SIEMENS

SIMATIC

S7-300
CP 341 Point-to-Point Communication, Installation and Parameter Assignment

Manual

04/2011
A5E02191071-03

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER

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

⚠WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the relevant information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E02191071-03
℗ 07/2011

Preface

Purpose of this manual

The information in this manual enables you to set up and commission a point-to-point connection.

Contents of the manual

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

The following subjects are covered:

- The basics of the point-to-point connection with the CP 341
- Commissioning the CP 341
- Mounting the CP 341
- Communication via the CP 341
- Troubleshooting
- Application example
- Properties and technical specifications

Scope of the manual

The manual is relevant for:

Product	Order number	as of product version
CP 341-RS 232C	6ES7 341-1AH02-0AE0	01
CP 341-20mA TTY	6ES7 341-1BH02-0AE0	01
CP 341-RS 422/485	6ES7 341-1CH02-0AE0	01

Note

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

Conventions

The abbreviation CP 341 is used in this documentation when information applies to all three module variants: CP 341-RS232C, CP 341-20mA TTY and CP 341-RS 422/485.

Navigation features of this manual

The manual's navigation features outlined below support quick access to specific information:

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

Electronic manuals

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

Recycling and disposal

The CP 341 is an environment-friendly product. It's characteristic features include:

- In spite of its excellent resistance to fire, the flame-resistant agent in the plastic used for the housing does not contain halogens.
- Laser inscriptions (i.e. no labels)
- Plastics identification in accordance with DIN 54840
- Fewer materials used due to size reduction; fewer parts due to integration in ASICs

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

Please contact a certified waste disposal company for eco-friendly recycling and to dispose of your old device.

Additional support

If you have any unanswered questions regarding use of the products described in this manual, please contact your local Siemens representative.

- You can locate your contact partner on this Internet URL:

  http://www.siemens.de/automation/partner (http://www.siemens.com/automation/partner)

- The portal to our technical documentation for the various SIMATIC products and systems is available at:

  http://www.siemens.de/simatic-doku (http://www.siemens.com/simatic-doku)

  The online catalog and the online ordering system are available at:

  http://www.siemens.de/automation/mall (http://www.siemens.com/automation/mall)

Training center

We offer a range of relevant courses to help you to get started with the SIMATIC S7 automation system. Please contact your regional training center or the central training center in D90327 Nuremberg, Germany.

- Internet: http://www.siemens.com/sitrain (http://www.siemens.com/sitrain)

Technical Support

How to contact Technical Support for all Industry Automation and Drive Technology products

- Use the Web form for the support request

  http://www.siemens.com/automation/support-request
  (http://www.siemens.com/automation/support-request)

Additional information about our technical support is available on the Internet at
http://www.siemens.de/automation/service&support
(http://www.siemens.com/automation/service&support)

Service & Support on the Internet

In addition to our documentation, we offer a comprehensive knowledge base on the Internet.
http://www.siemens.de/automation/service&support
(http://www.siemens.com/automation/service&support)

There you will find:

- The newsletter, which constantly provides you with up-to-date information on your products
- Relevant documentation for your application, which you can access via the search function in Product Support
- the bulletin board, a worldwide knowledge exchange for users and experts.
- Your local Industry Automation and Drive Technology representative in our contact database
- Information about repairs, spare parts and consulting.

See also

Literature on SIMATIC S7 (Page 237)

Table of contents

Product Description 1

1.1 Possible applications for the CP 341

The CP 341 communication processor enables you to exchange data between automation devices or computers by means of a point-to-point connection.

Functionality of the CP 341

The CP 341 communication processor provides the following functionality:

- Transmission rate up to 115.2 kbaud, half duplex
- Integration of the most important transmission protocols in the module firmware:
  - 3964(R) procedure
  - RK 512 computer link
  - ASCII driver
  - Printer driver
- Subsequent loading of other drivers (transmission protocols) with the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface.
- Adaptation of the transmission protocols using the **CP 341: Point-to-Point Communication, Parameter Assignment** 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 following table).

Module variants

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

Table 1- 1 CP 341 module variants

Module	Order number	Integrated interface
CP 341-RS 232C	6ES7 341-1AH02-0AE0	RS 232C interface
CP 341-20mA-TTY	6ES7 341-1BH02-0AE0	20mA-TTY interface
CP 341-RS 422/485	6ES7 341-1CH02-0AE0	X27 (RS 422/485) interface

Functions of module variants

Different driver functions can be used depending on the module variant of the CP 341:

Table 1- 2 Functions of CP 341 module variants

Function	CP 341- RS 232C	CP 341- 20mA-TTY	CP 341-RS 422/485	
			RS 422*	RS 485*
ASCII driver:	**Yes**	**Yes**	**Yes**	**Yes**
Control of RS 232C accompanying signals	Yes	No	No	No
Controlling/reading of RS 232C accompanying signals with FBs	Yes	No	No	No
RTS/CTS flow control	Yes	No	No	No
XON/XOFF flow control	Yes	Yes	Yes	No
3964(R) procedure	**Yes**	**Yes**	**Yes**	**No**
RK 512 computer link	**Yes**	**Yes**	**Yes**	**No**
Printer driver	**Yes**	**Yes**	**Yes**	**Yes**
* The RS 422 and RS 485 are differentiated through parameter assignment.				

Uses of the CP 341

The CP 341 communication processor enables a point-to-point connection with various Siemens modules and with non-Siemens products:

- SIMATIC S5 via the 3964(R) driver or RK 512 with corresponding interface module on S5 side
- Siemens PDA terminals from the ES 2 family via 3964(R) driver
- MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) and ES 030K data acquisition terminal via 3964R driver
- PCs via the 3964(R) procedure (the following development tools are available for programming on PCs for MS DOS or for Windows: PRODAVE S5 DOS/Win 64R (6ES5 897-2VD01))
- Barcode readers via the 3964(R) or ASCII driver
- Non-Siemens PLCs via the 3964(R) driver, ASCII driver, or RK 512
- Other devices with simple protocol structures by means of appropriate protocol adaptation with the ASCII driver
- Other devices that also have a 3964(R) driver or RK 512
- Printers (HP Deskjet, HP Laserjet, Postscript, Epson, IBM)

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

The CP 341 can be operated in a PROFINET IO network if

- the corresponding PROFINET IO controller is integrated in the SIMATIC STEP 7 CPU,
- FB7 (P_RCV_RK) / FB8 (P_SND_RK) - blocks with versions higher than or equal to V3.0 are used.

Note

The CP 341 modules (6ES7 341-1xH0y-0AE0) cannot be operated downstream from the external communication CPs CP 342-5 (PROFIBUS DP) and CP 343-1 (PROFINET IO).

1.2 Design of the CP 341

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

Positions of the module elements

The diagram shows the arrangement of the module elements on the front panel of the CP 341 communication processor.

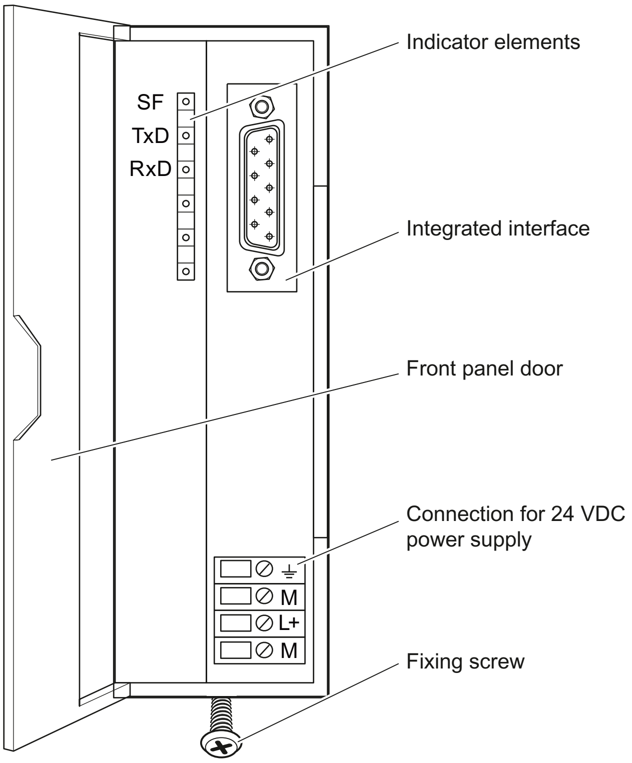


Figure 1-1 Positions of the module elements on the CP 341 communication processor

LED indicators

The following LED indicators are located on the front panel of the communication processor:

• SF	(red)	Error display
• TxD	(green)	Interface sending
• RxD	(green)	Interface receiving

Section "Diagnostics via the display elements of the CP 341 (Page 177)" describes the operating states and errors that these LEDs indicate. Section "Subsequent Loading of Firmware Updates (Page 118)" provides information on the LED indicators that occur when loading a firmware update.

Integrated interface

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

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

The interface type is indicated on the front of the CP 341. A detailed interface description can be found in Section "Properties of the Serial Interface (Page 18)".

Bus connector for the S7 backplane bus

A bus connector is supplied with the CP 341. The bus connector is inserted onto the rear panel connector of the CP 341 during installation. The S7-300 backplane bus is connected by means of the bus connector.

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

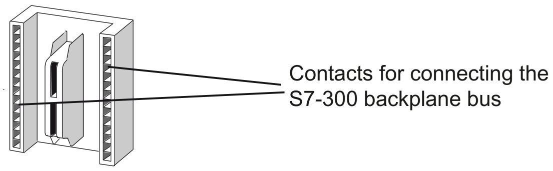


Figure 1-2 Bus connector

1.3 Components for a Point-to-Point Connection with the CP 341

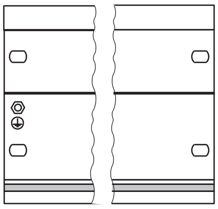
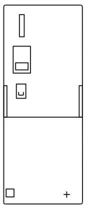
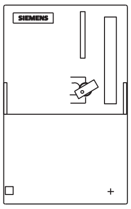
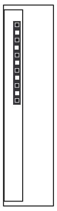
To establish a point-to-point connection between the communication processor and a communication partner, you require certain hardware and software components.

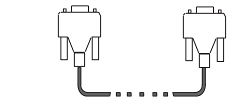
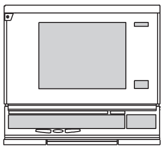
1.3.1 Required Hardware Components

Hardware components

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

Table 1- 3 Hardware components for a point-to-point connection

Component	Function	Illustration
Rack (mounting rail)	... provides the mechanical and electrical connections of the S7-300.	
Power supply module (PS)	... converts the line voltage (120/230 V AC) into the operating voltage of 24 V DC required to supply the S7-300.	
Central processing unit (CPU) Accessories: • Memory card • Backup battery	... executes the user program; communicates with other CPUs or with a programming device via the MPI interface.	
Communication processor	... communicates with a communication partner via the interface.	
Standard connecting cable	... connects the CP 341 communication processor to the communication partner.	

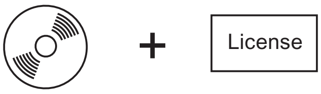
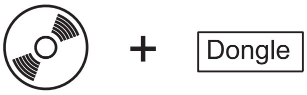
Component	Function	Illustration
PG cable	... connects a CPU to a programming device/PC.	
Programming device (PG) or PC	... communicates with the CPU of the S7-300.	

1.3.2 Required Software Components

Software components

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

Table 1- 4 Software components for a point-to-point connection with the CP 341

Component	Function	Illustration
STEP 7 software package	... configures, assigns parameters, programs and tests the S7-300.	
Parameter assignment interface: **Point-to-Point Communication, Parameter Assignment** parameter assignment interface	... assigns parameters for the interface of the CP 341.	
Function blocks (FB) with programming example	... controls the communication between CPU and CP 341.	
Loadable drivers	... with transmission protocols that can be loaded on the CP 341 in addition to the standard protocols in the module firmware.	

1.4 Properties of the Serial Interface

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

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

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

Definition

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

Properties

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

Type:	Voltage interface
Front connector:	9-pin subminiature D male connector with a screw-type fitting (compatible with the 9-pin COM port (PC/PG))
RS 232C signals:	TXD, RXD, RTS, CTS, DTR, DSR, RI, DCD, GND; all isolated against the S7-internal power supply (S7-300 backplane bus) and the external 24 V DC supply
Max. baud rate:	115.2 kbaud
Max. cable length:	15 m, cable type LIYCY 7 x 0.14 (6ES7 902-1Ax00-0AA0)
Standard:	DIN 66020, DIN 66259, EIA-RS 232C, CCITT V.24/V.28

RS 232C signals

The following table shows the meanings of the RS232C accompanying signals.

Table 1- 5 RS 232C interface signals

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

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

Definition

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

Properties

The 20mA-TTY interface has the following properties and meets the following requirements:

Type:	Current-loop interface
Front connector:	9-pin subminiature D female with screw interlock
20mA TTY signals:	two isolated 20 mA current sources, receive loop (RX) "-" and "+" transmit loop (TX) "-" and "+"; all isolated against the S7-internal power supply (S7-300 backplane bus) and the external 24 V DC supply
Max. baud rate:	19.2 kbaud
Max. cable length:	1000 m active at 9.6 kbaud[1)] (CP supplies the current loop), 1000 m passive at 9.6 kbaud [1)] (partner supplies the current loop), 500 m active, 500 m passive at 19.2 kbaud; cable type LIYCY 7 x 0.14 (6ES7 902-2Ax00-0AA0)
Standard:	DIN 66258 Part 1
[1)] The switch from active to passive is made possible through appropriate wiring on the cable connector.	

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

Definition

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

Properties

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

Type:	Differential voltage interface
Front connector:	15-pin sub-D female, with screwed interlock
RS 422 signals:	TXD (A), RXD (A), TXD (B), RXD (B), GND; all isolated against the S7-internal power supply
RS 485 signals:	R/T (A), R/T (B), GND; all isolated against the S7-internal power supply (S7-300 backplane bus) and the external 24 V DC supply
Max. baud rate:	115.2 kbaud
Max. cable length:	250 m at 115.2 kbaud 500 m at 38.4 kbaud 1200 m at 19.2 kbaud; cable type LIYCY 7 x 0.14 (6ES7 902-3Ax00-0AA0)
Standard:	DIN 66259 Parts 1 and 3, EIA-RS 422/485, CCITT V.11

Note

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

1.5 Cables for Connecting the Communication Processor to a Communication Partner

Standard Cables

Siemens offers standard cables in various lengths for point-to-point connection between the communication processor and a communication partner.

The order numbers and the length of the standard cables are listed in the appendix "Accessories and order numbers (Page 235)" of this manual.

Fabricating Your Own Cables

If you are fabricating your own cables, there are a few points to take into consideration: Refer to section "Connecting cables (Page 215)" of this manual for information about this and for the pin assignment of the sub D cable and the wiring diagrams.

Basic Principles of Serial Data Transmission 2

2.1 Serial Transmission of a Character

The system provides various networking options for the exchange of data between two or more communication partners. The simplest form of data interchange is via a point-to-point connection between two communication partners.

Point-to-point connection

With the point-to-point connection, the communication processor forms the interface between a programmable controller and a communication partner. The data is transmitted serially in the point-to-point connection with the communication processor.

Serial data transmission

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

Drivers for uni/bidirectional data traffic

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

Unidirectional data traffic:

- Printer driver

Bidirectional data traffic:

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

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

Unidirectional data traffic - printer output

In the case of printer output (printer driver), n bytes of user data is output to a printer. No characters are received. The only exception to this are individual data flow control characters (e.g., XON/XOFF).

Bidirectional data traffic - operating modes

The communication processor has two operating modes for bidirectional data traffic:

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

  Data is exchanged between one or more communication partners in both directions alternately. Half-duplex mode means that data is either being sent or received at any given moment. The exception to this may be individual data flow control characters (e.g., XON/XOFF), which can also be sent during a receive operation or received during a send operation.

- **Full-duplex mode** (ASCII driver)

  Data is exchanged between one or more communication partners in both directions simultaneously; both sending and receiving can take place at the same time. Every communication partner must be able to handle send and receive operations simultaneously.

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

Asynchronous data transmission

With the CP 341, serial data transmission occurs asynchronously. The so-called time grid synchronism (a fixed time grid for transmission of a fixed character string) is only maintained during transmission of a character. Each character to be sent is preceded by a synchronization pulse, or start bit. The length of the start-bit transmission determines the clock pulse. The end of the character transfer 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 data transmission can take place. These include:

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

Sections "Configuration data (Page 76)" and "Transmission Procedure with a Point-to-Point Connection (Page 27)" describe the role the declarations play in the various transmission procedures, and how they are parameterized.

Character frame

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

The figure below shows an example of the three data formats of the 10-bit character frame.

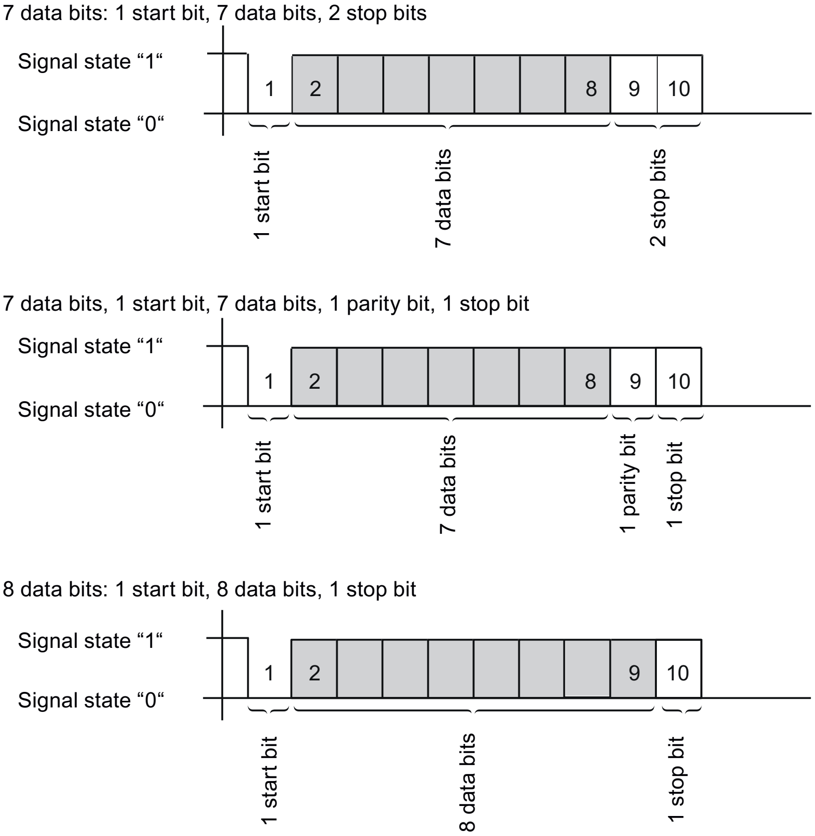


Figure 2-1 10-bit character frame

Character delay time

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

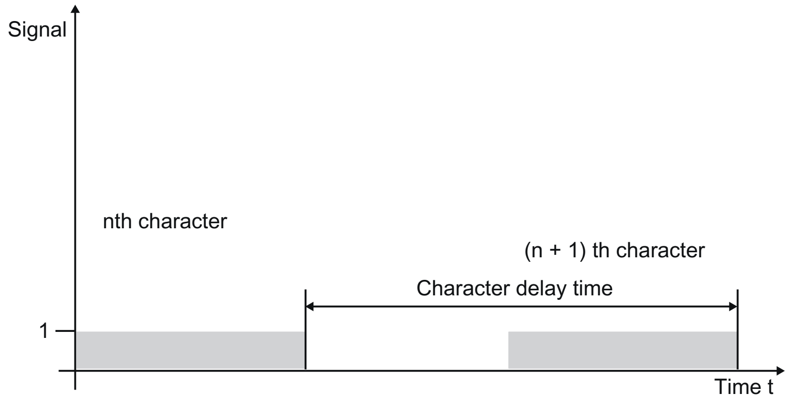


Figure 2-2 Character delay time

2.2 Transmission Procedure with a Point-to-Point Connection

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

Protocol

All communication partners must adhere to a fixed set of rules for handling and implementing data traffic. Such rules are called protocols.

A protocol defines the following:

- Operating mode

  Half-duplex or full-duplex mode
- Initiative

  Specifies which communication partner can initiate the transmission and under what conditions.
- Control characters

  Specifies which control characters are to be used for data transmission
- Character frame

  Specifies the character frame to be used for data transmission.
- Data backup

  Specifies the data backup procedure to be used
- Character delay time

  Specifies the time period within which a character to be received must be arrive.
- Transmission speed

  Specifies 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 defined as follows:

1. Physical layer
   - Physical requirements for data transmission, e.g., transmission medium, baud rate
2. Data link layer
   - Security procedure for the data transmission
   - Access method
3. Network layer
   - Definition of communication paths
   - Specification of the addressing for the data transmission between two communication partners
4. Transport layer
   - Error detection procedure
   - Corrective actions
   - Handshaking
5. Session layer
   - Setup of data transmission
   - Execution
   - Release of data transmission
6. Presentation layer
   - Implementation of the standardized communication system display type in a device-specific form (data interpretation guidelines)
7. Application layer
   - Specification of 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-oriented) to the lowest layer (No. 1, physical specifications) while the receiving communication partner processes the protocols in the reverse order, i.e., starting with layer 1.

Not every protocol has to take all seven layers into account. If the sending and receiving communication partner speak the same language, layer 6 is omitted.

2.3 Transmission integrity

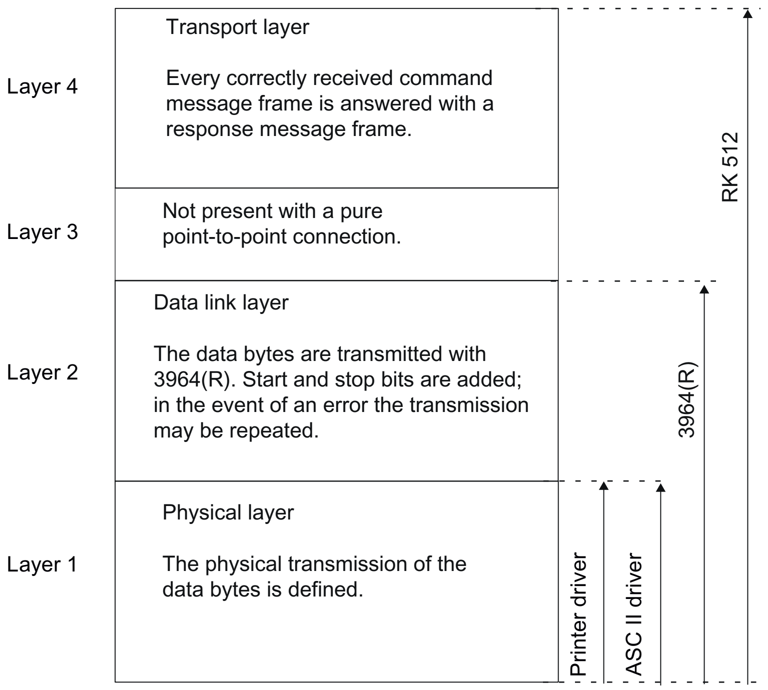
Transmission integrity plays an important role in the transmission of data and in selection of the transmission procedure. Generally speaking, the more layers of the reference model are applied, the greater the transmission integrity.

