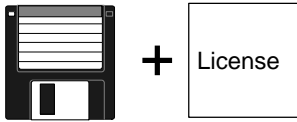

Table 1-3 Hardware Components for a Point-to-Point Connection with the CP 340, continued

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 340.

Table 1-4 Software Components for a Point-to-Point Connection with the CP 340

Components	Function	Diagram
STEP 7 software package	... configures, parameterizes, programs and tests the S7-300.	
<i>CP 340: Point-to-Point Communication, Parameter Assignment interface</i>	... parameterizes the interface of the CP 340.	
Function blocks (FBs) with programming example	... control communication between the CPU and the CP 340.	

1.3 Design of the CP 340

Introduction

The CP 340 communications processor is supplied with an RS 232C interface.

Position of Module Elements

Fig. 1-1 shows the positions of the module elements on the front panel of the CP 340 communications processor.

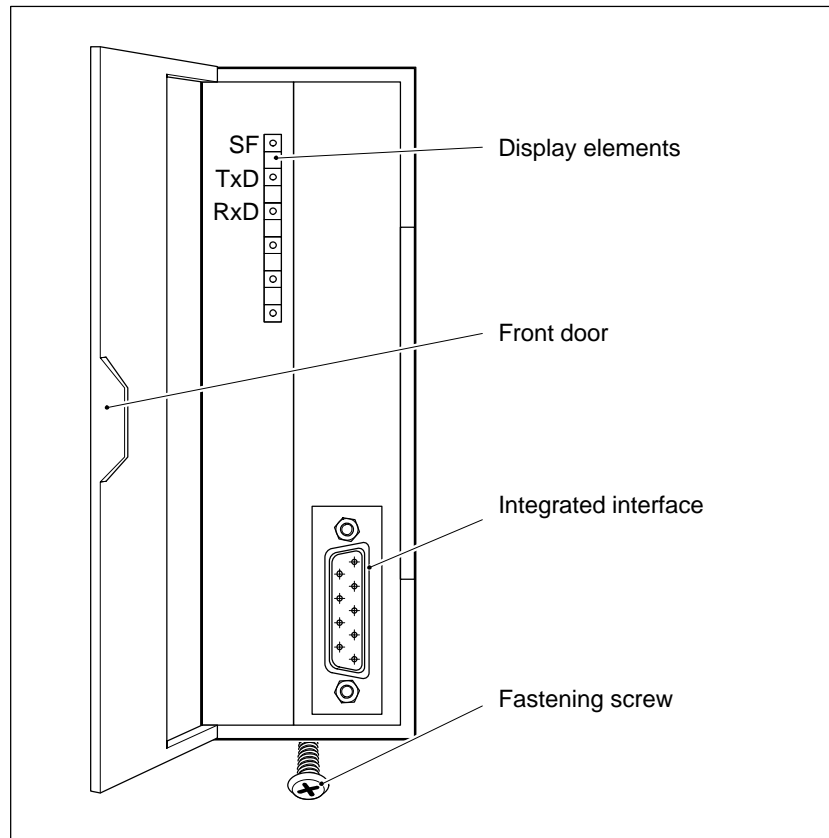


Figure 1-1 Position of the Module Elements on the CP 340 Communications Processor

LED displays

The following LED displays are located on the front panel of the CP 340:

- **SF** (red) Indicates fault
- **TxD** (green) Interface transmitting
- **RxD** (green) Interface receiving

The operating modes and errors indicated by these LEDs are described in Section 8.2.

Integrated Interface

The CP 340 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 340. You will find a detailed description of the interfaces in Section 1.4.

Backplane Connector for S7 Backplane Bus

A bus connector is supplied with the CP 340. The bus connector is plugged onto the back panel of the CP 340 when mounting the CP 340 (see Section 4.2). The S7-300 backplane bus is connected via the bus connector.

The S7-300 backplane bus is a serial data bus via which the CP 340 communicates with the modules of the programmable controller and is supplied with the necessary voltage.

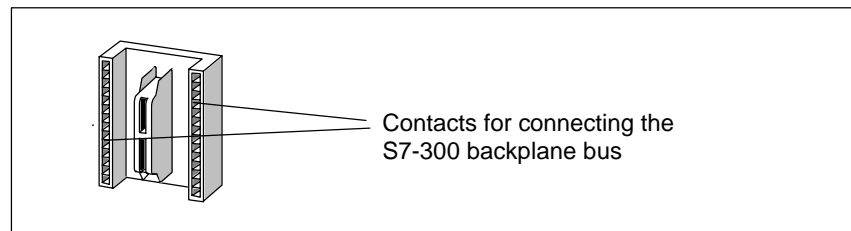


Figure 1-2 Bus Connector

1.4 Attributes of the Serial Interface

Introduction

Three module variants of the CP 340 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 340-RS 232C

Definition

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

Properties

The RS 232C interface has the following properties:

- Type: voltage interface
- Front connector: 9-pin sub D male with screw fixing
- RS 232C signals: TXD, RXD, RTS, CTS, DTR, DSR, RI, DCD, GND; All isolated against S7-internal power supply
- Max. transmission rate: 19.2 Kbaud (3964(R) procedure)
9.6 Kbaud (ASCII driver, printer driver)
- Max. cable length: 15 m, cable type LIYCY 7 × 0.14
- Standards: DIN 66020, DIN 66259, EIA-RS 232C, CCITT V.24/V.28
- Degree of protection: IP 00

RS 232C Signals

Table 1-3 shows the meaning of the RS 232C 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 340 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 340 is ready to send RTS "OFF": CP 340 does not send
CTS	Clear To Send	Communication partner can receive data from CP 340. The CP 340 expects the signal as response to RTS "ON".
DTR	Data Terminal Ready	DTR "ON": CP 340 is active and ready for operation DTR "OFF": CP 340 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 340-20mA TTY

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

Properties The 20 mA interface has the following properties and fulfills 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
- Max. transmission rate: 9.6 Kbaud
- Max. cable length: 100 m active, 1000 m passive; cable type LIYCY 7 × 0.14
- Standards: DIN 66258 Part 1
- Degree of protection: IP 00

1.4.3 X27 (RS 422/485) Interface of the CP 340-RS 422/485

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

Properties The X27 (RS 422/485) interface has the following properties and fulfills the following requirements:

- Type: Voltage-difference 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
- Max. transmission rate: 19.2 Kbaud (3964(R) procedure)
9.6 Kbaud (ASCII driver, printer driver)
- Max. cable length: 1200 m, cable type LIYCY 7 × 0.14
- Standards: DIN 66259 Parts 1 and 3,
EIARS 422/485, CCITT V.11
- Degree of protection: IP 00

Note

The X27 (RS 422/485) interface can only be run in 4-wire mode with the 3964 procedure.

1.5 Cables for Connecting the CP 340 to a Communication Partner

Standard Connecting Cables

For point-to-point connections between the CP 340 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

Section Overview This chapter is divided into the following sections:

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	2-7
2.3	Parameterization Data	2-36

2.1 Serial Transmission of a Character

Introduction	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 340 communications processor forms the interface between a programmable controller and a communication partner. In a point-to-point connection with the CP 340, the data is transmitted serially.
Serial Transmission	In serial transmission, the individual bits of each byte of information are transmitted one after the other in a fixed order.
Uni/Bidirectional Data Traffic	<p>The CP 340 itself handles data transmission with communication partners via the serial interface. The CP 340 is equipped with three different drivers for this purpose.</p> <ul style="list-style-type: none">• Unidirectional data traffic:<ul style="list-style-type: none">– Printer driver• Bidirectional data traffic:<ul style="list-style-type: none">– ASCII driver– 3964(R) procedure <p>The CP 340 handles data transmission via the serial interface in accordance with the interface type and the selected driver.</p>
Unidirectional Data Traffic - Printer Output	In the case of printer output (printer driver), n bytes of user data are output to a printer. No characters are received. The only exception to this are data flow control characters (e.g. XON/XOFF).

Bidirectional Data Traffic - Operating Modes

The CP 340 has two operating modes for bidirectional data traffic:

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

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. Every communication partner must be able to operate a send and a receive facility simultaneously.

You can choose between half-duplex operation (RS 485) and full-duplex operation (RS 422) when using the CP 340-RS 422/485 module variant.

Asynchronous Data Transmission

With the CP 441, 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 length of the start-bit transmission determines the clock pulse. 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 sending and receiving partners before serial transmission can take place. These include:

- Transmission speed (baud rate)
- Character and acknowledgment delay times
- Parity
- Number of data bits
- Number of stop bits
- Number of setup and transmission attempts permitted

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 340 and a communication partner via the serial interface in a 10-bit or 11-bit character frame. Three data formats are available for each character frame. You can parameterize the format you require using the parameterization interface *CP 340: Point-to-Point Communication, Parameter Assignment*.

10-Bit Character Frame

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

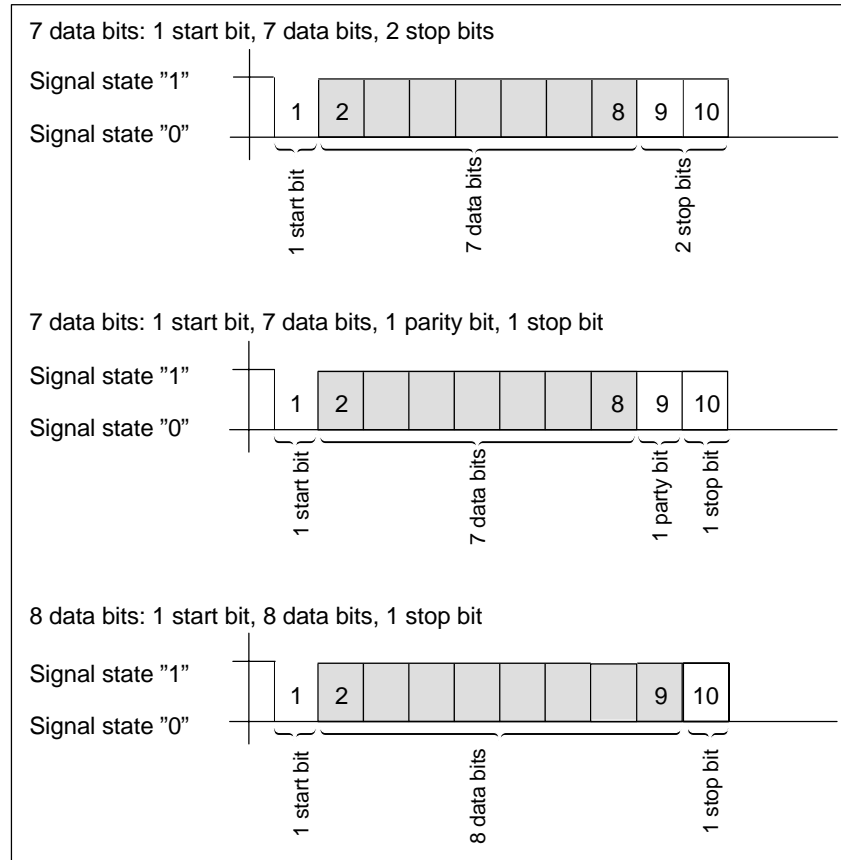


Figure 2-1 10-Bit Character Frame

11-Bit Character Frame

The figure below shows the three possible data formats for an 11-bit character frame.

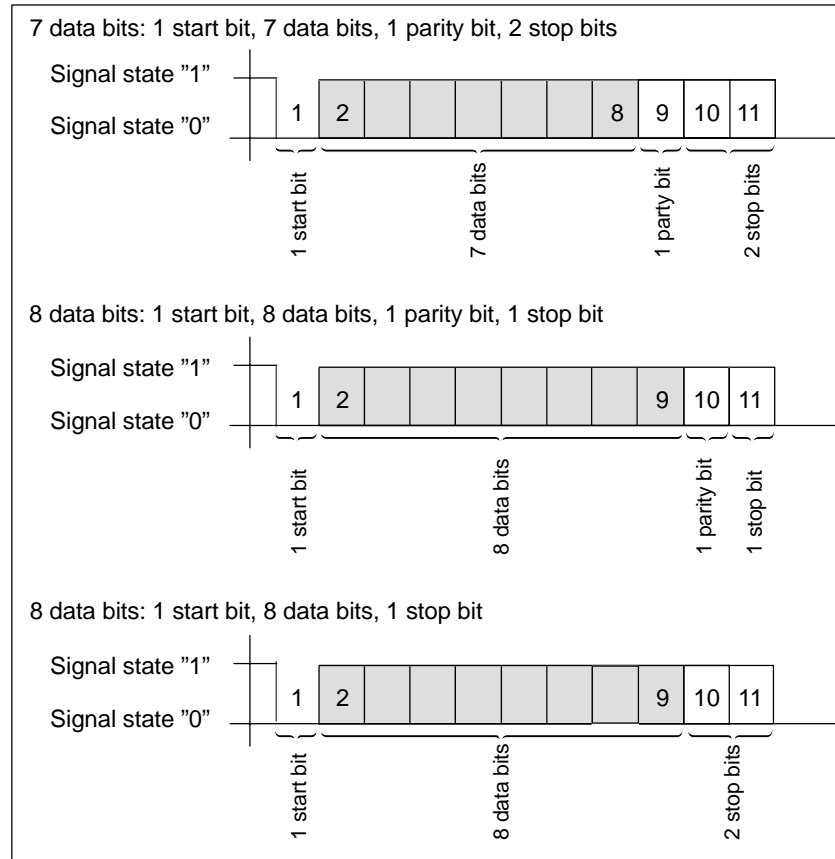


Figure 2-2 11-Bit Character Frame

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.

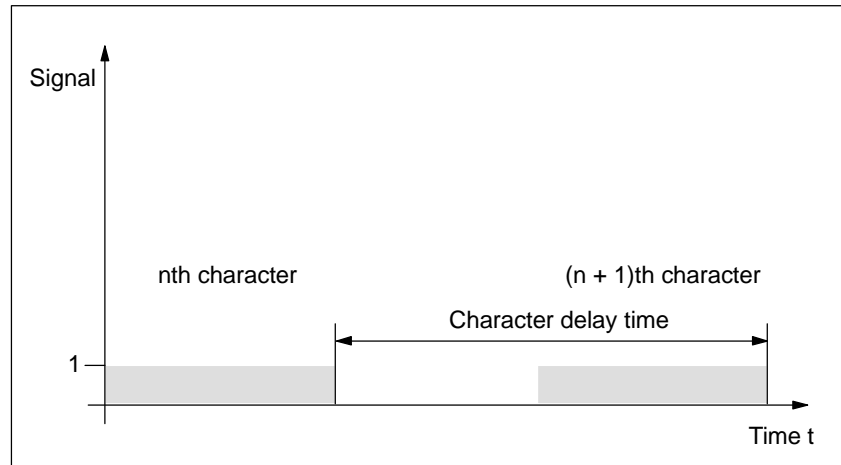


Figure 2-3 Character Delay Time

2.2 Transmission Procedure with a Point-to-Point Connection

Introduction 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 operation
- **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-link 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
- Handshakes

5. **Session layer**

- Establishing and terminating communication
- Communication control

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 340 governs the following protocols:

- 3964(R) procedure
- ASCII driver
- Printer driver

The figure below illustrates how these supplied protocols of the CP 340 fit into the ISO reference model:

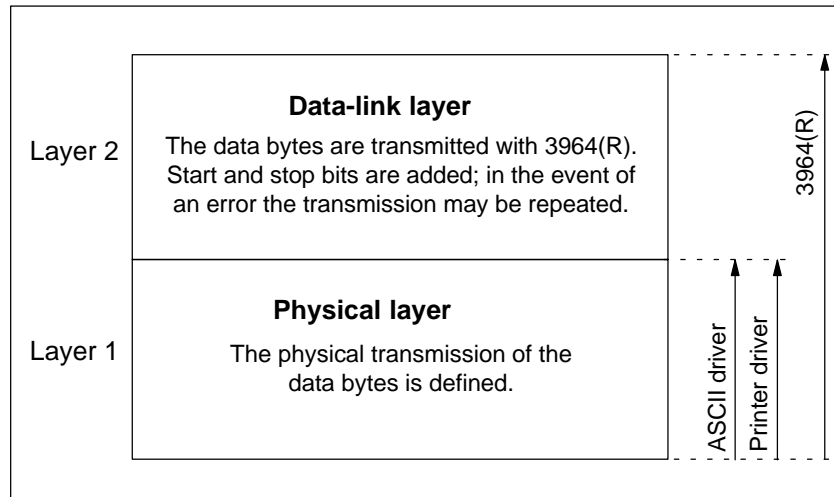


Figure 2-4 Position of the Supplied Protocols of the CP 340 in the ISO Reference Model

Transmission Integrity with the Printer Driver

Data Integrity When Using the Printer Driver:

- No data integrity precautions are taken for data transmission with the printer driver.
- To prevent data from being lost in the event of the printer receive buffer overflowing, you can work with data flow control (XON/XOFF, RTS/CTS).
- When data is output to the printer, the printer's BUSY signal is evaluated. The CP 340 receives the BUSY signal as a CTS signal and evaluates it in the same way (see ASCII driver). **Please note** that, when using CTS/RTS flow control, you must set the polarity of the BUSY signal to CTS = "OFF" on the printer.

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 also the case with high-level protocols for data communication (see ISO 7-layer reference model).

Transmission Integrity with 3964(R)

Enhanced Data Integrity with the 3964(R) Procedure:

- The Hamming distance with the 3964(R) is 3. This measures the integrity of data transmission.
- The 3964(R) procedure ensures high transmission integrity on the data line. This high integrity is achieved by means of a fixed message-frame set-up and clear-down 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 a block check character: **3964(R)**

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

**Performance
Limits with 3964(R)**

Performance Limits of the 3964(R) 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 3964(R) procedure (EXOR operation) cannot detect missing zeros (as a whole character) because a zero in the EXOR 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.

2.2.2 Data Transmission with the 3964(R) Procedure

Introduction The 3964(R) procedure controls data transmission via a point-to-point connection between the CP 340 and a communication partner. As well as the physical layer (layer 1), this procedure also incorporates the data-link layer (layer 2).

Control Characters During data transmission, the 3964(R) procedure adds control characters to the user 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
- **DLE** Data link escape
- **ETX** End of text
- **BCC** Block check character (with 3964(R) only)
- **NAK** Negative acknowledge

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 try to send at the same time, the partner with the lower priority will defer its send request.

Block Checksum

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

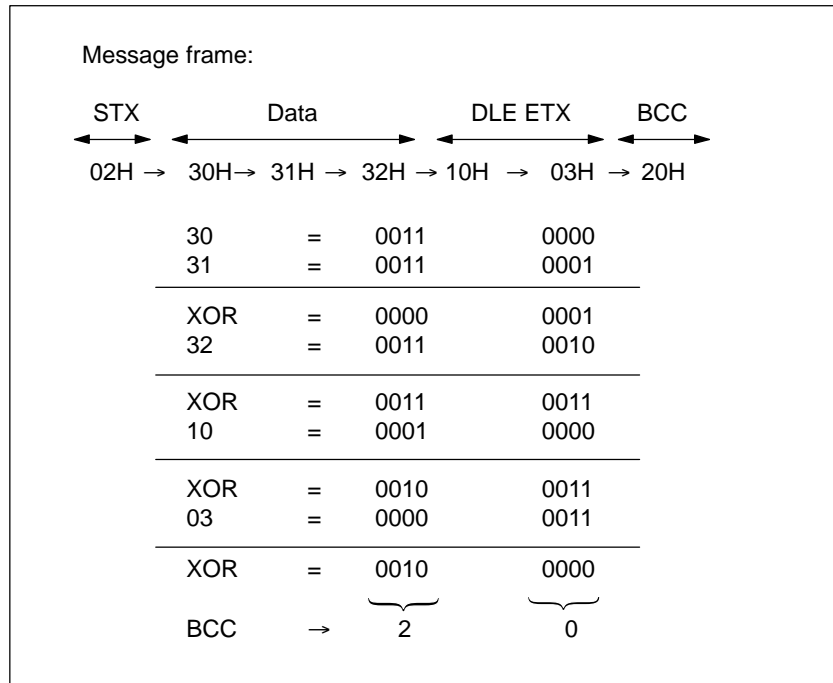


Figure 2-5 Block Checksum

The block checksum is the even longitudinal parity (EXOR operation on 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.

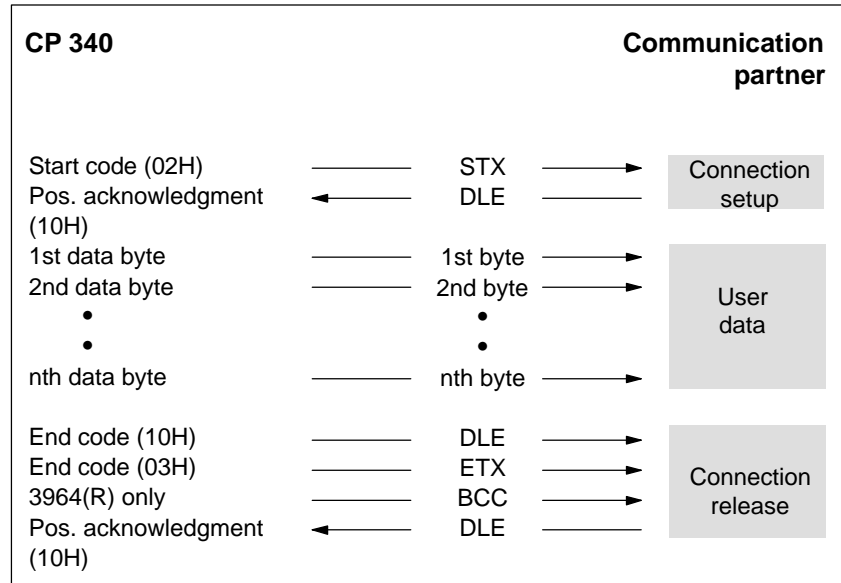


Figure 2-6 Data Traffic when Sending with the 3964(R) Procedure

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 expires, the procedure switches to send mode.

If the communication partner answers with NAK or with any other control code (except for DLE), 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 system program reports the error to the function block P_SEND (output parameter STATUS).

Sending Data

If a connection is successfully established, the user data contained in the output buffer of the CP 340 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.

Releasing a Send Connection

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. 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.

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 system program reports the error to the function block P_SEND (output parameter STATUS).

Receiving Data with 3964(R)

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

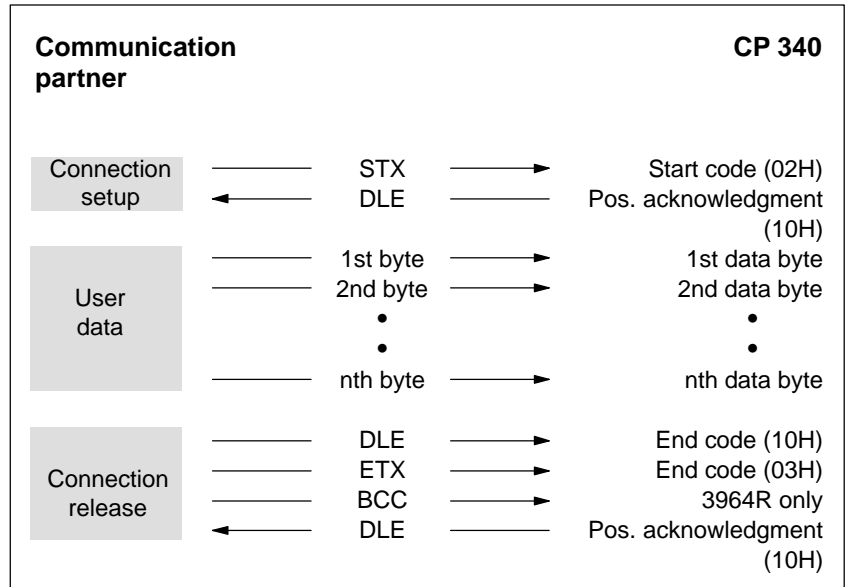


Figure 2-7 Data Traffic when Receiving with the 3964(R) Procedure

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 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.

Receiving Data

If the procedure receives the STX code and an empty receive buffer is available, it responds with DLE. Incoming receive characters are now 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, an NAK is sent to the communication partner. The system program reports the error to the function block P_RCV (output parameter STATUS).

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 system program 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 as described above.

Releasing a Receive Connection

If transmission errors occur during receiving (lost character, frame error, parity error, etc.), the procedure continues to receive until the connection is released, then an NAK is sent to the communication partner. A repetition is then expected. If the undamaged block still cannot be received after the number of repeat attempts defined on parameter assignment, 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 system program then reports the error to the function block P_RCV (output parameter STATUS).

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, or an NAK if it was damaged. 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, an NAK is sent to the communication partner. A repetition is then expected.

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.

Handling Errored Data

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

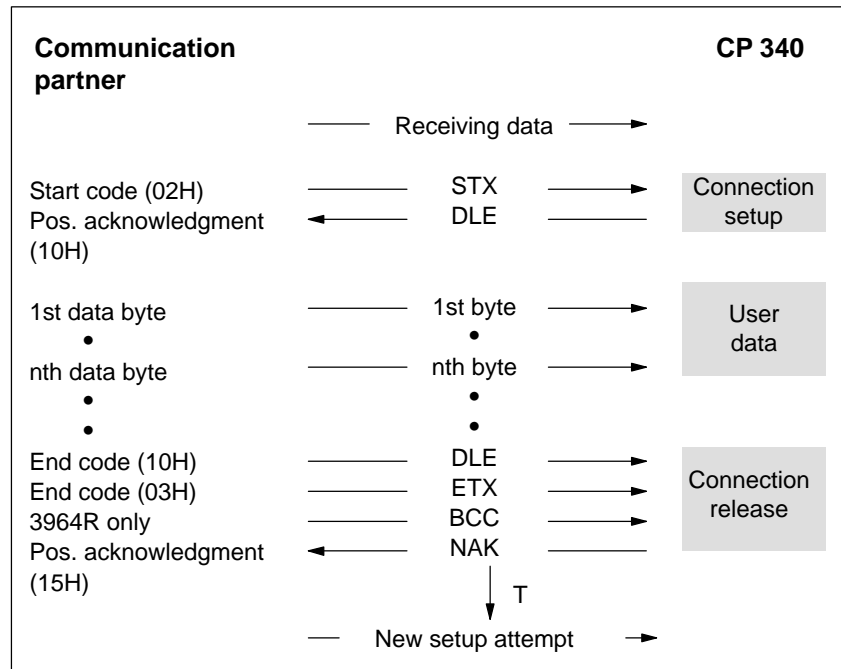


Figure 2-8 Data Traffic when Receiving Errored Data

When DLE, ETX, BCC is received, the CP 340 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 340 responds with DLE.

Otherwise, it 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 340 aborts the receive operation.

Initialization Conflict

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

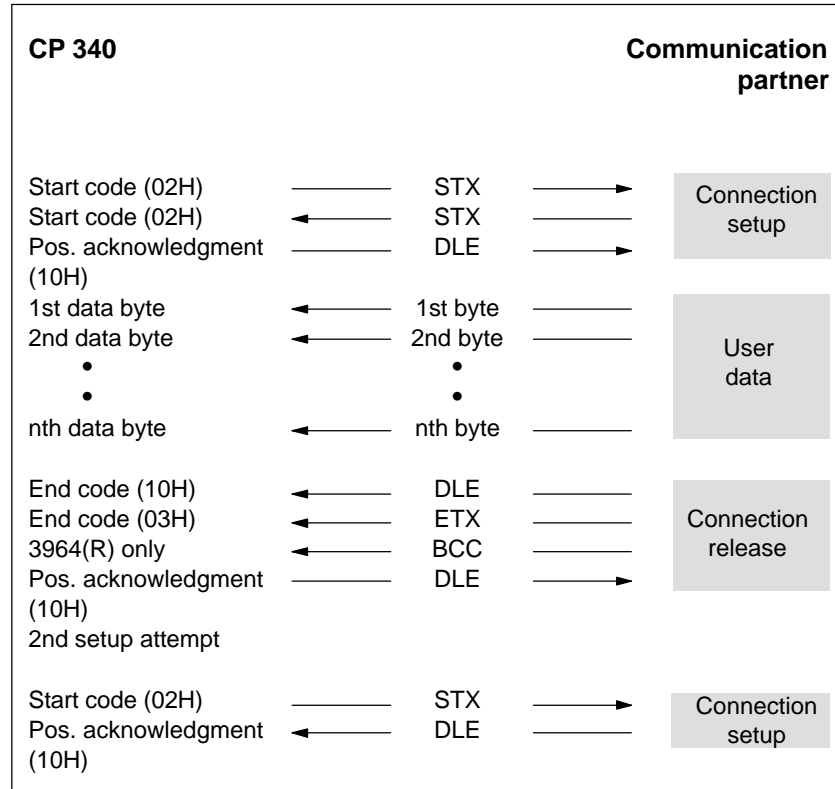


Figure 2-9 Data Traffic during an Initialization Conflict

If a device responds to the communication partner's send request (code STX) within the acknowledgment delay time 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 repeat 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 high repetition frequency balances this out, however. In this case you are advised to check the transmission link 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 link is interrupted, the system program reports a BREAK status (a break is displayed via the diagnostics alarm of the CP 340) (see Section 8.4) and 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. A BREAK evaluation occurs only if the BREAK monitoring is not deactivated with the parameter assignment user interface.

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.

2.2.3 Data Transmission with the ASCII Driver

Introduction

The ASCII driver controls data transmission via a point-to-point connection between the CP 340 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 340. 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 sending data, specify the number of bytes of user data to be transmitted for the parameter LEN when you call the function block P_SEND. The user data must contain any required start-of-text and end-of-text characters.

If you are working with the end criterion "character delay time expired" when receiving data, the ASCII driver pauses between two message frames, also when sending. You can call the P_SEND 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.

Note

When the flow control XON/XOFF 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.

Sending Data

The figure below illustrates a send operation.

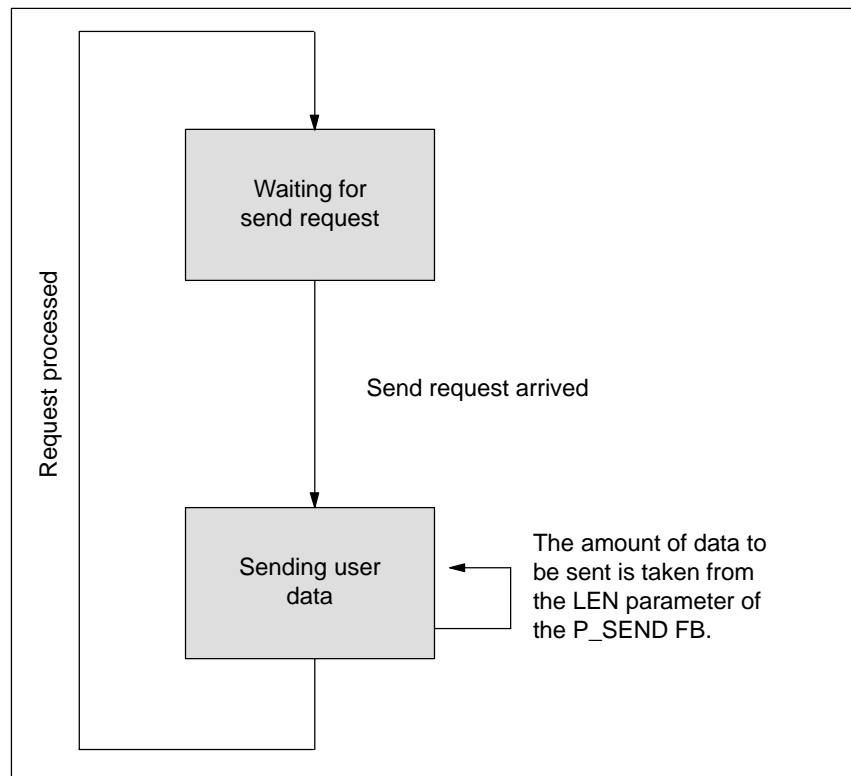


Figure 2-10 Sequence of a Send Operation

Receiving Data with the ASCII Driver

For data transmission using the ASCII driver you can choose between three different end criteria. 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 choice of parameterized end criterion and 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 the flow control XON/XOFF 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 taken over by the CPU with the function block P_RCV.

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 link takes a send pause within a message frame.

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

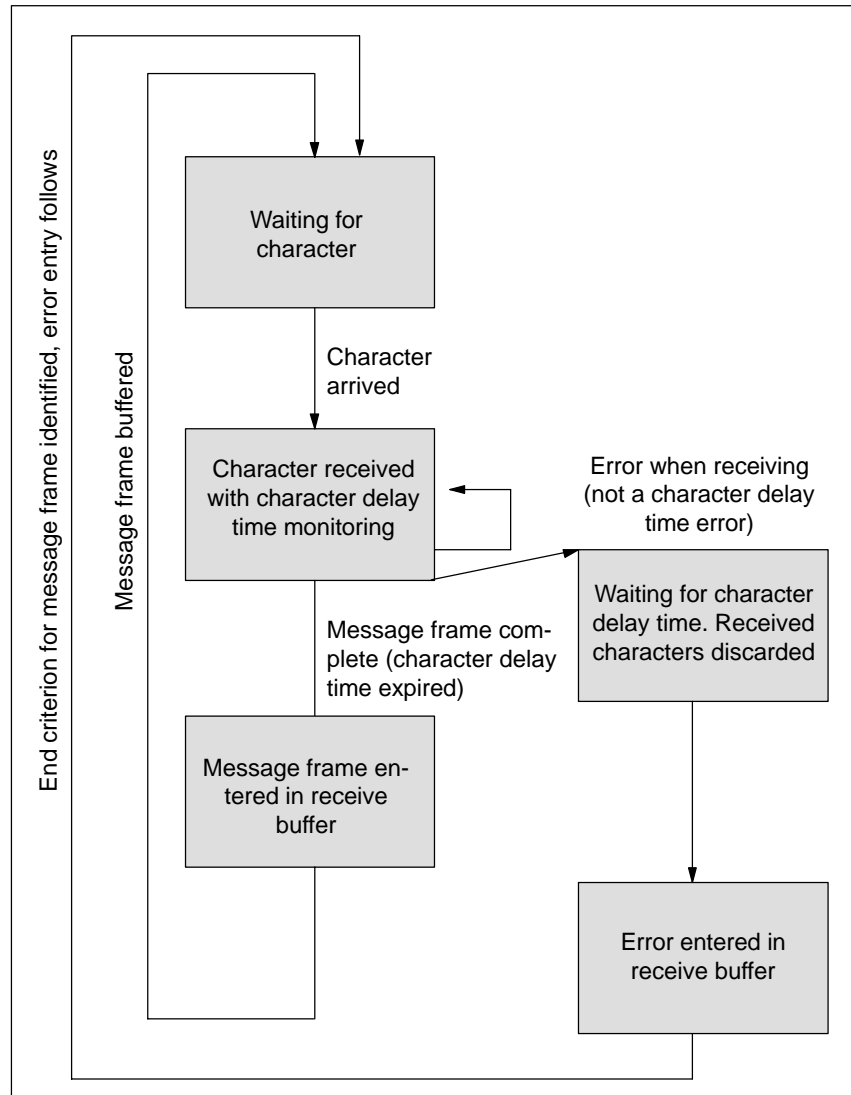


Figure 2-11 Sequence of Receive Operation with End Criterion "Expiry of Character Delay Time"

**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, is taken over by the CPU with the function block P_RCV.

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) do not appear in the user data of the user.

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

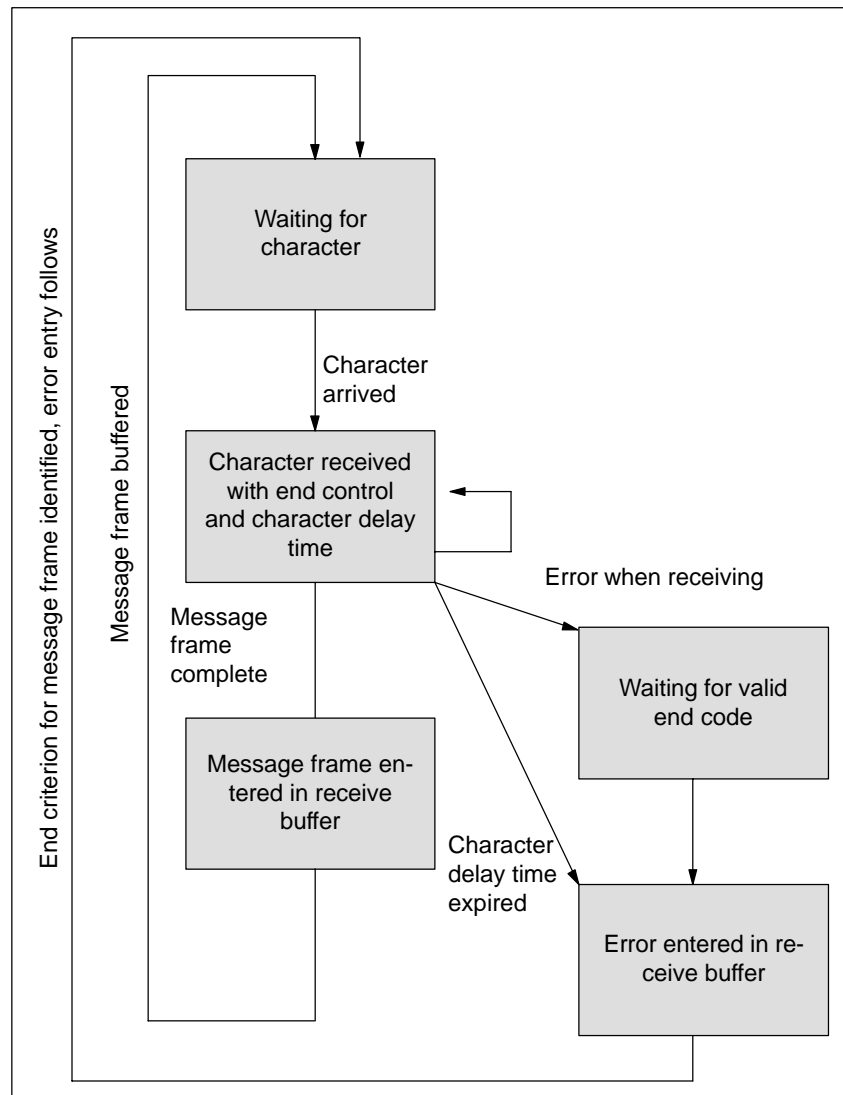


Figure 2-12 Sequence of Receive Operation with End Criterion "End-of-Text Character"

**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 taken over by the CPU with the function block P_RCV.

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.

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

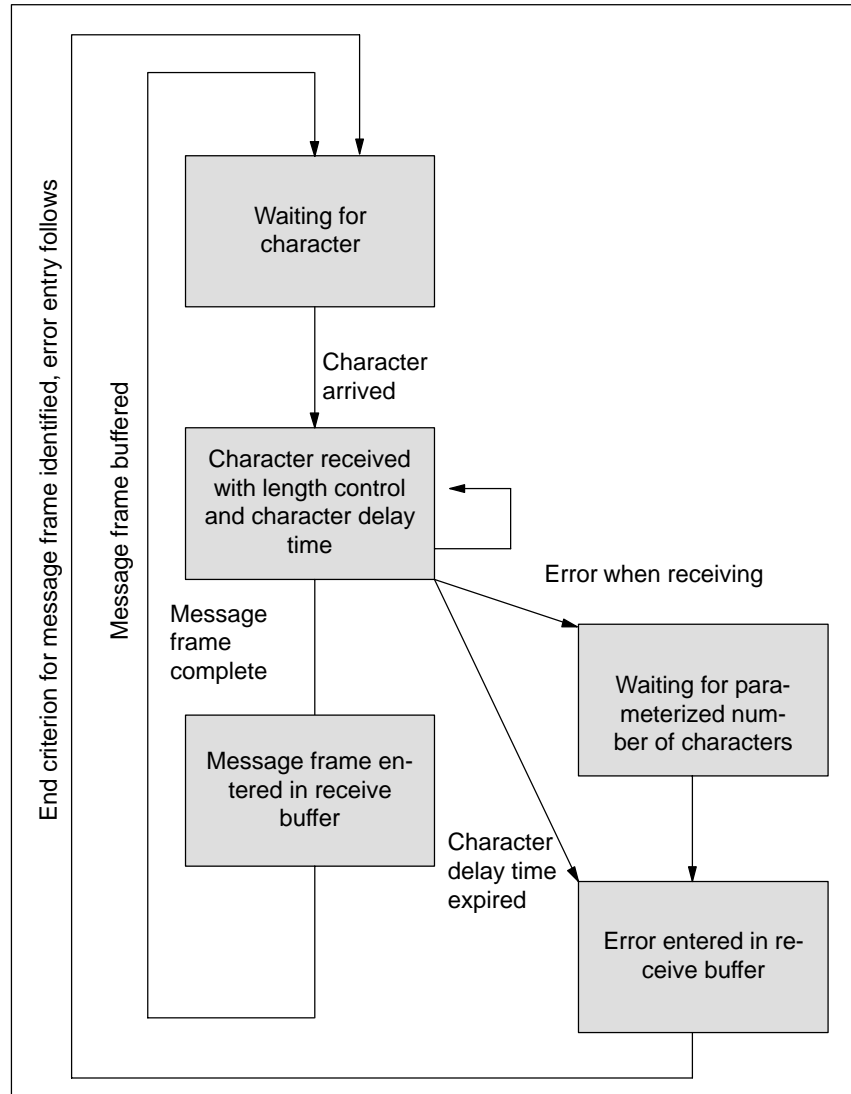


Figure 2-13 Sequence of Receive Operation with End Criterion "Fixed Message Frame Length"

BREAK - Monitoring on CP340

A BREAK evaluation only occurs if the BREAK monitoring is not deactivated with the parameter assignment user interface.

Receive Buffer on CP 340

The CP 340 receive buffer accommodates 1024 bytes. On parameterization, you can specify whether the CP receive buffer is to be deleted at start-up and whether the overwriting of data in the receive buffer is to 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 340 is a ring buffer:

- If several message frames are entered in the receive buffer of the CP 340, it is always the oldest one that is sent from the CP 340 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 received data from the user program is interrupted for a while, you may find that when the received data is requested again, the CPU first receives an old message frame from the CP 340 before it receives the most recent one.

The old message frame is the one that was on its way when transmission between the CP 340 and the CPU was interrupted, or which had already been received by the FB.

For further information on receive buffers, see Section 7.3.

**RS 232C
Secondary Signals**

The following RS 232C secondary signals exist on the CP 340-RS 232C (see also Appendix B):

- **DCD** (input) Data carrier detect;
Data carrier detected
- **DTR** (output) Data terminal ready;
CP 340 ready for operation
- **DSR** (input) Data set ready;
Communication partner ready for operation
- **RTS** (output) Request to send;
CP 340 ready to send
- **CTS** (input) Clear to send;
Communication partner can receive data from the
CP 340 (response to RTS = ON of the CP 340)
- **RI** (input) Ring Indicator;
Indication of an incoming call

When the CP 340-RS 232C is switched on, the output signals are in the OFF state (inactive).

You can parameterize the use of the DTR/DSR and RTS/CTS control signals by means of the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface or control them in the user program by means of functions (FCs).

**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 340 is implemented as follows:

- As soon as the CP 340 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 340 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 an 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 340 threatens to overflow, the CP 340 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 340.

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.

Time Diagram

Figure 2-14 illustrates the chronological sequence of a send request.

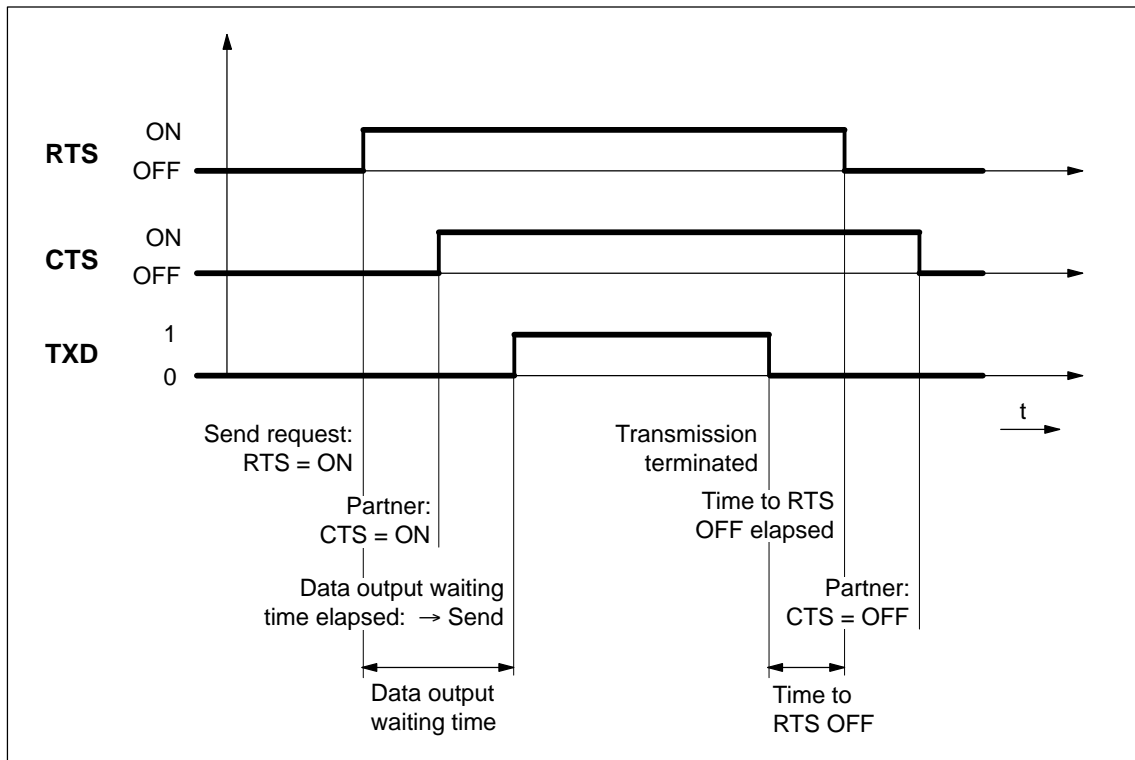


Figure 2-14 Time Diagram for Automatic Use of the RS 232C Secondary Signals

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 340:

- As soon as the CP 340 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 340 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 340 sends the XON character or sets the RTS line to ON.
- If the CP 340 receives the XOFF character, or the CTS control signal is set to OFF, the CP 340 interrupts the transmission. If neither an XON character is received nor CTS 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 FC V24_STAT and FC V24_SET

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.

2.2.4 Data Transmission with the Printer Driver

Introduction

The printer driver allows you to output message texts with the date and time to a printer. This enables you to monitor simple processes, print error or fault messages or issue instructions to the operating personnel, for example.

The printer driver contains the physical layer (layer 1).

Message Texts and Parameters for Printer Output

You use the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface to configure the message texts and set the parameters (page layout, character set, control characters) for printer output. Message texts and printer output parameters are transmitted to the CP 340 together with the module parameters when it starts up.

Message texts:

You can configure message texts with variables and control statements (e.g. for bold, condensed, expanded or italic type and underlining). Each message text is assigned a number during configuration. A message text is printed if its number is specified in a format string when the P_PRINT function block is called.

You must have stored the format string and variables in data blocks beforehand (see Chapter 6).

Page layout:

You can configure the margins, possible line breaks and headers and footers.

Character set:

The ANSI character set is converted to the printer character set by STEP 7 by means of a character conversion table. You can change a character conversion table suggested for a printer type in order to include special characters required for a particular language, for example.

Control characters:

By means of a control character table you can change the control statements in the message text for the printer emulation for switching on and off bold, condensed, expanded or italic type and underlining and to add control characters.

Variables

Up to 4 variables (3 + a message text number) can be displayed in a message text. The values of variables can be transmitted from the CPU to the CP 340. The following can be displayed as variables: values calculated by the application program (e.g. levels), date and time, strings (string variables) or other message texts.

A conversion statement must be specified in the configured message text or in the format string for each variable, and the meaning and output format of the variable value must be encoded in this statement.

Format String

The format string allows you to define the format and composition of a message text. The format string can consist of:

- Text (e.g. The level ... l was reached at ... hours.). All printable characters are permitted.
- Conversion statements for variables (e.g. %N = pointer to message text number x, where x is the value of a variable (see example 2 below))

For each variable there must be one conversion statement in the format string or in the configured message text. The conversion statements are applied to the variables in the sequence in which they occur.

- Control statements with control characters for bold, condensed or italic type and underlining (e.g. \B = bold type on) or with additional control characters you have defined.

You can use additional control characters if you enter them in the control character table of the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface and reparameterize the CP 340.

You will find all the conversion and control statements that are possible in the format string and in configured message texts in Section 2.3.5.

Additional Functions

In addition to outputting message texts, you can use the following functions for printer output. If you want one of these functions to be executed, you specify it in the format string in the same way.

- Set page number (format string = %P)
- Begin new page (form string = \F)
- Print with/without line break (\x at the end of the format string)

Please note that a line feed is carried out by default after each output.

Examples

Example 1: The level "200" l was reached at "17.30" hours.

Format string = The level %i l was reached at %Z hours.
 Variable 1 = Time
 Variable 2 = Level

Example 2: The pressure in the chamber "is falling"

Format string = %N %S
 Variable 1 = 17 (message text no. 17: The pressure in the chamber ...)
 Variable 2 = Reference to string (string variable: ... is falling)

Example 3: (Set page number to 10)

Format string = %P
 Variable 1 = 10 (page number: 10)

Printer Output

To output n bytes of user data to a printer, you specify the block number of a pointer DB when calling the P_PRINT function block. The pointers to the data blocks are stored in the pointer DB together with the format string and the variables and in a specific order (see Section 6.3.2).

During output the data is edited for printing. The print editing is performed as parameterized by means of the CP 340: *Point-to-Point Communication, Parameter Assignment* parameterization interface (page layout, character set, control characters, etc.).

Characters are not received during printer output. The exception to this are any flow control characters that have been parameterized. Any characters received are not adopted.

Note

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

Message Text Output

The figure below illustrates the sequence of operations at printer output.

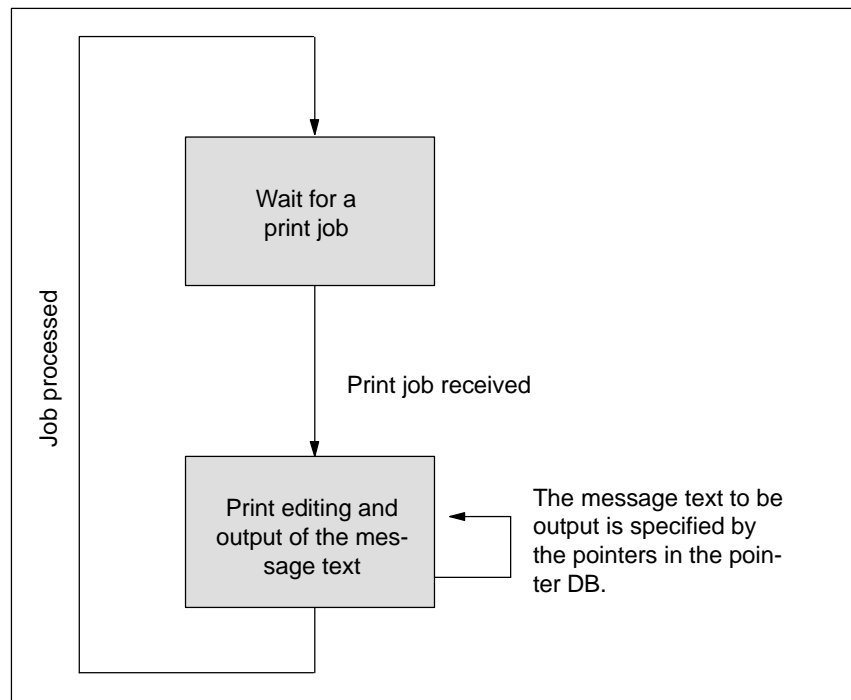


Figure 2-15 Flow Chart of Printer Output

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.

You can also send message texts with data flow control during printer output. 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 340 during printer output:

- As soon as the CP 340 is switched by parameterization to the operating mode with flow control, it sends the XON character or sets the RTS line to ON.
- If the CP 340 receives the XOFF character, or the CTS control signal is set to OFF, the CP 340 interrupts the output of characters. If neither an XON character is received nor CTS is set to ON before a parameterized time has elapsed, printer output is aborted and an appropriate error message (0708H) is generated at the STATUS output of the PRINT SFB.

Note

When RTS/CTS flow control is parameterized, you must fully wire the interface signals in the plug connection (see Appendix B.1).

BUSY Signal

The CP 340 evaluates the printer's "BUSY" control signal. The printer indicates to the CP 340 that it is ready to receive:

- In the case of the CP 340-20mA TTY: by means of current on the RXD line
- In the case of the CP 340-RS 232C and CP 340-RS 422/485: by means of the signal CTS = "ON"

Note

When you parameterize with RTS/CTS flow control, you must set the polarity of the BUSY signal on the printer as follows:

- BUSY signal: CTS = "OFF"

Please note that some printers use the DTR signal to display the BUSY signal. In such cases you must wire the connecting cable to the CP 340 appropriately.

2.3 Parameterization Data

Introduction

By setting the basic parameter, you can define the diagnosis behavior of the CP 340. By selecting different protocols, you can adjust your CP 340 communications processor to suit the properties of the communication partner.

The sections that follow describe the basic parameter of the CP 340 and the parameterization data for the 3964(R) procedure, ASCII driver and printer driver.

2.3.1 Basic Parameter of the CP 340

Introduction

You can define in this parameter whether the CP 340 generates a diagnostics alarm if serious errors occur.

Basic Parameter

Enter the basic parameter using the STEP 7 dialog “Properties - CP 340”. Double-click on the CP 340 in the STEP 7 configuration table to open the dialog. How to enter the basic parameter of the CP 340 is described in Section 5.2.

The basic parameters are described in the table below.

Table 2-1 Basic Parameter

Parameter	Description	Value Range	Default Value
Diagnostics alarm	If a serious error occurs, the CP 340 can generate a diagnostics alarm (see Section 8.4).	<ul style="list-style-type: none">• yes• no	<ul style="list-style-type: none">• no

2.3.2 Parameterization Data of the 3964(R) Procedure

Introduction Using the parameterization data of the 3964(R) procedure, you can adjust the CP 340 to suit the properties of its communication partner.

Parameterization Data of the 3964(R) Procedure In the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface you specify the parameters for the physical layer (layer 1) and the data-link layer (layer 2) of the 3964(R) procedure. In the following you will find a detailed description of the parameters.

How to enter the parameterization data using the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface is described in Section 5.2.

X27 (RS 422/485) Interface Please note the following in relation to the X27 (RS 422/485) interface:

Note

In the case of the CP 340-RS 422/485 module variant, the 3964(R) procedure can only be used with RS 422.

Protocol The following table describes the protocol.

Table 2-2 3964(R) Protocol

Parameter	Description	Default Value
3964 with default values and no block check	<ul style="list-style-type: none"> • The protocol parameters are set to default values. • If the CP 340 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 default values and block check: Character delay time = 220 ms Acknowledgment delay time = 2000 ms Setup attempts = 6 Transmission attempts = 6
3964R with default values and block check	<ul style="list-style-type: none"> • The protocol parameters are set to default values. • If the CP 340 recognizes the string DLE ETX BCC, it stops receiving and 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 340 sends the code DLE to the communication partner. (In the event of an error, the NAK code is sent). 	
3964 parameterized without block check	<ul style="list-style-type: none"> • The protocol parameters are freely selectable. • If the CP 340 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 parameterized with block check	<ul style="list-style-type: none"> • The protocol parameters are freely selectable. • If the CP 340 recognizes the string DLE ETX BCC, it stops receiving and 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 340 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)

Parameter	Description	Value Range	Default Value
Character delay time	The character delay time defines the maximum amount of time permitted between two incoming characters in a message frame.	20 ms to 655350 ms in 10-ms increments	220 ms
Acknowledgment delay time	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).	10 ms to 655350 ms in 10-ms increments	2000 ms (550 ms with 3964 without block check)
Setup attempts	This parameter defines the maximum number of attempts the CP 340 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)

Parameter	Description	Value Range	Default Value
Baud rate	Speed of data transmission in bits/s	<ul style="list-style-type: none"> • 2400 • 4800 • 9600 • 19200 	<ul style="list-style-type: none"> • 9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	<ul style="list-style-type: none"> • 1 (fixed value) 	<ul style="list-style-type: none"> • 1
Data bits	Number of bits to which a character is mapped.	<ul style="list-style-type: none"> • 7 • 8 	<ul style="list-style-type: none"> • 8
Stop bits	During transmission, a stop bit is appended to every character to be sent to signal the end of the character.	<ul style="list-style-type: none"> • 1 • 2 	<ul style="list-style-type: none"> • 1
Parity	<p>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 enhancing data integrity.</p> <p>A parity of "none" means that no parity bit is sent.</p> <p>A parity of "any" means that when data is sent the CP sets the parity to 0. When data is received, the parity bit is not checked.</p>	<ul style="list-style-type: none"> • none • odd • even • any 	<ul style="list-style-type: none"> • even

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

Parameter	Description	Value Range	Default Value
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.	<ul style="list-style-type: none"> • high • low 	<ul style="list-style-type: none"> • high
Activate BREAK monitoring	<p>You can choose whether the monitoring on an interrupted receive line should be activated or deactivated.</p> <p>If a BREAK monitoring is deactivated during a Break,</p> <p>a) an entry is not made into the diagnostic buffer and the FB-RECV is not triggered with the set ERROR-Bit and the corresponding STATUS entry)</p> <p>b) triggered send requests are output to the user without an error message.</p>	<ul style="list-style-type: none"> • yes • no 	<p>depending on the HW variant used and the selected operating mode:</p> <p>RS232: <ul style="list-style-type: none">• yes</p> <p>TTY: <ul style="list-style-type: none">• yes</p> <p>RS422 for R(A)5V/R(B)0V:</p> <ul style="list-style-type: none"> • yes (cannot be activated) <p>RS422 bei R(A)0V/R(B)5V:</p> <ul style="list-style-type: none"> • no (cannot be activated) <p>For this HW variant RS422, this parameter is directly influenced via the selected initial state of the receive line in the “Interface” folder.</p>

Receive Buffer on CP

You will find the description of the parameters for the X27 (RS 422) interface in the following table:

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

Parameter	Description	Value Range	Default Value
Delete CP receive buffer during start-up	Here you can specify whether the CP receive buffer is to be deleted during start-up, or whether an existing (old) message frame is to be passed on to the CPU.	<ul style="list-style-type: none"> • yes • no 	<ul style="list-style-type: none"> • yes
Initial State of the receive line	<p>R(A)5V/R(B)0V: a BREAK recognition is possible for this initial state and cannot be deactivated.</p> <p>R(A)0V/R(B)5V: a BREAK recognition is not possible for this initial state</p>	<ul style="list-style-type: none"> • R(A)5V / R(B)0V • R(A)0V / R(B)5V 	<ul style="list-style-type: none"> • R(A)5V / R(B)0V

2.3.3 Parameterization Data of the ASCII Driver

Introduction

Using the parameterization data of the ASCII driver, you can adjust the CP 340 to suit the properties of the communication partner.

Parameterization Data of the ASCII Driver

Using the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface, specify the parameters for the physical layer (layer 1) of the ASCII driver. The parameters are described in detail below.

Section 5.2 describes how to enter parameterization data via the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface.