Classifying the supplied protocols

The CP 341 can use the following protocols:

- 3964(R) procedure
- RK 512 computer link
- ASCII driver
- Printer driver

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



Every correctly received command message frame is answered with a response message frame.

Not present with a pure point-to-point connection.

The data bytes are transmitted with 3964(R). Start and stop bits are added; in the event of an error the transmission may be repeated.

The physical transmission of the data bytes is defined.

Figure 2-3 Position of the supplied protocols of the CP 341 in the reference model

Transmission Integrity with the Printer Driver

Data integrity when using the printer driver:

- No data integrity precautions are taken for data transport 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 341 receives the BUSY signal as a CTS signal and evaluates it in the same way (see ASCII driver). 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 makes it possible to detect an inverted bit in a character that is to be transmitted. If two or more bits of a character are inverted, this error can no longer be detected.
- To increase transmission integrity, a checksum and length specification for a message frame can be employed. These measures must be implemented by the user.
- A further increase in data integrity can be achieved by means of acknowledgment message frames in response to send or receive message frames. This is the case with high-level protocols for data communication (ISO 7-layer reference model).

Transmission Integrity with 3964(R)

Enhanced data integrity through use of the 3964(R) procedure:

- The Hamming distance with the 3964(R) is 3. This measures the integrity of a data transmission.
- The 3964(R) procedure ensures high transmission integrity on the transmission line. This high transmission integrity is achieved by means of a specified message-frame setup and release 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: **3964R**

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

Performance limits with 3964R

- Further processing of the send/receive data by the PLC program in the communication partner is not guaranteed. You can only ensure this by using a programmable acknowledgment mechanism.
- The block check of the 3964R procedure (EXOR 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 frame along with the data itself, and having the length checked in the communication partner.
- Such transmission errors are ruled out when the RK 512 computer link is used for data transmission, because here (unlike the 3964(R) procedure) further processing (e.g., storage in the destination data block) is acknowledged via response message frames and the send data length is recorded in the message frame header. This enables the RK 512 to achieve a higher Hamming distance (of 4) than the 3964R.

Transmission Integrity with RK 512

Very high data integrity through use of RK 512:

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

Performance limits with RK 512

- Using the RK 512 computer link provides maximum data integrity! You can improve this still further by, for example, using other block check mechanisms (such as CRC checks).

2.4 Data Transmission with the 3964(R) Procedure

2.4.1 Principle of the Data Transmission with the 3964(R) Procedure

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

2.4.2 Control characters of the 3964(R) procedure

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 characters:

- STX Start of text;
  Start of character string to be transmitted
- DLE Data Link Escape;
  Data transmission switchover
- ETX End of Text;
  End of character string to be transmitted
- BCC Block check character (3964R only);
  Block check character
- NAK Negative Acknowledge;
  negative acknowledgment

Note

If DLE is transmitted as an information character, it is sent twice (DLE duplication) so that it can be distinguished from the DLE control character on the send line for connection setup and release. 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 communication partners issue a send request at the same time, the partner with the lower priority will defer its send request.

2.4.3 Block Checksum

Block Checksum

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

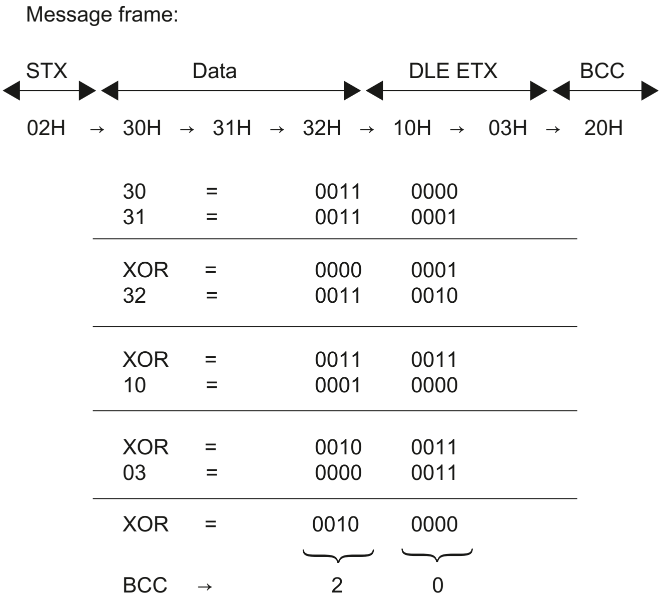


Figure 2-4 Block Checksum

The block checksum is the even longitudinal parity (EXOR 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.

2.4.4 Sending Data with 3964(R)

Sending data with 3964(R)

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

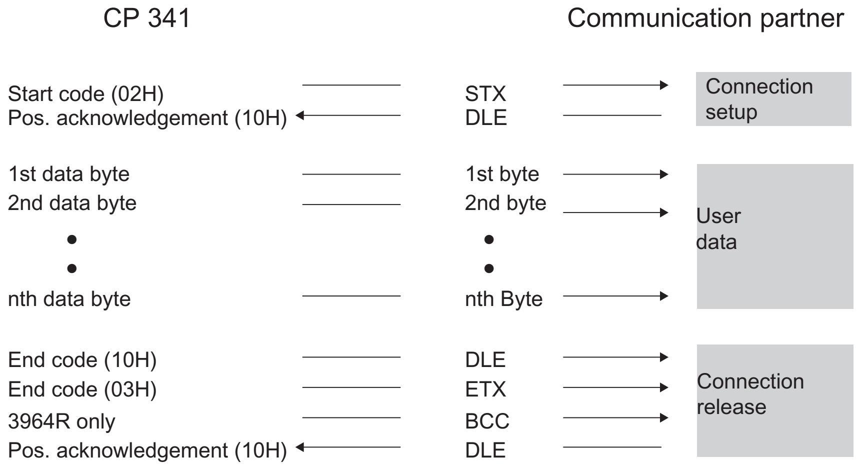


Figure 2-5 Data traffic when sending with the 3964(R) procedure

Connection setup for sending

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

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

Sending data

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

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

Connection release for sending

Once the contents of the buffer have been sent, the procedure adds the DLE and ETX characters and, with 3964R only, the block checksum BCC as the end identifier, and waits for an acknowledgment character. If the communication partner sends the DLE character within the acknowledgment delay time, it means that the data block has been received without errors. If the communication partner responds with NAK, any other character (except DLE), or a damaged character, 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 assigned number of attempts to send the data block, the procedure stops trying and sends a NAK to the communication partner. The CP 341 reports the error to the P_SND_RK function block (STATUS output parameter).

Sending with the 3964(R) Procedure

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

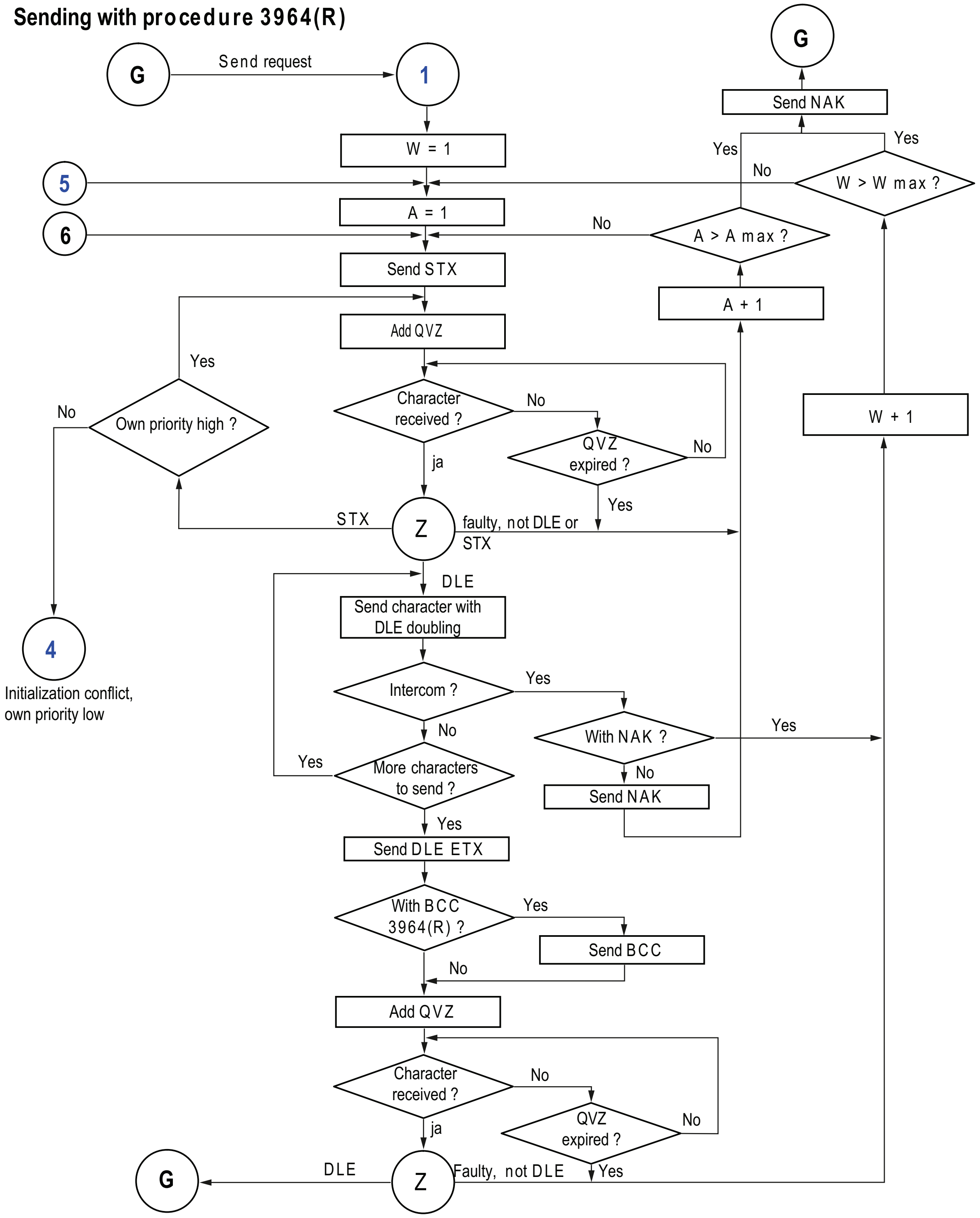


Figure 2-6 Flow diagram of sending with the 3964(R) procedure

C: Counter for connection attempts

R: Counter for retries

D: Default state

W: Waiting for character reception

2.4.5 Receiving Data with 3964(R)

Receiving data with 3964(R)

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

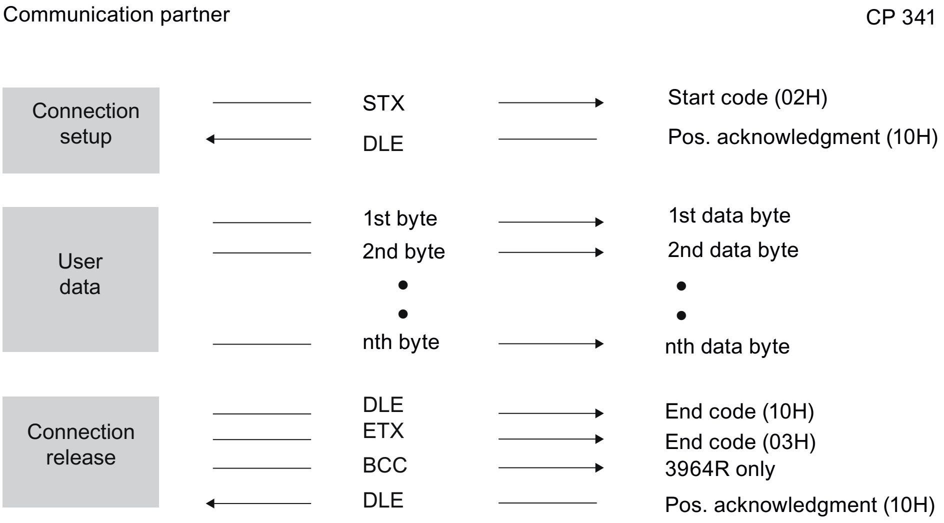


Figure 2-7 Data traffic when receiving with the 3964(R) procedure

---

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.

Connection setup for receiving

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

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

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

Receiving data

After a successful connection setup, the receive characters that arrive are stored in the receive buffer. If two consecutive DLE characters 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, the NAK character is sent to the communication partner. The system program reports the error to the P_RCV_RK function block (STATUS output parameter). The 3964(R) procedure does not initiate a repetition.

If transmission errors occur during receiving (lost character, frame error, parity error, etc.), the procedure continues to receive until the connection is released, then a NAK is sent to the communication partner. A repetition is then expected. If the block still cannot be received without errors after the number of transmission attempts defined in the static parameter set, or if the communication partner does not start the repetition within a block wait time of 4 seconds, the procedure cancels the receive operation. The CP 341 reports the first erroneous transmission and the final cancelation in the P_RCV_RK function block (STATUS output parameter).

Connection release for receiving

When the **3964** procedure detects a DLE ETX character string, it stops receiving and confirms that the block has been successfully received by sending a DLE character to the communication partner. If an error occurs while receiving, the procedure sends a NAK character to the communication partner. A repetition is then expected.

If the **3964R** procedure detects the string DLE ETX BCC, it stops receiving. It compares the received BCC block check character with the internally calculated longitudinal parity. If the BCC is correct and no other receive errors have occurred, the 3964R procedure sends the DLE character and returns to idle mode. If the BCC is faulty or a different receive error occurs, the NAK character 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.

Receiving with the 3964(R) Procedure

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

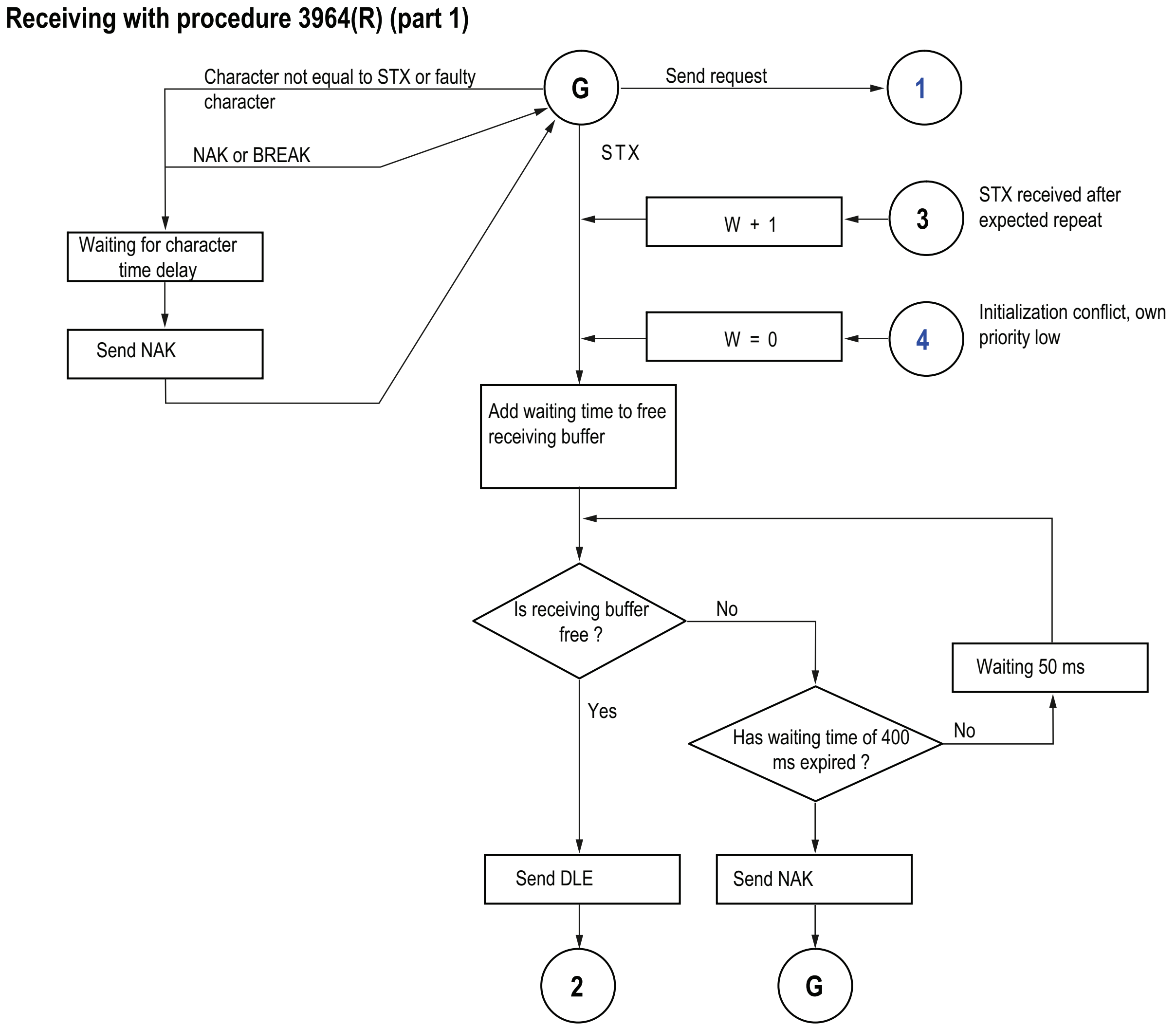

Receiving with procedure 3964(R) (part 1)
Character not equal to STX or faulty character
G
Send request
1
NAK or BREAK
STX
W + 1
3
STX received after expected repeat
Waiting for character time delay
Send NAK
W = 0
4
Initialization conflict, own priority low
Add waiting time to free receiving buffer
Is receiving buffer free ?
No
Waiting 50 ms
Yes
Has waiting time of 400 ms expired ?
No
Send DLE
Send NAK
2
G

Figure 2-8 Flow Diagram for Receiving with the 3964(R) Procedure (Part 1)

R: Counter for retries

D: Default state

Receiving with 3964(R) procedure (Part 2)

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

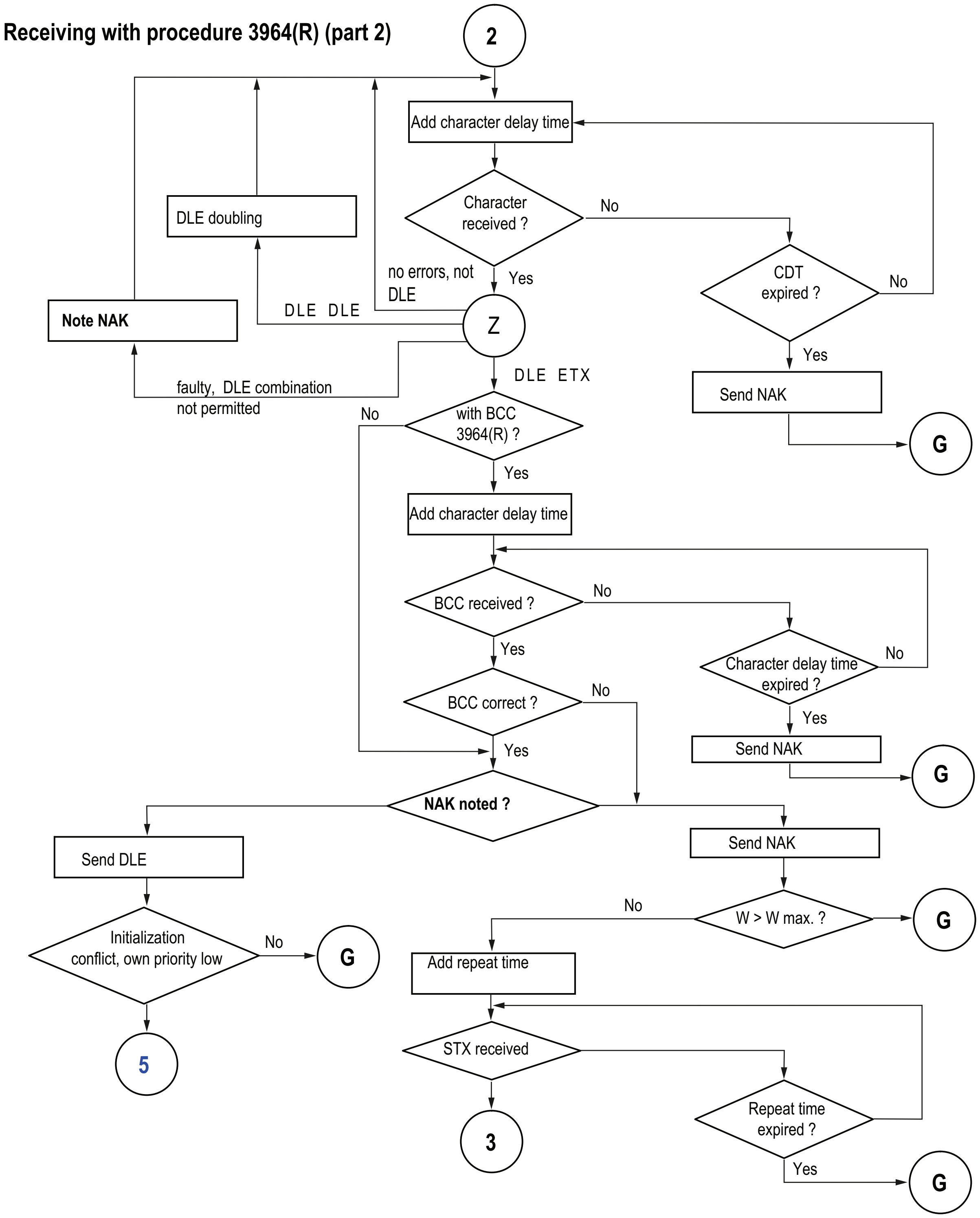


Figure 2-9 Flow diagram for receiving with the 3964(R) procedure (Part 2)

R: Counter for retries

D: Default state

W: Waiting for character reception

2.4.6 Handling Erroneous Data

Handling erroneous data

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

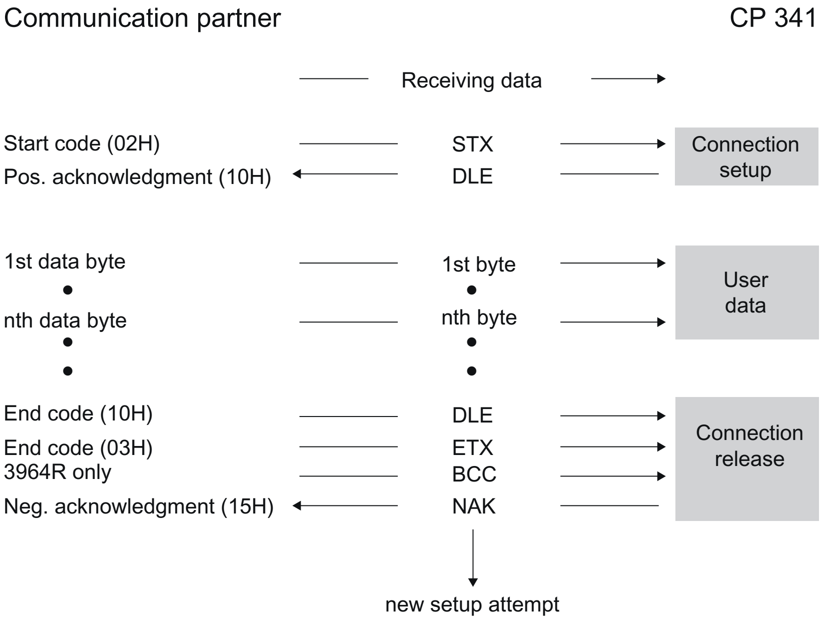


Figure 2-10 Data traffic when erroneous data is received

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

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

Extended error display at the receive FB

Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK.