X27 (RS 422/485) Interface

Please note the following in relation to the X27 (RS 422/485) interface:

Note

In the case of the CP 340-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)

Parameter	Description	Value Range	Default Value
Indicator for end of receive message frame	Defines which criterion signals the end of each message frame.	<ul style="list-style-type: none"> On expiry of character delay time On receipt of end-of-text character On receipt of fixed number of characters 	<ul style="list-style-type: none"> On expiry of character delay time
Character delay time	The character delay time defines the maximum permitted time between 2 consecutively received characters. ¹	4 to 65535 ms	<ul style="list-style-type: none"> 4 ms
End-of-text character 1 ²	First end code.	<ul style="list-style-type: none"> With 7 data bits³: 0 to 7FH (Hex) With 8 data bits³: 0 to FFH (Hex) 	<ul style="list-style-type: none"> 3
End-of-text character 2 ²	Second end code, if specified.	<ul style="list-style-type: none"> With 7 data bits³: 0 to 7FH (Hex) With 8 data bits³: 0 to FFH (Hex) 	<ul style="list-style-type: none"> 0
Message frame length when received ⁴	When the end criterion is "fixed message frame length", the number of bytes making up a message frame is defined.	1 to 1024 (bytes)	<ul style="list-style-type: none"> 240

¹ The shortest possible character delay time is the time needed to transmit 4 characters.

² Can only be set if the end criterion is an end-of-text character.

³ Depending on the parameterization of the character frame (7 or 8 data bits) (see Table 2-7).

⁴ 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)

Parameter	Description	Value Range	Default Value
Baud rate	Speed of data transmission in bits/s Note that the ASCII driver can be operated in full-duplex mode.	<ul style="list-style-type: none"> • 2400 • 4800 • 9600 	<ul style="list-style-type: none"> • 9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	<ul style="list-style-type: none"> • 1 (fixed value) 	<ul style="list-style-type: none"> • 1
Data bits	Number of bits onto which a character is mapped.	<ul style="list-style-type: none"> • 7 • 8 	<ul style="list-style-type: none"> • 8
Stop bits	During transmission, stop bits are appended to every character to be sent, indicating the end of the character.	<ul style="list-style-type: none"> • 1 • 2 	<ul style="list-style-type: none"> • 1
Parity	<p>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 enhancing data integrity.</p> <p>A parity of "none" means that no parity bit is sent.</p> <p>A parity of "any" means that when data is sent the CP 340 sets the parity to 0. When data is received, the parity is not checked.</p>	<ul style="list-style-type: none"> • none • odd • even • any 	<ul style="list-style-type: none"> • even

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

Parameter	Description	Value Range	Default Value
Activate BREAK monitoring	<p>You can choose whether the monitoring on an interrupted receive line should be activated or deactivated.</p> <p>If a BREAK monitoring is deactivated during a Break,</p> <p>a) an entry is not made into the diagnostic buffer and the FB-RECV is not triggered with the set ERROR-Bit and the corresponding STATUS entry)</p> <p>b) triggered send requests are output to the user without an error message.</p>	<ul style="list-style-type: none"> • yes • no 	<p>depending on the HW variant used and the selected operating mode:</p> <p>RS232: • yes</p> <p>TTY: • yes</p> <p>RS422 for R(A)5V/R(B)0V:</p> <ul style="list-style-type: none"> • yes (cannot be deactivated) <p>RS422/RS485 for R(A)0V/R(B)5V:</p> <ul style="list-style-type: none"> • no (cannot be activated) <p>For this HW variant RS422, this parameter is directly influenced via the selected initial state of the receive line in the “Interface” folder.</p>

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)

Parameter	Description	Value Range	Default Value
Data flow control	Defines which type of data flow control is used.	<ul style="list-style-type: none"> • None • XON/XOFF • RTS/CTS • Automatic use of the V24 signals 	<ul style="list-style-type: none"> • None
XON character ¹	Code for XON character	<ul style="list-style-type: none"> • With 7 data bits²: 0 to 7FH (Hex) • With 8 data bits²: 0 to FFH (Hex) 	<ul style="list-style-type: none"> • 11 (DC1)
XOFF character ¹	Code for XOFF character	<ul style="list-style-type: none"> • With 7 data bits²: 0 to 7FH (Hex) • With 8 data bits²: 0 to FFH (Hex) 	<ul style="list-style-type: none"> • 13 (DC3)
Waiting for XON after XOFF (wait time for CTS=ON) ³	Period of time for which the CP 340 shall wait for the XON code or for CTS="ON" of the communication partner when sending.	20 to 655350 ms in 10-ms increments	<ul style="list-style-type: none"> • 2000 ms
Time to RTS OFF ⁴	Time to elapse after the transmission before the CP 340 sets the RTS line to OFF.	0 to 655350 ms in 10 ms steps	<ul style="list-style-type: none"> • 10 ms
Data output waiting time ⁴	Time that the CP 340 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	<ul style="list-style-type: none"> • 10 ms

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

² Depending on whether you parameterize 7 or 8 data bits for the character frame (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

More 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 2.2.3 as of the subsection entitled "RS 232C Secondary Signals".

Receive Buffer on CP 340 The table below describes the parameters for the CP receive buffer.

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

Parameter	Description	Value Range	Default Value
Delete CP receive buffer at start-up ¹	Here you can specify whether the CP receive buffer is to be deleted during start-up, or whether an existing (old) message frame is to be passed on to the CPU.	<ul style="list-style-type: none"> • yes • no 	<ul style="list-style-type: none"> • yes
Buffered receive message frames ¹	<p>Here you can specify the number of receive message frames to be buffered in the CP receive buffer.</p> <p>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.</p>	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.	<ul style="list-style-type: none"> • yes • no (only if "buffered receive message frames" = "1") 	<ul style="list-style-type: none"> • yes

¹ If several message frames are buffered, the CP 340 always sends only the oldest to the CPU (ring buffer)

Further Information In Section 2.2.3 you can find further information on handling the receive buffer under "Receive Buffer on CP 340".

X27 (RS 422/485) 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)

Parameter	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).	<ul style="list-style-type: none"> Full-duplex (RS 422) four-wire mode Half-duplex (RS 485) two-wire mode 	<ul style="list-style-type: none"> Full-duplex (RS 422) four-wire mode
Initial state of the receive line	See Figure 2-16. R(A)0V/R(B)5V: This initial state corresponds to the idle state (no sender active) in half-duplex (RS485) two-wire mode".	<ul style="list-style-type: none"> R(A)5V / R(B)0V R(A)0V / R(B)5V 	<ul style="list-style-type: none"> R(A)5V / R(B)0V

Initial State of the Receive Line

Figure 2-16 illustrates the wiring of the recipient at the X27 (RS 422/ 485) interface:

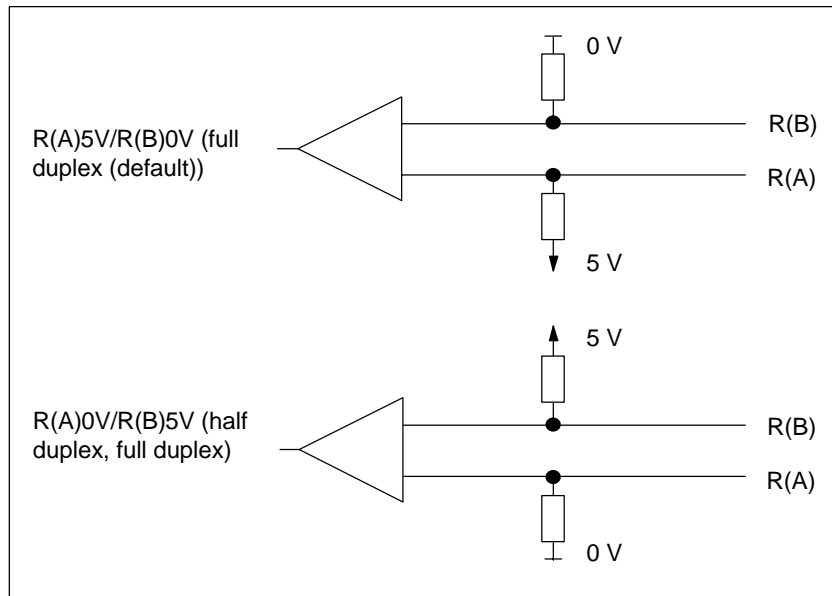


Figure 2-16 Wiring of the Recipient at the X27 (RS 422/485) Interface

2.3.4 Parameterization Data of the Printer Driver

Introduction You can use the parameterization data of the printer driver to generate the transmission-specific parameters and the message texts for printer output.

Parameterization Data of the Printer Driver You use the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface to specify:

- The parameters for the physical layer (layer 1) of the printer driver
- The message texts fro printer output
- The page layout, character set and control characters for the message texts

You will find a detailed description of the parameters below. Section 5.2 describes how to use the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface to enter the parameterization data.

Baud Rate / Character Frame The table below contains descriptions of and specifies the value ranges of the relevant parameters.

Table 2-11 Baud Rate/Character Frame (Printer Driver)

Parameter	Description	Value Range	Default Value
Baud rate	Speed of the data transmission in bits/s	<ul style="list-style-type: none"> • 2400 • 4800 • 9600 	<ul style="list-style-type: none"> • 9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	<ul style="list-style-type: none"> • 1 (fixed value) 	<ul style="list-style-type: none"> • 1
Data bits	Number of bits onto which a character is mapped.	<ul style="list-style-type: none"> • 7 • 8 	<ul style="list-style-type: none"> • 8
Stop bits	During transmission, stop bits are appended to every character to be sent, indicating the end of the character.	<ul style="list-style-type: none"> • 1 • 2 	<ul style="list-style-type: none"> • 1

Table 2-11 Baud Rate/Character Frame (Printer Driver)

Parameter	Description	Value Range	Default Value
Parity	<p>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 enhancing data integrity.</p> <p>A parity of "none" means that no parity bit is sent.</p> <p>A parity of "any" means that when data is sent the CP 340 sets the parity to 0.</p>	<ul style="list-style-type: none"> • none • odd • even • any 	<ul style="list-style-type: none"> • even
Activate BREAK monitoring	<p>You can choose whether the monitoring on an interrupted receive line should be activated or deactivated.</p> <p>If a BREAK monitoring is deactivated during a Break,</p> <p>a) an entry is not made into the diagnostic buffer and the FB-RECV is not triggered with the set ERROR-Bit and the corresponding STATUS entry)</p> <p>b) triggered send requests are output to the user without an error message.</p>	<ul style="list-style-type: none"> • yes • no 	<p>depending on the HW used and the selected operating mode:</p> <p>RS232: • yes</p> <p>TTY: • yes</p> <p>RS422 for R(A)5V/R(B)0V:</p> <ul style="list-style-type: none"> • yes (cannot be deactivated) <p>RS422 for R(A)0V/R(B)5V:</p> <ul style="list-style-type: none"> • no (cannot be activated) <p>For this HW variant RS422, this parameter is directly influenced via the selected initial state of the receive line in the "Interface" folder.</p>

Data Flow Control The table below contains descriptions of the parameters for data flow control. Data flow control is not possible with the RS 485 interface. RTS/CTS data flow control is only possible with the RS 232C interface (see also Table 1-2).

Table 2-12 Data Flow Control (Printer Driver)

Parameter	Description	Value Range	Default Value
Data flow control	Defines which data flow control procedure is used.	<ul style="list-style-type: none"> • None • XON/XOFF • RTS/CTS 	<ul style="list-style-type: none"> • None
XON character ¹	Code for XON character	<ul style="list-style-type: none"> • With 7 data bits²: 0 to 7FH (Hex) • With 8 data bits²: 0 to FFH (Hex) 	<ul style="list-style-type: none"> • 11 (DC1)
XOFF character ¹	Code for XOFF character	<ul style="list-style-type: none"> • With 7 data bits²: 0 to 7FH (Hex) • With 8 data bits²: 0 to FFH (Hex) 	<ul style="list-style-type: none"> • 13 (DC3)
Waiting for XON after XOFF (wait time for CTS=ON) ³	Time for which the CP 340 is to wait for the XON code or for CTS = "ON" from the communication partner before sending.	20 to 655350 ms in 10 ms steps	<ul style="list-style-type: none"> • 2000 ms

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

² Depending on whether you parameterize 7 or 8 data bits for the character frame (see Table 2-11)

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

More Information You will find more information on data flow control with XON/XOFF and RTS/CTS in Section 2.2.4 in the subsection entitled "Data Flow Control/Handshaking."

X27 (RS 422/485)-interface You will find the description of the parameters for the X27 (RS 422) interface in the following table:

Table 2-13 X27 (RS 422/485) interface (ASCII driver)

Parameter	Description	Value Range	Default Value
Initial state of receive line	<p>R(A)5V/R(B)0V: a BREAK recognition is possible for this initial state and cannot be deactivated.</p> <p>R(A)0V/R(B)5V: a BREAK recognition is not possible for this initial state</p>	<ul style="list-style-type: none"> • R(A)5V / R(B)0V • R(A)0V / R(B)5V 	<ul style="list-style-type: none"> • R(A)5V /R(B)0V

Page Layout

The table below contains descriptions of the parameters for the page layout.

Table 2-14 Page Layout (Printer Driver)

Parameter	Description	Value Range	Default Value
Left margin (number of characters)	Number of spaces to precede each line in the body of the text, header or footer. You yourself must ensure that a line is not too long for the printer.	0 to 255	<ul style="list-style-type: none"> • 3
Lines per page (with header and footer)	Number of lines to be printed on each page. The number of printed lines is determined on the basis of the separators output, which means that all header and footer lines must be included in the count.	<ul style="list-style-type: none"> • 1 to 255 • 0 (continuous printing) 	<ul style="list-style-type: none"> • 50
Separators/line end	Characters used to conclude each line in the body of the text, header or footer. The defined separator must be contained in the text, header and footer to be output.	<ul style="list-style-type: none"> • CR (carriage return) • LF (line feed) • CR LF (carriage return and line feed) • LF CR (line feed and carriage return) 	<ul style="list-style-type: none"> • CR LF (carriage return and line feed)
Header lines	Text for up to 2 header and footer lines; a header or footer line is output when the entry field in the parameterization software contains a text or at least a blank. If a text is specified only for the 2nd header or footer line, the 1st header or footer line is automatically padded with a blank and printed.	<ul style="list-style-type: none"> • ASCII characters (text) • %P (conversion statement for outputting a page number) 	-
Footer lines	A blank line is output before and after the headers/footers.	(max. 60 characters)	

Character Set The table below contains descriptions of the parameters for the character set.

Table 2-15 Character Set (Printer Driver)

Parameter	Description	Value Range	Default Value
Printer character set	<p>"IBM" converts the Windows ANSI character set to the printer character set.</p> <p>If you set "User-Defined", you can adapt the character set to include special characters for a particular language.</p>	<ul style="list-style-type: none"> • IBM • User-Defined 	<ul style="list-style-type: none"> • IBM

Control Characters The table below contains a description of the parameter for control characters.

Table 2-16 Control Characters (Printer Driver)

Parameter	Description	Value Range	Default Value
Printer emulation	<p>Sets the printer emulation (printer commands for the following control characters: bold, condensed, expanded and italic type and underlining).</p> <p>If you set "User-Defined", you can modify the printer emulation and include additional control characters. The characters A to Z and a to z are permissible as control characters.</p>	<ul style="list-style-type: none"> • HP DeskJet • HP LaserJet • IBM Proprinter • User-defined 	<ul style="list-style-type: none"> • HP DeskJet

Features Conditions when configuring message texts:

- Size of the text SDB: 8 kbytes
- Max. length of the message texts without variables: 150 characters
- Max. length of the message texts with variables displayed: 250 characters
- Max. number of variables in message texts: 4 (3 + message text number)

Message Texts The following table contains descriptions of the parameters for configuring message texts (using the *CP 340: Point-to-Point Communication, Parameter Assignment*) parameterization interface.

Table 2-17 Message Texts (Printer Driver)

Parameter	Description	Value Range	Default Value
Name of text SDB/text file	The message texts for a CP 340 (serial interface) must be stored in a text SDB for parameterization. You can also store configured message texts in an external text file.	ASCII characters (max. 8 characters)	-
Version number	Version number of the text SDB/text file	0.1 to 255.9	-
Message texts	All the message texts stored in the text block are displayed here together with their message text numbers; you can change a selected message text line by means of the "Edit Message" parameter.	ASCII characters (unchangeable)	-
Edit message	You can transfer message texts edited here to the "Message Texts" list by clicking the "Enter" button.	Message text number: 0 to 999 Message text (max. 150 characters): <ul style="list-style-type: none"> • ASCII characters (text) • Conversion statements (for variables) • Control characters (all those defined in the control character table) 	-
Type style	You can easily assign control characters to text selected in the "Edit Message" entry box by using buttons B to U.	<ul style="list-style-type: none"> • B (bold type) • C (condensed type) • E (expanded type) • I (italic type) • U (underlining) 	-

More Information Section 2.3.5 contains detailed descriptions of the conversion statements and control characters that are possible in message texts and explains their purpose.

2.3.5 Conversion and Control Statements for Printer Output

Introduction

The output of a message text with variables and control statements (e.g. for bold, condensed, expanded or italic type and underlining) is defined by means of a format string.

In the format string you can also define statements to execute other useful functions for printer output (e.g. to set a page number or begin a new page).

All the permissible characters and representation modes for the format string are described below. You can also configure all the control statements (except \F “start new page” and \x “print without page break”) and conversion statements for variables (except for %P “set page number”) in the message texts using the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface.

Format String

Figure 2-17 illustrates the structure of the format string schematically.

A format string can contain normal text and/or conversion statements for variables and/or control statements. Normal text, conversion statements and control statements can occur in any sequence in the format string.

There must be a conversion statement (and only one) for each variable in the format string or message text. The conversion statements are applied to the variables in the sequence in which they occur.

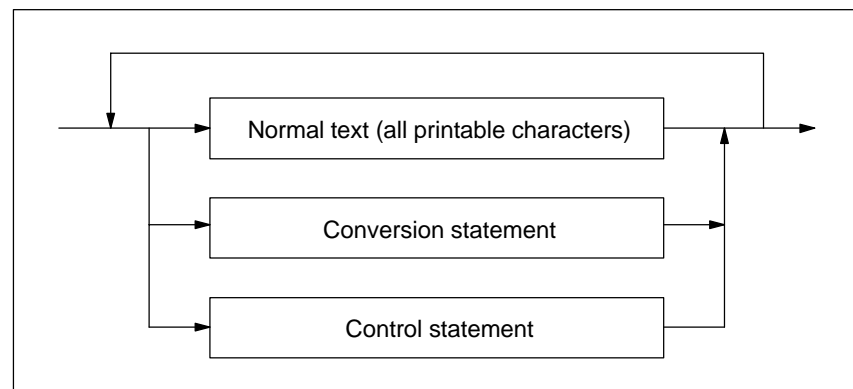


Figure 2-17 Schematic Structure of the Format String

Permissible Characters for Text

The following can be used for text:

- All printable characters
- All characters preceded by \$ at the language interface (ICE 1131-3). The language compilers convert these characters to the corresponding hex code. **Exception:** The character \$N is not permissible.

Example: Carriage return ODH = \$R in the format string

Conversion Statement

Figure 2-18 illustrates the structure of a conversion statement schematically.

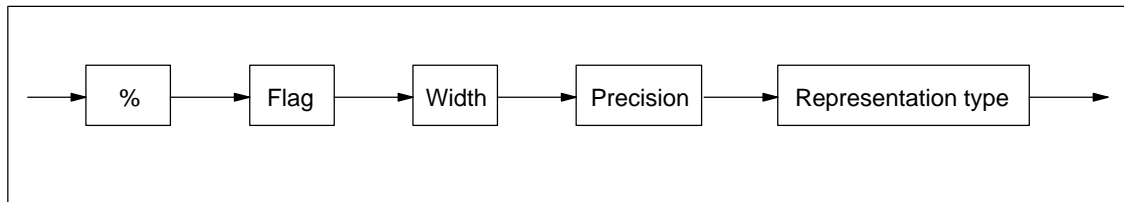


Figure 2-18 Schematic Structure of a Conversion Statement

Flag

Without = Right-justified output
 - = Left-justified output

Width

Without = Standard output representation (see Table 2-18)
 n = Exactly n characters are output (up to a maximum of 255 characters); blanks may be added before (right-justified output) or after (left-justified output)

Precision

Precision is only relevant to representation types A, D, F and R. In all other cases, it is ignored.

Without = Standard output representation (see Table 2-18)

.0 = No output of decimal point and places after the decimal point in the Real (R) and Floating point (F) representation types.

.n = Output of decimal point and n (1 ... 99) significant places after the decimal point in the Real (R) and Floating point (F) representation types. In the case of dates (= representation types A and D), precision relates to the number of digits used for the year. Only 2 and 4 are permitted for dates.

Please note that the precision is always preceded by a period. The period serves to identify it and separate it from the width.

Representation Type

The table below describes the possible representation types of the values of the variables. Representation types N and P are exceptions. They are explained below the table. Both upper- and lower-case characters are permitted for the representation type.

Table 2-18 Representation Types in the Conversion Statement

Representation Type	Associated Data Type	Standard Representation	Width of the Standard Representation	Description
A	DATE, WORD	10.06.1992 (German)	10	German date format
C	CHAR, BYTE WORD DWORD ARRAY OF CHAR ARRAY OF BYTE	A, B AB ABCD ABCDE ... ABCDE ...	1 2 4 - -	Alphanumeric characters
D	DATE, WORD	1996-06-10 (American)	10	ICE data format 1131-3
F	REAL, DWORD	0.123456	8	Floating point, without exponent
H	All data types incl. ARRAY OF BYTE	In accordance with the data type	In accordance with the data type	Hexadecimal format
I	INT, WORD DINT, DWORD	-32767 -2147483647	Max. 6 Max. 11	Integer range
N ¹	WORD (text number)	Message text output	-	Integer 0 to 999
P ²	INT, WORD	Page number	5	-
R	REAL, DWORD	0.12E-04	8	Floating point, without exponent
S	STRING	Text output	-	Text strings
T ¹	TIME, DWORD	2d_3h_10m_5s_250ms	Max. 22	Duration (negative duration is marked by an initial (-) minus sign)
U	BYTE WORD DWORD	255 65535 4294967295	Max. 3 Max. 5 Max. 10	Integer range without plus/minus sign
X	BOOL BYTE WORD DWORD	1 11101100 11001... (16) 11001... (32)	1 8 16 32	Binary representation
Y ³	DATE_AND_TIME_OF_DAY, DT	10.06.1992 - 15:42:59.723	25	Date and time
Z	TIME_OF_DAY DWORD	15:42:59.723	12	Time

¹ If there is no message text number or system time in these representation types, 6 * characters appear in the printout instead (the CP 340 does not keep the time).

All conversion statements except %N are permitted in the message text.

² The P representation type is only permitted in the format string. P is not permitted in the configured message texts.

³ The current time and date must be read first by means of the "READ_CLOCK" system function (SFC 1) and stored in the user memory (flag, data).

Output by Means of Message Text Number (%N)

You use the N representation type when you want to start printing message texts stored on the CP 340. The conversion statement variable contains the number of the message text.

Example: The pressure in the chamber "is falling"

Format string = %N %S

Variable 1 = 17 (message text no. 17: The pressure in the chamber ...)

Variable 2 = Reference to string (string variable: ... is falling)

Note

All conversion statements up to %N and control statements up to "\F" and "\X" within a message text are allowed. An explicit width range at %N restricts the printed length of the referenced message text to the specified width.

Setting the Page Number (%P)

You use the P representation type to change the page number in the printout.

The CP 340 always begins a printout at page 1. This conversion statement allows you to set the page number to a specific value. The conversion statement variable contains the number to be set.

Example: (Setting the page number to 10)

Format string = %P

Variable 1 = 10 (page number: 10)

Note

In the case of the P representation type, there must be no further text, conversion or control statements in the format string.

The P representation type is not permitted in configured message texts.

Notes on the Conversion Statement

Please note the following in relation to conversion statements:

- Whenever a maximum length is specified for the standard representation, the actual output can also be shorter. Example: The output of the integer 10 consists of only 2 characters.
- The length of the data to be printed depends on the length of the variables. For example, in the case of the I representation type a maximum of 6 characters can be output for the INT data type and a maximum of 11 characters for the DINT data type.
- A width "0" is not permissible in conversion statements. This is printed out as "*****" with the valid conversion statement.
- If the specified width is too small, in the case of text-based output (representation types A, C, D, S, T, Y and Z), only the number of characters corresponding to the specified width are output (the output is truncated). In all other cases, * characters are output corresponding to the width.
- Undefined or invalid conversion statements are not executed. This is printed out as "*****" (e.g. representation type missing: %2.2).

The rest of the conversion statement (e.g. everything after the character identified as incorrect) is output. This allows the exact cause of the error to be determined.

- Conversion statements without an associated variable are ignored. Variables for which there is no conversion statement are not output.
- Conversion statements that are not supported in a header or footer are not executed. Instead, they are forwarded to the printer transparently.
- You have to use control statements to specify formatting (line feed, tabs, etc.) in a message text or in the printer output of a long conversion statement.
- If both the format string and the message text contain conversion statements, the format string is expanded first followed by the message text.

Example: Voltage 3 V - Current 2 A

Message text 1 = Voltage %I V

Format string = '%N Current: %I A'

Variable 1 = 1

Variable 2 = 2

Variable 3 = 3

**Examples of
Incorrect
Conversion
Statements**

Here are some examples of incorrect conversion statements:

Example 1: *****.2R

Format string = %303.2R
Variable 1 = 1.2345E6

Error: Invalid width in the R representation type. The maximum permissible value for all representation types is 255.

Example 2: ****

Format string = %4.1I
Variable 1 = 12345 DEC

Error: The selected width was too small for the variable value to be output. The precision is not relevant to representation type I.

Example 3: 96-10-3

Format string = %7.2D
Variable 1 = D#1996-10-31

Error: The format string is formally correct, but the selected width was too small to print the date out fully.

Example 4: *****

Format string = %.3A
Variable 1 = D#1996-10-31

Error: The standard width of representation type A was selected but with invalid precision. The possible values here are 2 and 4.

Example 5: *****

Format string = %3.3
Variable 1 = 12345 HEX

Error: A representation type was not specified.

**Examples of
Correct
Conversion
Statements**

Here are some examples of correct conversion statements:

Example 1:31.10.1996

Format string = %15.4A
Variable 1 = D#1996-10-31

A width of 15 with a precision of 4 (width of the year) and right-justified formatting were selected.

Example 2: 12345.

Format string = %-6I
Variable 1 = 12345 DEC

The selected width was one character greater than the variable value to be output; left-justified formatting.

Example 3: 12d_0h_0m_23s_348ms

Format string = %T
Variable 1 = T#12D23S348MS

The IEC time is in the standard format; unspecified time units are inserted with zeros.

Example 4: 1.234560E+02

Format string = %12.6R
Variable 1 = 123.456

A width of 12 is available to display the whole variable, with the precision (number of places after the decimal point) taking up 6 characters.

Example 5: TEST..

Format string = %-6C
Variable 1 = TEST

Left-justified formatting of the text variable

Control Statements

Control statements are used to achieve specific results in the printout (e.g. underlining).

In addition to the standard control statements (for bold, condensed, expanded or italic type and underlining), you can also use other control characters if you enter them in the control character table of the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface before parameterizing the CP 340 (e.g. K for small capitals in Figure 2-19).

Figure 2-19 illustrates the structure of a control statement schematically.

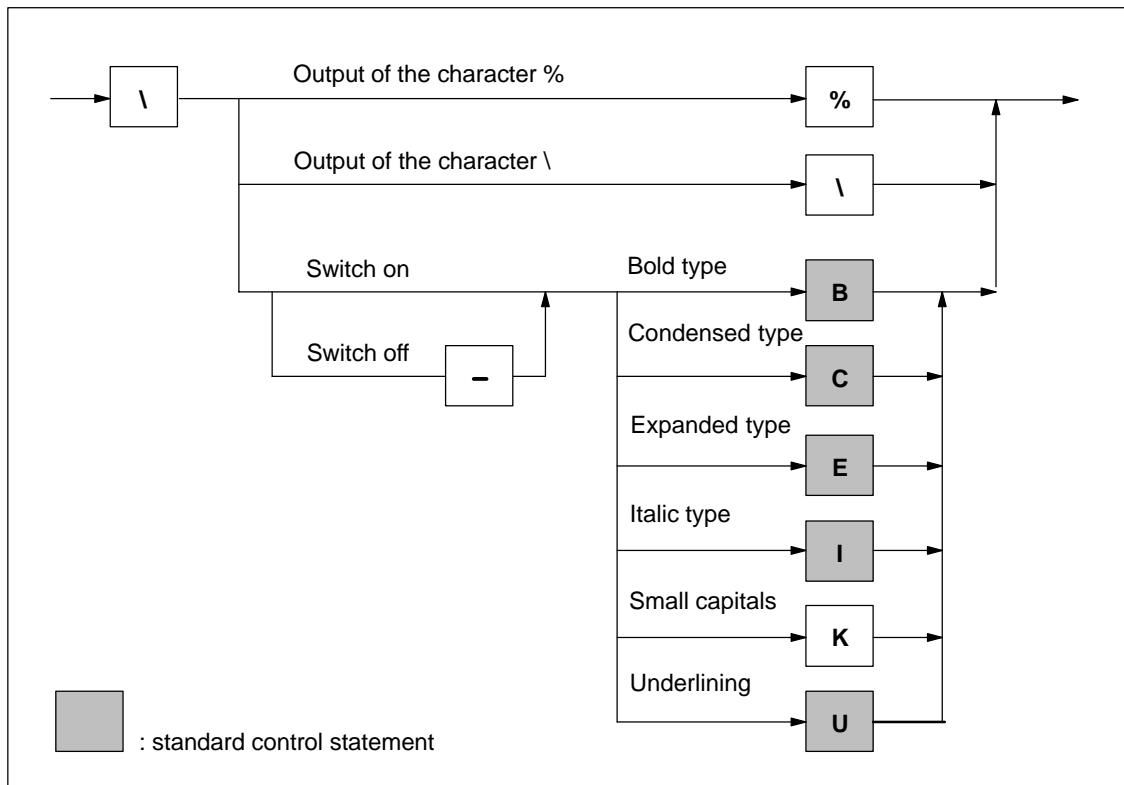


Figure 2-19 Schematic Structure of a Control Statement

Examples

Here are some examples with control statements:

Example 1:

To print the text "**Bold type** and underlining are ways of highlighting a text", you have to enter the following:

```
\Bbold type\B and \Uunderlining\U are ways of highlighting a text
```

Example 2:

To print the format string with the conversion statement "Message text no. %i of %8.2A" transparently, you have to enter the following:

```
'Message text no. \%i of \%8.2A'
```

Beginning a New Page (\F)

Given the parameterized page layout, i.e. the configured headers and footers and the number of lines per page, the \F control statement can be used to begin a new page. This differs from a pure form feed on the printer.