If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.

Initialization conflict

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

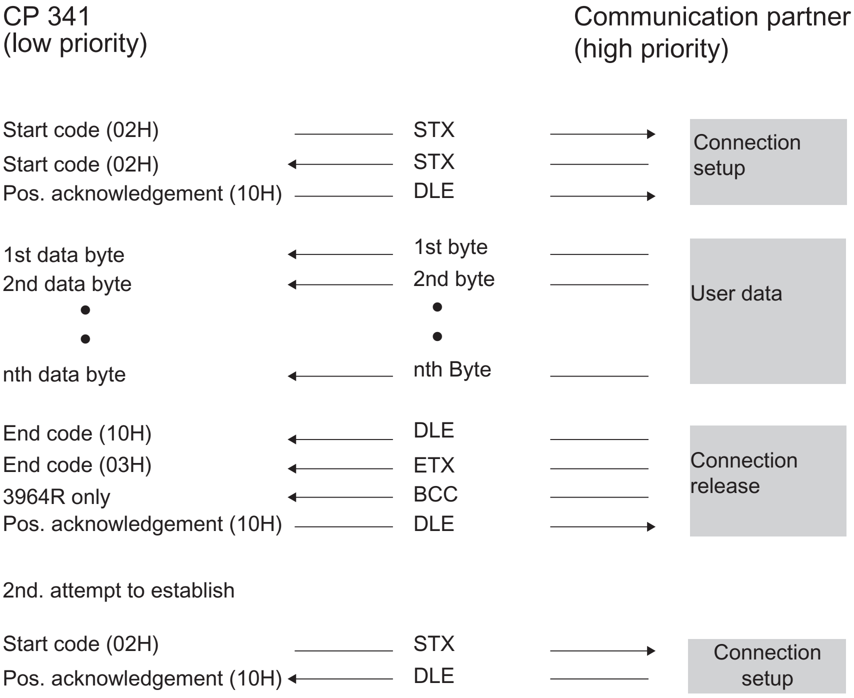


Figure 2-11 Data traffic in the event of an initialization conflict

If a device responds to the communication partner's send request (STX) within the acknowledgment delay time by sending the STX character instead of the DLE or NAK acknowledgment, an initialization conflict occurs. Both devices want to execute a pending send request. The device with the lower priority withdraws its send request and responds with the DLE character. 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.

In order to resolve initialization conflicts you must assign different priorities for the communication partners.

Procedure errors

The procedure recognizes both errors caused by faulty communication partner behavior and errors caused by faults on the line.

In both cases, the procedure initially attempts to send/receive the data block correctly. If the data block cannot be sent or received error-free within the maximum specified number of transmission attempts (or if a new error status occurs), the procedure cancels the send or receive operation. It reports the error number of the first error detected and returns to the idle state. These error messages are displayed in the STATUS output of the FB.

If the system program frequently reports an error number at the STATUS output of the FB for send and receive repetitions, this implies occasional disturbances in data traffic. The high number of transmission attempts 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. However, the disturbance could also be the result of a malfunction on the part of the communication partner.

In the event of a BREAK on the receive line (receive line interrupted), an error message is indicated at the STATUS output of the FB. No repetition is started. The BREAK status is automatically reset as soon as the connection is restored on the line.

For every detected 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, if previous repetition attempts have failed.

3964(R) procedure startup

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

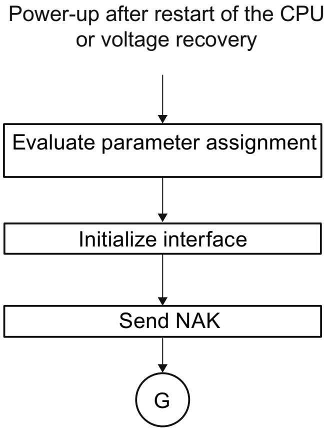

Power-up after restart of the CPU
or voltage recovery
Evaluate parameter assignment
Initialize interface
Send NAK
G

Figure 2-12 Flow diagram of the start-up of the 3964(R) Procedure

2.5 Data Transmission with the RK 512 Computer Connection

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

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

Response message frame

The RK 512 computer link answers every command message frame it receives correctly with a response message frame to the CPU (transport layer). This allows the sender to check whether its data has arrived error-free at the CPU or whether the the requested data is available on the CPU.

Command message frame

Command message frames are either SEND or FETCH message frames.

Refer to section "Communication via Function Blocks (Page 123)" for information on how to initiate a SEND or FETCH message frame.

SEND message frame

In the case of a SEND message frame, the CP 341 sends a command message frame with user data, and the communication partner replies with a response message frame without user data.

FETCH message frame

In the case of a FETCH message frame, the CP 341 sends a command message frame without user data, and the communication partner replies with a response message frame with user data.

Continuation message frame

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

Message frame header

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

Structure of the message frame header

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

Table 2- 1 Structure of command message frame header (RK 512)

Byte	Meaning	
1	Message frame ID in command message frames (00H), In continuation command message frames (FFH)	
2	Message frame ID (00H)	
3	'O' (41H)	for SEND request with destination DB or
	'O' (4FH)	for SEND request with destination DX or
	'I' (45H)	for FETCH request
4	Data to be transmitted are off (only 'D' is possible when sending):	
	'D' (44H)	=Data block
	'X' (58H)	=Expanded data block
	'I' (45H)	=Input bytes
	'O' (41H)	=Output bytes
	'M' (4DH)	=Memory bytes
	T' (54H)	=Timer cells
	'C' (5AH)	=counter cells
5	Data destination of SEND request or data source of FETCH request, e.g., byte 5 = DB no., byte 6 = DW no. (RK 512 addressing describes the data source and destination with word limits. Conversion to byte addresses in SIMATIC S7 is automatic.)	
6	Length high byte Length of data to be transmitted according to type in bytes or	
7	Length low byte Words	
8		
9	Byte number of interprocessor communication flag; FFH is displayed if you have not specified an interprocessor communication flag.	
10	Bits 0 to 3: Bit number of the interprocessor communication flag; the protocol enters FH here if you have not specified an interprocessor communication flag. Bits 4 to 7: CPU number (number from 1 to 4); if you have not specified a CPU number (number = 0) but you have specified an interprocessor communication flag, 0H is displayed here; if you specified neither a CPU number nor an interprocessor communication flag, FH is shown here.	

The letters in bytes 3 and 4 are ASCII characters.

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

Response message frame

Once the command message frame has been transmitted, the RK 512 waits for a response message frame from the communication partner within the monitoring time. The duration of the monitoring time depends by default on the transmission rate 20 s. This monitoring time can be reduced by setting parameters for the user in the "RK512" dialog of the parameter assignment interface. When the option "dependent on transmission rate" is selected, the following maximum waiting times are used for monitoring:

Table 2- 2 Monitoring time for response message frame

Transmission rate	Monitoring time
• 300 bps	10 s
• 600 bps	7 s
• 1200 bps	5 s
• as of 38400 bps	3 s

The "grayed" field "maximum waiting time" is only used to display the monitoring time used and cannot be edited!

Structure and contents of the response message frame

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

Table 2- 3 Structure of response message frame header (RK 512)

Byte	Meaning
1	Message frame ID in response message frames (00H), in continuation response message frames (FFH)
2	Message frame ID (00H)
3	Displays 00H
4	Error number of the communication partner in the response message frame:* 00H if transmission was error-free > 00H error number
* The error number in the response message frame automatically triggers an event number in the STATUS output of the function blocks (see chapter "Diagnostics Messages of the Function Blocks (Page 178)").	

2.5.1 Sending Data with RK 512

Sending data with RK 512

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

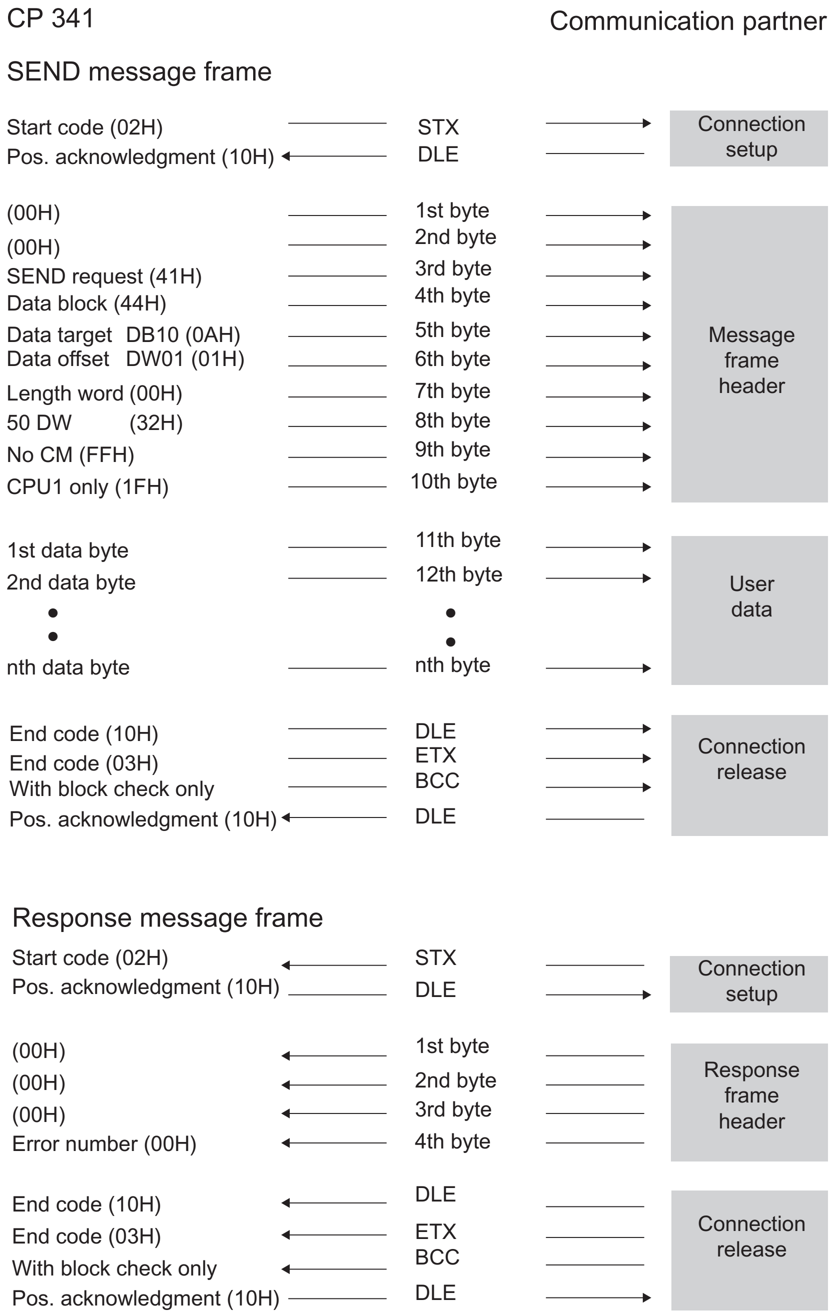


Figure 2-13 Data traffic when sending with a response message frame

Sending data

The SEND request is executed in the following sequence:

- Active partner

  Sends a SEND message frame. This contains the message frame header and data.
- Passive partner

  Receives the message frame, checks the message frame header and the data, transfers the data to the CPU, and then acknowledges this with a response message frame.
- Active partner

  Receives the response message frame.

  Sends user data.

  If the volume of user data exceeds 128 bytes, the active partner sends a continuation SEND message frame.
- Passive partner

  Receives the continuation SEND message frame, checks the message frame header and the data, transfers the data to the CPU, and then acknowledges this with a continuation response message frame.

Note

If the CPU does not receive the SEND message frame error-free or if an error has occurred in the message frame header, the communication partner enters an error number in the 4th byte of the response message frame. This does not apply when protocol errors occur.

Continuation SEND message frames

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

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

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

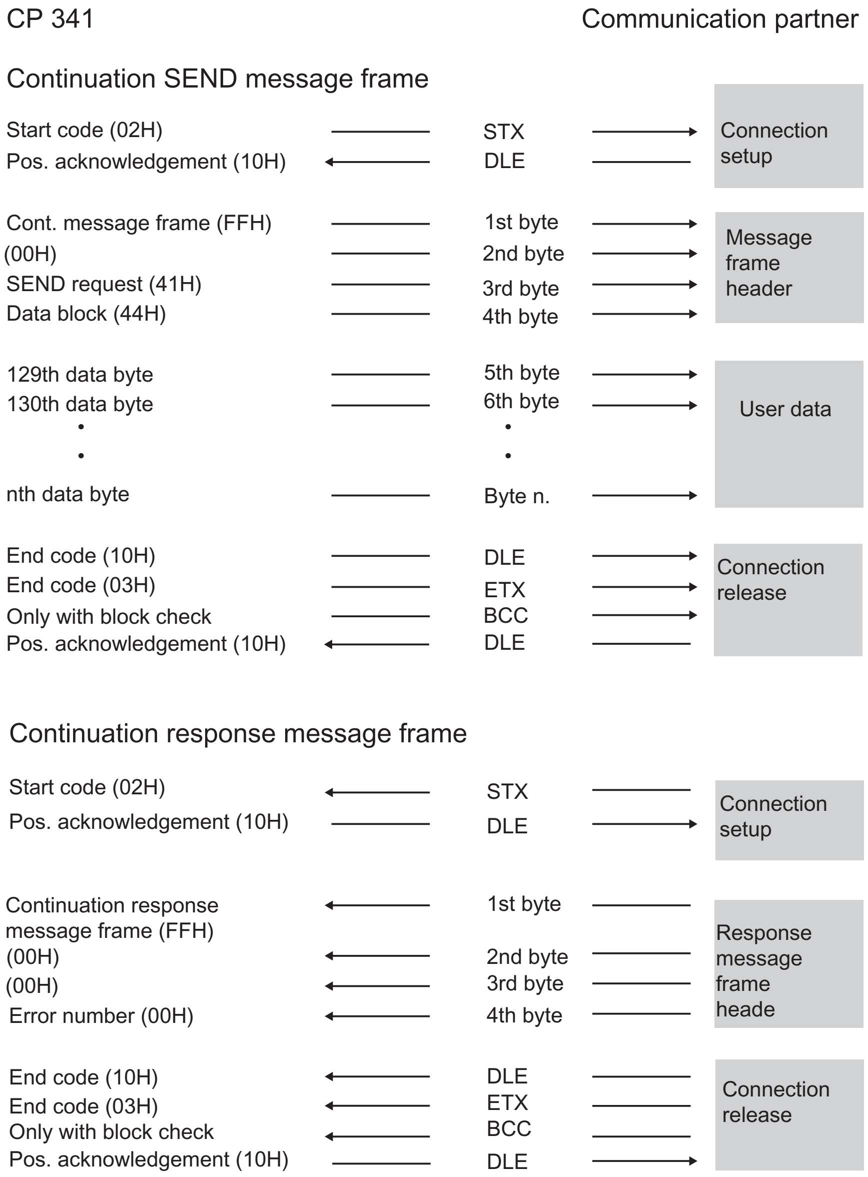


Figure 2-14 Sequence of a continuation SEND message frame with a continuation response message frame

2.5.2 Fetching Data with RK 512

Fetching data with RK 512

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

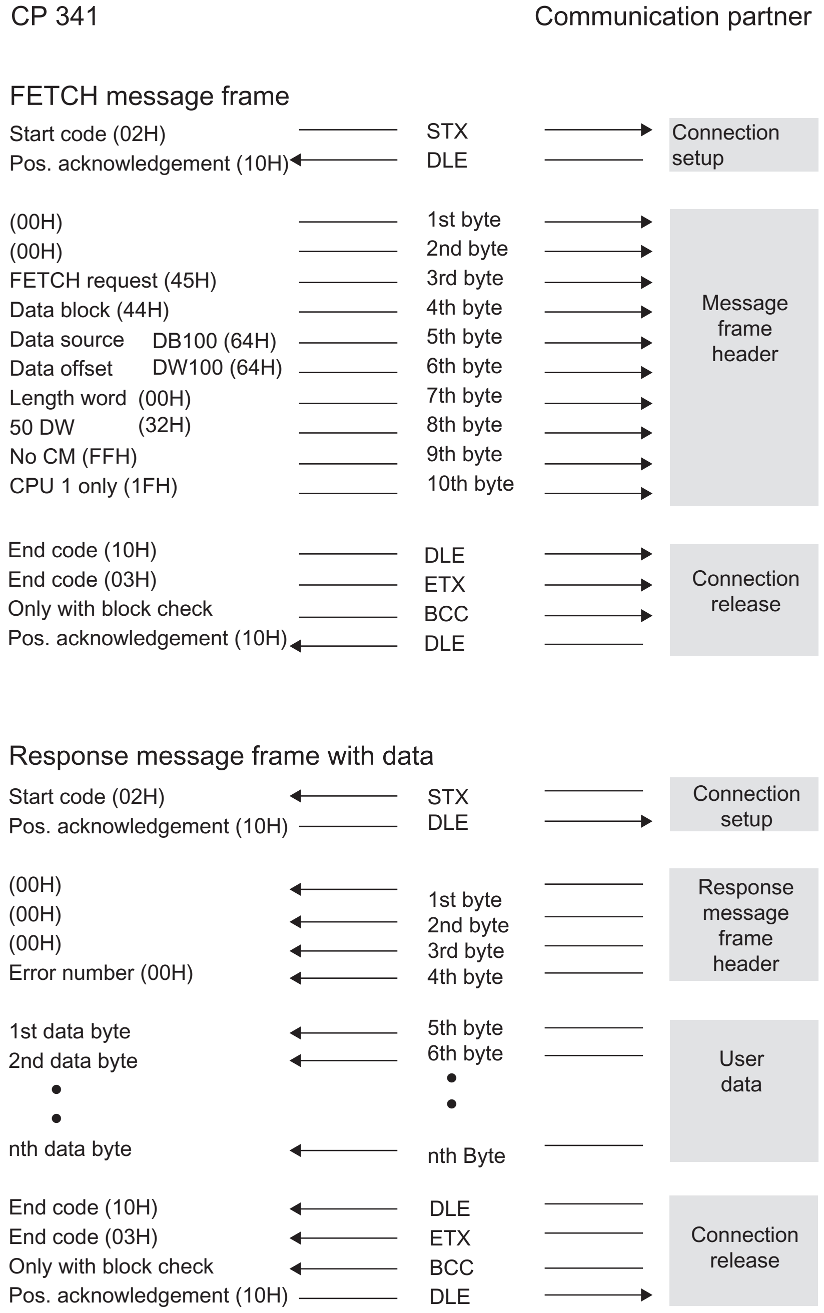


Figure 2-15 Data traffic when fetching with a response message frame

Fetching data

The FETCH request is executed in the following sequence:

- Active partner

  Sends a FETCH message frame. This contains the message frame header.

- Passive partner

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

- Active partner

  Receives the response message frame.

  If the volume of user data exceeds 128 Bytes, the active partner sends a continuation FETCH message frame. This contains byte 1 to 4 of the message frame header.

- Passive partner

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

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

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

Note

If the CPU does not receive the FETCH message frame error-free or if an error has occurred in the message frame header, the communication partner enters an error number in the 4th byte of the response message frame. This does not apply when protocol errors occur.