Example: (Beginning a new page)

Format string = \F

Note

In the case of the \F control statement, there must be no further text, conversion or control statements in the format string. The variables remain unassigned.

Printing Without a Line Break (\x)

The CP 340 normally appends the parameterized end-of-line character (CR, LF, CR LF, LF CR) when it sends a message text. The \x control statement cancels the line break after a message text. This means that you can print several messages in a single line in order, for example, to display more variables in a line. The \x control statement is appended at the end of the format string.

Example: The level "200" l was reached at "17.30 " hours. ...

Format string = The level %i l was reached at %Z hours.\x

Variable 1 = Time

Variable 2 = Level

Note

Please note that when you use the \x control statement, the new line always begins without a left margin.

Notes on Control Statements

Please note the following in relation to control statements:

- If the deactivation of an effect is specified without it previously having been activated, or if the output device is incapable of producing the effect, the control statement is ignored.
- The % and \ characters required to define the format string can be printed by means of the control statement.
- Undefined or incorrect control statements are not executed.

3

Starting Up the CP 340

Introduction

Before starting up the CP 340 you will need to perform the following operations in the order given.

1. Mounting the CP 340
2. Configuring the CP 340
3. Parameterizing the CP 340
4. Storing the parameterization data
5. Creating a user program for the CP 340

Mounting the CP 340

Mounting the CP 340 involves inserting it into the mounting rack of your programmable controller.

For a detailed description, see Chapter 4.

Configuring the CP 340

Configuring the CP 340 involves entering it in the configuration table. The CP 340 is configured using the STEP 7 software.

For a detailed description, see Section 5.1.

Parameterization of the CP 340

Parameterizing the CP 340 involves creating the specific parameters of the protocols and configuring message texts for printer output. You use the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface to parameterize the CP 340.

For a detailed description, see Section 5.2.

Storing the Parameterization Data

Storing the parameterization data of the CP 340 involves using the STEP 7 software to save the parameters, load them into the CPU, and transfer them to the CP 340.

For a detailed description, see Section 5.3.

**Creating a User
Program for the
CP 340**

Programming the CP 340 involves configuring it for the associated CPU via the STEP 7 user program. The CP 340 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/.

/1/ *Programming with STEP 7 Manual*

Mounting the CP 340

4

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
4.1	CP 340 Slots	4-2
4.2	Mounting and Dismounting the CP 340	4-3

4.1 CP 340 Slots

Introduction

The following section describes the rules you must observe when positioning the CP 340 in the rack.

Position of the CP 340 in the Rack

The following rules apply when positioning the CP 340 in the rack:

- A maximum of 8 communications modules can be inserted to the right of the CPU.
- The number of pluggable communications modules is limited:
 - by the current consumption of the CP 340 from the S7-300 backplane bus (depending on CPU and IM) and
 - by the expandability of the CPU (for instance CPU 312 IFM in first tier) or of the ET 200M (IM 153) in a distributed configuration (single-tier mounting only).

For further information on slots, see /2/.

Note

The CP 340 can be operated in expansion tiers only from the following CPU versions:

- CPU 314: Version 6
 - CPU 614: Version 6
 - CPU 315: Version 3
 - CPU 315DP: Version 3
-

4.2 Mounting and Dismounting the CP 340

Introduction When mounting and dismantling the CP 340, you must observe certain rules.

Tool For mounting and dismantling the CP 340 you require a 4.5 mm cylindrical screwdriver.

Mounting Sequence To insert the CP 340 in a rack, proceed as follows:

1. Switch the CPU to STOP mode.
2. The CP 340 is accompanied by an expansion bus. Plug this onto the backplane connector of the module to the left of the CP 340.
3. If more modules are to be mounted to the right of the CP 340, plug the expansion bus of the next module onto the right backplane connector of the CP 340.
4. Mount the CP 340 on the rail and tilt it downward.
5. Screw the CP 340 tight.

Dismounting Sequence To dismantle the CP 340 from the rack, proceed as follows:

1. Switch the CPU to STOP mode.
2. Open the front doors.
3. Detach the sub D connector from the integrated interface.
4. Undo the fastening screw on the module.
5. Tilt the module and remove it from the rail, and then remove it from the PLC.

Note

Before you mount or dismount the CP 340, you must switch the CPU to STOP mode. The power supply module does not need to be in POWER OFF mode.

You can plug in or unplug the cable to the integrated submodule on the CP 340 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.

Configuring and Parameterizing the CP 340

5

Section Overview

This chapter is divided into the following sections:

In Section	You Will Find	on Page
5.1	Configuring the CP 340	5-2
5.2	Parameterizing the Communications Protocols	5-3
5.3	Managing the Parameter Data	5-4

Parameterization Options

You configure and parameterize the module variants of the CP 340 using STEP 7 or the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface.

Table 5-1 Configuration Options for the CP 340

Product	Order Number	Parameterized Using the Parameterization Interface	Under STEP 7
CP 340-RS 232C	6ES7 340-1AH00-0AE0*	As of V1.0	As of V2.1
CP 340-RS 232C	6ES7 340-1AH01-0AE0**	As of V3.0	As of V3.0
CP 340-20mA TTY	6ES7 340-1BH00-0AE0**		
CP 340-RS 422/485	6ES7 340-1CH00-0AE0**		

* You can also configure this module using the parameterization interface as of V3.0, but only the drivers integrated in the module firmware can be parameterized (not the printer driver) in this case.

** You can also configure these modules using the parameterization interface V1.0, but only the drivers offered in the parameterization interface can be parameterized (not the printer driver).

5.1 Configuring the CP 340

Introduction	Once you have mounted the CP 340 you must inform the programmable controller that it is there. This process is known as "configuration".
Configuration	<p>In the following, "configuration" refers to the entry of the CP 340 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 340. STEP 7 then automatically assigns an address to the CP 340.</p> <p>The CPU is now able to find the CP 340 in its slot in the rack by way of its address.</p>
Prerequisite	Before you can enter the CP 340 in the configuration table of the STEP 7 software, you must have created a project and a terminal with STEP 7.
Further Information	<p>How to configure S7-300 modules is described in detail in the STEP 7 manual /3/.</p> <p>In addition, the on-line help for STEP 7 provides sufficient support for configuring an S7-300 module.</p>

5.2 Parameterizing the Communications Protocols

- Introduction** Once you have entered the CP 340 in the configuration table, you must supply its interface with parameters. In the case of the printer driver, you can also configure message texts for printer output. This process is known as "parameterization".
- Parameterization** The term "parameterization" is used in the following to describe the setting of protocol-specific parameters and the configuration of message texts. This is done using the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface.
- Double-click on the CP 340 in the configuration table of STEP 7 or mark the CP 340 and select the menu item **Edit > Object Properties** to call the "Properties - CP 340" dialog.
- Here you can set the basic parameter of the CP 340 (see Section 2.3.1). By selecting the "Parameter" button, you can start the *CP 340: Point-to-Point Communication, Parameter Assignment* interface.
- Prerequisite** The *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface must be installed on the programming device/PC under STEP 7 V (see Table 5-1).
- Installation** The *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface, the function blocks and the programming example are stored on a 3.5" diskette. Process as follows to install the parameterization interface:
1. Insert the diskette into the diskette drive of your programming device/PC.
 2. Under Windows 95, start the dialog for installing software by double-clicking on the "software" icon in "Control panel".
 3. In the dialog, select the diskette drive and the **setup.exe** file and start installation.
 4. Now follow the step-by-step instructions of the installation program.
- Further Information** The basic operation of the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface is the same for all communications processors and is self-explanatory. 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

Introduction The configuration and parameterization data of the CP 340 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 the Configuration and Parameters You can now load the configuration and parameterization data on-line from the programming device onto the CPU (menu item **PLC > Download**). The CPU puts the parameters into effect as soon as they are loaded.

The module parameters are automatically transmitted to the CP 340

- when they are loaded onto the CPU and as soon as the CP 340 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 STEP 7 user manual /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 the parameters.

Communication via Function Blocks

6

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
6.1	Communication via Function Blocks	6-2
6.2	Overview of the Function Blocks	6-3
6.3	Using the Function Blocks	6-4
6.4	Parameterizing the Function Blocks	6-21
6.5	General Information on Program Processing	6-25
6.6	Technical Specifications of the Function Blocks	6-26

6.1 Communication via Function Blocks

Introduction	Communication between the CPU, the CP 340 and a communication partner takes place via the function blocks and the protocols of the CP 340.
Communication between CPU and CP 340	The function blocks form the software interface between the CPU and the CP 340. They must be called up cyclically from the user program.
Communication between CP 340 and a Communication Partner	<p>The protocol conversion takes place on the CP 340. Via the protocol (3964(R) procedure, ASCII driver or printer driver) the interface of the CP 340 is adjusted to suit the interface of the communication partner.</p> <p>This enables the CP 340 to be linked to all communication partners which support the standard protocols (3964(R) procedure, ASCII driver or printer driver).</p>

6.2 Overview of the Function Blocks

Introduction The S7-300 programmable controller provides you with a number of function blocks which initiate and control communication between the CPU and the CP 340 communications processor in the user program.

Function Blocks / Functions of the CP 340 The table below shows the function blocks / functions of the CP 340 and their meanings.

Table 6-1 Function Blocks / Functions of the CP 340

FB/FC	Meaning
FB 2 P_RCV	The P_RCV function block allows you to receive data from a communication partner and store it in a data block.
FB 3 P_SEND	The P_SEND function block allows you to send all or part of a data block to a communication partner.
FB 4 P_PRINT	The P_PRINT function block allows you to output a message text containing up to 4 variables to a printer.
FC 5 V24_STAT	The V24_STAT function allows you to read the signal states at the RS 232C interface of the CP 340-RS 232C.
FC 6 V24_SET	The V24_SET function allows you to set/reset the outputs at the RS 232C interface of the CP 340-RS 232C.

Scope of Supply and Installation The function blocks of the CP 340, together with the parameterization interface and the programming example, are supplied on a 3.5" diskette 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 following library:

CP340

Open the library using the STEP 7 SIMATIC Manager by calling the menu command **File > Open > Library** under CP Ptp\CP 340\Blocks.

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

6.3 Using the Function Blocks

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

6.3.1 Connection to a Communication Partner

Introduction The following function blocks are available for linking with a communication partner: P_SEND (FB 3) for sending data, and P_PRCV (FB 2) for receiving data in the cyclical program.

Requests which can be Processed Simultaneously Only one FB P_SEND and one FB P_RCV may be inserted in the user program for each CP 340 used. In addition, you may only program one instance data block for the FB P_SEND and the FB P_RCV, since the instance data blocks contain the necessary status information for the internal FB sequence.

S7 Sends Data to a Communication Partner The P_SEND FB transmits a subframe from a data block, specified by the parameters DB_NO, DBB_NO and LEN, to the CP 340. The P_SEND FB is called statically (without conditions) for data transmission in the cycle or **alternatively** in a time-controlled program. 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_SEND FB can be called in the cycle with the signal state "1" at the parameter input R. This aborts the transmission to the CP 340 and sets the P_SEND FB back to its initial state. Data that has already been received by the CP 340 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 340 to be addressed. 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 there were no errors, STATUS has the value "0". DONE and ERROR/STATUS are also output when the P_SEND FB is reset (see Fig. 6-1). 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 function block P_SEND does not have a parameter check, that is if there are invalid parameters, the CPU branches to the STOP mode.

Before the CP 340 can process an activated request after the CPU has changed from STOP to RUN mode, the CP-CPU start-up mechanism of the P_SEND FB must be completed (see Section 6.5) Any requests initiated in the meantime do not get lost, but are transmitted once the start-up coordination with the CP 340 is finished.

What to Do

Block call

STL representation		LAD representation	
CALL	P_SEND, I_SEND	I_SEND	
REQ:	=	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> P_SEND </div>	
R:	=	— EN	ENO —
LADDR:	=	— REQ	DONE —
DB_NO:	=	— R	ERROR —
DBB_NO:	=	— LADDR	STATUS —
LEN:	=	— DB_NO	
DONE:	=	— DBB_NO	
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_SEND FB works with an instance DBI_SEND of 40 bytes, whose number is specified in the call. 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.

**FB 3 P_SEND
Parameters**

The table below lists the parameters of P_SEND (FB 3).

Table 6-2 FB 3 P_SEND Parameters

Name	Type	Data Type	Description	Permitted Values, Comment
REQ	INPUT	BOOL	Initiates request with positive edge	
R	INPUT	BOOL	Aborts request	Current request is aborted and sending is blocked.
LADDR	INPUT	INT	Basic address of CP 340	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 ≤ DBB_NO ≤ 8190 Transmitted data as of data word
LEN	INPUT	INT	Data length	1 ≤ LEN ≤ 1024, specified in number of bytes
DONE ¹	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
STATUS	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details

¹ After a correct send request, the DONE parameter is available for **one** CPU cycle!

Time Sequence Chart for FB 3 P_SEND

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

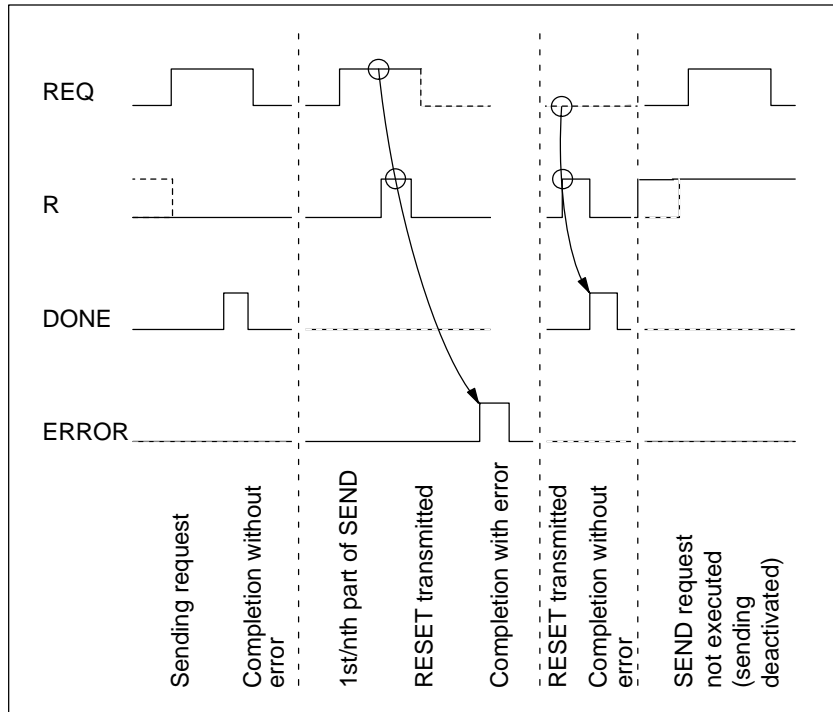


Figure 6-1 Time Sequence Chart for FB 3 P_SEND

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**

The P_RCV FB transmits data from the CP 340 to an S7 data area specified by the parameters DB_NO, DBB_NO and LEN. The P_RCV FB is called statically (without conditions) for data transmission in the cycle or alternatively in a time-controlled program.

With the (static) signal state "1" at parameter EN_R, the software checks whether data can be read by the CP 340. 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 FB is set to the initial state. Receiving is deactivated as long as the R parameter shows the signal state "1". If signal state "0" returns, the aborted message frame is received again from the beginning.

The LADDR parameter defines the CP 340 to be addressed.

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 error number is displayed under STATUS (see Section 8.3). If there were no errors, STATUS has the value "0". NDR and ERROR/STATUS are also output when the P_RCV FB is reset (parameter LEN == 16#00) (see Fig. 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 function block P_RCV does not have a parameter check, i.e. if there are invalid parameters, the CPU can branch to STOP mode.

Before the CP 340 can receive a request after the CPU has changed from STOP to RUN mode, the CP-CPU start-up mechanism of the P_RCV FB must be completed (see Section 6.5).

What to Do

Block call

STL representation		LAD representation	
CALL	P_RCV, I_RCV	I_RCV	
EN_R:	=	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> P_RCV EN ENO EN_R NDR R ERROR LADDR LEN DB_NO STATUS DBB_NO </div>	
R:	=	—	—
LADDR:	=	—	—
DB_NO:	=	—	—
DBB_NO:	=	—	—
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 FB works with an instance DBI_RCV of 40 bytes, whose number is specified in the call. 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.

**FB 2 P_RCV
Parameters**

The table below lists the parameters of P_RCV (FB 2).

Table 6-3 FB 2 P_RCV Parameters

Name	Type	Data Type	Description	Permitted Values, Comment
EN_R	INPUT	BOOL	Enables data read	
R	INPUT	BOOL	Aborts request	Active request is aborted and receiving is blocked.
LADDR	INPUT	INT	Basic address of CP 340	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	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details
LEN	OUTPUT	INT	Length of message frame received	$1 \leq \text{LEN} \leq 1024$, specified in number of bytes
STATUS	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details.

¹ After a correct receive request, the NDR parameter is available for **one** CPU cycle!

Time Sequence Chart for FB 2 P_RCV

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

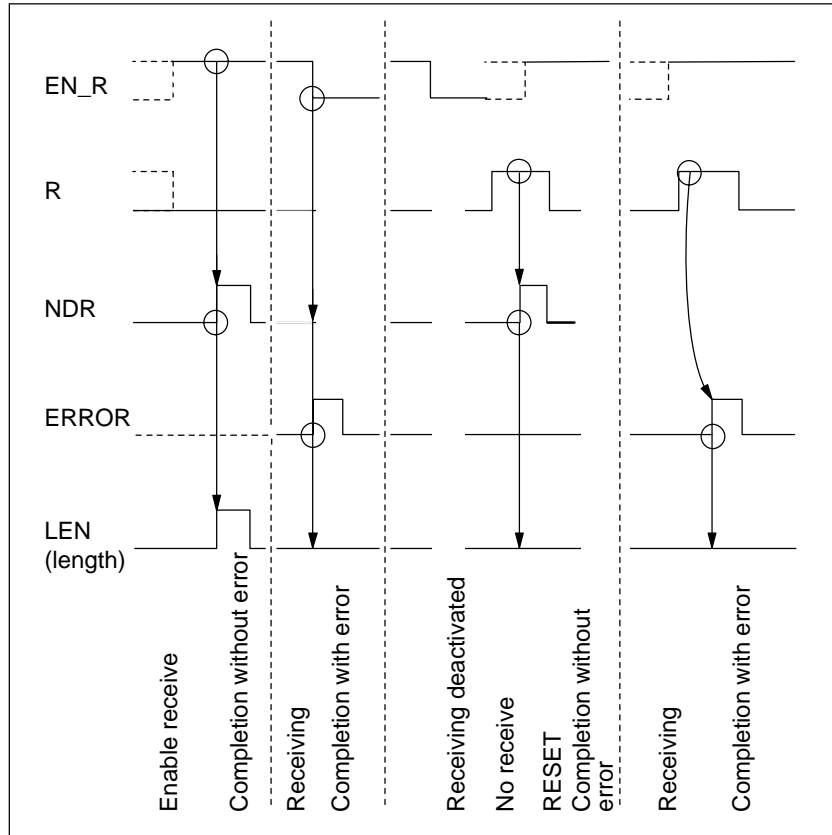


Figure 6-2 Time Sequence Chart for FB 2 P_RCV

Note

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

6.3.2 Outputting Message Texts to a Printer

Introduction

The P_PRINT function block (FB 4) is available to you for outputting message texts to a printer. The P_PRINT function block (FB 4) sends a process message to the CP340, for example. The CP340 logs the process message on the connected printer.

Outputting Message Texts

The P_PRINT FB sends a message text containing up to four variables to the CP340. You configure the message texts using the *CP340: Point-to-Point Communication, Parameter Assignment* parameterization interface (see Sections 2.2.4 and 2.3.5). The P_PRINT FB is called statically (without conditions) for data transmission either in the cycle or in a time-controlled program.

The pointers (to data blocks) for the format string and the four variables can be reached by means of the DB_NO and DBB_NO parameters. The pointers must be stored without gaps and in a specific sequence in the parameterized data block. This is the pointer DB (see Figure 6-3).

The transmission of the message text is initiated by a positive edge at the REQ input. The format string of the message text is sent first, followed by variables 1 to 4.

Depending on the volume of data involved, a data transmission may run over several calls (program cycles).

The P_PRINT 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 340 and sets the P_PRINT FB back to its initial state. Data already received by the CP 340 is sent to the communication partner. If a signal state of "1" is applied statically at the R input, the transmission of print requests is deactivated.

The LADDR parameter specifies the address of the CP 340 to be addressed. 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 STATUS (see Section 8.3). If there were no errors, STATUS has a value of 0. DONE and ERROR/STATUS are also output when the P_PRINT FB is reset (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 a status of "1".

Note

The P_PRINT 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 340 can process an activated request after the CPU has changed from STOP to RUN mode, the CP CPU start-up mechanism of the P_PRINT FB must be completed (see Section 6.5). Any requests initiated in the meantime do not get lost. They are transmitted to the CP 340 once the start-up coordination is completed.

What to Do

Block Call

STL Representation		LAD Representation	
CALL	P_PRINT, I_PRINT	I_PRINT	
REQ:	=	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> P_PRINT EN ENO REQ DONE R ERROR LADDR STATUS DB_NO DBB_NO </div>	
R:	=	—	—
LADDR:	=	—	—
DB_NO:	=	—	—
DBB_NO:	=	—	—
DONE:	=	—	—
ERROR:	=	—	—
STATUS:	=	—	—

Note

The EN and ENO parameters 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, Instance DB

The P_PRINT FB works with an I_PRINT instance DB of 39 bytes, whose DB number is specified in the call. 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 symbolic access to the instance DB.

Assignment in the Data Area, Pointer DB

The P_PRINT FB uses the DB_NO and DBB_NO parameters to access a pointer DB in which the pointers to the data blocks containing the message texts and variables are stored in a fixed order. You have to create the pointer DB.

Figure 6-3 shows the structure of the pointer DB addressed by means of the DB_NO and DBB_NO parameters of the P_PRINT DB.

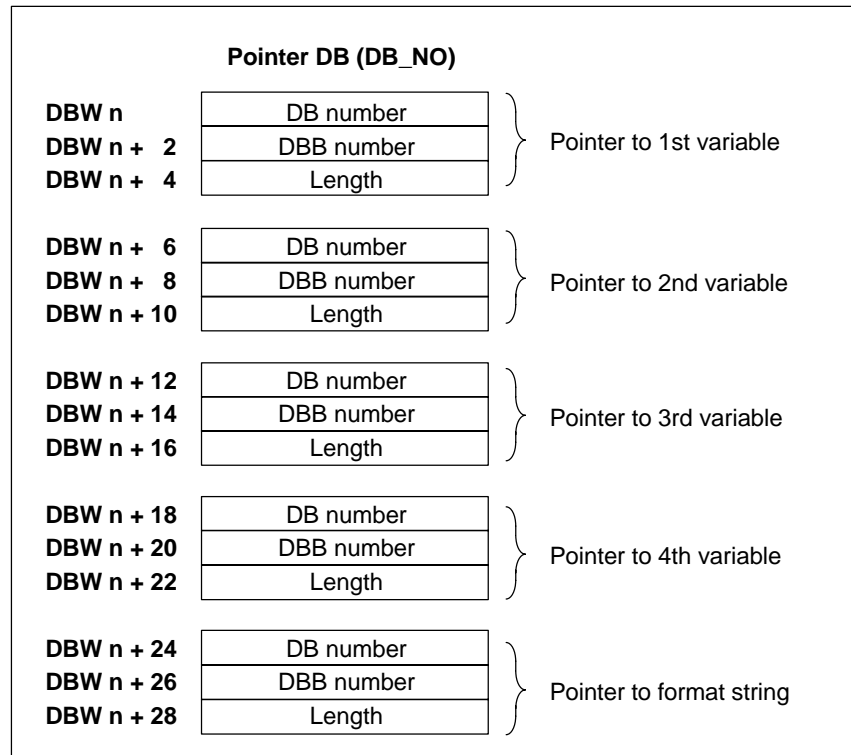


Figure 6-3 Structure of the Pointer DB for the P_PRINT Function Block (FB)

Permissible DB Number

The permissible DB numbers are CPU-specific. If a DB number of 16#00 is specified for a pointer to a variable, this variable is interpreted as not being there and the pointer is set on the next variable or the format string.

If the DB number for the pointer to the format string is 16#00, the print request is aborted and the event number 16#1E43 displayed at the STATUS parameter output of the P_PRINT FB.

Permissible DBB Number

The variable or format string begins at the parameterized DBB number. The variables can have a maximum length of 32 bytes, and the format string can have a maximum length of 150 bytes.

If the maximum length is exceeded, the print request is aborted and the event number 16#1E41 displayed at the STATUS parameter output of the P_PRINT FB.

Permissible Length

The entry length in the pointer DB is to be set for each display type (data type) independently from the precision used.

**FB 4 P_PRINT
Parameters**

The table below lists the parameters of the P_PRINT function block (FB 4).

Table 6-4 FB 4 P_PRINT Parameters

Name	Type	Data Type	Description	Permitted Values, Comment
REQ	INPUT	BOOL	Initiates request at positive edge	
R	INPUT	BOOL	Aborts request	Current request is aborted, and printer output is blocked.
LADDR	INPUT	INT	Basic address of the CP 340	The basic address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Pointer to pointer DB ¹ : CPU-specific (zero not permitted)
DBB_NO	INPUT	INT	Data byte number	0 ≤ DBB_NO ≤ 8162 Pointer as of data byte
DONE ²	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR	OUTPUT	BOOL	Request completed with errors	STATUS parameter contains error details.
STATUS	OUTPUT	WORD	Error specification	If ERROR == 1, STATUS parameter contains error details.

¹ The pointers to variables and format string are stored in the pointer DB in a fixed order (see Figure 6-3).

² The DONE parameter is available for a **single** CPU cycle after the send request has been completed correctly.

**Time Sequence
Chart for FB
4 P_PRINT**

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

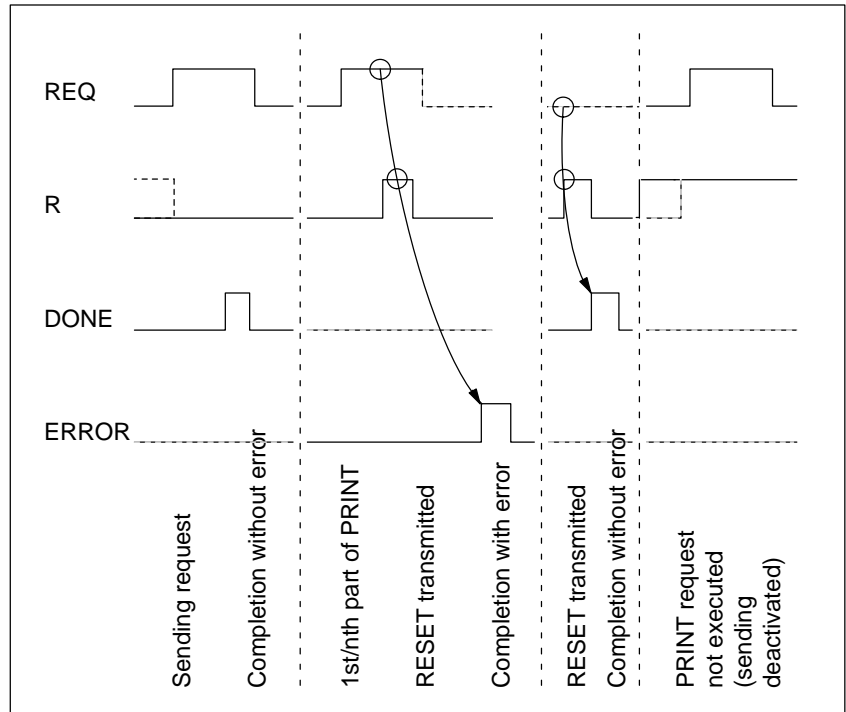


Figure 6-4 Time Sequence Chart for FB 4 P_PRINT

Note

The REQ input is edge-triggered. A positive edge at the REQ input is enough to trigger it. It need not have the signal state "1" during the entire transmission.

6.3.3 Reading and Controlling the RS 232C Secondary Signals

Introduction

The functions available for reading and controlling the RS 232C secondary signals are V24_STAT (FC 5) for checking the interface statuses and V24_SET (FC 6) for setting/resetting the interface outputs.

Checking the Interface Statuses of the CP 340

The V24_STAT FC reads the RS 232C secondary signals from the CP 340 and makes them available to the user in the block 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 340 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 340 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.

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 function V24_STAT 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 340 and the response time of the communication partner.

FC 5 V24_STAT Parameters

The table below lists the parameters of the V24_STAT function (FC 5).

Table 6-5 FC 5 V24_STAT Parameters

Name	Type	Data Type	Description	Permitted Values, Comment
LADDR	INPUT	INT	Basic address of CP 340	The basic address is taken from STEP 7.
DTR_OUT	OUTPUT	BOOL	Data terminal ready, CP 340 ready for operation	(CP 340 output)
DSR_IN	OUTPUT	BOOL	Data set ready, communication partner ready for operation	(CP 340 input)
RTS_OUT	OUTPUT	BOOL	Request to send, CP 340 ready to send ¹	(CP 340 output)
CTS_IN	OUTPUT	BOOL	Clear to send, communication partner can receive data from the CP 340 (response to RTS = ON of the CP 340) ¹	(CP 340 input)
DCD_IN	OUTPUT	BOOL	Data Carrier Detect, receive signal level	(CP 340 input)
RI_IN	OUTPUT	BOOL	Ring Indicator, pole signal	(CP 340 input)

¹ For further information on this RS 232C secondary signal, see Section 2.2.3.

Setting/Resetting Interface Outputs of the CP 340

The user can set or reset the interface outputs via the corresponding parameter inputs of the V24_SET FC. 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 340 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.

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 V24_SET function does not occupy any data areas.

V24_SET (FC 6) Parameters The table below lists the parameters of the V24_SET function (FC 6).

Table 6-6 FC 6 V24_SET Parameters

Name	Type	Data Type	Description	Permitted Values, Comment
LADDR	INPUT	INT	Basic address of CP 340	The basic address is taken from STEP 7.
RTS	INPUT	BOOL	Request to send , CP 340 ready to send ¹	(Control CP 340 output)
DTR	INPUT	BOOL	Data terminal ready , CP 340 ready for operation ¹	(Control CP 340 output)

¹ For further information on the RS 232C secondary signals, see Section 2.2.3.

6.4 Parameterizing the Function Blocks

Introduction

This section is aimed at users who are upgrading from SIMATIC S5 to SIMATIC S7. The following subsections describe what you need to observe when 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.

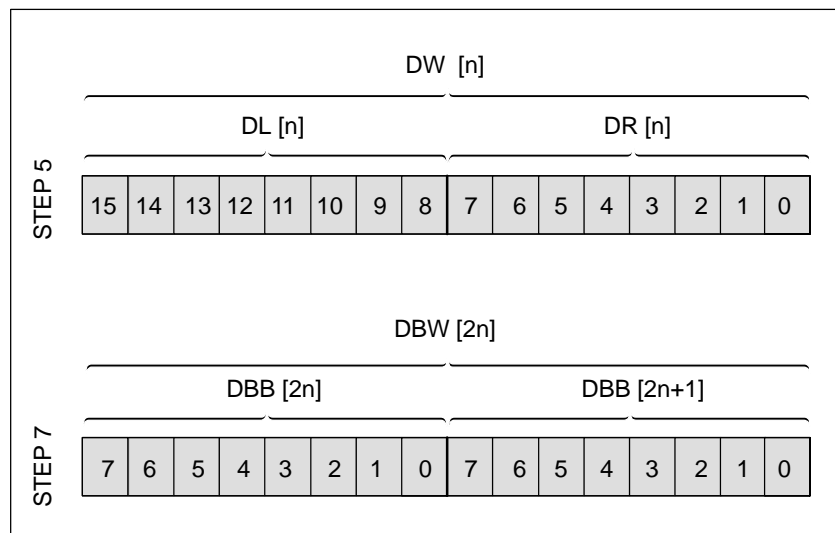


Figure 6-5 Contrasting Data Addressing in STEP 5 and STEP 7

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

Calling FB 3 with direct parameterization:

STL	Explanation
Network 1:	
CALL FB 3, DB3	
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
DONE := M 26.0	//Terminated without error
ERROR := M 26.1	//Terminated with error
STATUS := MW 27	//Status word

Example of Indirect Parameterization

Calling FB 3 with indirect parameterization:

STL	Explanation
Network 1:	
CALL FB 3, DB3	
REQ := M 0.6	//Activates SEND
R := M 5.0	//Activates RESET
LADDR := MW21	//Basic address in MW21
DB_NO := MW40	//DB no. in MW40
DBB_NO := MW42	//DBB no. in MW42
LEN := MW44	//Length in MW44
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 3 with absolutely addressed actual operands:

STL	Explanation
Network 1:	
CALL FB 3, DB3	
REQ := M 0.6	//Activates SEND
R := M 5.0	//Activates RESET
LADDR := MW21	//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
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 3 with symbolically addressed actual operands:

STL	Explanation
Network 1:	
CALL FB 3, DB3	
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
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 Behavior: PLC-CP 340 connection	The parameterization data is generated using the <i>CP 340: Point-to-Point Communication, Parameter Assignment</i> parameterization interface and transmitted to the CPU with the STEP 7 software. Each time the CPU is started up, the current parameters are transferred to the CP 340 by the system service of the CPU.
Start-up Behavior: FB-CP 340 connection	<p>Once the connection between the CPU and the CP 340 has been established, the CP 340 must be initialized.</p> <p>For each function block, P_SEND, P_RCV, P_PRINT, there is a separate start-up coordination. Before requests can be actively processed, the accompanying start-up procedure must be completed.</p>
Disable Interrupts	In the function blocks the interrupts are not disabled.
Interrupt Behavior	<p>The CP 340 module can trigger a diagnostics alarm in the CPU. When this happens, the operating system provides the user with 4 bytes of alarm information. Interpretation of the alarm information must be programmed by the user (OB 82).</p> <p>The CP 340 function blocks cannot be called in the process or diagnostics interrupt program.</p>
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 340 function blocks / functions.

Table 6-7 Memory Requirements of the Function Blocks / Functions (in byte)

Block	Name	Version	Load Memory	Work Memory	Local Data
FB 2	P_RCV	1.0	1776	1410	28
FB 3	P_SEND	1.0	1706	1352	28
FB 4	P_PRINT	1.0	1966	1586	32
FC 5	V24_STAT	1.0	188	72	2
FC 6	V24_SET	1.0	148	40	2

Runtimes The table below shows the runtimes of the CP 340 function blocks/functions.

Table 6-8 Runtimes of the Function Blocks / Functions in μ s

Block	Name	Version	Function	CPU 312 IFM	CPU 313/ CPU 314	CPU 315/ CPU 315DP	CPU 614
FB 2	P_RCV	1.0	Idling	450	250	230	210
			Polling receive	480	300	260	230
			Receiving*	2700	2200	1900	1700
FB 3	P_SEND	1.0	Idling	420	250	230	200
			Sending*	2800	2300	2000	1800
FB 4	P_PRINT	1.0	Idling	450	250	230	200
			Sending*	2800	2300	2000	1800
FC 5	V24_STAT	1.0	Reading RS 232C secondary signals	160	150	120	100
FC 6	V24_SET	1.0	Setting RS 232C secondary signals	150	140	110	90

* The runtimes are based on a block size of 1 to 14 bytes.

Minimum Number of CPU Cycles

The table below describes the minimum number of CPU cycles (FB/FC calls) required to process a "minimum request" (14 bytes SEND, 13 bytes RECEIVE for the transported user data set per program cycle). This only applies in centralized operation.

Table 6-9 Minimum Number of CPU Cycles

	Number of CPU Cycles for Processing ...		
	Termination without Error	Termination with Error	RESET/RESTART
RECEIVE	≥ 3	≥ 3	≥ 4
SEND	≥ 3	≥ 3	≥ 4
PRINT	≥ 3	≥ 3	≥ 4
V24_STAT	1	-	-
V24_SET	2	>> 2	-

Before the CP 340 can process an activated request after the CPU has changed from STOP to RUN mode, the CP-CPU start-up mechanism P_SEND or P_PRINT must be completed. Any requests initiated in the meantime do not get lost, but are transmitted once the start-up coordination with the CP 340 is finished.

Before the CP 340 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 RCV 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 340

7

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
7.2	Operating Modes of the CP 340	7-2
7.1	Start-up Characteristics of the CP 340	7-3
7.3	Behavior of the CP 340 on Operating Mode Transitions of the CPU	7-4

7.1 Operating Modes of the CP 340

Introduction	The CP 340 has the operating modes STOP, reparameterization and RUN.
STOP	<p>When the CP 340 is in STOP mode, no protocol driver is active and all send and receive requests from the CPU are given a negative acknowledgment.</p> <p>The CP 340 remains in STOP mode until the cause of the stop is removed (e.g. break, invalid parameter).</p>
Reparameterization	<p>For reparameterization, the protocol driver is initialized. The SF LED is on during reparameterization.</p> <p>Sending and receiving are not possible, and send and receive message frames stored in the CP 340 are lost when the driver is restarted. Communication between the CP and the CPU is started afresh (active message frames are aborted).</p> <p>At the end of the reparameterization, the CP 340 is in RUN mode and is ready to send and receive.</p>
RUN	The CP 340 processes the send requests from the CPU and provides the message frames received by the communications partner to be fetched by the CPU.

7.2 Start-up Characteristics of the CP 340

Introduction

The CP 340 start-up is divided into two phases:

- Initialization (CP 340 in POWER ON mode)
- Parameterization

Initialization

As soon as the CP 340 is connected to the power supply, the serial interface is supplied with default parameters (the interface parameters are given preset values at the factory) of the module.

As soon as the initialization is complete, the CP 340 automatically starts up with the 3964R driver with block check by default. The CP 340 is now ready for operation.

Parameterization

During parameterization the CP 340 receives the module parameters assigned to the current slot, which were generated using the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface.

Reparameterization is performed. The default parameters are overwritten by the newly set module parameters.

7.3 Behavior of the CP 340 on Operating Mode Transitions of the CPU

- Introduction** Once the CP 340 has been started up, all data is exchanged between the CPU and the CP 340 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 RS 232C interface of the CP 340-RS 232C is continued with the ASCII driver and the printer 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** At start-up, the CPU sends off the parameters generated with the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface. The CP 340 only reparameterizes if the parameters have changed.
- Through appropriate parameterization with the parameterizing software, you can have the receive buffer on the CP 340 deleted automatically at CPU start-up.
- 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 340 and the corresponding FBs are synchronized. No new P_SEND, P_RCV or P_PRINT FB is executed until this is finished.

**Points to Note
when Sending
Message Frames,
Printer Output**

Transmission of message frames and printer output are possible only in CPU RUN mode.

If the CPU switches to STOP mode during CPU > CP data transmission, the P_SEND or P_PRINT FB reports the error "current program interrupted, request aborted due to BREAK/restart/reset" after restart.

Note

The CP 340 does not send data to the communications partner until it has received all data from the CPU.

**Points to Note
when Receiving
Message Frames**

The *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface allows you to parameterize "delete CP receive buffer at start-up = yes/no".

- If you select "yes", the receive buffer on the CP 340 is automatically deleted when the CPU mode changes from STOP to RUN.
- If you select "no", as many message frames as you have parameterized are stored in the CP 340 receive buffer (1 to 250).

If the CPU changes to STOP mode during transmission CP > CPU, the P_RCV FB reports the error "current program interrupted, request aborted due to BREAK/restart/ reset" after restart. If "delete CP receive buffer at start-up = no" is set, the message frame is retransmitted from the CP 340 to the CPU.

For further information on receive buffers, see Chapter 2.

Diagnostics with the CP 340

8

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
8.1	Diagnostics Functions of the CP 340	8-2
8.2	Diagnosis via the Display Elements of the CP 340	8-3
8.3	Diagnostics Messages of the Function Blocks P_SEND, P_RCV and P_PRINT	8-4
8.4	Diagnosis via the S7-300 Backplane Bus	8-12
8.5	Diagnosis by means of the Diagnostic Buffer of the CP 340	8-15

8.1 Diagnostics Functions of the CP 340

Introduction

The diagnostics functions of the CP 340 enable you to quickly localize any errors which occur. The following diagnostics options are available:

- Diagnosis via the display elements of the CP 340
- Diagnosis via the STATUS output of the function blocks
- Diagnosis via the S7-300 backplane bus
- Diagnosis via the diagnostic buffer of the CP 340

Display Elements (LED)

The display elements show the operating mode or possible error states of the CP 340. The display elements give you an initial overview of any internal or external errors as well as interface-specific errors (see Section 8.2).

STATUS Output of the FBs

The P_SEND, P_RCV and P_PRINT 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 340 also enters the diagnostic events at the STATUS output in its diagnostic buffer.

S7-300 Backplane Bus

The CP 340 can trigger a diagnostics alarm on the CPU assigned to it. The CP 340 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 (see Section 8.4).

The CP 340 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.

Diagnostic Buffer of the CP 340

All the CP 340's 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).

8.2 Diagnosis via the Display Elements of the CP 340

Introduction

The display elements of the CP 340 provide information on the CP 340. The following display functions are distinguished:

- **Group alarm LED**
 - SF (red) Fault has occurred or reparameterization
- **Special LEDs**
 - TXD (green) Sending active; lights up when the CP 340 is sending user data via the interface
 - RXD (green) Receiving active; lights up when the CP 340 is receiving user data via the interface

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 340, 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 errors
- BREAKs (receive cable between CP 340 and communications partner becomes disconnected). The BREAK display on the Group alarm LED SF only occurs if the BREAK monitoring was not deactivated with the parameter assignment interface.

Note

In the event of a BREAK on the RXD line, the group alarm LED SF **and** the special LED RXD light up.

Event Classes The table below describes the various event classes and numbers.

Table 8-1 Event Classes and Event Numbers

Event Class 5 (05H): "Error while processing CPU request"		
Event No.	Event	Remedy
(05)02H	Request not permitted in this operating mode of CP (e.g. device interface not parameterized).	Analyze diagnostics alarm and recover error accordingly.
(05)05H	With printer driver only: System data block with message texts not available on the CP	Use the parameterization software to configure the message texts, and then carry out a restart.
(05)06H	With printer driver only: Message text not available	Use the parameterization software to configure the message texts, and then carry out a restart.
(05)07H	With printer driver only: Message text too long	Change the message text so that it has no more than 150 characters (or no more than 250 characters if it contains variables)
(05)08H	With printer driver only: Too many conversion statements	You have configured more conversion statements than variables. The conversion statements without associated variables are ignored.
(05)09H	With printer driver only: Too many variables	You have configured more variables than conversion statements. Variables for which there is no conversion statement are not output.
(05)0AH	With printer driver only: Unknown conversion statement	Check the conversion statement. Undefined or unsupported conversion statements are replaced in the printout with *****.
(05)0BH	With printer driver only: Unknown control statement	Check the control statement. Undefined or incorrect control statements are not executed. The control statement is not output as text either.
(05)0CH	With printer driver only: Conversion statement not executable	Check the conversion statement. Conversion statements that cannot be executed appear in the printout in accordance with the defined width and the valid part of the conversion statement or in the standard representation with * characters.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(05)0DH	With printer driver only: Width in conversion statement too small or too great	Correct the specified width of the variable in the conversion statement on the basis of the variable's maximum number of characters in text-based representation types (A, C, D, S, T, Y, Z). Only as many characters as will fit in the specified width appear in the printout; the text is truncated to this width. In all other cases, * characters are output corresponding to the width.
(05)0EH	With 3964(R) and ASCII driver only: Invalid message frame length	The message frame is > 1024 bytes in length. The rest of the message frame (> 1024 bytes) is received by the CP 340, and the first part of the message frame is thus rejected. Select a smaller message frame length.
(05)1BH	With printer driver only: Precision invalid	Correct the specified precision in the conversion statement. The precision is always preceded by a period to identify it and separate it from the width (e.g. ".2" to output a decimal a point and two places after the decimal point). Precision is only relevant to representation types F, R A, and D. In all other cases, it is ignored.
(05)1CH	With printer driver only: Variable invalid (Variable length incorrect/incorrect type)	Correct the specified variable. Table 2-18 indicates the data types possible for each representation type.
Event Class 7 (07H): "Send error"		
Event No.	Event	Remedy
(07)01H	Only with 3964(R): Sending the first repetition: <ul style="list-style-type: none"> An error was recognized when sending the the telegram or the partner requested a repetition via a negative acknowledgment (NAK). 	A repetition is not an error, however, it can be an indication that there are disturbances on the transmission line or a malfunction of the partner device. If the telegram can still not be transmitted after the maximum number of repetition, an error number which describes the error that first occurred is sent.
(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 using interface test device (FOXPG) which is switched into the transmission line.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(07)03H	<p>With 3964(R) only:</p> <p>Acknowledgment delay time exceeded:</p> <p>After STX was sent, no response came from partner within acknowledgment delay time.</p>	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 using interface test device (FOXPG) which is switched into the transmission line.
(07)04H	<p>With 3964(R) only:</p> <p>Termination by partner:</p> <p>During current send operation, one or more characters were received by partner.</p>	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. Check for malfunction at partner device, possibly using interface test device (FOXPG) which is switched into the transmission line.
(07)05H	<p>With 3964(R) only:</p> <p>Negative acknowledgment during sending</p>	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. Check for malfunction at partner device, possibly using interface test device (FOXPG) which is switched into the transmission line.
(07)06H	<p>With 3964(R) only:</p> <p>Error at end of connection:</p> <ul style="list-style-type: none"> • 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. Check for malfunction at partner device, possibly using interface test device (FOXPG) which is switched into the transmission line.
(07)07H	<p>With 3964(R) only:</p> <p>Acknowledgment delay time exceeded at end of connection or response monitoring time exceeded after send message frame:</p> <p>After connection release with DLE ETX, no response received from partner within acknowledgment delay time.</p>	Partner device faulty or too slow. Check this, possibly using interface test device (FOXPG) which is switched into the transmission line.
(07)08H	<p>With ASCII driver and printer driver only:</p> <p>The waiting time for XON or CTS = ON has elapsed.</p>	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)0BH	<p>With 3964(R) only:</p> <p>Initialization conflict cannot be solved because both partners have high priority.</p>	Change parameterization.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(07)0CH	<p>With 3964(R) only:</p> <p>Initialization conflict cannot be solved because both partners have low priority.</p>	Change parameterization.
Event Class 8 (08H): ”Receive error”		
Event No.	Event	Remedy
(08)01H	<p>Only with 3964(R):</p> <p>Expecting the first repetition:</p> <p>An error was recognized on receiving a telegram and the CP requested repetition from the partner via a negative acknowledgment (NAK).</p>	A repetition is not an error, however, it can be an indication that there are disturbances on the transmission line or a malfunction of the partner device. If the telegram can still not be transmitted after the maximum number of repetitions, an error number which describes the error that first occurred is sent.
(08)02H	<p>With 3964(R) only:</p> <p>Error during connection setup:</p> <ul style="list-style-type: none"> • 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. <p>After the partner has signaled POWER ON:</p> <ul style="list-style-type: none"> • While partner is being activated, CP receives an undefined code. 	Check for malfunction at partner device, possibly using interface test device (FOXPG) which is switched into the transmission line.
(08)05H	<p>With 3964(R) only:</p> <p>Logical error during receiving:</p> <p>After DLE was received, a further random code (other than DLE or ETX) was received.</p>	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 using interface test device (FOXPG) which is switched into the transmission line.
(08)06H	<p>Character delay time exceeded:</p> <ul style="list-style-type: none"> • Two successive characters were not received within character delay time, or <p>With 3964(R) only:</p> <ul style="list-style-type: none"> • 1st character after sending of DLE during connection setup was not received within character delay time. 	Partner device too slow or faulty. Check for malfunction at partner device, possibly using interface test device (FOXPG) which is switched into the transmission line.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(08)07H	Unpermitted message frame length: A message frame of length 0 was received.	Receipt of a message frame of length 0 is not an error. Check why communications partner is sending message frames without user data.
(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 using interface test device (FOXPG) which is switched into the transmission line.
(08)09H	With 3964(R) only: Number of repetitions set must be identical.	Parameterize same block wait time at communications partner as at CP 340. Check for malfunction at communications partner, possibly using interface test device (FOXPG) which is 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 FB must be called more frequently.
(08)0CH	Transmission error: <ul style="list-style-type: none"> Transmission error (parity error, stop bit error, overflow error) detected. With 3964(R) only: <ul style="list-style-type: none"> If this occurs during send or receive operation, repetitions are initiated. If faulty character is received in idle mode, the error is reported immediately so that disturbances on the transmission line can be detected early. If SF (red) and RXD (green) LEDs are lit up, there is a break on the line between the two communications partners. 	Faults 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)10H	With ASCII driver only: Parity error: <ul style="list-style-type: none"> If SF (red) and RXD (green) LEDs are lit up, there is a break on the line between the two communications partners. 	Check connecting cable of communications partner or check whether both devices have same setting for baud rate, parity and number of stop bits. Change your system setup or the line routing.

Table 8-1 Event Classes and Event Numbers, continued

Event No.	Event	Remedy
(08)11H	<p>With ASCII driver only:</p> <p>Character frame error:</p> <ul style="list-style-type: none"> If SF (red) and RXD (green) LEDs are lit up, there is a break on the line between the two communications partners. 	<p>Check connecting cable of communications partner or check whether both devices have same setting for baud rate, parity and number of stop bits.</p> <p>Change your system setup or the line routing.</p>
(08)12H	<p>With ASCII driver only:</p> <p>More characters were received after the CP had sent XOFF or set CTS to OFF.</p>	<p>Reparameterize communications partner or read data from CP more quickly.</p>
(08)18H	<p>With ASCII driver only:</p> <p>DSR = OFF or CTS = OFF</p>	<p>The partner has switched the DSR or CTS signal to "OFF" before or during a transmission.</p> <p>Check the partner's control of the RS 232C secondary signals.</p>

Displaying and Interpreting 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 340 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.
(1E)41H	P_PRINT FB: Number of bytes specified in the pointer DB for variable or format string is impermissible.	Comply with the permitted lengths: 32 bytes for variables, 150 bytes for a format string
(1E)43H	P_PRINT FB: No pointer for the format string.	Specify in the pointer DB the data block number and data word number for the format string (see Section 6.3.2).

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 Diagnosis via the S7-300 Backplane Bus

Introduction

The CP 340 can trigger a diagnostics alarm on the assigned CPU, thus indicating a malfunction of the CP 340. You can specify at parameterization whether the CP 340 is to trigger a diagnostics alarm or not in the event of serious errors (see Section 2.3.1).

By default DIAGNOSTICS ALARM = NO is set.

Diagnostics Alarm

In the event of an error, the CP 340 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.

If a diagnostics alarm 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 the 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 340 provides 4 bytes of diagnostics information. To display the error that has occurred, these bytes are occupied as follows:

2nd byte:

The 2nd byte of diagnostics data contains the class ID of the CP 340 in bits 0 to 3.

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

1st, 3rd and 4th bytes:

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	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No parameter	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RAM error	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
ROM error	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
System error	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Diagnosis Information (in KH Format)

The following table shows the 4 bytes diagnosis information of the CP 340 in the KH format.

Event	1st Byte	2nd Byte	3rd Byte	4th Byte
Wire break	25H	8CH	02H	00H
Incorrect parameter	81H	8CH	00H	00H
No parameter	41H	8CH	00H	00H
RAM error	03H	8CH	00H	08H
ROM error	03H	8CH	00H	04H
System error	03H	8CH	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:

- All events entered in the diagnostics buffer of the CPU are deleted,
- 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.

8.5 Diagnosis by Means of the Diagnostic Buffer of the CP 340

Diagnostic Buffer of the CP 340

The CP 340 has its own diagnostic buffer, in which all the diagnostic events of the CP 340 are entered in the sequence in which they occur.

The following are displayed in the diagnostic buffer of the CP 340:

- The operating status of the CP 340
- Hardware/firmware errors on the CP 340
- Initialization and parameterization errors
- Errors during execution of a CPU job
- 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 340 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. The most recent entry always comes first.

When the power of the CP 340 is switched off, the contents of the diagnostic buffer are lost.

Reading the Diagnostic Buffer at the Programming Device

The contents of the diagnostic buffer of the CP 340 can be read by means of the STEP 7 information functions.

Note

Diagnostic events in the diagnostic buffer of the CP 340 can be read using **STEP 7 as of Version 3.1.**

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.

Prerequisite: 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.

Result: The window containing the configuration table appears.

2. Select the CP 340 in the configuration table.
3. Choose the **PLC > Module Information** menu command.

Result: The "Module Information" dialog box appears for the CP 340. The "General" tab is displayed by default the first time you call it.

4. Select the "Diagnostic Buffer" tab.

Result: The "Diagnostic Buffer" tab displays the most recent diagnostic events of the CP 340. Any additional information on the cause of the problem appears in the lower part of the tab.

If you click the "Update" button, the current data is read from the CP 340. By clicking the "Help on Event" button you can display a help text on the selected diagnostic event with information on error correction.

Programming Example for Standard Function Blocks

9

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
9.1	General Information	9-2
9.2	Device Configuration	9-3
9.3	Settings	9-4
9.4	Blocks Used	9-5
9.5	Example "Point-to-Point Connection"	9-6
9.6	Example "Printing" and "Reading and Controlling the CP 340 Inputs/Outputs"	9-8
9.7	Installation, Error Messages	9-10
9.8	Activation, Start-Up Program and Cyclic Program	9-11

9.1 General Information

Introduction	<p>The programming example given here describes standard functions for operating the CP 340 communications processor.</p>
Objective	<p>The programming example</p> <ul style="list-style-type: none">• 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 easily be extended for your own purposes. <p>The example shows how a connection to a communications partner can be configured using the standard function blocks P_SEND and P_RCV (to send and receive data respectively).</p> <p>The example also shows how data can be output to a printer using the P_PRINT function block and how the inputs and outputs of the CP 340 can be controlled and monitored using the V24_STAT and V24_SET standard functions.</p> <p>There are two SIMATIC stations in the example because the CP 340 has to be parameterized differently for sending/receiving data and for printer output:</p> <ul style="list-style-type: none">• 1st station: Computer link with the P_SEND and P_RCV function blocks• 2nd station: Printing and reading and controlling RS 232C secondary signals with the P_PRINT function block and the V24_STAT and V24_SET functions <p>The CP 340 is parameterized by the CPU when the latter is started up (system service).</p>
Prerequisite	<p>The example can be executed with the minimum hardware equipment (2 bytes for inputs, 2 bytes for outputs). The STEP 7 function Monitor/Modify Variables is also used (e.g. to modify transmitted data).</p>
Program Example	<p>The program example of the CP 340 is supplied on a 3.5" diskette which comes with this manual. The diskette also contains the parameterization interface and the function blocks.</p> <p>It is available both compiled and as an ASCII source file. A list of all the symbols used in the example is also included.</p>

9.2 Device Configuration

Application

To try out the sample program, you could use the following devices:

- One S7-300 PLC (mounting rack, power supply, CPU)
- One CP 340 module with a communications partner (e.g. a second CP) or printer, or you could plug in a "short-circuit connector", i.e. the send line is bridged to the receive line
- One simulator module with inputs and outputs (or alternatively one digital input module and one digital output module)
- One programming device (e.g. PG 740).

Note

You do not need the digital inputs and outputs if all functions are executed with the STEP 7 function **Monitor/Modify Variables**. In this case, you must change the program in organization block OB 1.

9.3 Settings

Settings in the CPU via STEP 7

You must configure your controller setup with STEP 7:

- Slot 1: power supply
- Slot 2: CPU
- Slot 4: Digital input, IB0 and IB1
- Slot 5: Digital output, QB4 and QB5
- Slot 6: CP 340, start address P288

Settings on the CP 340

You cannot make any hardware settings on the CP 340. All relevant data is configured via STEP 7, including the parameters for the CP 340, using the *CP 340: Point-to-Point Communication, Parameter Assignment* interface, and loaded in the CPU.

You can run the program example for the computer link 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 program example for the printer can only be run with the printer driver.

The functions for reading and controlling RS 232C secondary signals can only be executed with the ASCII driver. The prerequisite for this is that the “Automatic Use of V24 Signals” parameter on the “Transmission” tab is not selected.

9.4 Blocks Used

Blocks Used

The table below shows the blocks used for the sample program.

Block	Symbol	Explanation
OB 1	CYCLE	Cyclic program processing
OB 100	COLD RESTART	Cold restart processing
DB 2	DB_P_RCV	Instance DB for P_RCV FB
DB 3	DB_P_SEND	Instance DB for P_SEND FB
DB 4	DB_P_PRINT	Instance DB for P_PRINT FB
DB 10	SEND_DB	Send data block
DB 20	RCV_DB	Receive data block
DB 30	DB_PRT_AUFTR	Pointer DB for format string and variables
DB 31	DB_VAR1	DB data for variable 1
DB 32	DB_VAR2	DB data for variable 2
DB 33	DB_VAR3	DB data for variable 3
DB 34	DB_VAR4	DB data for variable 4
DB 35	DB_STRING	DB data for format string
FB 2	P_RCV	Receive standard FB for data
FB 3	P_SEND	Send standard FB for data
FB 4	P_PRINT	Standard FB for message output
FC 5	V24_STAT	Standard FC for reading CP outputs
FC 6	V24_SET	Standard FC for writing to CP outputs
FC 8	DT_TOD	Standard FC for converting the DT format to TOD
FC 9	CPU_TIME	Standard FC for reading the CPU time and converting it to TOD
FC 10	ORG_CYC	Organization of cycle
FC 11	SEND_CYC	Send data
FC 12	RCV_CYC	Receive data
FC 13	PRN_ZYK	Output message
FC 14	V24_CYC	Control RS 232C secondary signals

9.5 Example "Point-to-Point Connection"

Introduction The inputs and outputs are mapped to memory bits at the beginning and end of OB 1. Only the memory bits are used in the test program.

Inputs and Outputs Used for FB 2 and FB 3 In the table below, the assignment of the inputs/outputs and memory bits is shown.