Continuation FETCH message frame

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

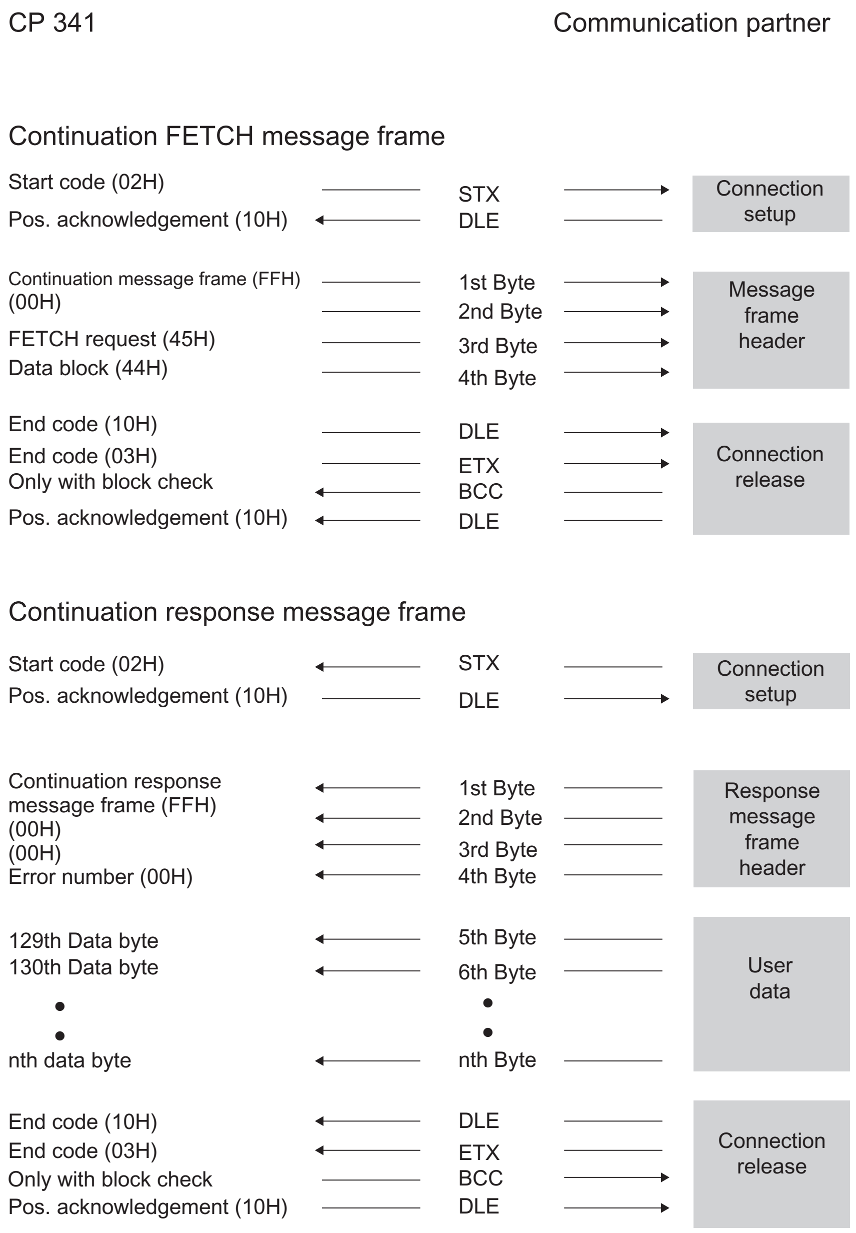


Figure 2-16 Sequence of a continuation FETCH message frame with a continuation response message frame

2.5.3 Quasi-Full-Duplex Operation

Quasi-full-duplex mode

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

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

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

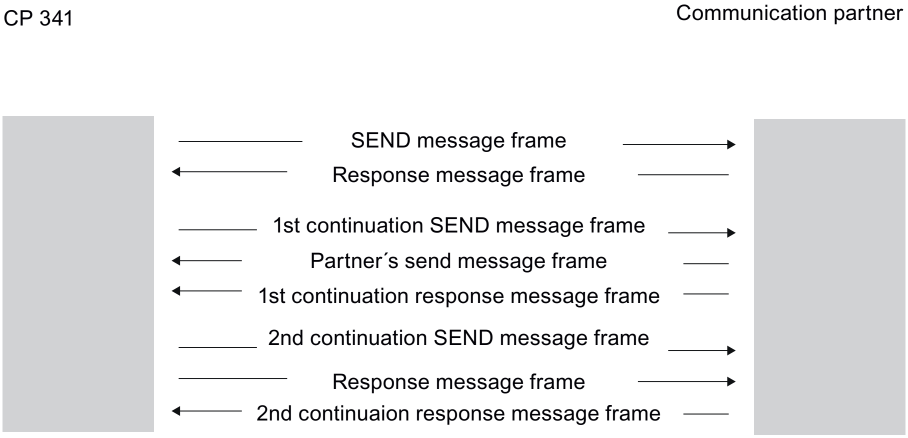

CP 341
Communication partner
SEND message frame
Response message frame
1st continuation SEND message frame
Partner´s send message frame
1st continuation response message frame
2nd continuation SEND message frame
Response message frame
2nd continuaion response message frame

Figure 2-17 Quasi-full-duplex mode

2.5.4 RK 512 CPU Requests

Processes involved in the RK 512 when CPU requests are made

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

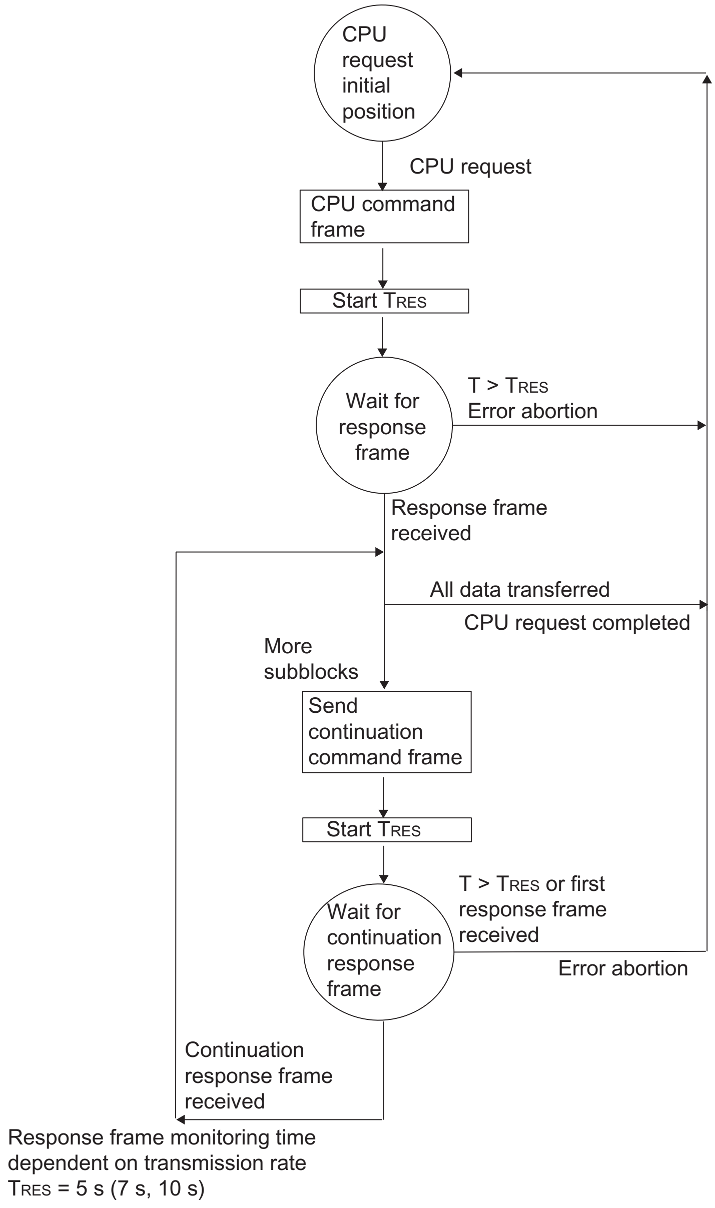

CPU request initial position
CPU request
CPU command frame
Start TRES
Wait for response frame
T > TRES
Error abortion
Response frame received
All data transferred
CPU request completed
More subblocks
Send continuation command frame
Start TRES
Wait for continuation response frame
T > TRES or first response frame received
Error abortion
Continuation response frame received
Response frame monitoring time dependent on transmission rate
TRES = 5 s (7 s, 10 s)

Figure 2-18 Flow diagram of data transmission with the RK 512 when CPU requests are made

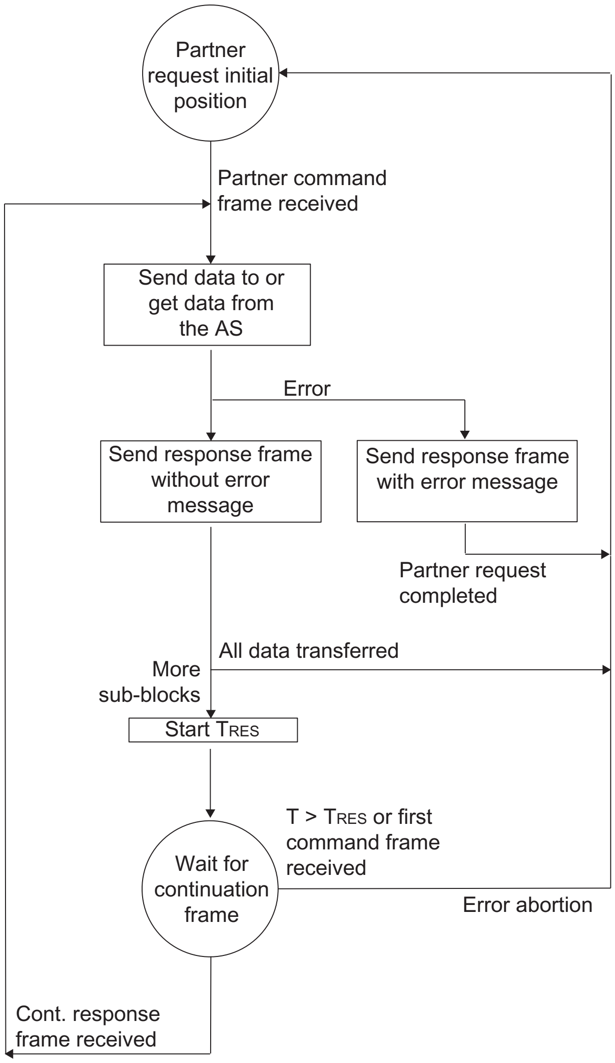
Extended error display at the receive FB

Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK.

If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.

RK 512 Partner Requests

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


Partner request initial position
Partner command frame received
Send data to or get data from the AS
Error
Send response frame without error message
Send response frame with error message
Partner request completed
All data transferred
More sub-blocks
Start TRES
T > TRES or first command frame received
Wait for continuation frame
Error abortion
Cont. response frame received

Response frame monitoring time dependent on transmission rate
T_{RES} = 5 s (7 s, 10 s)
AS = automation system

Figure 2-19 Flow diagram of data transmission with the RK 512 when partner requests are made

Extended error display at the receive FB

Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK.

If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.

2.6 Data Transmission with the ASCII Driver

2.6.1 Principle of the Data Transmission with the ASCII Driver

The ASCII driver controls data transmission via a point-to-point connection between the communication processor and a communication partner. This driver contains the physical layer (layer 1).

The structure of the message frames is left open through the S7 user passing on the complete send message frame to the communication processor. For the receive direction, the end criterion of a message must be configured. 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.

2.6.2 Sending data with the ASCII driver

Sending data with ASCII driver

For sending, you specify the number of user data bytes to be transferred when the P_SND_RK function block is called as the "LEN" parameter.

If you are working with the end criterion "Expiration of character delay time", the ASCII driver pauses between two message frames when sending. You can call the FB P_SND_RK at any time, but the ASCII driver does not begin its output until a period longer than the assigned character delay time has elapsed since the last message frame was sent.

If you are working with the "End-of-text character" end criterion, you have a choice of three options:

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

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

- Send up to length assigned in the FB

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

  The message frame will be sent without error message even if the data to be sent do not contain the end-of-text character.

- Send up to the length assigned in the FB and automatically append the end-of-text character(s)

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

If you are working with the "**Fixed message frame length**" end criterion, the amount of data transferred in the send direction is as specified in the "LEN" parameter of the FB P_SND_RK. The amount of data transferred in the receive direction, i.e. in the receive DB, is as specified for the receiver using the "Fixed message frame length" parameter in the parameter assignment interface. The two parameter settings must be identical, in order to ensure correct data traffic. A pause equal to the length of the character delay time is inserted between two message frames when sending to allow the partner to synchronize (recognize start of message frame).

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

Note

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

Sending data

The figure below illustrates a send operation.

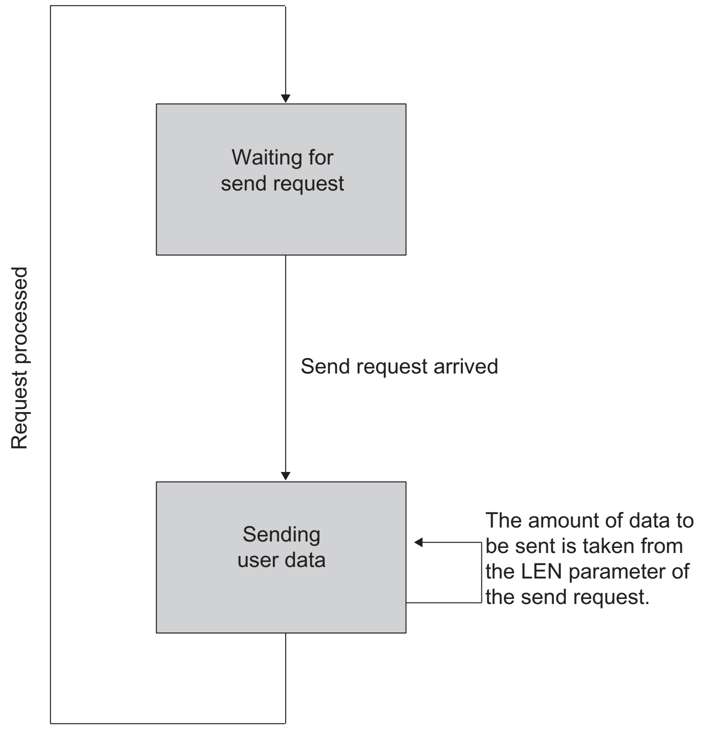

Waiting for send request
Send request arrived
Sending user data
The amount of data to be sent is taken from the LEN parameter of the send request.
Request processed

Figure 2-20 Flowchart of a send operation

2.6.3 Receiving data with the ASCII driver

Receiving data with ASCII driver

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

- Expiration of the character delay time

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

- Receipt of the end-of-text character(s)

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

- 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 the assigned end criterion and the flow control:

- With one or two end-of-text characters
  - Not code-transparent
- When end criterion is character delay time or fixed message frame length
  - Code-transparent
- Code-transparent operation is not possible when 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 "Expiration of character delay time"

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

In this case the character delay time must be set such that it is certain to expire between two consecutive message frames. But it should be so long that the end of the message frame is falsely identified in the case of a send pause of the link within a message frame.

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

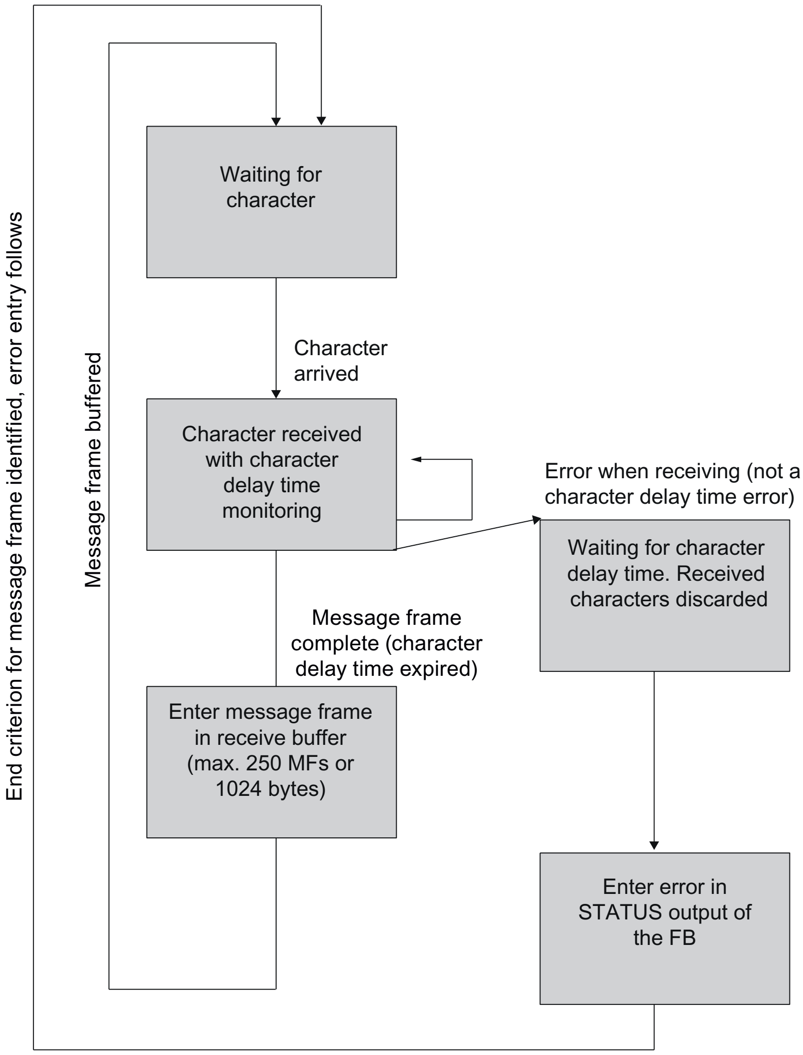

Waiting for character
Character arrived
Character received with character delay time monitoring
Error when receiving (not a character delay time error)
Waiting for character delay time. Received characters discarded
Message frame complete (character delay time expired)
Enter message frame in receive buffer (max. 250 MFs or 1024 bytes)
Enter error in STATUS output of the FB
Message frame buffered
End criterion for message frame identified, error entry follows

Figure 2-21 Flow diagram for receiving with end criterion "Expiration of character delay time"

End criterion end-of-text character

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

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

If you are working with endoftext characters, transmission is not code transparent. You must then make sure that the end character(s) are not included in the user data of the user.

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

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

  All characters, including the end-of-text character, are written to the receive DB. The characters located after the end-of-text character are

  - Discarded if the character delay time expires at the end of the message frame.
  - Merged with the next message frame if a new message frame is received before the character delay time expires.
- End-of-text character **not** included in message frame:

  The message frame is either:

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

The figure below illustrates a receive operation with the end criterion "Endoftext character".

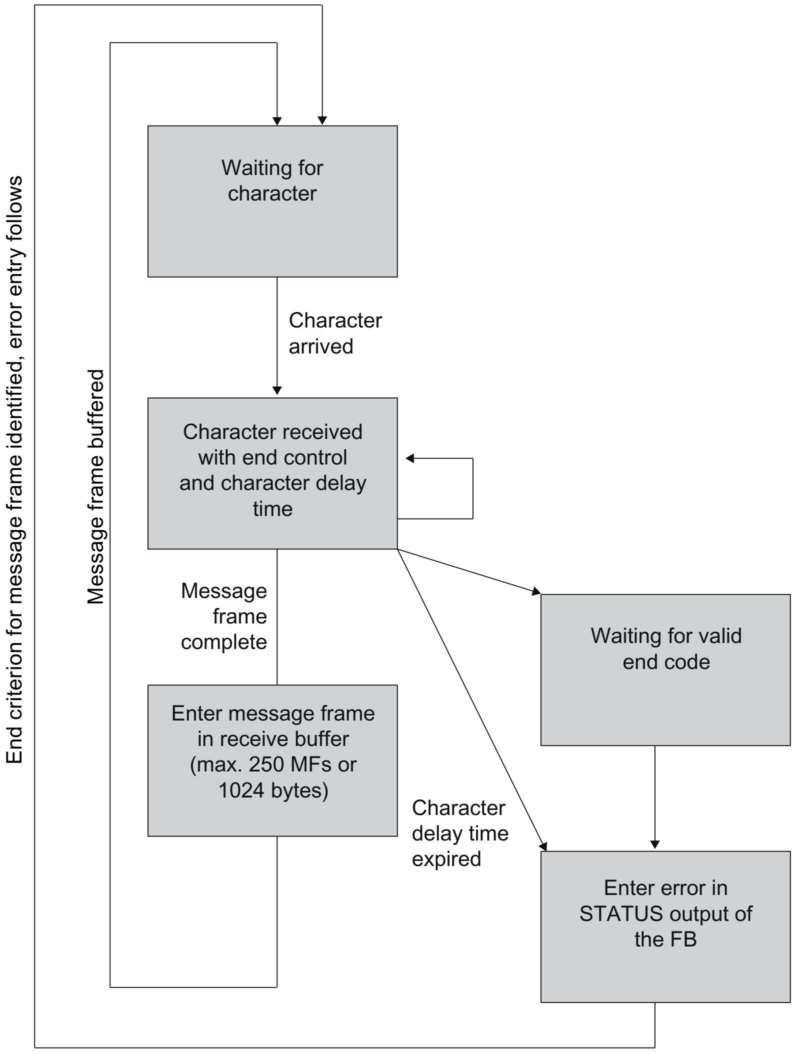

Waiting for character
Character arrived
Character received with end control and character delay time
Message frame complete
Enter message frame in receive buffer (max. 250 MFs or 1024 bytes)
Waiting for valid end code
Character delay time expired
Enter error in STATUS output of the FB
Message frame buffered
End criterion for message frame identified, error entry follows

Figure 2-22 Flow diagram for receiving with end criterion "End-of-text character"

Fixed message frame length end criterion

When receiving data, the end of the message frame is recognized when the assigned number of characters have been received. The received data is accepted by the CPU.

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

Note the following if the message frame length of the received characters does not conform with the fixed assigned message frame length:

- Message frame length of the received characters is greater than the fixed assigned message frame length:

  All characters received after reaching the fixed assigned message frame length are either:

  – Discarded if the character delay time expires at the end of the message frame.

  – Merged with the next message frame if a new message frame is received before the character delay time expires.

- Message frame length of the received characters is less than the fixed assigned message frame length:

  The message frame is either:

  – Discarded if the character delay time expires at the end of the message frame.

  – Merged with the next message frame if a new message frame is received before the character delay time expires.

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

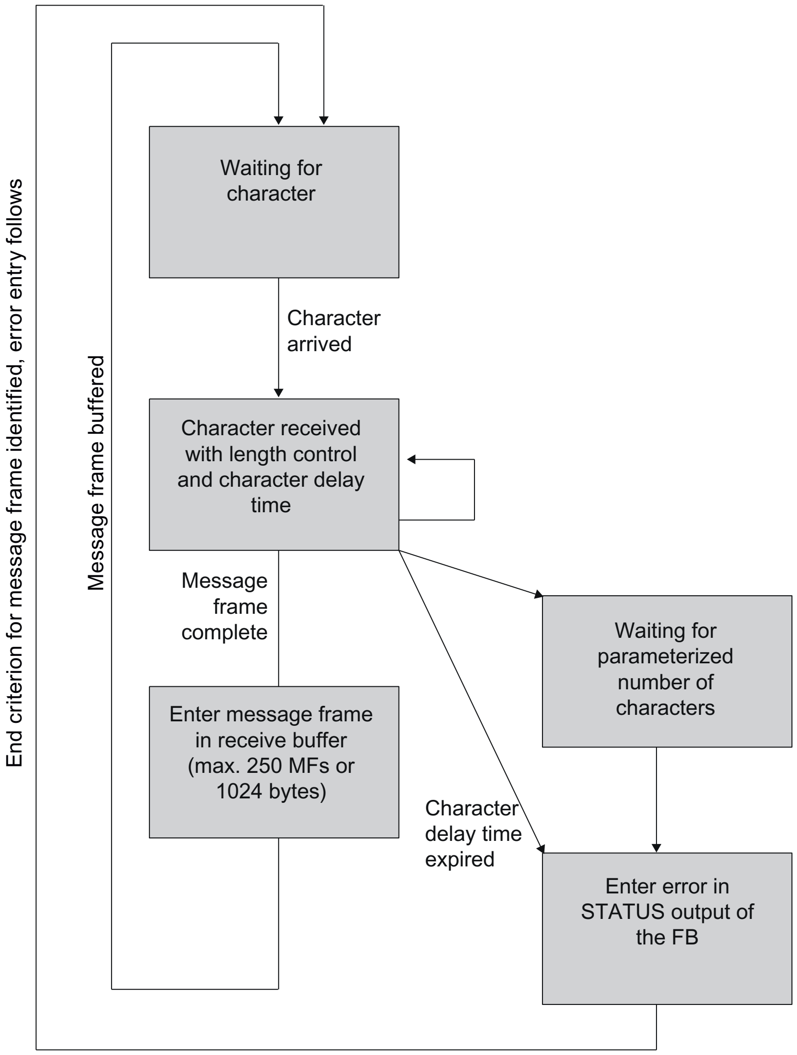

End criterion for message frame identified, error entry follows
Message frame buffered
Waiting for character
Character arrived
Character received with length control and character delay time
Message frame complete
Enter message frame in receive buffer (max. 250 MFs or 1024 bytes)
Waiting for parameterized number of characters
Character delay time expired
Enter error in STATUS output of the FB

Figure 2-23 Flow diagram for receiving with end criterion "Fixed message frame length"

Receive buffer on CP 341

The CP 341 receive buffer accommodates 4096 bytes. During parameter assignment, you can specify whether overwriting of data in the receive buffer is to be prevented. You can also specify the range of values (1 to 250) for the number of buffered receive message frames.