Symbol	Input/Output	Memory Marker	Explanation	
ANW_RECH	I 0.0	M 0.0	"1" = select point-to-point connection	
ANW_DRUCK	I 0.1	M 0.1	"0" signal	
RESET_SP	I 0.2	M 0.2	Start RESET SEND	
RESET_R	I 0.3	M 0.3	Start RESET RCV	
	I 0.4	M 0.4	Free	
	I 0.5	M 0.5	Free	
REQ_SP	I 0.6	M 0.6	Start SEND request	
EN_R_R	I 0.7	M 0.7	Enable receipt	
AUFTR_1_SP	I 1.0	M 1.0	Send request selection; "1" = request 1	
AUFTR_2_SP	I 1.1	M 1.1	Send request selection; "1" = request 2	
AUFTR_1_R	I 1.2	M 1.2	Enable receipt of request selection; "1" = request 1	
AUFTR_2_R	I 1.3	M 1.3	Enable receipt of request selection; "1" = request 2	
ANZ_SFCERR_SP	I 1.4	M 1.4	"1" = display of SEND-SFCERR, "0" = display of the FB parameters	
ANZ_SFCERR_R	I 1.5	M 1.5	"1" = display of RCV-SFCERR, "0" = display of the FB parameters	
	I 1.6	M 1.6	Free	
	I 1.7	M 1.7	Free	
			Display of SFCERR (or)	Display of FB parameters
A_DONE_SP	A 4.0	M 8.0	Event class	SEND-DONE
A_ERROR_SP	A 4.1	M 8.1	Event class	SEND-ERROR
A_BIE_SP	A 4.2	M 8.2	Event class	SEND binary result
	A 4.3	M 8.3	Event class	"0"
A_NDR_R	A 4.4	M 8.4	Event class	RCV-NDR
A_ERROR_R	A 4.5	M 8.5	Event class	RCV-ERROR
A_BIE_R	A 4.6	M 8.6	Event class	RCV binary result
	A 4.7	M 8.7	Event class	"0"

Continued

Symbol	Input/ Output	Memory Marker	Explanation	
			Display of SFCERR (or)	Display of FB parameters
	A 5.0	M 9.0	Error message	"0"
	A 5.1	M 9.1	Error message	"0"
	A 5.2	M 9.2	Error message	"0"
	A 5.3	M 9.3	Error message	"0"
	A 5.4	M 9.4	Error message	"0"
	A 5.5	M 9.5	Error message	"0"
	A 5.6	M 9.6	Error message	"0"
	A 5.7	M 9.7	Error message	"0"

**Input/Output
Parameters for
FB 2 and FB 3**

In the table below, the input/output parameters for FB 2 and FB 3 are mapped to memory bits.

Symbol	Address	Data Format	Explanation
SEND_DONE	M26.0	BOOL	SEND: DONE parameter
SEND_ERROR	M26.1	BOOL	SEND: ERROR parameter
SEND_BIE	M26.2	BOOL	SEND: binary result
RCV_NDR	M29.0	BOOL	RCV: NDR parameter
RCV_ERROR	M29.1	BOOL	RCV: ERROR parameter
RCV_BIE	M29.2	BOOL	RCV: binary result
BGADR	MW21	INT	Logical basic address
SEND_STATUS	MW27	WORD	SEND: STATUS parameter
RCV_STATUS	MW30	WORD	RCV: STATUS parameter
SEND_DBNR	MW40	INT	SEND: DB_NO parameter
SEND_DWNR	MW42	INT	SEND: DBB_NO parameter
SEND_LAE	MW44	INT	SEND: LEN parameter
RCV_DBNR	MW50	INT	RCV: DB_NO parameter
RCV_DWNR	MW52	INT	RCV: DBB_NO parameter
RCV_LAE	MW54	INT	RCV: LEN parameter

9.6 Example "Printing" and "Reading and Controlling the CP 340 Inputs/Outputs"

Introduction

The inputs and outputs are mapped to memory bits at the beginning and end of OB 1. Only the memory bits are used in the test program.

Inputs and Outputs Used for FB 5 and FB 6

In the table below, the assignment of the inputs/outputs to memory bits is shown.

Symbol	Input/Output	Memory Marker	Explanation	
ANW_RECH	I 0.0	M 0.0	"0" signal	
ANW_DRUCK	I 0.1	M 0.1	"1" = select printing and control/status	
RESET_SP	I 0.2	M 0.2	Start RESET-PRINT	
	I 0.3	M 0.3	Free	
	I 0.4	M 0.4	Free	
	I 0.5	M 0.5	Free	
REQ_SP	I 0.6	M 0.6	Start PRINT request	
EN_R_R	I 0.7	M 0.7	Execute SET request	
AUFTR_1_DR	I 1.0	M 1.0	Print request selection; Request 1 = "1"	
AUFTR_2_DR	I 1.1	M 1.1	Print request selection; Request 2 = "1"	
AUFTR_3_DR	I 1.2	M 1.2	Print request selection; Request 3 = "1"	
AUFTR_4_DR	I 1.3	M 1.3	Print request selection; Request 4 = "1"	
ANZ_SFCERR_SP	I 1.4	M 1.4	"1" = display of PRINT-SFCERR, "0" = display of the FB parameters	
	I 1.5	M 1.5	Free	
STEU_DTR	I 1.6	M 1.6	Control signal DTR, signal for V24_SET FC	
STEU_RTS	I 1.7	M 1.7	Control signal RTS, signal for V24_SET FC	
			Display of SFCERR (or)	Display of FB parameters
A_DONE_SP	A 4.0	M 8.0	Event class	PRINT-DONE
A_ERROR_SP	A 4.1	M 8.1	Event class	PRINT-ERROR
A_BIE_SP	A 4.2	M 8.2	Event class	PRINT binary result
	A 4.3	M 8.3	Event class	"0"
	A 4.4	M 8.4	Event class	"0"
	A 4.5	M 8.5	Event class	"0"
	A 4.6	M 8.6	Event class	"0"
	A 4.7	M 8.7	Event class	"0"

Continued

Symbol	Input/ Output	Memory Marker	Explanation	
			Display of SFCERR (or)	Display of FB parameters
A_V24_STAT_DTR_OUT	A 5.0	M 9.0	Error number	STAT_DTR_OUT
A_V24_STAT_DSR_IN	A 5.1	M 9.1	Error number	STAT_DSR_IN
A_V24_STAT_RTS_OUT	A 5.2	M 9.2	Error number	STAT_RTS_OUT
A_V24_STAT_CTS_IN	A 5.3	M 9.3	Error number	STAT_CTS_IN
A_V24_STAT_DCD_IN	A 5.4	M 9.4	Error number	STAT_DCD_IN
A_V24_STAT_RI_IN	A 5.5	M 9.5	Error number	STAT_RI_IN
	A 5.6	M 9.6	Error number	"0"
	A 5.7	M 9.7	Error number	"0"

**Input/Output
Parameters of FB 5
and FB 6**

The table below shows how the input/output parameters of FC 5 and FC 6 are mapped to memory bits.

Symbol	Address	Data Format	Explanation
BGADR	MW21	INT	Logical base address
V24_STAT_DTR_OUT	M9.0	BOOL	STAT: DTR_OUT parameter
V24_STAT_DSR_IN	M9.1	BOOL	STAT: DSR_IN parameter
V24_STAT_RTS_OUT	M9.2	BOOL	STAT: RTS_OUT parameter
V24_STAT_CTS_IN	M9.3	BOOL	STAT: CTS_IN parameter
V24_STAT_DCD_IN	M9.4	BOOL	STAT: DCD_IN parameter
V24_STAT_RI_IN	M9.5	BOOL	STAT: RI_IN parameter
PRINT_DBNR	MW56	INT	PRINT: DB_NO parameter
PRINT_DWNR	MW58	INT	PRINT: DBB_NO parameter
PRINT_STATUS	MW61	WORD	PRINT: STATUS parameter
PRINT_DONE	M60.0	BOOL	PRINT: DONE parameter
PRINT_ERROR	M60.1	BOOL	PRINT: ERROR parameter
PRINT_BIE	M60.2	BOOL	PRINT: Binary result

9.7 Installation, Error Messages

Scope of Supply and Installation

The CP 340 program example, together with the *CP 340: Point-to-Point Communication, Parameter Assignment* parameterization interface and the function blocks, is supplied on a 3.5" diskette 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:

CP340p

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.

Downloading to the CPU

The hardware for the example is completely 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.

After every block call in the cyclic program, if an error has occurred (BR = "0"), an error memory bit is set, which can only be displayed at the outputs Q 4.2/4.6 with the computer connection.

If there is an error message, the parameter output ERROR (Q 4.1/4.5) of the modules is also set. A more detailed description of the error is then stored in the STATUS parameter of the modules. 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.

The FB parameters DONE, NDR and ERROR as well as the binary result are displayed if the two inputs I 1.4 and I 1.5 have the signal state "0". If one of these inputs has the signal state "1", the associated error message SFCERR from the instance DB is displayed at the output location QW4.

9.8 Activation, Start-Up Program and Cyclic Program

Activation, Start-Up Program

The start-up program is located in the OB 100.

At start-up, only the logical basic address of the CP 340 is entered in the MW BGADR (MW21).

Cyclic Program

The cyclic program is located in the OB 1.

At the beginning of the program, first the inputs used are mapped onto memory bits, which are then used during the rest of the program. At the end of the program, the control memory bits set are mapped onto the outputs and displayed.

In the example, the standard function blocks P_RCV (FB 2) and P_SEND (FB 3) work with functions FC 11 and FC 12, and with data blocks DB 2 and DB 3 as instance DBs, and DB 10 and DB 20 as send and receive DBs respectively.

The P_PRINT standard function block (FB 4) works with FC 13, with DB 4 as the instance DB, and with DB 30 to DB 35 as data DBs.

The standard functions V24_STAT (FC 5) and V24_SET (FC 6) work with FC 14.

The cyclic processing is organized in FC 10.

In the example, the standard function blocks are parameterized partly with constants and partly with symbolically addressed actual operands.

Description
"Point-to-Point
Connection"

For the "point-to-point connection", the input I 0.0 must be showing the signal "1" and the input I 0.1 must be showing the signal "0". Using the inputs I 1.0 and I 1.1, you can choose between two SEND requests. Request 1 sends the data of DB 10 from DBB 2 to DBB 11, while request 2 sends it from DBB 14 to DBB 113.

The data is sent to the communications partner when the input I 0.6 (SEND-REQ) has a signal change from "0" to "1".

For data to be received, the receive enable (EN_R parameter in block P_RCV), input I 0.7, must have the signal state "1".

If the receiving of request 1 has been enabled (I 1.2 = signal "1"), this data is stored in DB 20 as of DBB 0. If request 2 has been enabled (I 1.3 = signal "1"), the data is stored in DB 20 as of DBB 50.

The signal states of the FB parameters DONE/NDR and ERROR as well as the binary results are displayed when inputs I 1.4 and I 1.5 have the signal state "0". If one of these inputs has the signal state "1", the associated SFCERR variable (error code of the integrated SFCs) from the instance DB is displayed.

If the signal state is "1" at inputs I 0.2 and I 0.3, a RESET-SEND or RESET-RCV can be passed to the CP 340. If the signal state is statically "1", sending or receiving data has been deactivated.

Description
"Reading and
Controlling the
CP 340 Inputs/
Outputs"

The functions for reading and controlling the RS 232C secondary signals can only be executed with the ASCII driver. The prerequisite for this is that the "Automatic Use of V24 Signals" parameter in the "Transmission" tab is not selected.

To enable the CP 340 inputs/outputs to be read and controlled, input I 0.0 must be showing the "0" signal and input I 0.1 the "1" signal. The signal states SET_DTR and SET_RTS can be preselected by means of inputs I 1.6 and I 1.7. When the signal at I 0.7 changes from "0" → "1", this state is transferred to the CP by the V24_SET function.

The V24_STAT standard function is called cyclically. The state of the CP 340 inputs/outputs is displayed at output byte AB 5.

Description
"Printing"

To enable printing, input I 0.0 must be showing the "0" signal and input I 0.1 the "1" signal. Inputs I 1.0 and I 1.1 allow you to select between two PRINT requests. The PRINT requests are located in the data block DB 30. This points to the actual data (variables 1 to 4 and the format string) to be sent to the CP 340.

Request 1 sends the data of variables 1 to 4 and the format string. The data is read:

- Variable 1 from DB 31 as of data double word DBD 0
- Variable 2 from DB 32 as of data word DBW 0
- Variable 3 from DB 33 as of data word DBW 0
- Variable 4 from DB 34 as of data word DBW 0
- Format string from DB 35 from DBB 2 to DBB 43

Request 2 sends the data of variables 1 to 4 and the format string. The data is read:

- Variable 1 from DB 31 as of data double word DBD 8
- Variable 2 from DB 32 as of data word DBW 2
- Variable 3 from DB 33 as of data word DBW 2
- Variable 4 from DB 34 as of data word DBW 2
- Format string from DB 35 from DBB 2 to DBB 43

Request 3 sends the data of variables 1 to 4 and the format string. The data is read:

- Variable 1 from DB 31 as of data double word DBD 16
- Variable 2 from DB 32 as of data word DBW 4
- Variable 3 from DB 33 as of data word DBW 4
- Variable 4 from DB 34 as of data word DBW 4
- Format string from DB 35 from DBB 2 to DBB 43

Request 4 sends the data of variable 1 and the format string. The data is read:

- Variable 1 from DB 31 as of data double word DBD 24
- Format string from DB 35 from DBB 68 to DBB 111

The data is sent to the communication partner when there is a signal change from "0" to "1" at input I 0.6 (PRINT-REQ).

The signal states of the FB parameters DONE and ERROR and the binary result are displayed when input I 1.4 has the signal state "0". If input I 1.4 has the signal state "1", the associated SFCERR variable (error code of the integrated SFCs) from the instance DB is displayed.

When input I 0.2 has the signal state "1", a RESET-PRINT can be sent to the CP 340. If a signal state of "1" is applied statically, the transmission of data is deactivated.

In the example, the V24_STAT standard function is also called cyclically. The state of the CP 340 inputs/outputs is displayed at output byte AB 5.

Example

Here is an example of a printout with the sample program for the CP 340:

SIMATIC S7 / CP340

Example for printing mode

At 23:32:07.754 hours: Level reached upper limit

At 23:32:16.816 hours: Level above upper limit

At 23:32:21.681 hours: Level below lower limit

At 23:32:26.988 hours the 200 l level was reached

Copyright © Siemens AG 1996. All rights reserved.

Technical Specifications

A

Section Overview This chapter is divided into the following sections:

In Section	You Will Find	on Page
A.1	Technical Specifications of the CP 340	A-2
A.2	Certification and Application Areas	A-7
A.3	Recycling and Disposal	A-9

A.1 Technical Specifications of the CP 340

General Technical Specifications

The following table contains the general technical specifications of the CP 340.

Table A-1 General Technical Specifications

Technical Specifications	
Power supply via S7-300 backplane bus (5V)	CP 340-RS 232C: max. 220 mA CP 340-20mA TTY: max. 220 mA CP 340-RS 422/485: max. 220 mA
Operating temperature	0°C to +60°C
Storage temperature	-40°C to +70°C
Power loss	typ. 0.85 W
Dimensions W × H × D	40 × 125 × 120 mm
Weight	0.3 kg
Display elements	LEDs for transmitting (TXD), receiving (RXD) and group faults (SF)
Supplied protocol drivers	ASCII driver 3964(R) driver Printer driver
Transmission speed with 3964(R) protocol	2400, 4800, 9600, 19200 bits/s (half-duplex)
Transmission speed with ASCII driver	2400, 4800, 9600 bits/s (full-duplex)
Transmission speed with printer driver	2400, 4800, 9600 bits/s
Character frames (10 bit or 11 bit)	No. of bits per character (7 or 8) No. of start/stop bits (1 or 2) Parity (none, even, odd, any)
Quantity of user data transported per program cycle	Sending: 14 bytes Receiving: 13 bytes
Storage space requirements of the standard blocks (FBs)	Sending and receiving approx. 2700 bytes

Technical Specifications of the RS 232C Interface

The table below contains the technical specifications of the RS 232C interface of the CP 340-RS 232C.

Table A-2 Technical Specifications of the RS 232C Interface

RS 232C Interface	
Interface	RS 232C, 9-pin sub D male connector
RS 232C signals	TXD, RXD, RTS, CTS, DTR, DSR, RI, DCD, GND All isolated against the S7 internal power supply
Max. transmission distance	15 m

Technical Specifications of the 20 mA TTY Interface

The table below contains the technical specifications of the 20 mA TTY interface of the CP 340-20mA 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 connector
TTY signals	Two isolated 20 mA current sources Receive loop (RX) ”-” and ”+” Send loop (TX) ”-” and ”+” All isolated against the S7 internal power supply
Max. transmission distance	100 m active, 1000 m passive

Technical Specifications of the X27 (RS 422/485) Interface

The table below contains the technical specifications of the X27 (RS 422/ 485) interface of the CP 340-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 connector
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
Max. transmission distance	1200 m

Technical Specifications of the 3964(R) Protocol

The table below contains the technical specifications of the 3964(R) protocol.

Table A-5 Technical Specifications of the 3964(R) Protocol

3964(R) Protocol with Default Values	
Max. message frame length	1024 bytes
Parameters	The following can be parameterized: <ul style="list-style-type: none"> • With/without block check character • Priority: low/high • Transmission speed: 2400, 4800, 9600, 19200 bits/s • Character frame: 10 or 11 bits • Delete CP receive buffer at start-up: yes/no
3964(R) Protocol with Parameterization	
Max. message frame length	1024 bytes
Parameters	The following can be parameterized: <ul style="list-style-type: none"> • With/without block check character • Priority: low/high • Transmission speed: 2400, 4800, 9600, 19200 bits/s • Character frame: 10 or 11 bits • Character delay time: 20 ms to 655350 ms in a 10 ms matrix • Acknowledgment delay time: 10 ms to 655350 ms in a 10 ms matrix • Number of setup attempts: 1 to 255 • Number of transmission attempts: 1 to 255 • Delete CP receive buffer at start-up: yes/no

Technical Specifications of the ASCII Driver

The table below 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	<p>The following can be parameterized:</p> <ul style="list-style-type: none"> • Transmission speed: 2400, 4800, 9600 bits/s • Character frame: 10 or 11 bits • Character delay time: 4 ms to 65535 ms in 1 ms steps • Flow control: None, XON/XOFF, RTS/CTS; automatic use of the V.24 signals • XON/XOFF character (only when flow control = "XON/XOFF") • Wait for XON after XOFF (waiting time for CTS=ON): 20 ms to 655350 ms in 10 ms steps • Time to RTS OFF: 20 ms to 655350 ms in 10 ms steps (only in the case of automatic use of the V.24 signals) • Data output waiting time: 20 ms to 655350 ms in 10 ms steps (only in the case of automatic use of the V.24 signals) • Delete CP receive buffer at start-up: yes/no • Number of message frames to be buffered: 1 to 250 • Prevent overwriting: yes/no (only when buffered receive message frames = 1) • End detection of a receive message frame: <ul style="list-style-type: none"> – After expiration of the character delay time – After receipt of the end code(s) – After receipt of a fixed number of characters
ASCII Driver with Message Frame End Detection After Expiration of the Character Delay Time	
Parameters	No further parameters need to be set. The end of the message frame is detected when the parameterized character delay time expires.
ASCII Driver with Message Frame End Detection on the Basis of Parameterized End Codes	
Parameters	<p>The following can also be parameterized:</p> <ul style="list-style-type: none"> • Number of end codes: 1 or 2 • Hex code for first/second end identifier
ASCII Driver with Message Frame End Detection on the Basis of the Configured Message Frame Length	
Parameters	<p>The following can also be parameterized:</p> <ul style="list-style-type: none"> • Message frame length: 1 to 1024 bytes

Technical Specifications of the Printer Driver

The table below contains the technical specifications of the printer driver.

Table A-6 Technical Specifications of the Printer Driver

Printer Driver	
Size of the text SDB	8 kbytes
Parameters	<p>The following can be parameterized:</p> <ul style="list-style-type: none"> • Transmission speed: 2400, 4800, 9600 bits/s • Character frame: 10 or 11 bits • Flow control: none, XON/XOFF, RTS/CTS • XON/XOFF character (only when flow control = "XON/XOFF") • Wait for XON after XOFF (waiting time for CTS=ON): 20 ms to 655350 ms in 10 ms steps • Message text: max. 150 characters (max. 250 characters with display of variables) • Left margin (number of blanks): 0 to 255 • Lines per page (with header and footer): 1 to 255 or 0 (continuous printing) • Separators/line end: CR, LF, CRLF, LF CR • Character set: IBM-Proprinter or user-defined • Printer emulation for bold, condensed, expanded and italic type and underlining: HP DeskJet, HP LaserJet, IBM-Proprinter or user-defined • 1/2 header and/or footer lines

A.2 Certification and Application Areas

Introduction

This section contains information for the CP 340 on:

- The most important standards with which the CP 340 complies
- Certificates and approvals of the CP 340

IEC 1131

The CP 340 communications processor meets the requirements and criteria of IEC 1131, Part 2.

CE Marking



Our products fulfill the requirements and safety objectives of the following EC Directives and comply with the harmonized European standards (EN) published for stored-program controllers in the official journals of the European Communities:

- 89/336/EEC Electromagnetic Compatibility Directive (EMC Directive)
- 73/23/EEC Low Voltage Directive (for electrical equipment)

The EC Declarations of Conformity are available to the relevant authorities at the following address:

Siemens Aktiengesellschaft
 Bereich Automatisierungstechnik
 AUT E 14
 Postfach 1963
 D-92209 Amberg

EMC Directive

SIMATIC products are designed for industrial use.

SIMATIC products can also be used in a domestic environment (domestic, business and commercial) subject to individual approval being obtained. You must obtain individual approval from the relevant authorities or testing institute. In Germany approval is granted by the Federal Office for Post and Telecommunications and its agencies.

Area of Application	Requirements	
	Emitted Interference	Noise Immunity
Industry	EN 50081-2 : 1993	EN 50082-2 : 1995
Domestic	Individual approval	EN 50082-1 : 1992

UL Recognition

UL Recognition Mark
Underwriters Laboratories (UL) to
Standard UL 508, File No. 116536

CSA Certification

CSA Certification Mark
Canadian Standard Association (CSA) to
Standard C22.2 No. 142, File No. LR 48323

FM Approval

Factory Mutual Approval Standard Class Number 3611, Class I, Division 2,
Group A, B, C, D.



Warning

There is a risk of injury and damage to property.

In potentially explosive atmospheres, there is a risk of injury to people and damage to property if you disconnect connectors when an S7-300 is in operation.

Always de-energize the S7-300 in potentially explosive atmospheres before disconnecting any connectors.



Warning

**WARNING - DO NOT DISCONNECT WHILE CIRCUIT IS LIVE
UNLESS LOCATION IS KNOWN TO BE NONHAZARDOUS**

A.3 Recycling and Disposal

Recycling and Disposal

The SIMATIC S7-300 is an environment-friendly product. It is exceptional for the following:

- 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 SIMATIC S7-300 is recyclable due to its low contaminant content.

To recycle and dispose of your old SIMATIC equipment in an environment-friendly manner, contact:

Siemens Aktiengesellschaft
Technische Dienstleistungen
ANL A 44 Kreislaufwirtschaft
Postfach 32 40
D-91052 Erlangen

Phone: +49 91 31/7-3 26 98

Fax: +49 91 31/7-2 66 43

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.

Connecting Cables

B

Section Overview

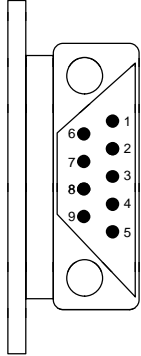
This chapter is divided into the following sections:

In Section	You Will Find	on Page
B.1	RS 232C Interface	B-2
B.2	20 mA TTY Interface	B-10
B.3	X27 (R422/485) Interface	B-17

B.1 RS 232C Interface of the CP 340-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 340-RS 232C.

Table B-1 Pin Allocation for the 9-Pin Sub D Male Connector of the Integrated Interface of the CP 340-RS 232C

Male Connector on CP 340*	Pin	Designation	Input/Output	Meaning
	1	DCD Received Detector	Input	Receiver signal level
	2	RXD Received Data	Input	Received data
	3	TXD Transmitted Data	Output	Transmitted data
	4	DTR Data Terminal Ready	Output	Communication terminals ready
	5	GND Ground	–	Signal ground (GND _{int})
	6	DSR Data Set Ready	Input	Ready for operation
	7	RTS Request To Send	Output	Activate transmitter
	8	CTS Clear To Send	Input	Ready for sending
	9	RI Ring Indicator	Input	Receiving call

* Front view

Connecting Cables

If you make your own connecting cables you must remember that unconnected inputs at the communication partner may have to be connected to open-circuit potential.

Please 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 340-RS 232C and S7 modules or SIMATIC S5.

**RS 232C
Connecting Cables
(S7/M7 (CP 340) –
S7/M7 CP
340/CP 441))**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 340/CP 441.

For the connecting cables you require the following female connectors:

- At CP 340: 9-pin sub D female with screw fixing
- At communication partner: 9-pin sub D female with screw fixing

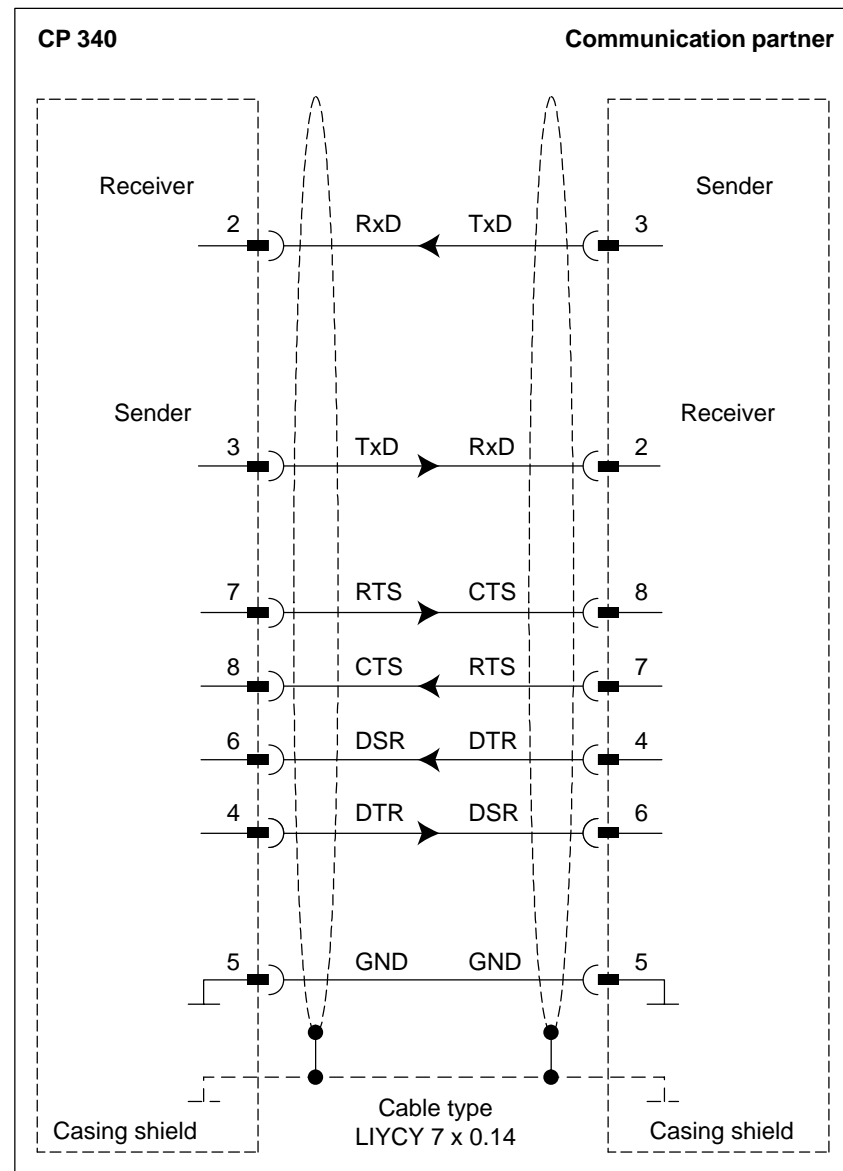


Figure B-1 RS 232C Connecting Cable CP 340 – CP 340/CP 441

You can order the connecting cable under the order number given in Appendix D.

**RS 232C
Connecting Cables
(S7/M7 (CP 340) –
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 340 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 CP 340: 9-pin sub D female with screw fixing
- At communication partner: 25-pin sub D male with screw fixing

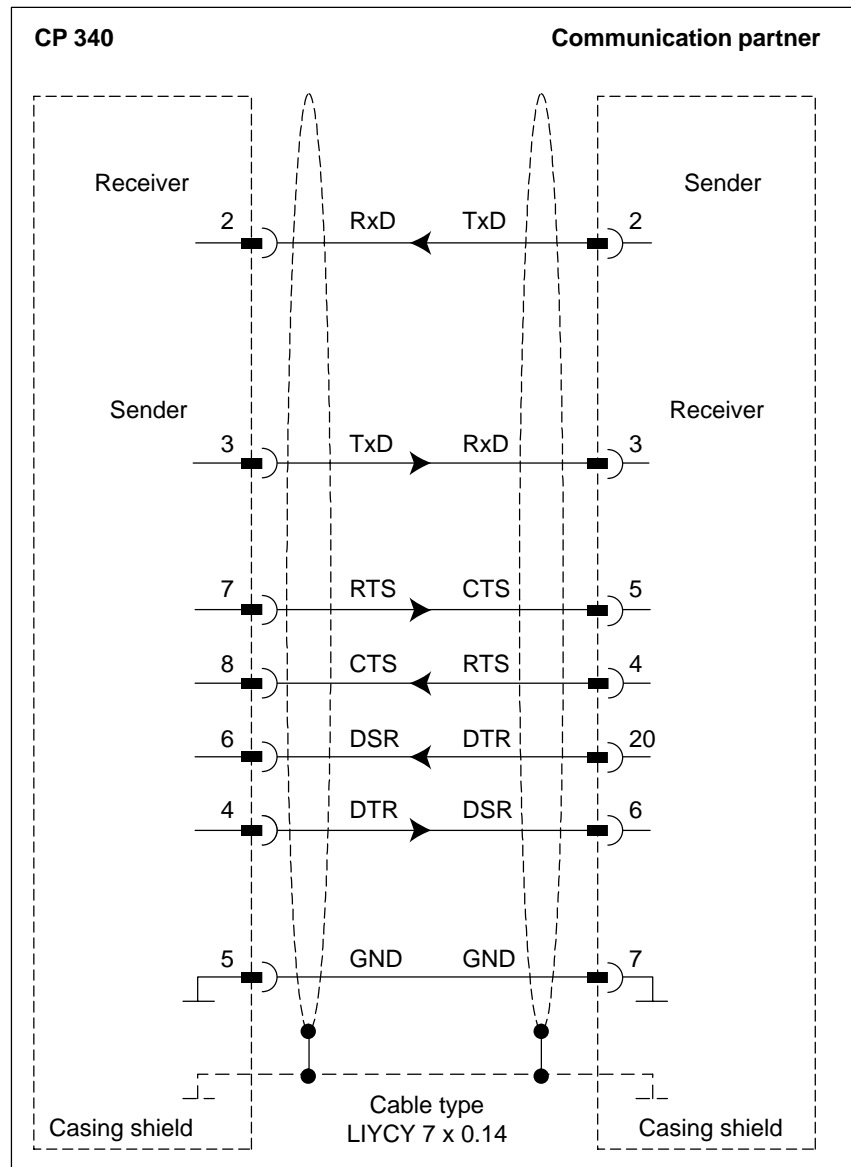


Figure B-2 RS 232C Connecting Cable CP 340 – CP 544, CP 524, CPU 928B, CPU 945, CPU 948

**RS 232C
Connecting Cables
(S7/M7 (CP 340) –
CP 521SI/
CP 521 BASIC)**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 521SI/CP 521 BASIC.

For the connecting cables you require the following female/male connectors:

- At CP 340: 9-pin sub D female with screw fixing
- At communication partner: 25-pin sub D male with screw fixing

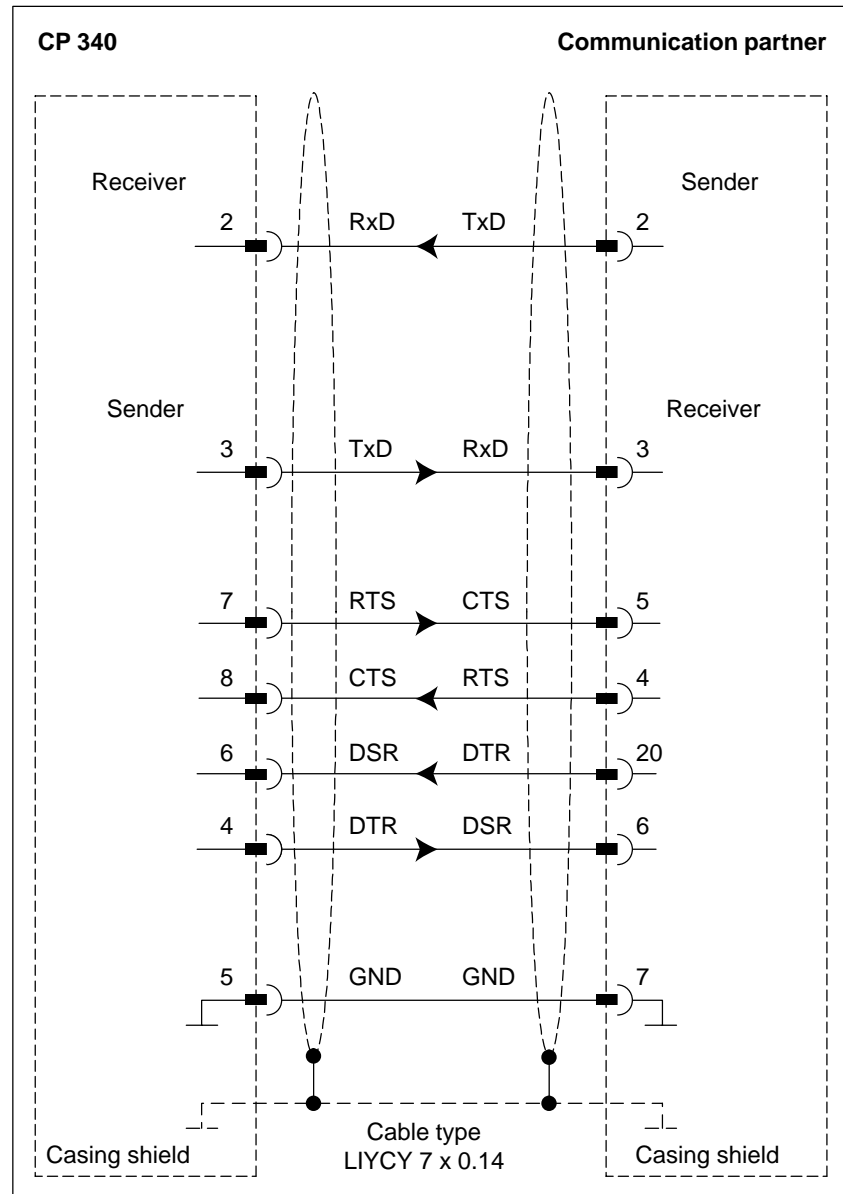


Figure B-3 RS 232C Connecting Cable CP 340 – CP 521 SI/CP 521 BASIC

**RS 232C
Connecting Cables
(S7/M7 (CP 340) –
CP 523)**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 523.

For the connecting cables you require the following female/male connectors:

- At CP 340: 9-pin sub D female with screw fixing
- At communication partner: 25-pin sub D male with screw fixing

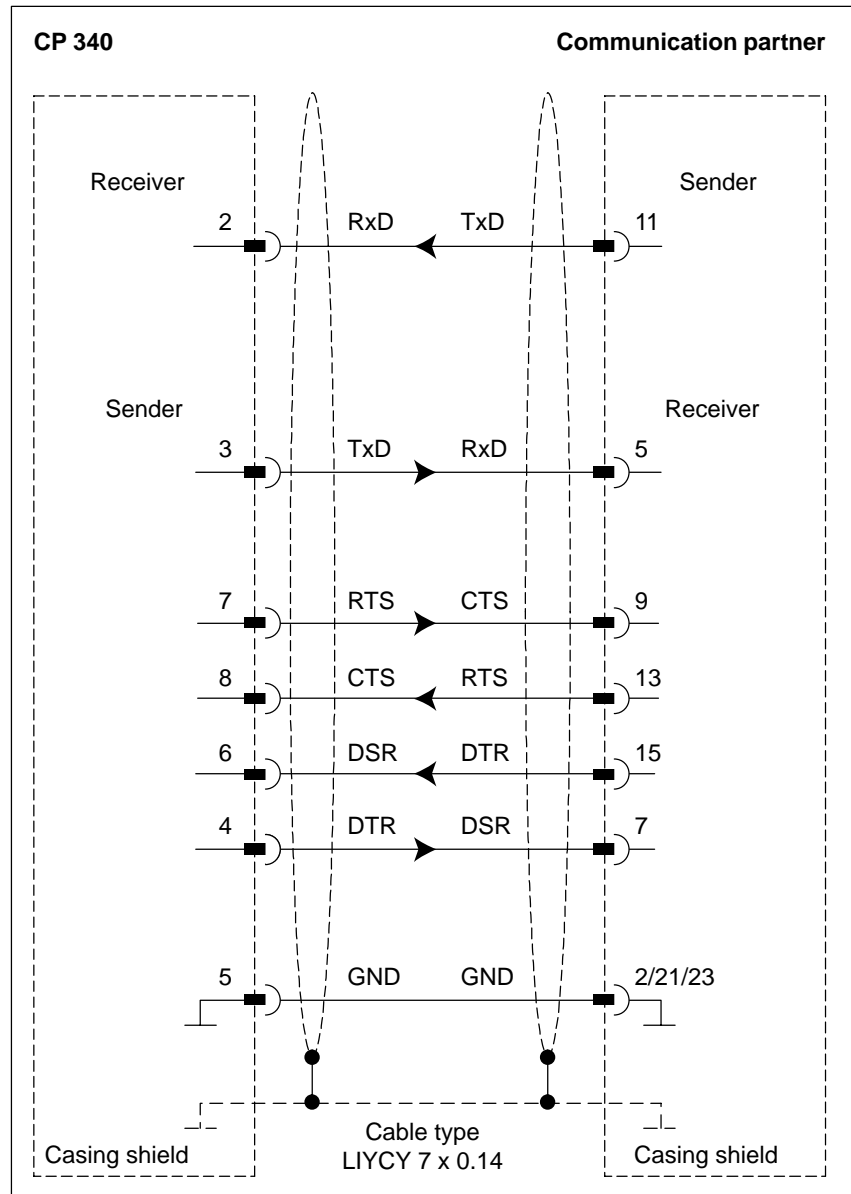


Figure B-4 RS 232C Connecting Cable CP 340 – CP 523

**RS 232C
Connecting Cable
(S7/M7 (CP 340) –
DR 2xx)**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a DR 2xx printer with a serial interface.

For the connecting cable you require the following female/male connectors:

- At CP 340: 9-pin sub D female
- At DR 2xx: 25-pin sub D male

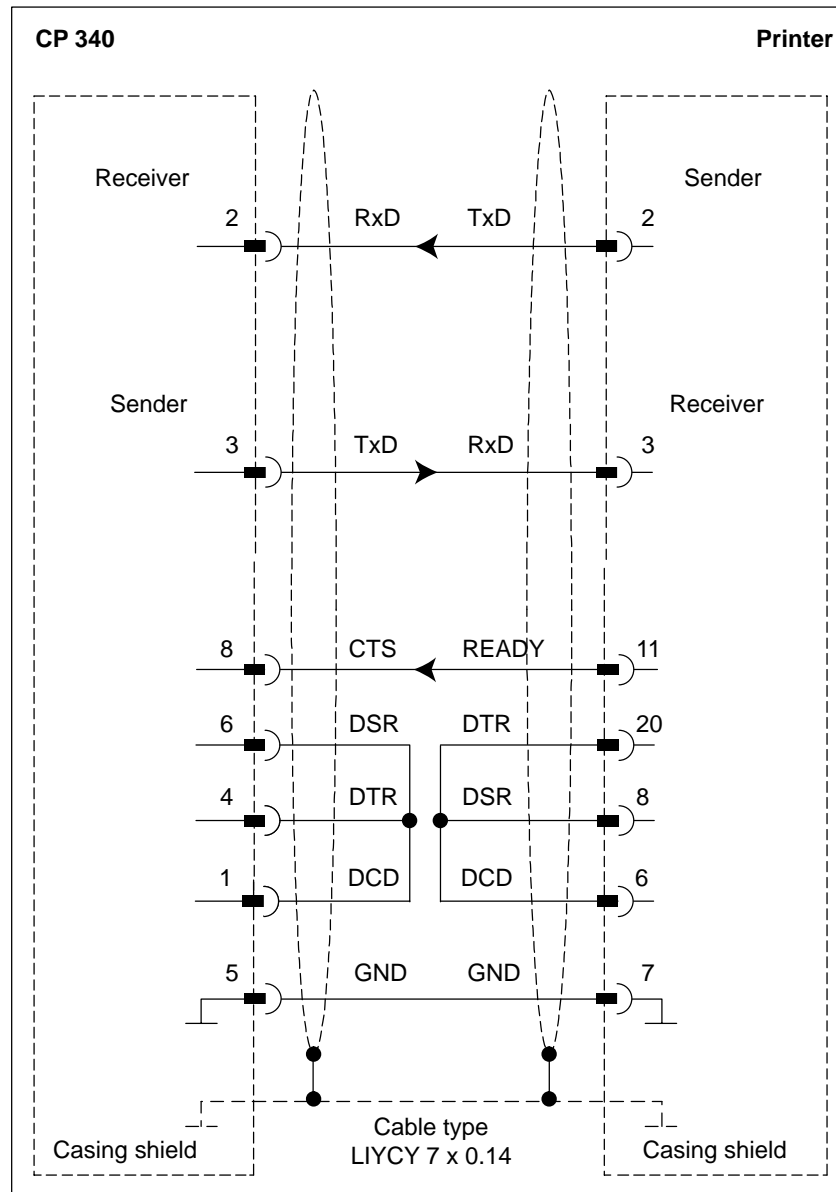


Figure B-5 RS 232C Connecting Cable CP 340 – DR 2xx

**RS 232C
Connecting Cable
(S7/M7 (CP 340) –
IBM-Proprinter
(PT 88))**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 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 CP 340: 9-pin sub D female
- At IBM Proprinter: 25-pin sub D male

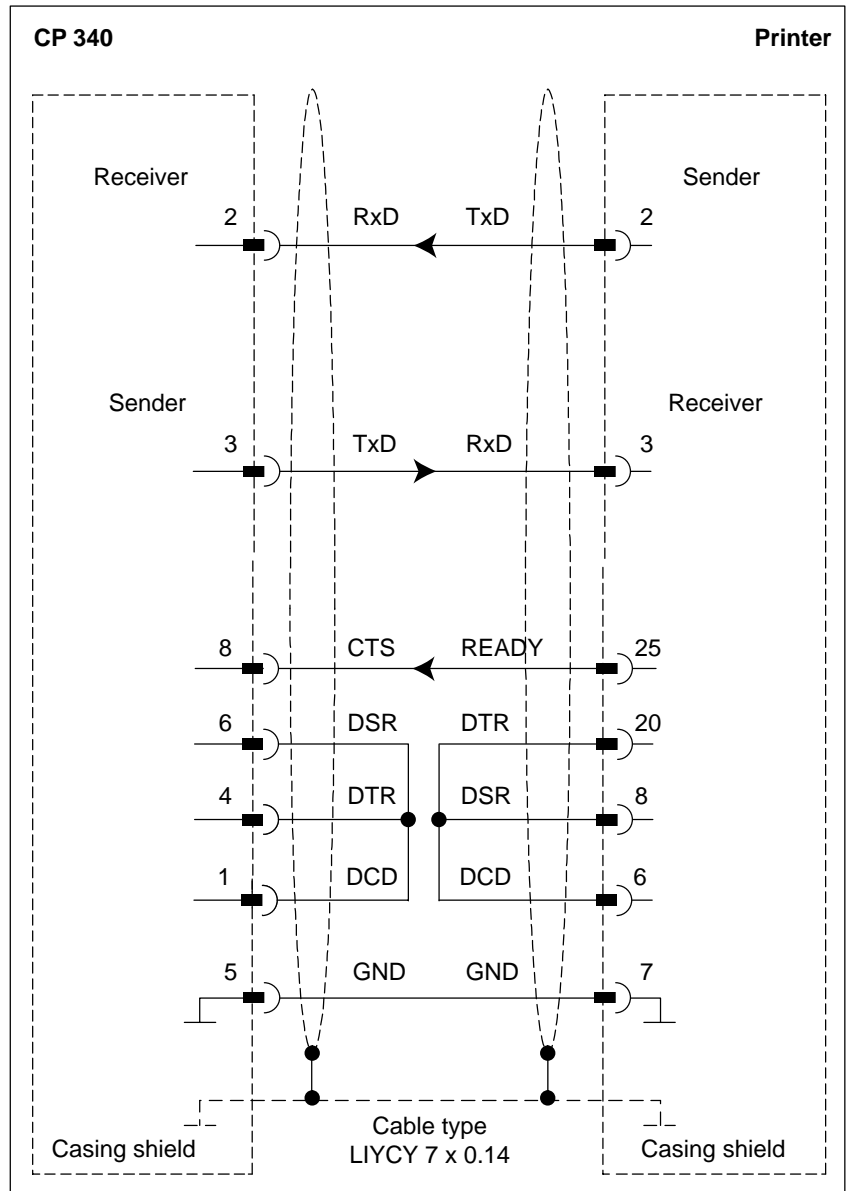


Figure B-6 RS 232C Connecting Cable CP 340 – IBM Proprinter

**RS 232C
Connecting Cable
(S7/M7 (CP 340) –
Laser Printer)**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 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 CP 340: 9-pin sub D female
- At IBM Proprinter: 25-pin sub D male

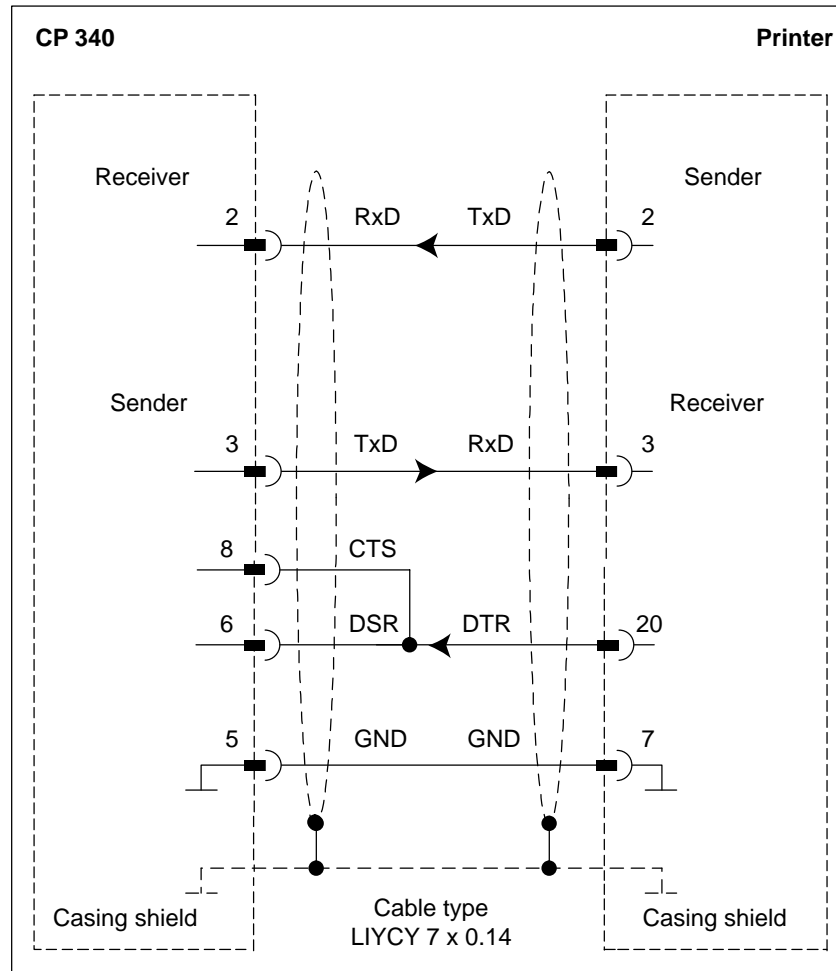
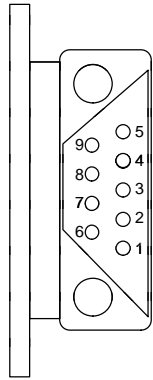


Figure B-7 RS 232C Connecting Cable CP 340 – Laser Printer

B.2 20 mA TTY Interface of the CP 340-20mA 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 340-20mA TTY.

Table B-2 Pin Allocation for the 9-Pin Sub D Female Connector of the Integrated Interface of the CP 340-20mA TTY

Female Connector on CP 340-20mA TTY*	Pin	Designation	Input/Output	Meaning
	1	TxD –	Output	Transmitted data
	2	20 mA –	Input	5 V ground
	3	20 mA + (I ₁)	Output	20 mA generator 1
	4	20 mA + (I ₂)	Output	20 mA generator 2
	5	RxD +	Input	Received data +
	6	–		
	7	–		
	8	RxD –	Output data	Received data –
	9	TxD +	Input	Transmitted data +

* View from the front

Block Diagram

The figure below is a block diagram of an IF963-TTY 20 mA TTY interface.

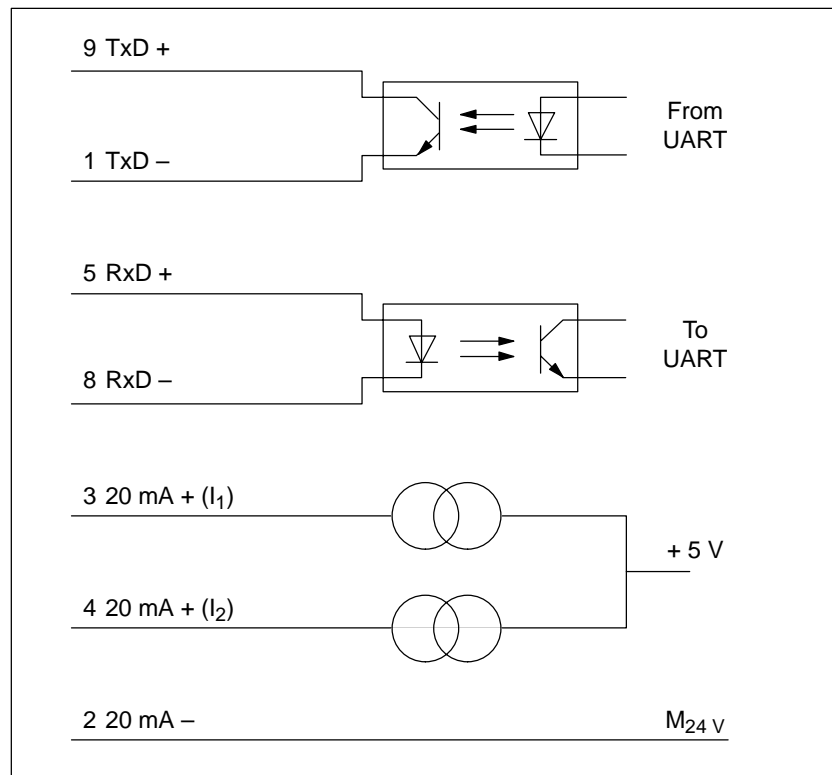


Figure B-8 Block Diagram of the IF963-TTY 20 mA TTY Interface

Connecting Cables

If you make your own connecting cables you must remember that unconnected inputs at the communication partner may have to be connected to open-circuit potential.

Please 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.

In the Following

On the following pages you will find examples of connecting cables for a point-to-point connection between the CP 340-20mA TTY and S7 modules or SIMATIC S5.

**20 mA TTY
Connecting Cable
(S7/M7 (CP 340) –
S7/M7
(CP 340/CP 441))**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 340/CP 441.

For the connecting cables you require the following male connectors:

- At CP 340: 9-pin sub D male with screw fixing
- At communication partner: 9-pin sub D male with screw fixing

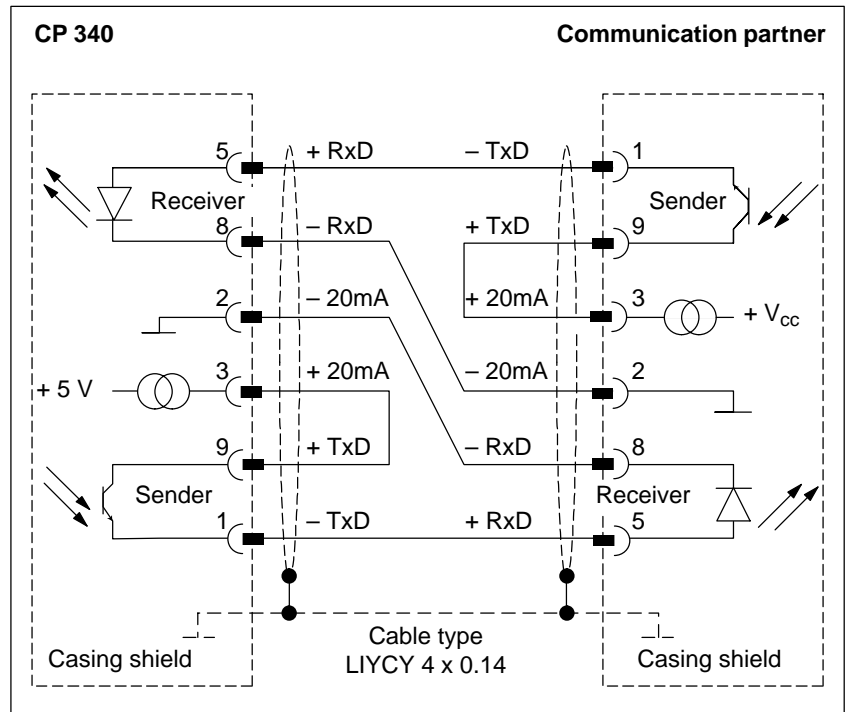


Figure B-9 20 mA TTY Connecting Cable CP 340 – CP 340/CP 441

You can order the connecting cable using the order number given in Appendix D.

**20 mA TTY
Connecting Cable
(S7/M7 (CP 340) –
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 340 and a CP 544, CP 524, CPU 928B, CPU 945, CPU 948.

For the connecting cables you require the following male connectors:

- At CP 340: 9-pin sub D male with screw fixing
- At communication partner: 25-pin sub D male with screw fixing

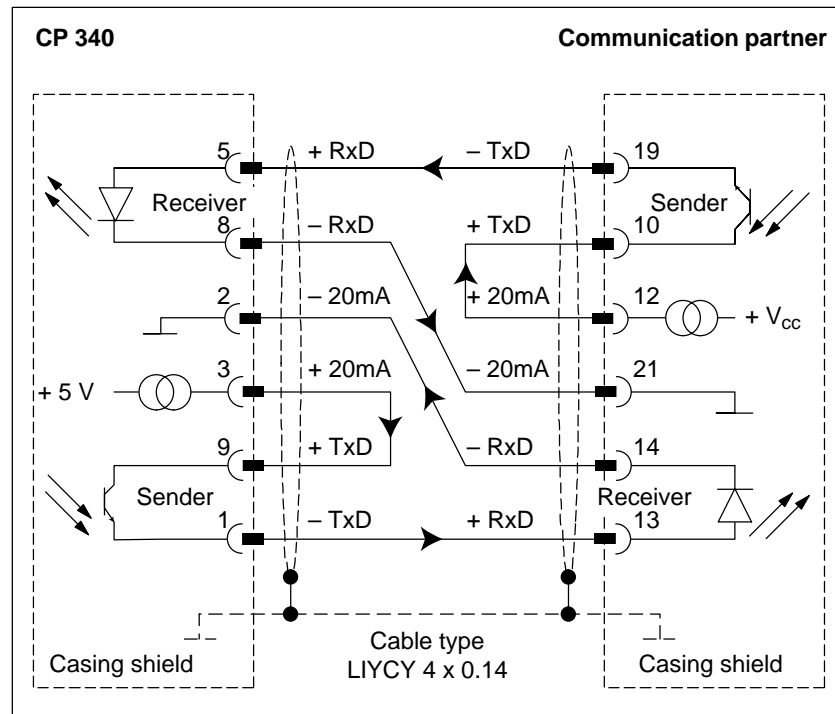


Figure B-10 20 mA TTY Connecting Cable CP 340 – CP 544, CP 524, CPU 928B, CPU 945, CPU 948

**20 mA TTY
Connecting Cable
(S7/M7 (CP 340) –
CP 523)**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 523.

For the connecting cables you require the following male connectors:

- At CP 340: 9-pin sub D male with screw fixing
- At communication partner: 25-pin sub D male with screw fixing

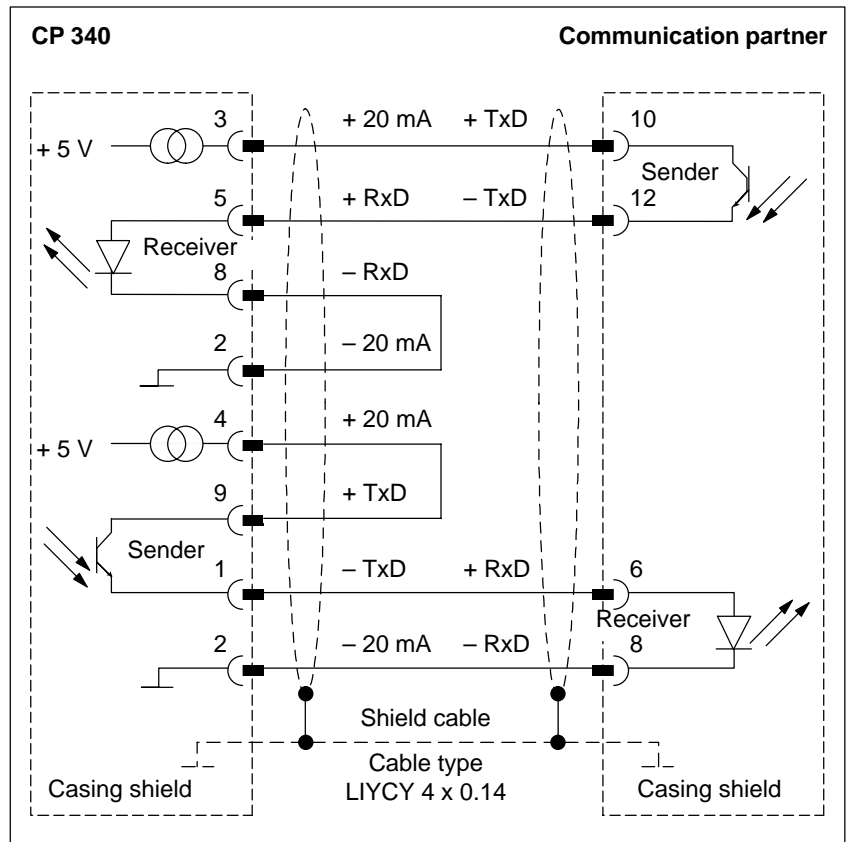


Figure B-11 20 mA TTY Connecting Cable CP 340 – CP 523

**20 mA TTY
Connecting Cable
(S7/M7 (CP 340) –
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 340 and a CP 521 SI/CP 521 BASIC.