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

- If multiple message frames are written to the CP 341 receive buffer: The CP 341 always sends the oldest message frame to the CPU.
- If you want to transfer only the latest received frame to the CPU, you must assign the value "1" for the number of buffered message frames and deactivate overwrite protection.

Note

If continuous reading of the receive data is interrupted for a certain time in the user program and new receive data is requested, the CP 341 might first receive an old message frame from the CPU and then the latest message frame. The old message frames are those that were on the bus between the CP 341 and the CPU at the time of interruption or that had already been received by the FB.

Extended error display at the receive FB

Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK.

If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.

2.6.4 RS 485 mode

RS 485 mode

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

Switchover times for RS485 module in half-duplex mode

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

This value is applicable to modules as of order number 6ES7 341–1xH01–0AE0.

2.6.5 RS 232C operation

RS 232C accompanying signals

The following RS 232C accompanying signals are available on the CP 341 RS 232C:

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

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

You can control the DTR/DSR and RTS/CTS control signals using the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface or via functions (FCs) in the user program.

Controlling RS 232C accompanying signals

The RS 232C accompanying signals can be controlled as follows:

- When automatic control of all RS 232C accompanying signals is assigned
- When data flow control (RTS/CTS) is assigned
- By means of the FC V24_STAT and FC V24_SET functions

Note

When automatic control of the RS 232C accompanying signals is assigned, neither RTS/CTS data flow control nor RTS and DTR control by means of the FC V24_SET function are possible. When RTS/CTS data flow control is assigned, RTS control by means of the FC V24_SET function is not possible. On the other hand, it is always possible to read all RS 232C accompanying signals by means of the FC V24_STAT function.

The sections that follow describe the basic principles for controlling and evaluating RS 232C accompanying signals.

Automatic control of accompanying signals

Automatic control of the RS 232C accompanying signals is implemented on the CP 341 as follows:

- As soon as the CP 341 is assigned for operation in a mode with automatic control of the RS 232C accompanying signals, it sets the RTS line to OFF and the DTR line to ON (CP 341 ready for operation).

  This prevents sending and receiving of message frames until the DTR line is set to ON. No data is received at the RS 232C interface as long as DTR = OFF. Any send requests will be cancelled with a corresponding error message.
- When a **send job** is queued, the module sets RTS=ON, and triggers the configured data output wait time. After the data output time has elapsed and CTS = ON, the data is sent via the RS 232C interface.
- If during sending the CTS line is not set to ON within the data output wait time or if CTS changes to OFF during the send operation, the send request is canceled and a corresponding error message is generated.
- Once the data has been sent and the assigned Clear RTS time has elapsed, the RTS line is set to OFF. There is no waiting time for the transition from CTS to OFF.
- Data can be **received** via the RS 232C interface when DSR=ON. If the receive buffer of the CP 341 is close to overflow, the CP 341 will not respond.
- If DSR changes from ON to OFF, an active send request as well as the receipt of data will be canceled with an error message. The message "DSR = OFF (automatic use of V24 signals)" is entered in the diagnostic buffer of the CP 341.

Note

Automatic control of the RS 232C accompanying signals is only possible in half-duplex mode. When automatic control of the RS 232C accompanying signals is assigned, neither RTS/CTS data flow control nor RTS and DTR control by means of the FC V24_SET function are possible.

Note

The "Clear RTS time" must be set in the parameter assignment interface so that the communication partner can receive the last characters of the message frame in their entirety before RTS, and thus the send request, is canceled. The "data output wait time" must be set so that the communication partner can be ready to receive before the time elapses.

Timing diagram

The figure illustrates the chronological sequence of a send request.

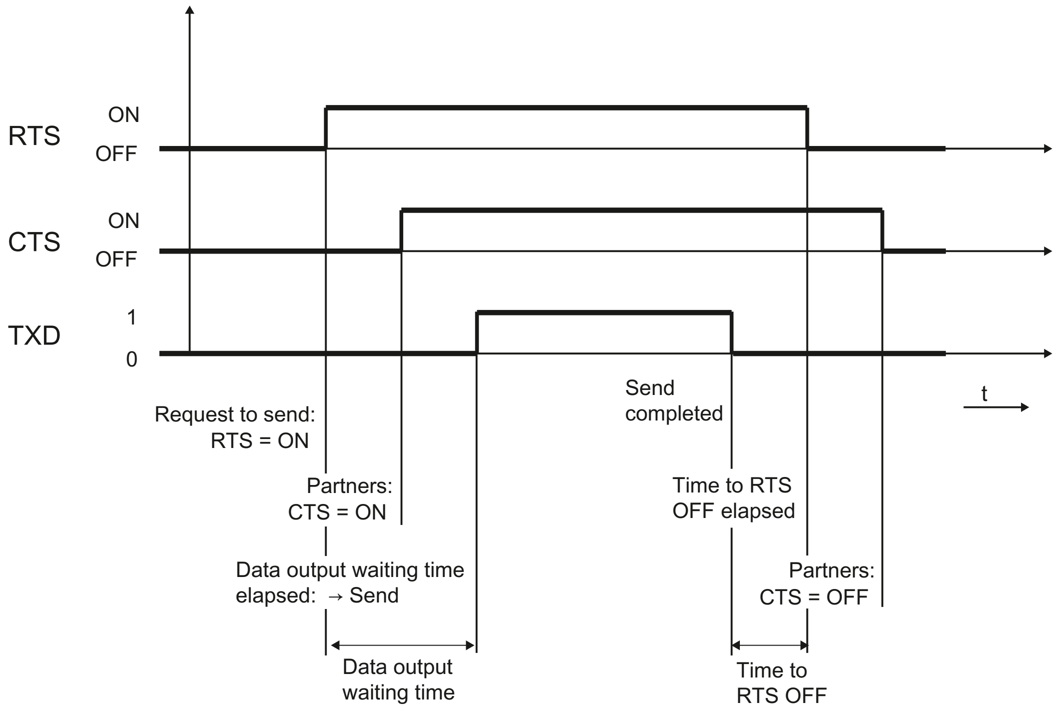


Figure 2-24 Timing diagram of automatic control of the RS 232C accompanying signals

Reading/controlling via FC V24_STAT and FC V24_SET

The FC V24_STAT function can be used to determine the status of each RS 232C accompanying signal. The FC V24_SET function can be used to control the DTR and RTS output signals.

2.6.6 Data flow control/Handshaking

Data flow control/Handshaking

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

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

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

- As soon as the CP 341 is assigned for operation in a mode with flow control, it sends the XON character or sets the RTS line to ON.
- If the assigned number of message frames or 50 characters are reached before the receive buffer overflows (size of the receive buffer: 4096 bytes), the CP 341 sends the XOFF character or sets the RTS line to OFF. If the communication partner ignores this state and continues transmission, an error message is generated if the receive buffer overflows. The data received in the last message frame will be discarded.
- As soon as a message frame is fetched by the S7 CPU and the receive buffer is ready to receive, the CP 341 sends the XON character or sets the RTS line to ON.
- The CP 341 interrupts the send operation when it receives the XOFF character or the CTS control signal of the communication partner is set to OFF. If an XON character is not received or the partner does not set CTS to ON within an assigned time period, the send operation is canceled and a corresponding error message (0708H) is generated at the STATUS output of the function blocks.

Note

When RTS/CTS data flow control is assigned, you must fully wire the interface signals in the plug connection. When RTS/CTS data flow control is assigned, RTS control by means of the FC V24_SET function is not possible.

2.7 Data transmission with the printer driver

Introduction

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

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

Message texts and parameters for printout

With the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface, you can configure the message texts and assign the parameters (page layout, character set, control characters) for printout. Message texts and printout parameters are transmitted to the CP 341 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.

Page layout:

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

Character set:

The ANSI character set is converted from STEP 7 to the printer characters set 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:

You can use a control character table to 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 other 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 341. The following can be displayed as variables: Calculated values of the user program, such as: fill 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 display type and composition of a message text. The format string can consist of:

- Text (all printable characters, for example: The level ... l was reached at ... hours.)
- 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))

  There must be one (and only one) conversion statement for each variable in the format string or 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, expanded, italic, and underlining (e.g., \B = bold type on) or with additional control characters you have defined

You can use other control characters if you enter them in the control character table in the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface and reassign the CP 341 parameters.

Additional functions

In addition to outputting message texts, you can use the following functions for printout. To execute one of these functions, simply specify it in the format string in the same way.

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

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: (Setting the page number to 10)

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

Printout

To output n bytes of user data to a printer, 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.

During output the data is edited for printing. Print editing is performed as configured in the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface (page layout, character set, control characters, etc.).

Characters are not received during printout, with the exception of flow control characters, provided this has been assigned. Any characters received are not accepted.

Note

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

Outputting a message text

The figure below illustrates the sequence of operations for a printout.

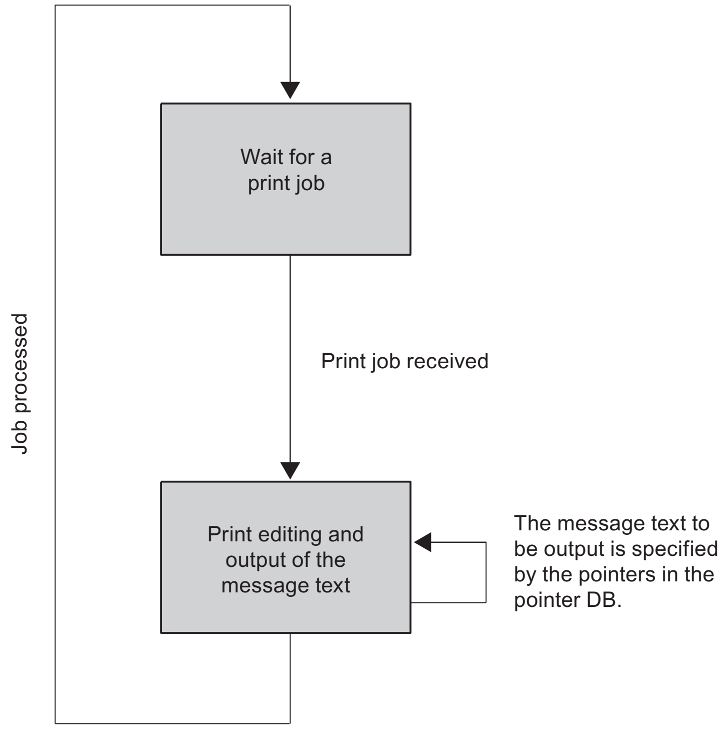

Wait for a print job
Print job received
Print editing and output of the message text
The message text to be output is specified by the pointers in the pointer DB.
Job processed

Figure 2-25 Flow chart of printout

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 printout. There are essentially two types of handshaking:

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

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

- As soon as the CP 341 is assigned for operation with flow control, it sends the XON character or sets the RTS line to ON.
- CP 341 interrupts the output of characters when it receives the XOFF character, or when control signal CTS = OFF is set. If neither an XON character is received nor CTS is set to ON once a configured time has elapsed, printout is canceled and an appropriate error message (0708H) is generated at the STATUS output of the SFB PRINT.

Note

When RTS/CTS flow control is assigned, you must fully wire the interface signals in the plug connection.

BUSY signal

The CP 341 evaluates the printer's "BUSY" control signal. The printer signals to the CP 341 that it is ready to receive:

- For CP 341-20mA-TTY: With current on RxD line
- For CP 341–RS 232C and CP 341-RS 422/485: CTS signal = "ON".

Note

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

- BUSY signal: CTS = "OFF"

Note that some printers use the DTR signal to display the BUSY signal. In such cases you must wire the cable to the CP 341 appropriately.

See also

RS 232C Interface of the CP 341-RS 232C (Page 215)

2.8 Configuration data

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

The sections that follow describe the parameter assignment data for the 3964(R) procedure, RK 512 computer link, ASCII driver, and printer driver.

2.8.1 Configuration Data of the 3964(R) Procedure

Using the parameter assignment data of the 3964(R) procedure, you can adjust the CP 341 to suit the properties of its communication partner.

Parameter assignment data of the 3964(R) procedure

With the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface, you can specify the parameters for the physical layer (layer 1) and for the data-link layer (layer 2) of the 3964(R) procedure. You will find a detailed description of the parameters below.

Section "Assigning Parameters to the Communications Protocols (Page 113)" describes how to enter the parameter assignment data using the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface.

X27 (RS 422/485) interface

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

Note

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

Protocol

The table below describes the protocol.

Table 2- 4 3964(R) protocol

Parameters	Description	Default value
3964 with default values and no block check	• Default values are assigned to the protocol parameters. • If the CP 341 recognizes the string DLE ETX, it stops receiving and sends a DLE to the communication partner if the block was received error-free, or an NAK if an error occurred.	**3964R with default values and block check:** Character delay time = 220 ms Acknowledgment delay time = 2000 ms Connection attempts = 6 Transmission attempts = 6
3964R with default values and block check	• Default values are assigned to the protocol parameters. • If the CP 341 recognizes the string DLE ETX BCC, it stops receiving. The CP 341 compares the block check character (BCC) received with the length parity calculated internally. If the BCC is correct and no other receive errors have occurred, the CP 341 sends the DLE character to the communication partner (the NAK character is sent if an error occurred).	
3964 assignable without block check	• The protocol parameters can be freely assigned by the user. • If the CP 341 recognizes the string DLE ETX, it stops receiving and sends a DLE to the communication partner if the block was received error-free, or an NAK if an error occurred.	
3964R assignable with block check	• The protocol parameters can be freely assigned by the user. • If the CP 341 recognizes the string DLE ETX BCC, it stops receiving. The CP 341 compares the block check character (BCC) received with the length parity calculated internally. If the BCC is correct and no other receive errors have occurred, the CP 341 sends the DLE character to the communication partner (the NAK character is sent if an error occurred).	

Protocol parameters

You can only assign the protocol parameters if you have not specified the default values for the protocol.

Table 2- 5 Protocol parameters (3964(R) procedure)

Parameters	Description	Range of values		Default value
Character delay time	The character delay time defines the maximum permissible time that may elapse between two received characters in a frame.	20 ms to 65530 ms in 10 ms increments The shortest character delay time depends on the baud rate:		220 ms
		300 bps:	60 ms	
		600 bps:	40 ms	
		1200 bps:	30 ms	
		2400 to 115200 bps:	20 ms	
Acknowledgment delay time	The acknowledgment delay time defines the maximum permissible time that may elapse before the partner's acknowledgment arrives during connection setup (time between STX and partner's DLE acknowledgment) or release (time between DLE ETX and partner's DLE acknowledgment).	20 ms to 65530 ms in 10 ms increments The shortest acknowledgment delay depends on the baud rate:		2000 ms (550 ms for 3964 without block check)
		300 bps:	60 ms	
		600 bps:	40 ms	
		1200 bps:	30 ms	
		2400 to 115200 bps:	20 ms	
Connection attempts	This parameter defines the maximum number of attempts of the CP 341 to set up a connection.	1 to 255		6
Transmission attempts	This parameter defines the maximum number of attempts to transfer a message frame (including the first one) in the event of errors.	1 to 255		6

Baud rate/Character frame

The table below describes the baud rate/character frame.

Table 2- 6 Baud rate/Character frame (3964(R) procedure)

Parameters	Description	Range of values	Default value
Baud rate	Data transmission rate in bps (baud) **Note:** A maximum of 19200 bps is possible for the 20mA-TTY interface.	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800 • 115200	9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	1 (fixed value)	1
Data bits	Number of bits to which a character is mapped.	• 7 • 8	8
Stop bits	During transmission, the stop bits are appended to every character to be sent; this signals the end of a character.	• 1 • 2	1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value ("0" or "1") brings the value of all the bits up to a defined status. This improves data integrity. "No" parity means that no parity bit is sent.	• None • Odd • Even	Even
Priority	A partner has high priority if its send request takes precedence over the other partner's send request. A partner has low priority if its send request has to wait until the other partner's send request has been dealt with. With the 3964(R) procedure, you must configure both communication partners with different priorities, i.e., one partner is assigned high priority, the other low.	• Low • High	High

Extended error display at the receive FB

Table 2- 7 Receive buffer on CP (3964(R) procedure)

Parameters	Description	Range of values	Default value
Extended error display at the receive FB	Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK. If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.	• Yes • No	No

X27 (RS 422) interface

You will find a description of the parameters for the X27 (RS 422) interface in the table below. RS 485 operation is not possible in conjunction with the 3964(R) procedure.

Table 2- 8 X27 (RS 422) interface (3964(R) procedure)

Parameters	Description	Range of values	Default value
Receive line initial state	None: This setting only makes sense with bus-capable special drivers.	None	R(A) 5V / R(B) 0V
	R(A) 5V / R(B) 0V: Break detection is possible in this initial state.	R(A) 5V / R(B) 0V	
	R(A) 0V / R(B) 5V: Break detection is not possible in this initial state. (also see the following figure)	R(A) 0V / R(B) 5V	

Receive line initial state

The figure illustrates the wiring of the receiver at the X27 (RS 422) interface:

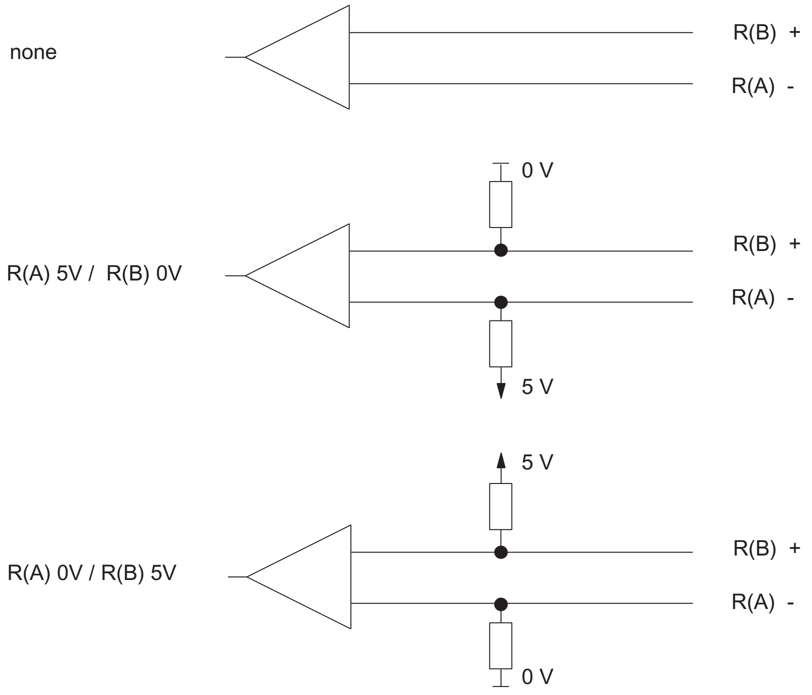


Figure 2-26 Wiring of the receiver at the X27 (RS 422) interface (3964(R) procedure)

2.8.2 Configuration Data of the RK 512 Computer Connection

You can use the parameter assignment data of the RK 512 computer link to adjust the CP 341 to suit the properties of the communication partner.

Parameter Assignment Data of the RK 512 Computer Link

The parameters are identical to those of the 3964(R) procedure because the 3964(R) procedure is a subset of the RK 512 computer link in the ISO 7-layer reference model (see Section "Configuration data (Page 76)").

Note

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

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

Waiting time for response message frames

Table 2- 9 RK512 computer connection

Parameters	Description	Range of values	Default value
Waiting time for response message frames Once the command message frame has been transmitted, the RK 512 waits for a response message frame from the communication partner within the monitoring time. The duration of the monitoring time depends by default on the transmission speed 20 s. This monitoring time can be reduced by setting parameters for the user in the "RK512" dialog of the parameter assignment interface.	Activate the parameter ""dependent on transmission rate" to monitor response message frames expected from the partner with the following waiting times: • 300 baud 10 s • 600 baud 7 s • 1200 baud 5 s • from 38400 baud 3 s The "grayed" field "maximum waiting time" is only used to display the monitoring time used and cannot be edited!	• Yes • No	No

Extended error display at the receive FB

Table 2- 10 Receive Buffer on CP (3964(R) Procedure)

Parameters	Description	Range of values	Default value
Extended error display at the receive FB	Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK. If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.	• Yes • No	No

2.8.3 Configuration data of the ASCII Driver

Using the parameter assignment data of the ASCII driver, you can adjust the communication processor to suit the properties of the communication partner.

Parameter assignment data of the ASCII driver

With the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface, you specify the parameters for the physical layer (layer 1) of the ASCII driver. You will find a detailed description of the parameters below.

Section "Assigning Parameters to the Communications Protocols (Page 113)" describes how to enter the parameter assignment data using the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface.

X27 (RS 422/485) interface

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

Note

In the case of the CP 341-RS 422/485 module variant, the ASCII driver can be used in four-wire mode (RS 422) and two-wire mode (RS 485). During parameter assignment, you must specify the type of interface (RS 422 or RS 485).

Protocol parameters

The table below describes the protocol parameters.

Table 2- 11 Protocol Parameters (ASCII Driver)

<table>
<tr><th>Parameters</th><th>Description</th><th colspan="2">Range of values</th><th>Default value</th></tr>
<tr><td>Indicator for end of receive message frame</td><td>Defines which criterion signals the end of each message frame.</td><td colspan="2">• After character delay time expires
• On receipt of end-of-text character(s)
• On receipt of a fixed number of characters</td><td>After character delay time expires</td></tr>
<tr><td rowspan="13">Character delay time</td><td rowspan="13">The character delay time defines the maximum permitted time between 2 consecutively received characters.</td><td colspan="2">2 ms to 65535 ms
The shortest character delay time depends on the baud rate</td><td rowspan="13">4 ms</td></tr>
<tr><td>Baud</td><td>Character delay time (ms)</td></tr>
<tr><td>300</td><td>130</td></tr>
<tr><td>600</td><td>65</td></tr>
<tr><td>1200</td><td>32</td></tr>
<tr><td>2400</td><td>16</td></tr>
<tr><td>4800</td><td>8</td></tr>
<tr><td>9600</td><td>4</td></tr>
<tr><td>19200</td><td>2</td></tr>
<tr><td>38400</td><td>2</td></tr>
<tr><td>57600</td><td>2</td></tr>
<tr><td>76800</td><td>2</td></tr>
<tr><td>115200</td><td>2</td></tr>
<tr><td>End-of-text character 1 [2]</td><td>Code of first end code.</td><td colspan="2">• With 7 data bits:
0 to 7FH (hex) [3]
• With 8 data bits:
0 to FFH (hex) [3]</td><td>3 (03H = ETX)</td></tr>
<tr><td>End-of-text character 2 [2]</td><td>Code of second end code, if specified.</td><td colspan="2">• With 7 data bits:
0 to 7FH (hex) [3]
• With 8 data bits:
0 to FFH (hex) [3]</td><td>0</td></tr>
<tr><td>Message frame length when received [1]</td><td>When the end criterion is "fixed message frame length", the number of bytes making up a message frame is defined.</td><td colspan="2">1 to 4096 (bytes)</td><td>240</td></tr>
<tr><td colspan="5">[1] Can only be set if the end criterion is "Fixed message frame length".
[2] Can only be set if the end criterion is "End-of-text character".
[3] Depending on whether you assign 7 or 8 data bits for the character frame.</td></tr>
</table>

Baud rate/Character frame

The table below contains descriptions of and specifies ranges of values for the relevant parameters.