For the connecting cables you require the following male connectors:

- At CP 340: 9-pin sub D male with screw fixing
- At communication partner: 25-pin sub D male with screw fixing

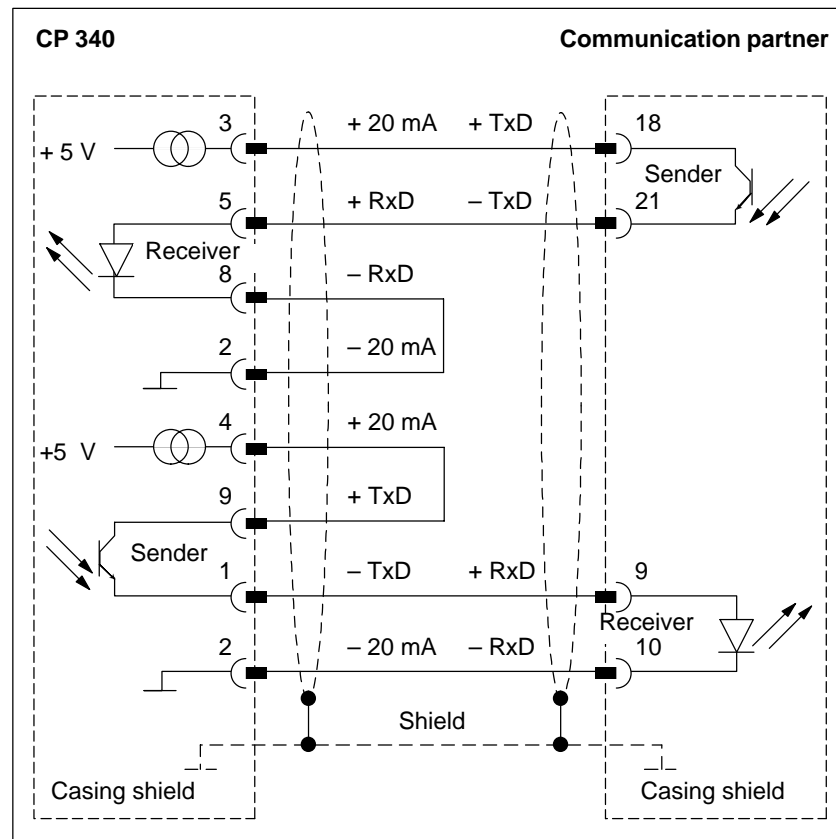


Figure B-12 20 mA TTY Connecting Cable CP 340 – CP 521SI/CP 521BASIC

**20 mA TTY
Connecting Cable
(S7/M7 (CP 340) –
CPU 944/AG 95)**

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CPU 944/AG 95.

For the connecting cables you require the following male connectors:

- At CP 340: 9-pin sub D male with screw fixing
- At communication partner: 15-pin sub D male with screw fixing

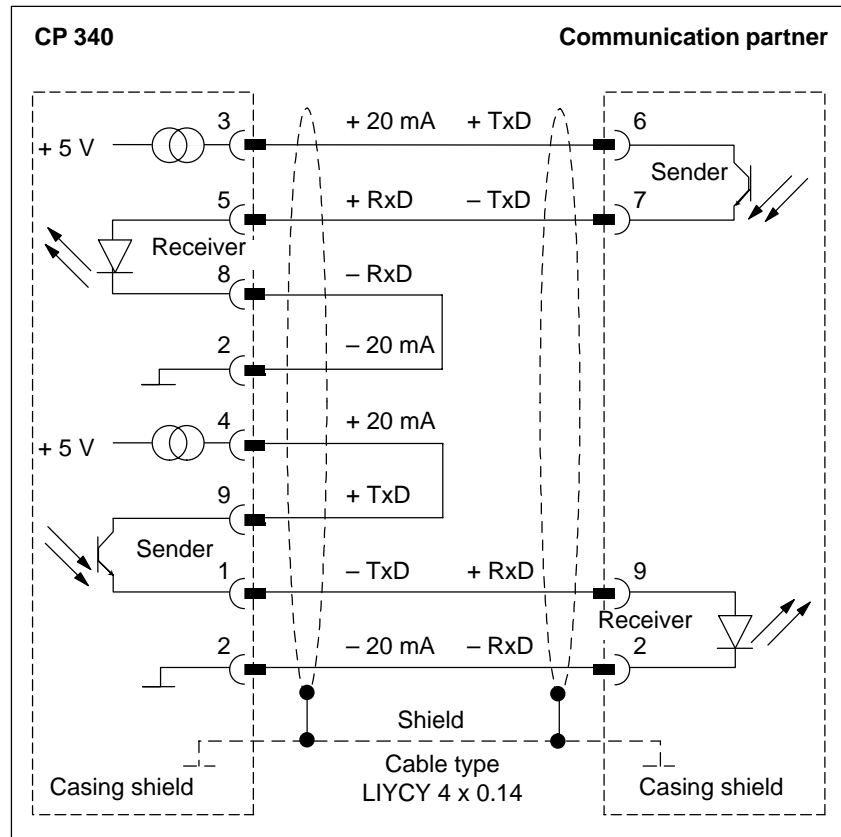
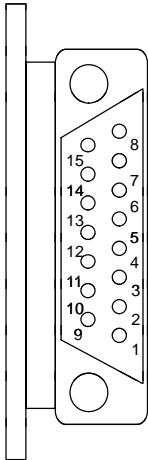


Figure B-13 20 mA TTY Connecting Cable CP 340 – CPU 944/AG 95

B.3 X27 (RS 422/485) Interface of the CP 340-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 340-RS 422/485.

Table B-3 Pin Allocation for the 15-Pin Female Connector of the Integrated Interface of the CP 340-RS 422/485

Female Connector on CP340-RS422/485*	Pin	Designation	Input/Output	Meaning
	1	–	–	–
	2	T (A)	Output	Transmitted data (four-wire mode)
	3	–	–	–
	4	R (A)/T (A)	Input Input/Output	Received data (four-wire mode) Received/transmitted data (two-wire mode)
	5	–	–	–
	6	–	–	–
	7	–	–	–
	8	GND	–	Functional ground (isolated)
	9	T (B)	Output	Transmitted data (four-wire mode)
	10	–	–	–
	11	R (B)/T (B)	Input Input/Output	Received data (four-wire mode) Received/transmitted data (two-wire mode)
	12	–	–	–
	13	–	–	–
	14	–	–	–
	15	–	–	–

* View from the front

Connecting Cables

If you make your own connecting cables you must remember that unconnected inputs at the communication partner may have to be connected to open-circuit potential.

Please 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 8), 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 340-RS 422485 and S7 modules or SIMATIC S5.

X 27 Connecting Cable (S7/M7 (CP 340) – CP 340/CP 441)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 340/CP 441 for RS 422 operation.

For the connecting cables you require the following male connectors:

- At CP 340: 15-pin sub D male with screw fixing
- At communication partner: 15-pin sub D male with screw fixing

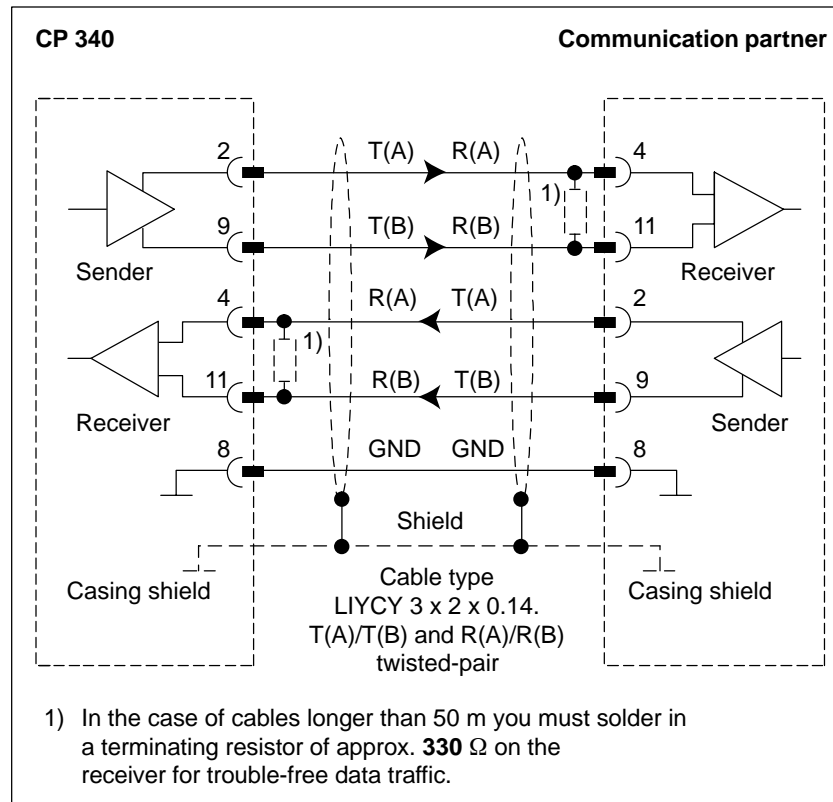


Figure B-14 X27 Connecting Cable CP 340 – CP 441/CP 340 for RS 422 Operation (Four-Wire Mode)

You can order the connecting cable using the order number given in Appendix D.

Note

The maximum length of this cable type at 19.6 Kbaud is 1200 m.

X 27 Connecting Cable (S7/M7 (CP 340) - CP 340/CP 441)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 340 and a CP 340/CP 441 for RS 485 operation.

For the connecting cables you require the following male connectors:

- At CP 340: 15-pin sub D male with screw fixing
- At communication partner: 15-pin sub D male with screw fixing

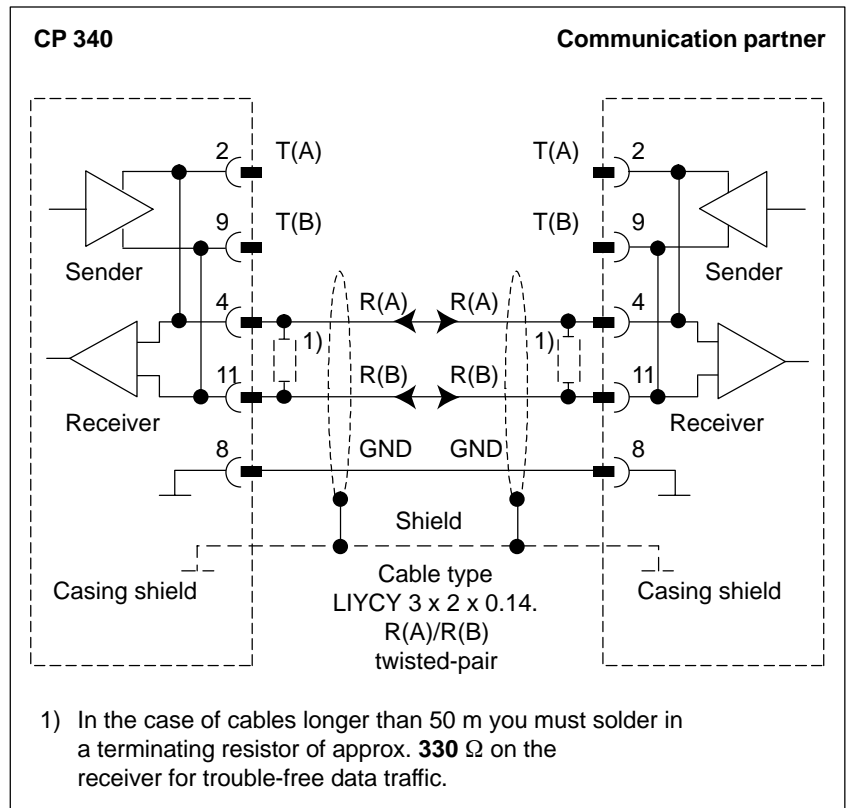


Figure B-15 X27 Connecting Cable CP 340 – CP 441/CP 340 for RS 485 Operation (Two-Wire Mode)

Note

Figure B-15 shows the wiring if you want to make the connecting cable yourself. In both RS 485 operation (two-wire mode) and RS 422 operation (four-wire mode) you can also use Siemens connecting cables (see Appendix D). If you do, the internal wiring in the connecting cable is as shown in Figure B-14.

X 27 Connecting Cable (S7/M7 (CP 340) – 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 340 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 CP 340: 15-pin sub D male with screw fixing
- At communication partner: 15-pin sub D male with screw fixing

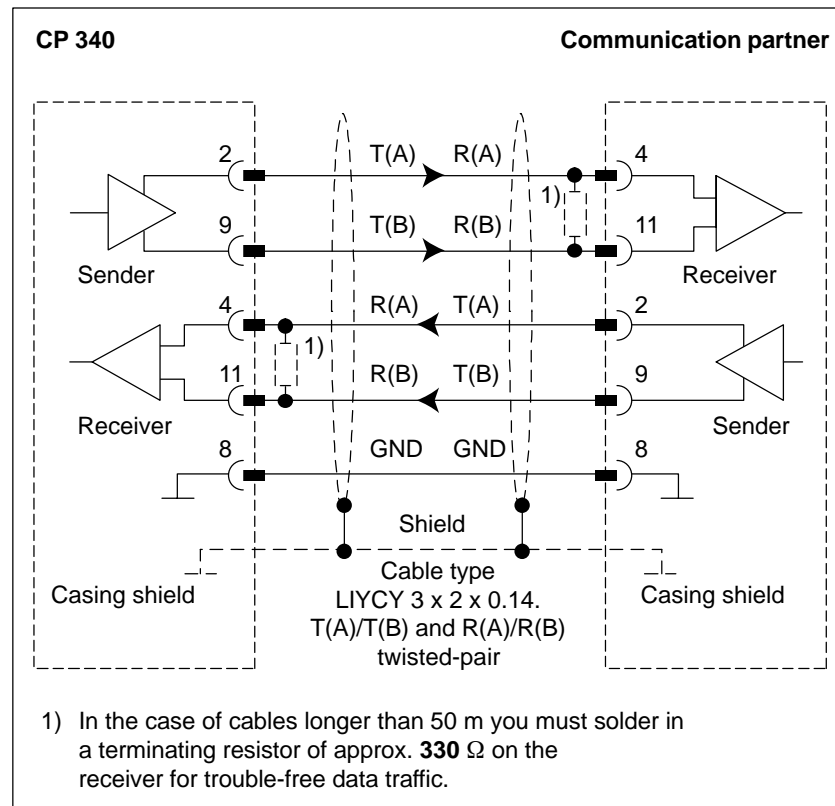


Figure B-16 X27 Connecting Cable CP 340 - CP 544, CP 524, CPU 928B, CPU 945, CPU 948 for RS 422 Operation (Four-Wire Mode)

Communication Matrix of the Protocols



Introduction

The CP 340 communications processor can communicate with the following CPs and CPUs of the SIMATIC S5 programmable controller.

Communication Matrix 3964(R)

The figure below shows the communication matrix of the 3964(R) procedure.

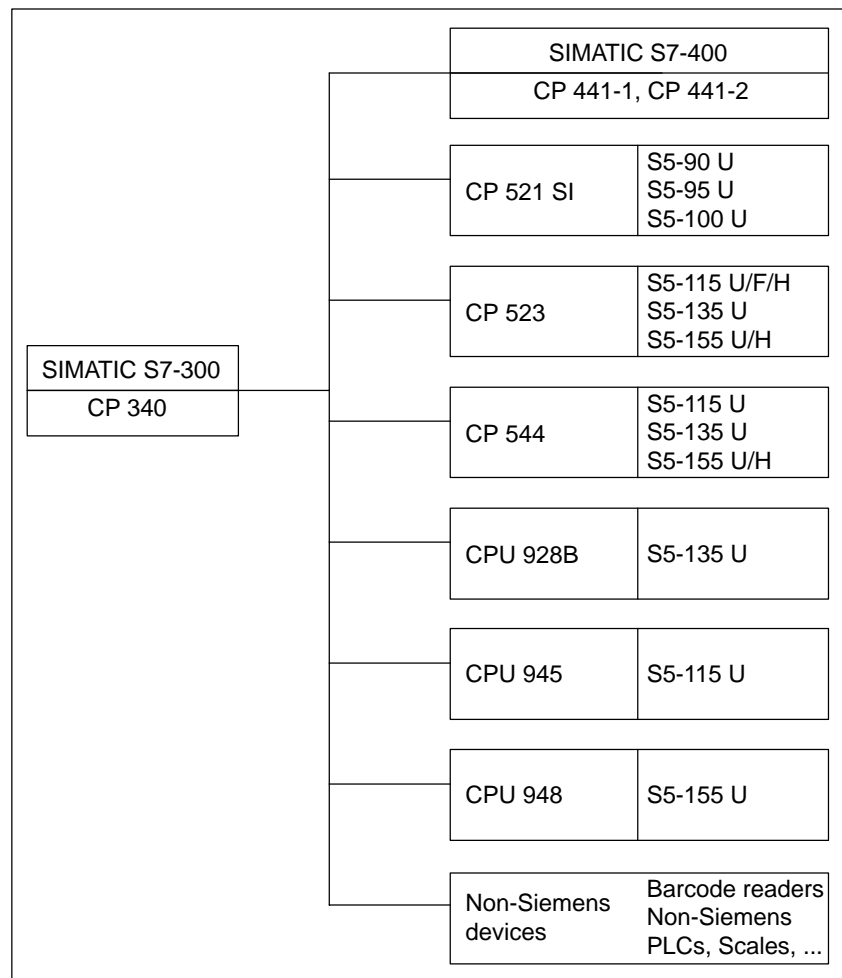


Figure C-1 Communication Matrix of the 3964(R) Procedure

Communication Matrix ASCII Driver

The figure below shows the communication matrix of the ASCII driver.

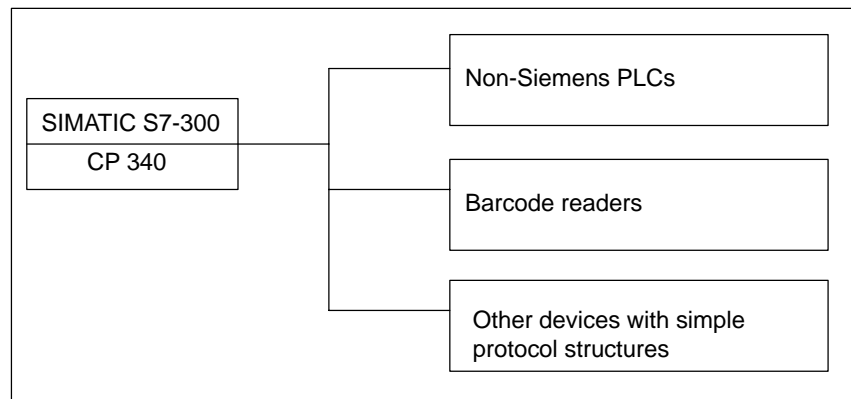


Figure C-2 Communication Matrix of the ASCII Driver

Communication Matrix Printer Driver

The figure below shows the communication matrix of the printer driver.

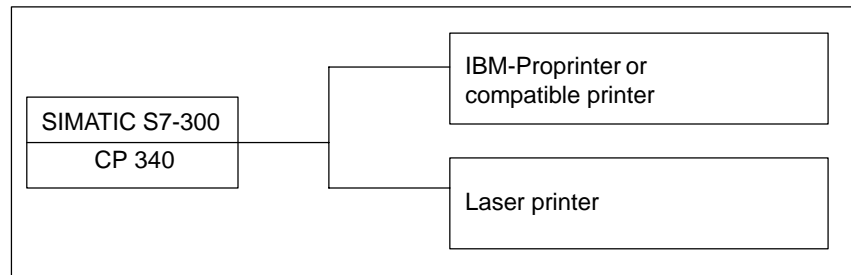


Figure C-3 Communication Matrix of the Printer Driver

Accessories and Order Numbers

D

Module Variants

The table below contains the different variants of the CP 340.

Table D-1 Order Numbers of the Module Variants of the CP 340

Product	Order Number
CP 340-RS 232C	6ES7 340-1AH01-0AE0
CP 340-20mA TTY	6ES7 340-1BH00-0AE0
CP 340-RS 422/485	6ES7 340-1CH00-0AE0

Connecting Cables

Connecting cables are available in the commonly preferred lengths: 5 m, 10 m and 50 m.

Table D-2 Order Numbers of the Connecting Cables

Connecting Cable for	Variant	Order Number
RS 232C interface	• 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	• 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 422/485) interface	• X27 (RS 422/485), 5 m	6ES7 902-3AB00-0AA0
	• X27 (RS 422/485), 10 m	6ES7 902-3AC00-0AA0
	• X27 (RS 422/485), 50 m	6ES7 902-3AG00-0AA0

SIMATIC S7 Reference Literature

E

Literature Referenced in This Manual

- /1/ *Programming with STEP 7, Manual*
- /2/ *S7-300 Programmable Controller, Hardware and Installation Manual*
- /3/ *Configuring Hardware and Communication Connections STEP 7, Manual*
- /4/ *System Software for S7-300 and S7-400, System and Standard Functions, 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>Working with STEP 7 Getting Started</i>	The Getting Started is an easy introduction to the methods for configuring and programming an S7-300/400. It is intended in particular for novice users of an automation system or an S7.
<i>Programming with STEP 7 Manual</i>	This 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 Manual</i>	The STEP 7 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 Programming Reference Manual</i>	The manuals for the STL, LAD 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 Programming Reference Manual</i>	
<i>Structured Control Language (SCL)¹ for S7-300/S7-400 Programming Manual</i>	
<i>S7-GRAPH¹ for S7-300 and S7-400 Programming Sequential Control Systems Manual</i>	With the S7-GRAPH, S7-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 Programming Continuous Function Charts Manual</i>	
<i>System Software for S7-300 and S7-400 System 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>

**Technical
Overviews**

Table E-3 contains technical overviews that provide you with an overview of the S7-300, STEP 7 and distributed I/Os in the S7/M7.

Table E-3 Technical Overviews for SIMATIC S7, STEP 7 and PROFIBUS-DP

Technical Overviews
<i>S7-300 Programmable Controller Configuration and Application</i>
<i>From SIMATIC S5 to SIMATIC S7 The Way to a Smooth Transition</i>
<i>S7-300/400 Programmable Controllers Programming</i>
<i>S7/M7 Programmable Controllers Distributed I/O with PROFIBUS-DP and AS-I</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 340 has its own diagnostic buffer, in which all the diagnostic events of the CP 340 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 340: Point-to-Point Communication, Parameter Assignment The CP 340: Point-to-Point Communication, Parameter Assignment parameterization interface is used to parameterize the submodules of the communications processor and configure the message texts for printer output.

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.

User Program 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-10
3964(R) procedure, 2-12
 block checksum, 2-13
 control characters, 2-12
 parameter, 2-37

A

Accessories, D-1
Acknowledgment delay time, 2-39
Actual operand, 6-24
 absolutely addressed, 6-24
 symbolically addressed, 6-24
Addressing the module, 6-25
ASCII driver, 2-21
 data flow control, 2-31, 2-35
 parameter, 2-43
 receive buffer, 2-27
 receiving data, 2-23
 RS 232C secondary signals, 2-28
 sending data, 2-22

B

Backplane connector for S7 backplane bus, 1-8
Basic parameter, 2-36
Baud rate, 2-40, 2-45, 2-50
Bit character frame, 11-bit character frame, 2-5
Block call
 P_PRINT, 6-13
 P_RCV, 6-9
 P_SEND, 6-5
 V24_SET, 6-19
 V24_STAT, 6-17
Block checksum, 2-13
BUSY signal, 2-35

C

CE, marking, A-7

Character delay time, 2-6, 2-39, 2-44
Character frame, 2-40, 2-45, 2-50
Character frames, 2-4
 10-bit-character frame, 2-4
 11-bit character frame, 2-5
Character set, in message texts, 2-32, 2-54
Checking the interface statuses of the CP 340,
 6-17
Communication, S7 sends data to a
 communication partner, 6-4
Communication matrix, C-1
 3964(R), C-1
 ASCII driver, C-2
 printer driver, C-2
Communication via function blocks, 6-2
Communications processor, 1-5
Configuration, 5-2
Connecting cables, B-2
Control characters, 2-12
 in message texts, 2-32, 2-54, 2-63
Conversion statements
 in message texts, 2-57
 representation types, 2-58
CP 340 parameterization screen forms
 configuration software, 1-6
CP 340 slots, 4-2
CPU, 1-5
CPU-RUN, 7-4
CPU-STOP, 7-4
CSA, A-8

D

Data bits, 2-40, 2-45, 2-50
Data block assignment, 6-21
Data flow control, 2-31, 2-35, 2-47, 2-52
Data transmission
 asynchronous, 2-3
 serial, 2-2
Diagnosis information, 8-14
Diagnostic buffer, 8-15
Diagnostics alarm, 2-36, 8-2, 8-12

- Diagnostics functions, 8-2
- Diagnostics information, 8-13
- Disable interrupts, 6-25
- Dismounting the CP 340, 4-3
- Display elements (LED), 8-2
- Displaying and interpreting STATUS output, 8-10
- Disposal, A-9

E

- EMC Directive, A-7
- EN/ENO Mechanism, 6-24
- End criterion, 2-24
 - end-of-text character, 2-25
 - expiry of character delay time, 2-24
 - fixed message frame length, 2-26
- End-of-text character, 2-44
- Event class, 8-4
- Event number, 8-4

F

- FM, approval, A-8
- Format string, 2-33, 2-56
- Full-duplex operation, 2-3
- Function blocks, 1-6, 6-3
 - FB 2 P_RCV, 6-10
 - FB 3 P_SEND, 6-6
 - FB 4 P_PRINT, 6-15
- Functions, 6-3
 - FC 5 V24_STAT, 6-18
 - FC 6 V24_SET, 6-20

G

- Group alarm LED, 8-3

H

- Half-duplex operation, 2-3
- Handshaking, 2-35

I

- IEC 1131, A-7
- Indicator for end of receive message frame, 2-44

- Initialization, 7-3
- Interface, 1-8
 - 20 mA TTY, 1-10, B-10
 - RS 232C, B-2
 - technical specifications, A-3
 - uses of, 1-3
 - X27 (RS 422/485), 1-11, B-17
- Interrupt behavior, 6-25
- ISO 7-layer reference model, 2-8

L

- LED displays, 1-8
- Loading the configuration and parameters, 5-4

M

- Managing the parameter data, 5-4
- Memory requirements, 6-26
- Message frame length when received, 2-44
- Message texts, 2-32, 2-55, 2-56, 6-12
 - "outputting", 6-12
 - character conversion table, 2-32
 - control character table, 2-32
 - control characters, 2-32
 - control statements, 2-63
 - conversion statements for variables, 2-57
 - format string, 2-33, 2-56
 - page layout, 2-32, 2-53
 - parameter, 2-55
 - variables, 2-32, 2-56
- Minimum number of CPU cycles, 6-27
- Mounting rack, 1-5
- Mounting the CP 340, 4-3

O

- Operating mode transitions, 7-4
- Operating modes, 7-2
- Order numbers, D-1
- Organization block OB 82, 8-12

P

- Page layout, in message texts, 2-32, 2-53
- Page number, setting, 2-59

Parameterization, 5-3, 6-22, 7-3

 direct, 6-22

 indirect, 6-22

 of data words, 6-23

Parameterization data, 2-36

 3964(R) procedure, 2-37

 ASCII driver, 2-43

 basic parameter, 2-36

 printer driver, 2-50

Parameters

 FB 2 P_RCV, 6-10

 FB 3 P_SEND, 6-6

 FB 4 P_PRINT, 6-15

 FC 5 V24_STAT, 6-18

 FC 6 V24_SET, 6-20

Parity, 2-40, 2-45, 2-51

Point-to-point connection, 2-2

Points to note

 when receiving message frames, 7-5

 when sending message frames, 7-5

Power supply module (PS), 1-5

Printer driver, 2-32

 BUSY signal, 2-35

 configuring message texts, 2-56

 message texts, 2-32, 2-56

 outputting a message text, 2-34

 parameters, 2-50

Priority, 2-41

Procedure, 2-7

 handling errored data, 2-18

 initialization conflict, 2-19

Procedure errors, 2-20

Programming device, 1-6

Programming device cable, 1-6

Protocol, 2-7, 2-38

Protocol parameters, 2-39, 2-44

R

Reading and controlling the RS 232C secondary signals, 6-17

Receive buffer, 2-27, 2-42, 2-48

Receiving data, 2-16

 ASCII driver, 2-23

 with 3964(R), 2-16

Recycling, A-9

Reparameterization, 7-2

RS 232C interface, 1-9, B-2

RS 232C secondary signals, 2-28

 automatic use, 2-29

RUN, 7-2

Runtimes, 6-26

S

S7 receives data from a communication partner, 6-8

S7-300 backplane bus, 8-2

Scope of This Manual, iii

Sending data, 2-14

 with 3964(R), 2-14

 with the ASCII driver, 2-22

Setting/resetting interface outputs of the CP 340, 6-19

Setup attempts, 2-39

SFCERR variable, "calling", 8-11

Special LEDs, 8-3

Standard connecting cable, 1-6, 1-12

Start bit, 2-40, 2-45, 2-50

Start-up characteristics, 7-3

Start-up behavior, 6-25

 FB-CP 340 connection, 6-25

 PLC-CP 340 connection, 6-25

STATUS output of the FBs, 8-2

STEP 7 software package, 1-6

STOP, 7-2

Stop bits, 2-40, 2-45, 2-50

Submodules, 1-5

System functions used, 6-27

T

Technical specifications, A-1

Transmission attempts, 2-39

Transmission integrity, 2-9

 with 3964(R), 2-10

 with the ASCII driver, 2-10

 with the printer driver, 2-10

U

UL, A-8

Uni-/bidirectional data traffic, 2-2

V

Variables, in message texts, 2-32

X

X27 (RS 422/485) interface, B-17

Siemens AG
A&D AS RD 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:

- | | |
|--|--|
| <input type="checkbox"/> Automotive | <input type="checkbox"/> Pharmaceutical |
| <input type="checkbox"/> Chemical | <input type="checkbox"/> Plastic |
| <input type="checkbox"/> Electrical Machinery | <input type="checkbox"/> Pulp and Paper |
| <input type="checkbox"/> Food | <input type="checkbox"/> Textiles |
| <input type="checkbox"/> Instrument and Control | <input type="checkbox"/> Transportation |
| <input type="checkbox"/> Nonelectrical Machinery | <input type="checkbox"/> Other _ _ _ _ _ |
| <input type="checkbox"/> Petrochemical | |