Table 2- 12 Baud rate/Character frame (ASCII driver)

Parameters	Description	Range of values	Default value
Baud rate	Data transmission rate in bps (baud) **Notes:** A maximum of 19200 bps is possible for the 20mA-TTY interface.	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800 • 115200	9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	1 (fixed value)	
Data bits	Number of bits to which a character is mapped.	• 7 • 8	8
Stop bits	During transmission, the stop bits are appended to every character to be sent; this signals the end of a character.	• 1 • 2	1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value ("0" or "1") brings the value of all the bits up to a defined status. This improves data integrity. "No" parity means that no parity bit is sent.	• None • Odd • Even	Even

Data flow control

The table below contains a description of the parameters for data flow control.

Data flow control is not possible with the RS 485 interface. Data flow control with "RTS/CTS" and "automatic operation of V24 signals" is only supported at the RS 232C interface (see section " Possible applications for the CP 341 (Page 11) ").

Table 2- 13 Data flow control (ASCII driver)

Parameters	Description	Range of values	Default value
Data flow control	Defines which data flow control procedure is used.	• None • XON/XOFF • RTS/CTS • Automat. control of V24 signals	None
XON character [1]	Code for XON character	• With 7 data bits: 0 to 7FH (hex) [4] • With 8 data bits: 0 to FFH (hex) [4]	11 (DC1)
XOFF character [1]	Code for XOFF character	• With 7 data bits: 0 to 7FH (hex) [4] • With 8 data bits: 0 to FFH (hex) [4]	13 (DC3)
Waiting for XON after XOFF (wait time for CTS = ON) [2]	Period of time for which the communication processor should wait for the XON code or for CTS="ON" of the communication partner when sending.	20 ms to 65530 ms in 10 ms increments	20000 ms
Clear RTS time [3]	Time allowed to elapse after the transmission before the communication processor sets the RTS line to OFF.	0 ms to 65530 ms in 10 ms increments	10 ms
Data output wait time [3]	Time that the communication processor should wait when transmitting for the communication partner to set CTS to ON after setting the RTS line to ON and before starting the transmission.	0 ms to 65530 ms in 10 ms increments	10 ms

[1] Only for data flow control with XON/XOFF.

[2] Only for data flow control with XON/XOFF or CTS/RTS.

[3] Only for automatic control of the RS 232C accompanying signals.

[4] Depending on whether you set 7 or 8 data bits for the character frame.

Further information

Additional information regarding data flow control with XON/XOFF or RTS/CTS and automatic control of the RS 232C accompanying signals can be found in Section "Data Transmission with the ASCII Driver (Page 59) " starting at "RS 232C Accompanying Signals".

Receive buffer on CP

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

Table 2- 14 Receive buffer on CP (ASCII driver)

Parameters	Description	Range of values	Default value
Delete CP receive buffer at startup	You can specify whether the CP receive buffer should be deleted on startup or an existing (old) frame should be sent to the CPU.	• Yes • No	No
Buffered receive message frames	You can specify the number of receive message frames to be buffered in the CP receive buffer. If you specify "1" here and deactivate the following parameter "prevent overwrite" **and** cyclically read the received data from the user program, a current message frame will always be sent to the CPU.	1 to 250	250
Prevent overwriting	You can deactivate this parameter if the "buffered receive message frames" parameter is set to "1". This authorizes the overwriting of the buffered receive message frame.	• Yes • No (only if "Buffered receive message frames" = "1")	Yes
Extended error display at the receive FB	Activate the parameter "Display receive error at FB" to also display a message frame received with errors at the status output of the function block P_RCV_RK. If the parameter is deactivated, an entry is only created in the diagnostic buffer of the CP 341.	• Yes • No	No

Further information

Additional information regarding handling of the receive buffer can be found in Section " Data Transmission with the ASCII Driver (Page 59) " in "Receive buffer on CP 341".

X27 (RS 422/485) interface

You will find a description of the parameters for the X27 (RS 422/485) interface in the table below.

Table 2- 15 X27 (RS 422/485) interface (ASCII driver)

Parameters	Description	Range of values	Default value
Operating mode	Specifies whether the X27 (RS 422/485) interface is to be operated in full-duplex (RS 422) or half-duplex (RS 485) mode. (See also Serial Transmission of a Character (Page 23))	• Full-duplex (RS 422) four-wire mode • Half-duplex (RS 485) two-wire mode	Full-duplex (RS 422) four-wire mode
Receive line initial state	**None**: This setting only makes sense with bus-capable special drivers. **R(A) 5V / R(B) 0V**: This default setting supports break detection in "Full-duplex (RS 422) four-wire mode". **R(A) 0V / R(B) 5V**: This default corresponds to idle state (no senders active) in "Half-duplex (RS 485) two-wire mode". Break detection is not possible in this initial state. (also see the following figure)	• None • R(A) 5V / R(B) 0V [1] • R(A) 0V / R(B) 5V	R(A) 5V / R(B) 0V [1] (In "Half-Duplex (RS 485) Two-Wire Mode", the default setting is R(A) 0V / R(B) 5V.)
Delete CP receive buffer at startup	You can specify whether the CP receive buffer should be deleted on startup or existing (old) message frames should be sent to the CPU.	• Yes • No	No
[1] Only in the case of "Full-Duplex (RS 422) Four-Wire Mode".			

Receive line initial state

The figure illustrates the wiring of the receiver at the X27 (RS 422/485) interface:

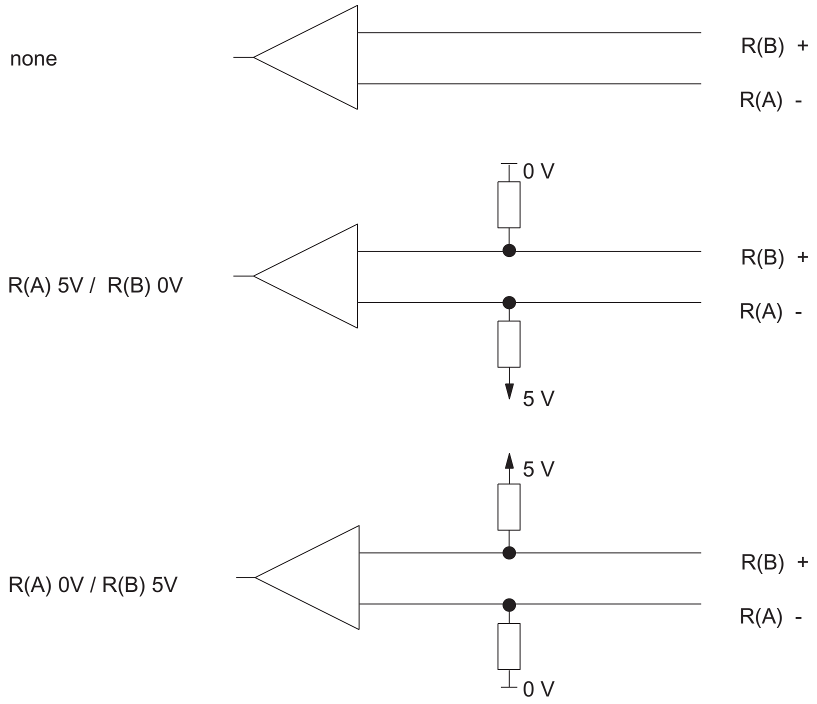


Figure 2-27 Wiring of the receiver at the X27 (RS 422/485) interface (ASCII driver)

2.8.4 Parameter assignment data of the printer driver

Introduction

You can use the parameter assignment data of the printer driver to generate the transmission-specific parameters and the message texts for printout.

Parameter assignment data of the printer driver

With the CP 341: Point-to-Point Communication, Parameter Assignment parameter assignment interface, you can specify:

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

You will find a detailed description of the parameters below.

Baud rate/Character frame

The table below contains descriptions of and specifies ranges of values for the relevant parameters.

Table 2- 16 Baud rate/Character frame (printer driver)

Parameters	Description	Range of values	Default value
Baud rate	Data transmission rate in bps	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800 • 115200	9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	1 (fixed value)	1
Data bits	Number of bits to which a character is mapped.	• 7 • 8	8

Parameters	Description	Range of values	Default value
Stop bits	During transmission, the stop bits are appended to every character to be sent; this signals the end of a character.	• 1 • 2	1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value ("0" or "1") brings the value of all the bits up to a defined status. This improves data integrity. "No" parity means that no parity bit is sent. "Any" parity indicates that the CP 341 has set the send parity to a value of "0".	• None • Odd • Even • Any	Even

Data flow control

The table below contains a description 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 supported with the RS 232C interface.

Table 2- 17 Data flow control (printer driver)

Parameters	Description	Range of values	Default value
Data flow control	Defines which data flow control procedure is used.	None XON/XOFF RTS/CTS	None
XON character (Only for data flow control with XON/XOFF.)	Code for XON character	• For 7 data bits: 0 to 7FH (hex) • With 8 data bits: 0 to FFH (hex) (Depending on whether you set 7 or 8 data bits for the character frame.)	11 (DC1)
XOFF character (Only for data flow control with XON/XOFF.)	Code for XOFF character	• For 7 data bits: 0 to 7FH (hex) • With 8 data bits: 0 to FFH (hex) (Depending on whether you set 7 or 8 data bits for the character frame.)	13 (DC3)
Wait for XON after XOFF (wait time for CTS = ON) (Only for data flow control with XON/XOFF or RTS/CTS.)	Period of time for which the CP 341 should wait for the XON code or for CTS="ON" of the communication partner when sending.	Up to 65530 ms in 10 ms increments	2000 ms

X27 (RS 422/485) interface

You will find a description of the parameters for the X27 (RS 422/485) interface in the table below.

Table 2- 18 X27 (RS 422/485) interface (ASCII driver)

Parameters	Description	Range of values	Default value
Receive line initial state	R(A)5V/R(B)0V: This initial state supports BREAK detection; it cannot be deactivated. R(A)0V/R(B)5V: This initial state does not support break detection.	R(A) 5V / R(B) 0V R(A) 0V / R(B) 5V	R(A) 5V / R(B) 0V

Page layout

The table below contains a description of the parameters for the page layout.

Table 2- 19 Page layout (printer driver)

Parameters	Description	Range of values	Default value
Left margin (number of characters)	Number of spaces to precede each line in the body of the text, header or footer. It is up to you to ensure that a line is not too long for the printer.	0 to 255	3
Lines per page (with header and footer)	Number of lines to be printed on each page. The number of lines printed is calculated on the basis of the separators output. In other words, all headers and footers must be counted.	1 to 255 0 (continuous printing)	50

Parameters	Description	Range of values	Default value
Separators/ Line end	Characters which end each line in the body of the text, header or footer. The body of the text, header and footer must contain the defined separator.	• CR (carriage return) • LF (line feed) • CR LF (carriage return and line feed) • LF CR (line feed and carriage return)	CR LF (carriage return and line feed)
Headers/Footers	Text for up to two header and footer lines; a header or footer line is output when the entry field in the parameter assignment software contains a text or at least a blank space. 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. A blank line is output before and after the header/footer.	• ASCII characters (text) • %P output conversion statement for page numbers) (max. 60 characters)	

Character set

The table below contains a description of the parameters for the character set.

Table 2- 20 Character set (printer driver)

Parameters	Description	Range of values	Default value
Printer character set	Set "IBM" to convert the set Windows ANSI character set into the printer character set. If you set "User-Defined", you can adapt the character set to include special characters for a particular language.	• IBM • User-defined	IBM

Control characters

The table below contains a description of the parameters for control characters.

Table 2- 21 Control characters (printer driver)

Parameters	Description	Range of values	Default value
Printer emulation	Sets the printer emulation (printer commands for the following control characters: bold, condensed, expanded, italics, and underlining). Set "User-Defined" to modify the printer emulation and include additional control characters. The characters A to Z and a to z are permissible as control characters.	• HP DeskJet • HP LaserJet • IBM Proprinter • User-defined	HP DeskJet

Performance features

Boundary conditions for configuring message texts:

- Size of the text SDB: 15 kbytes
- Max. length of a message text without variables: 150 characters
- Maximum length of message texts with variables displayed: 250 characters
- Maximum number of variables per message text: 4 (3 + message text number)

Message texts

The table below contains a description of the parameters for configuring message texts (using the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface).

Table 2- 22 Message texts (printer driver)

Parameters	Description	Range of values	Default value
Name of text SDB/text file	The message texts for a CP 341 (serial interface) must be saved to a text SDB for parameter assignment. 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	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 number:** 0 to 1999 **Message text (max. 150 characters):** • ASCII characters (text) • Conversion statements (for variables) • Control characters (all those defined in the control character table)	-
Font style	You can easily assign control characters to text selected in the "Edit Message" entry box by using buttons B to U.	• B (bold) • C (condensed) • E (expanded) • I (italic) • U (underlined)	-

2.8.5 Conversion and control statements for printout

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 printout (e.g., to set a page number or start a new page).

All the permissible characters and display types for the format string are described below. You can also configure all the described control statements (except \F "start new page" and \x "print without line break") and conversion statements for variables (except for %P "set page number") in the message texts using the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface.

Format string

The figure 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 one (and only one) conversion statement 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.

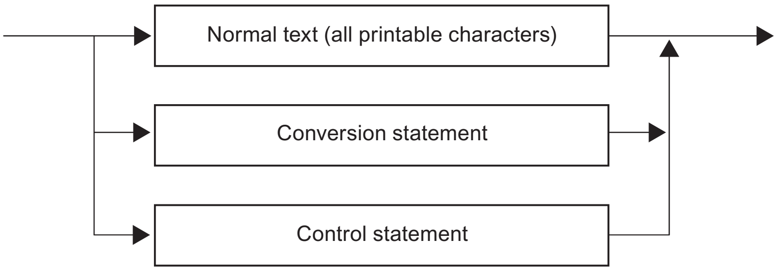


Figure 2-28 Schematic structure of the format string

Permissible characters for text

The following can be specified as text:

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

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

Conversion statement

The figure illustrates the structure of a conversion statement schematically.

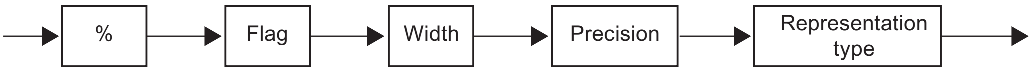


Figure 2-29 Schematic structure of a conversion statement

Flag

None = Right-justified output

– = Left-justified output

Width

None = Output in the default display

N = Exactly n characters are output (up to 255 characters are possible); blanks may be added before (right-justified output) or after (left-justified output)

Precision

Precision is only relevant to display types A, D, F, and R. It is ignored otherwise.

None = Output in the default display

.0 = No output of decimal point or decimal places in real (R) and floating point (F) display types.

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

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

Display type

The table below describes the possible display types for the values of the variables. Display types N and P are exceptions and are explained below the table.

The display type supports both uppercase and lowercase letters.

Table 2- 23 Display types in the conversion statement

Display type	Associated data type	Default display	Width of the default display	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 (US English)	10	Date format compliant with ICE 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 1999
P(2)	INT, WORD	Set page number	5	–
R	REAL, DWORD	0.12E-04	8	Floating point, with exponent
S	STRING	Text output	–	Text strings
T(1)	TIME, DWORD	2d_3h_10m_5s_250ms	Max. 22	Duration (negative duration is identified by a leading (-) minus sign)
U	BYTE WORD DWORD	255 65535 4294967295	Max. 3 Max. 5 Max. 10	Integer range, unsigned
X	BOOL BYTE WORD DWORD	1 11101100 11001... (16) 11001... (32)	1 8 16 32	Binary format

Display type	Associated data type	Default display	Width of the default display	Description
Y[3]	DATE_AND_TIME_ OF_DAY, DT	10.06.1992 –15:42:59.723	25	Date and time of day
Z	TIME_OF_DAY DWORD	15:42:59.723	12	Time
[1] If there is no message text number or system time in these display types, 6 "*" characters appear in the printout instead (the CP 341 does not keep the time). %N is the only conversion statement which cannot be used in the message text.				
[2] The P display type is only permitted in the format string. P is not permitted in configurable message texts.				
[3] The current time and date must be read first by means of the SFC 1 "READ_CLOCK" system function and stored in the user memory (bit memory, data).				

Output by means of message text number (%N)

Use the N display type to start printing message texts stored on the CP 341. 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

Within a message text, all conversion statements except for %N and all control statements except for "\F" and "\x" are allowed! An explicit width setting of %N limits the printed length of the referenced message text to the width indicated.

Setting the page number (%P)

Use the P display type to change the page number in the printout.

The CP 341 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 page 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 display type, there must be no further text, conversion statements, or control statements in the format string. The P display type is not permitted in configured message texts.

Notes about conversion statements

Please note the following in relation to conversion statements:

- Whenever a maximum length is specified for the default display, 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 display type a maximum of 6 places can be output for the INT data type and a maximum of 11 places for the DINT data type.
- A width of "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 (display types A, C, D, S, T, Y, and Z), only the number of characters corresponding to the specified width are output (the remainder are 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. display type missing: %2.2).

  The rest of the conversion statement (i.e., everything after the character identified as incorrect) is output. This allows the exact cause of the error to be determined.
- Conversion statements without associated variables will be ignored. Variables for which there is no conversion statement are not output.
- Conversion statements in a header or footer that are not supported are not executed. Instead, they are forwarded to the printer transparently.

- You must use control statements to specify formatting (line feed, tabs, etc.) in a message text or in the printout of a longer 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 invalid conversion statements

Below are several examples of invalid conversion statements.

Example 1: ******.2R

Format string = %303.2R
Variable 1 = 1.2345E6

Error: Invalid width in the R display type. The maximum permissible value for all display 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 display 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 in full.

Example 4: **********

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

Error: The default width of display 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 display type was not specified.

Examples of correct conversion statements

Below 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 decimal places) taking up 6 places.

Example 5: TEST..

Format string = %–6C
Variable 1 = TEST

Left-justified formatting of the text variables

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 on the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface prior to assigning parameters to CP 341.

The figure illustrates the structure of the control statement schematically.

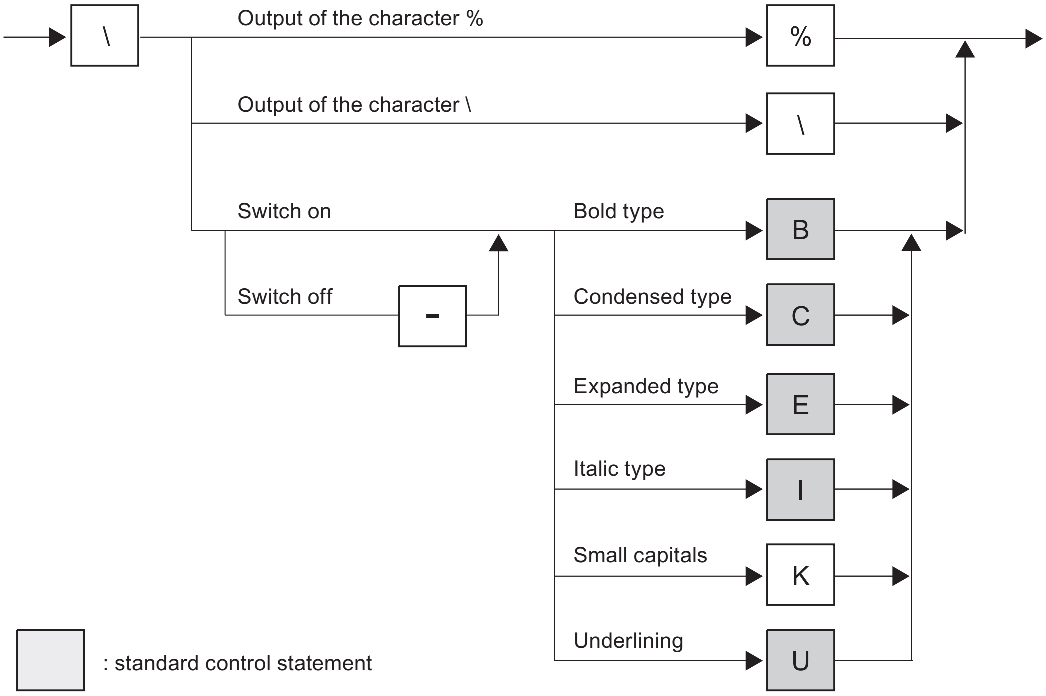


Figure 2-30 Schematic structure of control statements

Examples

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

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

Starting a new page (\F)

Taking into account the assigned 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: (Starting a new page)

Format string = \F

Note

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

Printing without a line break (\x)

The CP 341 normally appends the assigned 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 message texts 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

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

Notes about control statements

Note the following in relation to control statements:

- If the deactivation of an effect is requested 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 invalid control statements are not executed.

Commissioning the CP 341 3

To commission the communication processor, you will need to perform the following operations in the order given:

1. Mounting the CP 341
2. Configuring the CP 341
3. Assigning Parameters to the CP 341
4. Saving the parameter data
5. Create a user program for the CP 341

Mounting the CP 341

Mounting of the CP 341 involves its integration into the mounting rail (rack) of your programmable controller.

You can find a detailed description in Section "Configuring the CP 341 (Page 112)" of this manual.

Configuring the CP 341

Configuration of the CP 341 involves its layout in the configuration table. You configure the CP 341 using the STEP 7 software.

You can find a detailed description in Section "Configuring the CP 341 (Page 112)" of this manual.

Assigning Parameters to the CP 341

Parameter assignment of the CP 341 involves the creation of specific parameters of the protocols and the configuration of message texts for printout. You carry out the parameter assignment of the CP 341 using the **CP 341:Point-to-Point Communication, Parameter Assignment** parameter assignment interface.

You can find a detailed description in Section "Assigning Parameters to the Communications Protocols (Page 113)" of this manual.

Saving the parameter data

Storage of the parameter assignment data of the CP 341 involves saving the parameters, loading the parameters to the CPU, and transferring the parameters to the communication processor. You store the parameter assignment data using STEP 7 software.

You can find a detailed description in Section "Managing the Parameter Data (Page 116)" of this manual.

Creating a User Program for the CP 341

Programming of the CP 341 involves the program interface of the CP 341 to the associated CPU using the STEP 7 user program. You program the CP 341 using the language editors of the STEP 7 software.

A comprehensive programming example is available in Section "Programming Example for Standard Function Blocks (Page 199)". A detailed description of programming with STEP 7 is contained in the *Programming with STEP 7* manual.

Mounting the CP 341 4

4.1 CP 341 slots

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

Position of the CP 341 in the rack (mounting rail)

The following rules apply when positioning the CP 341 in the rack (mounting rail):

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

> **Note**
>
> Prior to removing or inserting the CP 341, you must switch the CPU to STOP mode and switch off the power supply. There are no restrictions for removing or inserting the cable for the integrated interface on the CP 341. 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.

Further information

Further information about slots is available in the Manual *S7–300 Programmable Controller, CPU Data, Installation*.

4.2 Mounting and Dismounting the CP 341

When installing and removing the CP 341, you must observe certain rules.

Tool

For installing and removing the CP 341, you require a 4.5 mm cylindrical screwdriver.

24 V DC load power supply

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

Only a safe, isolated extra-low voltage of ≤ 60 V DC may be used as the load current supply. The reliable electrical insulation can be implemented in compliance with the requirements of

- VDE 0100 Part 410 / HD 60364.4.41:2007 / IEC 60364–4–41:2005
  (as functional extra-low voltage with safe electrical isolation) or
- VDE 0805 / EN 60950 / IEC 950
  (as safety extra-low voltage SELV) or VDE 0106 Part 101.

4.2.1 Installation steps

Inserting the rack (mounting rail)

To insert the CP 341 in a rack (mounting rail), proceed as follows:

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

Connection terminals

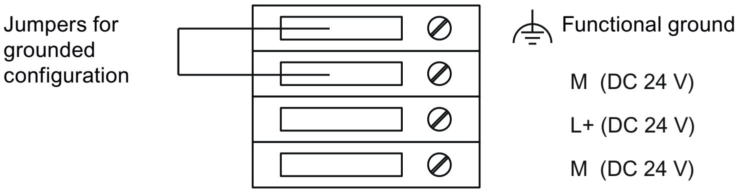


Figure 4-1 Connection terminal

- Connect the positive cable of the 24 V supply to terminal L+.
- Connect the negative cable of the 24 V supply to terminal M.
- The two M terminals are interconnected internally. The 24-V connection has polarity reversal protection.
- If you do not want to ground the 24 V ground cable, remove the jumper between the functional ground and M terminals.

4.2.2 Removal steps

Dismounting the rack (mounting rail)

To dismount the CP 341 from the rack (mounting rail), proceed as follows:

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

4.2.3 Installation guidelines

To be Observed

The general installation guidelines for the S7-300 must be observed (see the S7-300 Programmable Controller, CPU Data, Installation manual).

To comply with EMC (electromagnetic compatibility) values, the shield of the cables must be connected to a shielding bus.

Configuring and Parameter Assignment the CP 341 5

5.1 Parameter Assignment Options

Configuration options

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

Table 5- 1 Configuration options for the CP 341

Product	Order number	Assignable using the parameter assignment interface	Under STEP 7
CP 341-RS 232C	6ES7 341-1AH02-0AE0	as of version V5.1.7	as of version V5.3
CP 341-20mA-TTY	6ES7 341-1BH02-0AE0		
CP 341-RS 422/485	6ES7 341-1CH02-0AE0		

5.2 Installing the Programming Interface

Installation

The **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface is located on the CD along with the function blocks and programming example. To install the parameter assignment interface:

1. Insert the CD into the CD drive of your programming device or PC.
2. In **Microsoft Windows**, start the dialog for installing software by double-clicking the "Add and Remove Programs" icon in the "Control Panel".
3. In the dialog, select the CD drive and the "Setup.exe" file and start installation.
4. Follow the on-screen instructions provided by the setup program.

5.3 Configuring the CP 341

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

Requirements

The CP 341:Point-to-Point Communication, Parameter Assignment: parameter assignment interface is installed in the STEP 7 software on your programming device or PC (see section "Parameter Assignment Options (Page 111)").

Before you can enter the communication processor in the configuration table of the STEP 7 software, you must have created a project and a station with STEP 7.

Configuring

In the following, "configuring" refers to the placement of the CP 341 in the configuration table of the STEP 7 software. You enter the rack, the slot, and the order number of the CP 341 in the configuration table. STEP 7 then automatically assigns an address to the CP 341.

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

NOTICE
Prior to starting up a configured CP 341, you have to assign parameters to the module with a communication protocol - as described in the following section Assigning Parameters to the Communications Protocols (Page 113). A (non explicitly assigned CP 341) which is only connected is not automatically assigned specific default parameters!

Requirement

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

Further information?

How to configure S7-300 modules is described in detail in the *Configuring Hardware and Communication Connections STEP 7* manual.

In addition, the STEP 7 online help offers you comprehensive support when configuring an S7-300 module.

5.4 Assigning Parameters to the Communications Protocols

You must parameterize the CP 341 and its serial interface after having entered the CP 341 in the configuration table. In the case of the printer driver, you can also configure message texts for printer output.

Parameter assignment

The term "parameter assignment" is used in the following to describe the setting of protocol-specific parameters. The parameter assignment is carried out with the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface.

You start the parameter assignment interface by double-clicking the order number (CP 341) in the configuration table or by selecting the CP 341 and then the **Edit > Object Properties** menu command. The "Properties – CP 341" dialog will appear.

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

Further information?

The basic operation of the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface is the same for all communication processors and is self-explanatory. For this reason, the parameter assignment interface is not described in detail here.

Also, the online help will support you when working with the parameter assignment interface.

5.5 Identification data

Definition

Identification data represent information stored on the module and support you in:

- Troubleshooting a plant
- Verifying your plant configuration
- Locating hardware modifications in a plant

This ID data allows the unambiguous identification of modules in online mode. Starting with order no. 6ES7 341–1xH02–0AE0, this data is available on the CP 341.

To view the identification data, select **PLC > Module Information**, or **Read Data Record** (see below).

Reading the identification data

Users can access specific ID data by selecting **Read Data Record**.

The element of the ID data which is assigned to the corresponding index is found under the associated data record number.

All data records which contain ID data have a length of 64 bytes.

The table below shows the structure of those data records.

Table 5- 2 Data record structure

Content	Length (bytes)	Coding (hex)
Header information		
SZL ID	2	F1 11
Index	2	00 0x
Length of identification data	2	00 38
Number of blocks which contain ID data	2	00 01

Table 5- 3 Identification data

Identification data		
Index	2	00 0x
Identification data associated with the relevant index	54	

Identification data of the CP 341 module

Table 5- 4 Identification data of the CP 341 module

Identification data	Access	Default setting	Description
Index 1 (data record 231/read only)			
Manufacturer	Read (2 bytes)	00 2A hex (= 42 dec)	The name of the manufacturer is saved to this parameter (42 dec = Siemens AG).
Device name	Read (20 bytes)	6ES7 341–1xH02–0AE0	Order number of the module x = A(RS232), B(TTY), C(RS422/485)
Device serial number	Read (16 bytes)	The serial number of the module is saved to this parameter. This structure allows the unique identification of the module.	
Hardware revision	Read (2 bytes)	Provides information about the product version of the module.	
Software revision	Read (4 bytes)	Provides information about the firmware version of the module.	
Statistical revision no.	Read (2 bytes)	-	Not supported
Profile_ID	Read (2 bytes)	F6 00 hex	Internal parameter (to PROFIBUS DP)
Profile–specific type	Read (2 bytes)	00 04 hex (= 4 dec)	Internal parameter (communication module, to PROFIBUS DP)
I&M version	Read (2 bytes)	00 00 hex (= 0 dec)	Internal parameter (to PROFIBUS DP)
I&M supported	Read (2 bytes)	00 01 hex (= 1 dec)	Internal parameter (I&M0 and I&M1, to PROFIBUS DP)
Index 2 (data record 232/read and write)			
HID	Read/write (max. 32 characters)	-	Plant (higher-level) designation of the module.
LD	Read/write (max. 22 characters)	-	Location designation of the module.
Index 3 (data record 233/read and write)			
Device installation date	Read/write (max.16 characters)	-	Installation date
Index 4 (data record 234/read and write)			
Descriptor	Read/write (max. 54 characters)	-	Additional information

5.6 Managing the Parameter Data

The configuration and parameter assignment data of the communication processor is saved in the current project (on the hard disk of the programming device/PC).

Data management

When you exit the configuration table by selecting the Station > Save or Station > Save As menu command, the configuration data and parameter assignment data (including the module parameters) are automatically stored in the project/user file you have created.

Loading the configuration and parameters

You can now download the configuration data and parameter assignment data online from the programming device to the CPU (menu command PLC > Download). The CPU accepts the parameters immediately after the download.

The module parameters are automatically transmitted to the communication processor,

- when they are loaded onto the CPU and as soon as the communication processor can be reached via the S7-300 backplane bus,

  or

- when the CPU changes from STOP to RUN mode (CPU start-up).

Default settings apply if parameters are not changed.

Further information

The *Configuring Hardware and Communication Connections with STEP 7* manual describes in detail how you

- Save the configuration and the parameters
- Download the configuration and the parameters to the CPU
- Read, modify, copy, and print the configuration and the parameters

5.7 Subsequent Loading of Drivers (Transmission Protocols)

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

The loadable drivers are not shipped with the CP 341 or the parameter assignment interface. You have to order them separately. (see Catalog ST 70, section "Loadable drivers")

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

Requirements

The prerequisites for subsequently loading the drivers are:

- STEP 7 V5.3 and higher
- CP 341: Point-to-Point Communication, Parameter Assignment parameter assignment interface, V5.0 or higher
- The driver dongle must be connected to the port at the rear of CP 341.
- The valid parameter assignment was saved in HW Config beforehand and has been downloaded to the CPU.

Introduction to the parameter assignment interface

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

After you have successfully installed the parameter assignment interface and loadable drivers, you select the driver you want and assign the protocol-specific parameters in the same way as you do for the standard protocols. Installation of the parameter assignment interface and selection of a transmission protocol: see section: "Assigning Parameters to the Communications Protocols (Page 113)".

To find out what parameters are assigned and how to download the drivers to the CP 341, consult the separate documentation for the loadable driver.

5.8 Firmware updates

5.8.1 Subsequent Loading of Firmware Updates

You can download the firmware update to the operating system memory of the CP 341 in order to expand its functionality and for error handling.

Requirements

The requirements for loading firmware updates are as follows:

- STEP 7 V5.3 and higher
- You must first create a valid project in HW Config and load it onto the CPU before you can update the firmware of the CP.
- The CP 341 must be available online on the PG/PC.
- Refer to the instructions for the firmware update package for the storage location of the firmware update files.

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

Load firmware in HW Config

(valid as of order number 6ES7 341-1xH02-0AE0)

Carry out a firmware update as follows:

1. Switch the CPU to STOP mode.
2. Open HW Config, then select the relevant CP 341 module.
3. Select the menu command PLC > Update Firmware.

For additional information on procedures, refer to the *STEP 7* online help.

The system outputs a message to indicate successful completion of the update and immediately activates the new firmware.

Loading firmware with the CP 341 parameter assignment interface

(valid for order numbers 6ES7 341-1xH00-0AE0 and 6ES7 341-1xH01-0AE0)

The firmware is transferred to the CP 341 using the **CP 341:Point-to-Point Communication, Parameter Assignment** parameter assignment interface (as of V5.0).

Proceed as follows:

1. Switch the CPU to STOP mode.
2. Start the parameter assignment interface:

   In SIMATIC Manager: **File > Open > Project >** Open Hardware Config > double-click on CP 341 > select the "Parameters" button.
3. Select the **Options > Firmware Update** menu command.

   Result:

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

   If no firmware is found on CP 341, the string " - - - - " will be returned. This can happen, for example, if you cancelled the firmware update. In this case, the old firmware will be deleted. You have to reload a firmware prior to commissioning.
4. Click the "Find File ..." button to select the firmware to be loaded (*.UPD).

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

   Result:

   The version of the selected firmware is displayed under "Selected FW version".
5. Click the "Load Firmware" button to start loading the firmware to the CP 341. You are prompted for confirmation. The load operation will be canceled immediately if you click the "Cancel" button.

   Note: Before the basic firmware is deleted from the module, the CP 341 checks the order no. of the firmware to be downloaded in order to ensure that the firmware is approved for the CP 341.

   Result:

   The new firmware is loaded into the operating system memory of the CP 341. "Done" shows the progress on a status bar and as a percentage. The module is immediately ready for operation once the firmware update is successfully completed.

Update successful

After you have completed the CP 341 firmware update, attach a new label showing the new firmware version.

LED indicators

LED indicators when loading a firmware update:

Table 5- 5 LED indicators for firmware update

Status	SF	TXD	RXD	Comment	Remedy
Firmware update in progress	On	On	On	-	-
Firmware update completed	On	Off	Off	-	-
CP 341 without module firmware	Flashing (2 Hz)	Off	Off	Module firmware deleted, firmware update canceled, firmware update still possible	Reload the firmware
Hardware fault during firmware update	Flashing (2 Hz)	Flashing (2 Hz)	Flashing (2 Hz)	Delete/write operation failed	Switch power supply to module off and then on again and reload the firmware. Check whether the module is defective.

5.8.2 Viewing the Firmware Version

Viewing the hardware and firmware version

You can view the current hardware and firmware version on the CP 341 in **STEP 7** in the "Module Information" dialog. Access this dialog box as follows:

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

Communication via Function Blocks 6

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

Communication between CPU and CP 341

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

Communication between CP 341 and a communication partner

The transmission protocol conversion takes place on the CP 341. The protocol is used to adapt the interface of the CP 341 to the interface of the communication partner.

This enables you to link an S7 programmable controller with any communication partner that can handle today's standard protocols available in SIMATIC S5 (ASCII driver, 3964(R) procedure, RK 512 computer link, or printer driver).

Interrupt response

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

The function blocks P_SND_RK and P_RCV_RK of the CP 341 may only be called in one and the same execution level.

6.1 Overview of the Function Blocks

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

Function blocks/functions

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

Table 6- 1 Function blocks / functions of the CP 341

FB/FC	Meaning	Protocol
FC 5 V24_STAT (Version 2.0)	The V24_STAT function allows you to read the signal states at the RS 232C interface of the CP 341-RS 232C.	ASCII driver
FC 6 V24_SET (Version 2.0)	The V24_SET function allows you to set/reset the outputs at the RS 232C interface of the CP 341-RS 232C.	ASCII driver
FB 7 P_RCV_RK	The P_RCV_RK function block allows you to receive data from a communication partner and place it in a data block or to provide data to the communication partner.	3964(R) procedure, ASCII driver, RK 512 computer link
FB 8 P_SND_RK	The P_SND_RK function block allows you to send an entire area or subarea of a data block to a communication partner or fetch data from the communication partner.	3964(R) procedure, ASCII driver, RK 512 computer link
FB 13 P_PRINT_RK	The P_PRINT_RK function block enables you to output a message text with up to 4 variables to a printer.	Printer driver

Scope of supply and installation

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

The function blocks are installed together with the parameter assignment interface. After installation, the function blocks can be found in the library:

- **CP 341:** FC 5 V24_STAT (Version 2.0), FC 6 V24_SET (Version 2.0), FB 7 P_RCV_RK, FB 8 P_SND_RK and FB 13 P_PRINT_RK

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

When working with the function blocks, you only have to copy each function block to your project.

Permissible versions of the FBs and FCs

Note the following regarding permissible function blocks and functions:

⚠ WARNING

For the CP 341, you can use only the FC 5 V24_STAT and FC 6 V24_SET functions with **version ≥ 2.0**. Use of Version 1.0 can result in data corruption. For data transmission with the CP 341, only the FB 7 P_RCV_RK and FB 8 P_SND_RK function blocks can be used. The FB 2 P_RCV and FB 3 P_SEND function blocks of the CP 340 must **not** be used because this can result in data corruption.

6.2 Using function blocks

The following sections describe the factors to be considered when assigning parameters for the function blocks.

STATUS Indicator on the FB

Note the following regarding the STATUS indicator on the function blocks:

Note

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

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

In other words:

- When using the ASCII driver: Request was sent to the communication partner. It is not ensured that all data were received by the communication partner.
- When using the 3964(R) procedure: Request was sent to the communication partner and positive acknowledgement was returned. It is not ensured that the data were also passed onto the partner CPU.
- With RK 512 computer link: Request was sent to the communication partner, which forwarded it without error to the partner CPU.

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

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

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

Requests that can be processed simultaneously

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

Bear in mind, too, that you can use only:

- 1 instance data block for FB P_SND_RK and
- 1 instance data block for FB P_RCV_RK

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

Data consistency

The block size limits the data consistency for data transmission between the CPU and CP 341 to 32 bytes.

If you want consistent data transmission exceeding 32 bytes, you must take the following into account:

- Sender: Only access the send DB when all data have been completely transferred (DONE = 1).
- Receiver: Only access the receive DB when all data are received (NDR = 1). Then you must disable the receive DB (EN_R = 0) until you have processed the data.

6.3.1 S7 Sends Data to a Communication Partner

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

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

The FB P_SND_RK function block can be called cyclically when the signal state at the R parameter input is "1". This cancels the transmission to the CP 341 and places the FB P_SND_RK in its initial state. Data that has already been received by the CP 341 is still sent to the communication partner. If the signal state at the R input remains static at "1", this means that sending is deactivated.

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

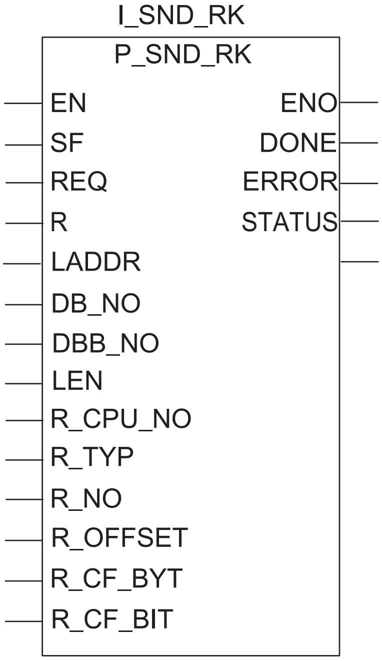
Error display at the FB P_SND_RK

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. If an error has occurred, the corresponding event number is displayed in STATUS (see section "Diagnostics Messages of the Function Blocks (Page 178)"). If no error has occurred, STATUS has the value 0. DONE and ERROR/STATUS are also output at RESET of the FB P_SEND RK (see figure in section "Receiving Data with FB P_RCV_RK (Passive Request) (Page 143)"). In the event of an error, the binary result BR is reset. If the block has been completed without errors, the binary result has the status "1".

Note

The P_SND_RK function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode. Before the CP 341 can process an activated request after the CPU has changed from STOP to RUN mode, the CP-CPU startup mechanism of the FB P_SND_RK must have been completed (see section "Diagnostics Messages of the Function Blocks (Page 178)"). Any requests initiated in the meantime are not lost. They are transmitted once the start-up coordination with the CP 341 is finished.

Block call

STL representation	**LAD representation**
CALL P_SND_RK, I_P_SND_RK SF: = REQ: = R: = LADDR: = DB_NO: = DBB_NO: = LEN: = R_CPU_NO: = R_TYP: = R_NO: = R_OFFSET: = R_CF_BYT = R_CF_BIT = DONE: = ERROR: = STATUS: =	 I_SND_RK P_SND_RK EN SF REQ R LADDR DB_NO DBB_NO LEN R_CPU_NO R_TYP R_NO R_OFFSET R_CF_BYT R_CF_BIT ENO DONE ERROR 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 FB P_SND_RK works together with an I_SND_RK instance DB. The DB number is passed on with the call. The instance DB requires 414 bytes load memory and 156 bytes RAM. Access to the data in the instance DB is not permitted.

Note

Exception: If the error STATUS == W#16#1E0F occurred, you can consult the SFCERR variable for additional details. This error variable can only be loaded via symbolic access to the instance DB.

FB P_SND_RK parameters

The table below lists the parameters of the FB P_SND_RK.

Note

The R_CPU_NO, R_TYP, R_NO, R_OFFSET, R_CF_BYT, and R_CF_BIT parameters are irrelevant for the 3964(R) procedure and do not have to be assigned. The SF parameter also does not have to be assigned since 'S' for sending is entered by default.

Table 6- 2 FB P_SND_RK parameters

Name	Sort	Data type	Comment	Permitted values, remark
REQ	INPUT	BOOL	Initiates request on positive edge	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Transmission is disabled.
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Send DB no.: CPU-specific, zero is not permitted
DBB_NO	INPUT	INT	Data byte number	0 ≤ DBB_NO ≤ 8190 Send data starting from data byte
LEN	INPUT	INT	Data length	1 ≤ LEN ≤ 4096, specified in number of bytes
DONE [1]	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR [1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
STATUS [1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter will contain error information.
[1] The parameter is available until the next time the FB is called.				

Time sequence chart of FB P_SND_RK

The figure below illustrates the behavior of the DONE and ERROR parameters depending on the input circuit of REQ and R.

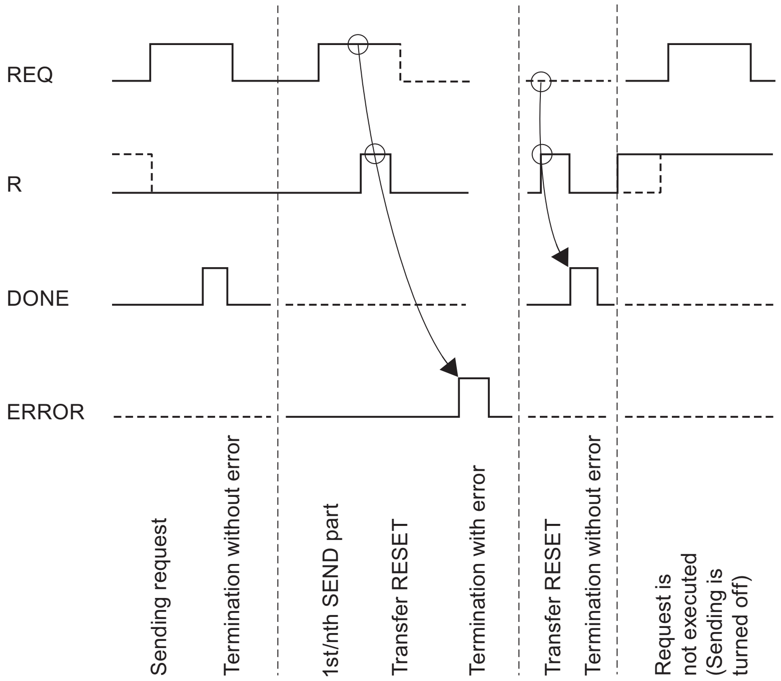

REQ
R
DONE
ERROR
Sending request
Termination without error
1st/nth SEND part
Transfer RESET
Termination with error
Transfer RESET
Termination without error
Request is not executed (Sending is turned off)

Figure 6-1 Time sequence chart of FB 8 P_SND_RK

Note

The REQ input is edge-triggered. A positive edge at the REQ input is sufficient. The result of the logic operation must not be at "1" at any point during transfer.

6.3.2 S7 receives data from a communication partner

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

A (static) signal state "1" at the EN_R, parameter enables a check to determine whether data can be read from the CP 341. An active transmission can be canceled with signal state "0" at the EN_R parameter. The canceled receive request is terminated with an error message (STATUS output). Receiving is disabled as long as the signal state at the EN_R parameter is "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 canceled and the FB P_RCV_RK is placed in its initial state. Receiving is disabled as long as the signal state at the R parameter is "1".

The LADDR parameter defines the CP 341 to be addressed.

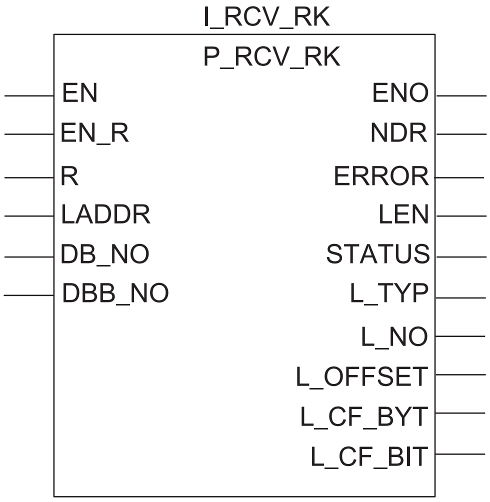
Error display at the FB P_RCV_RK

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

Note

The P_RCV_RK function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode. Before the CP 341 can receive a request after the CPU has changed from STOP to RUN mode, the CP-CPU start-up mechanism of the FB P_RCV_RK must have been completed.

Block call

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

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 FB P_RCV_RK works together with an I_RCV_RK instance DB. The DB number is passed on with the call. The instance DB requires 414 bytes load memory and 152 bytes RAM. Access to the data in the instance DB is not permitted.

Note

Exception: If the error STATUS == W#16#1E0E occurred, you can consult the SFCERR variable for additional details. This error variable can only be loaded via symbolic access to the instance DB.

FB P_RCV_RK parameters

The table below lists the parameters of the FB P_RCV_RK.

Note

The L_TYP, L_NO, L_OFFSET, L_CF_BYT, and L_CF_BIT parameters are irrelevant for the 3964(R) procedure and do not have to be assigned.

Table 6- 3 FB P_RCV_RK parameters

Name	Sort	Data type	Comment	Permitted values, remark
EN_R	INPUT	BOOL	Enables data reading	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Receiving is disabled.
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Receive DB No.: CPU-specific, zero is not permitted
DBB_NO	INPUT	INT	Data byte number	0 ≤ DBB_NO ≤ 8190 Receive data starting from data byte
NDR	OUTPUT	BOOL	Request completed without errors, data accepted	STATUS parameter == 16#00;
ERROR [1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
LEN [1]	OUTPUT	INT	Length of the message frame received	1 ≤ LEN ≤ 4096, specified in number of bytes
STATUS [1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter will contain error information.
[1] The parameter is available until the next time the FB is called.				

Time sequence chart of FB P_RCV_RK

The figure below illustrates the behavior of the NDR, LEN, and ERROR parameters depending on the input circuit of EN_R and R.

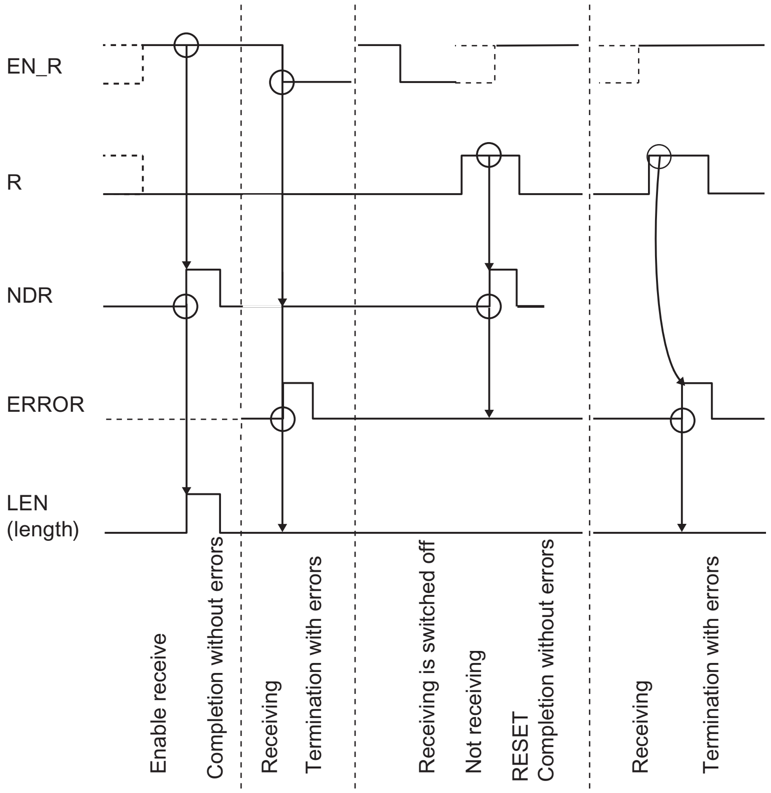


Figure 6-2 Time sequence chart of FB 7 P_RCV_RK

Note

Input EN_R must be set to static "1" state. The EN_R parameter must be supplied with result of logic operation "1" during the entire receive request.

6.4 Using the Function Blocks with the RK 512 Computer Connection

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

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

Data transmission options

Active requests:

The FB 8 P_SND_RK function block enables you to issue active requests to the CP 341 in the user program of the CPU. You can

- Send data from your automation system to a remote communication partner.
- Fetch data from a remote communication partner and store it in an S7 data area of your automation system
  Note: If you fetch data from a CP 341, you must always program an FB P_RCV_RK on the CP 341.

Passive requests:

The FB 7 P_RCV_RK function block enables you to use passive requests to coordinate the reading and providing of data on the CP 341. The communication partner is active. You can

- Read in data sent from the communication partner in an S7 data area of your automation system.
- Provide data in your automation system to a remote communication partner.

Requests that can be processed simultaneously

Only one active and one passive request can be programmed for each CP 341 in the user program. While the CP 341 is transacting an active request, it can simultaneously process a passive request.

Bear in mind, too, that you can use only:

- 1 instance data block for FB P_SND_RK and
- 1 instance data block for FB P_RCV_RK

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

Interprocessor communication flag

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

Data consistency

The block size limits the data consistency for data transmission between the CPU and CP 341 to 32 bytes.

If you want consistent data transmission exceeding 32 bytes, you must take the following into account:

- Sender: Only access the send DB when all data have been completely transferred (DONE = 1).
- Fetching data: Only access the send DB when all data have been completely transferred (DONE = 1).
- Receiver: Use the interprocessor communication flag. Do not access the receive DB until all data has been received (evaluation of the interprocessor communication flag specified for this job; the interprocessor communication flag is available in the FB for one cycle, if NDR = 1). After you have processed the received data, reset the interprocessor communication flag to "0".
- Providing data: Use the interprocessor communication flag. Do not access the provided data until all of the data have been fetched (evaluation of the interprocessor communication flag specified for this job). The interprocessor communication flag is available for one cycle in the FB, if NDR = 1). After you have processed the data to be fetched, reset the interprocessor communication flag to "0".

If your partner fetches data from the I (inputs), Q (outputs), M (bit memory), T (timers), or C (counters) areas, the data consistency is limited to 32 bytes, provided you cannot use the interprocessor communication flag to prevent other locations in the user program from accessing these areas during the transmission.

6.4.1 Sending Data with FB P_SND_RK (Active Request)

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

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

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

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

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

The interprocessor communication flag byte and bit on the partner CPU are specified in R_CF_BYT and R_CF_BIT.

The FB P_SND_RK function block can be called cyclically when the signal state at the R parameter input is "1". This cancels the transmission to the CP 341 and places the FB P_SND_RK in its initial state. Data that has already been received by the CP 341 is still sent to the communication partner. If the signal state at the R input remains static at "1", this means that sending is deactivated.

Error display at the FB P_SND_RK

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. In STATUS the event number is displayed in the event of an error. If no error has occurred, the value of STATUS is 0. DONE and ERROR/STATUS are also output in response to a RESET of FB P_SND_RK. In the event of an error, the binary result BR is reset. If the block has been completed without errors, the binary result has the status "1".

Note

The FB P_SND_RK function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode.

Special features for sending data

Note the following special features with regard to sending data:

- RK 512 allows only an even-numbered amount of data to be sent. If you specify an odd-numbered amount of data for the length (LEN), an additional filler byte with a value of "0" is sent at the end of the data.
- RK 512 allows only an even-numbered offset. If you specify an odd-numbered offset, the data is stored in the partner starting from the next lower even-numbered offset.

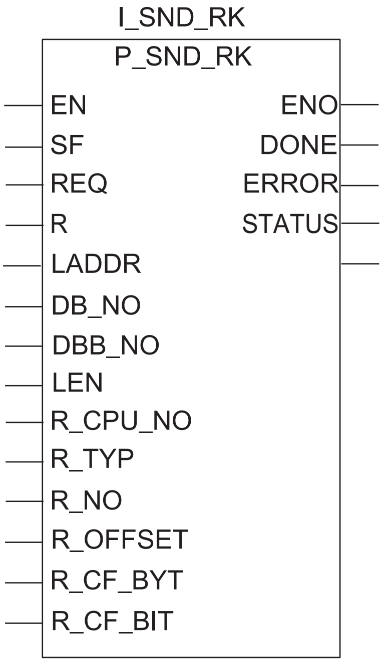
  Example: Offset is 7, data is stored from byte 6.

Block call

STL representation

```
CALL P_SND_RK, I_SND_RK
 SF: =
 REQ: =
 R: =
 LADDR: =
 DB_NO: =
 DBB_NO: =
 LEN: =
 R_CPU_NO: =
 R_TYP: =
 R_NO: =
 R_OFFSET: =
 R_CF_BYT =
 R_CF_BIT =
 DONE: =
 ERROR: =
 STATUS: =
```

LAD representation



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 FB P_SND_RK works together with an I_SND_RK instance DB. The DB number is passed on with the call. The instance data block is 62 bytes long. Access to the data in the instance DB is not permitted.

FB P_SND_RK parameters

The table below lists the parameters of the FB 8 P_SND_RK for the "Send data" request.

Table 6- 4 FB 8 P_SND_RK parameters for "Send data" request

Name	Sort	Data type	Comment	Permitted values, remark
SF	INPUT	CHAR	Selection for Send data or Fetch data	SF = 'S' (Send) Default: 'S'
REQ	INPUT	BOOL	Initiates request on positive edge	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Transmission is disabled. Default: 0
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number of source	Send DB no.: CPU-specific, zero is not permitted
DBB_NO	INPUT	INT	Data byte number of source	0 ≤ DBB_NO ≤ 8190 Send data starting from data byte
LEN	INPUT	INT	Data length of the message frame to be sent	1 ≤ LEN ≤ 4096, specified in number of bytes; only even-numbered values are appropriate
R_CPU_NO	INPUT	INT	CPU number of the partner CPU	0 ≤ R_CPU_NO ≤ 4, only with multiprocessor mode; default value: 1
R_TYP	INPUT	CHAR	Address type on the partner CPU	'D': Data block ‘X’: Expanded data block
R_NO	INPUT	INT	Data block number on the partner CPU	0 ≤ R_NO ≤ 255
R_OFFSET	INPUT	INT	Data byte number on the partner CPU	0 ≤ R_OFFSET ≤ 510, even-numbered values only
R_CF_BYT	INPUT	INT	Interprocessor communication flag byte on partner CPU	0 ≤ R_CF_BYTE ≤ 255 Default: 255 (means: without interprocessor communication flag)
R_CF_BIT	INPUT	INT	Interprocessor communication flag bit on the partner CPU	0 ≤ R_CF_BIT ≤ 7

Name	Sort	Data type	Comment	Permitted values, remark
DONE [1]	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR [1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
STATUS [1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter contains the error information.
[1] The parameter is available until the next time the FB is called.				

Information in the message frame header

The table below shows the information in the message frame header of the RK 512 message frame.

Table 6- 5 Information in the RK 512 message frame header for "Send data" request

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

Explanation of the abbreviations used:

D-DBNO: Destination data block number

D-Offset: Destination start address

DW: Offset in words

Time sequence chart of FB P_SND_RK

The figure below illustrates the behavior of the DONE and ERROR parameters depending on the input circuit of REQ and R.

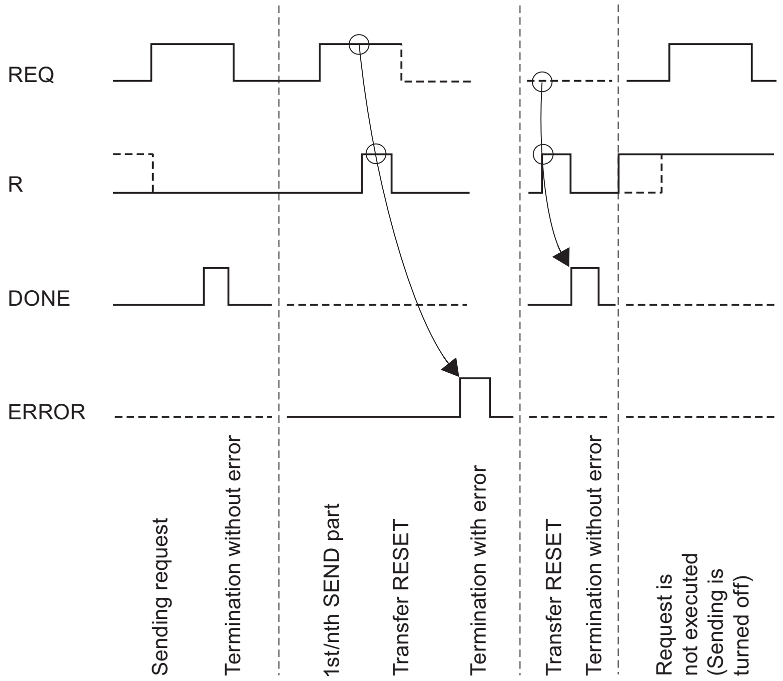

REQ
R
DONE
ERROR
Sending request
Termination without error
1st/nth SEND part
Transfer RESET
Termination with error
Transfer RESET
Termination without error
Request is not executed (Sending is turned off)

Figure 6-3 FB 8 P_SND_RK time sequence chart for "Send data" request

Note

The REQ input is edge-triggered. A positive edge at the REQ input is sufficient. The result of the logic operation must not be at "1" at any point during transfer.

6.4.2 Receiving Data with FB P_RCV_RK (Passive Request)

The FB P_RCV_RK transmits data from the CP 341 to an S7 data area. For purposes of data transmission, the FB P_RCV_RK is called cyclically or, alternatively, statically in a time-driven program (without conditions).

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

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

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

If the function block recognizes signal state "1" at the R parameter, the current transmission request is canceled and the FB P_RCV_RK is placed in its initial state. Receiving is disabled as long as the signal state at the R parameter is "1".

The NDR output shows "request completed without errors/data accepted" (all data read). The L_TYP, L_NO, and L_OFFSET parameters indicate for the duration of one cycle where the data are stored. In addition, the L_CF_BYT and L_CF_BIT parameters and length LEN of the respective request are indicated for one cycle.

Note

The P_RCV_RK function block does not detect if the partner sends a SEND or FETCH request.

Error display at the FB P_RCV_RK

ERROR indicates whether an error has occurred. In STATUS the event number is displayed in the event of an error. If no error has occurred, STATUS has the value 0. NDR and ERROR/STATUS are also output in response to a RESET of the FB P_RCV_RK (LEN parameter == 16#00). In the event of an error, the binary result BR is reset. If the block has been completed without errors, the binary result has the status "1".

Note

The P_RCV_RK function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode.

Use of interprocessor communication flags

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

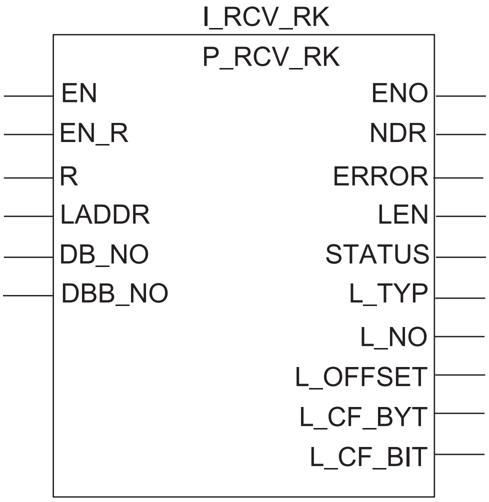
The user program can evaluate the interprocessor communication flag in order to find out if the transmitted data can be processed. As soon as the data have been processed, the user must reset the interprocessor communication flag to "0". The communication partner can now issue a SEND request again.

Block call

STL representation

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

LAD representation



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 FB P_RCV_RK works together with an I_RCV_RK instance DB. The DB number is passed on with the call. The instance data block is 60 bytes long. Access to the data in the instance DB is not permitted.

Note

Exception: If the error STATUS == W#16#1E0E occurred, you can consult the SFCERR variable for additional details. This error variable can only be loaded via symbolic access to the instance DB.

FB P_RCV_RK parameters

The table below lists the parameters of the FB 7 P_RCV_RK for the "Receive data" request.

Table 6- 6 FB 7 P_RCV_RK parameters for "Receive data" request

Name	Sort	Data type	Comment	Permitted values, remark
EN_R	INPUT	BOOL	Enables data receipt	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Receiving is disabled. Default: 0
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number of the receive data (destination)	Receive DB no.: CPU-specific, zero is not permitted (Relevant only for DX data destination)
DBB_NO	INPUT	INT	Data byte number of the receive data (destination)	0 ≤ DBB_NO ≤ 8190 Receive data starting from data byte (Relevant only for DX data destination)
L_TYP [1]	OUTPUT	CHAR	Type of area on local CPU (destination)	'D': Data block
L_NO [1]	OUTPUT	INT	Data block number on local CPU (destination)	0 ≤ L_NO ≤ 255
L_OFFSET [1]	OUTPUT	INT	Data byte number on local CPU (destination)	0 ≤ L_OFFSET ≤ 510
L_CF_BYT [1]	OUTPUT	INT	Interprocessor communication flag byte on local CPU	0 ≤ L_CF_BYTE ≤ 255255 means: without interprocessor communication flag
L_CF_BIT [1]	OUTPUT	INT	Interprocessor communication flag bit on local CPU	0 ≤ L_CF_BIT ≤ 7
NDR [1]	OUTPUT	BOOL	Request completed without errors, data accepted	STATUS parameter == 16#00;

Name	Sort	Data type	Comment	Permitted values, remark
ERROR [1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
LEN [1]	OUTPUT	INT	Length of the message frame received	0 ≤ LEN ≤ 4096, specified in number of bytes
STATUS [1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter contains the error information.
[1] The parameter is available until the next time the FB is called.				

Time sequence chart of FB P_RCV_RK

The figure below illustrates the behavior of the NDR, LEN, and ERROR parameters depending on the input circuit of EN_R and R.

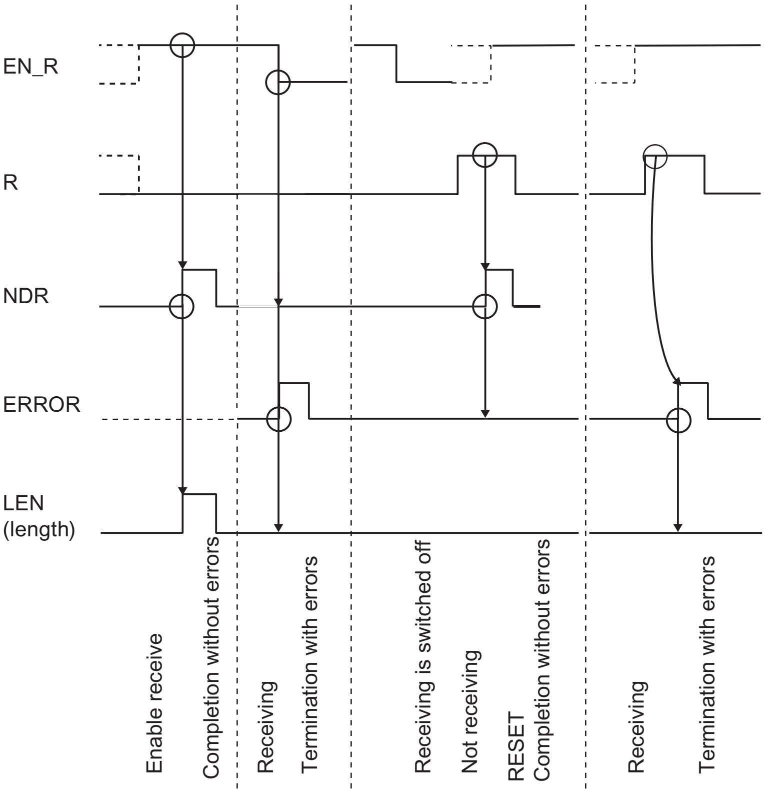


Figure 6-4 Time sequence chart of FB 7 P_RCV_RK for "Receive data" request

Note

Input EN_R must be set to static "1" state. The EN_R parameter must be supplied with result of logic operation "1" during the entire receive request.

6.4.3 Readying Data with FB P_RCV_RK (Passive Request)

The FB P_RCV_RK function block must be called if the communication partner executes a "Fetch data" request (FETCH request)..

The FB P_RCV_RK provides data from an S7 data area for the CP 341. For purposes of data transmission, the FB P_RCV_RK is called cyclically or, alternatively, statically in a time-driven program (without conditions).

A (static) signal state "1" at the EN_R, parameter enables a check to determine whether data can be provided to the CP 341. An active transmission can be canceled with signal state "0" at the EN_R parameter. The canceled request is terminated with an error message (STATUS output). The request is disabled as long as the signal state at the EN_R parameter is "0". A data transmission operation can run over several calls (program cycles), depending on the amount of data involved.

The type of source area (L_TYP), the source data block number (L_NO, relevant only when L_TYP = DB), the offset in the source area (L_OFFSET), and the length (LEN) of the data to be provided are determined from the first RK 512 message frame. The function block evaluates the information from this message frame and transfers the requested data to the CP 341. The DB_NO and DBB_NO parameters are irrelevant for the FB P_RCV_RK function block.

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

If the function block recognizes signal state "1" at the R parameter, the current transmission request is canceled and the FB P_RCV_RK is placed in its initial state. The request is disabled as long as the signal state at the R parameter is "1".

The NDR output shows "request completed without errors/data accepted" (all data read). The L_TYP, L_NO, and L_OFFSET parameters indicate for the duration of one cycle the location from which the data were fetched (possible data types: data blocks, input bytes, output bytes, timers and counters). In addition, the L_CF_BYT and L_CF_BIT parameters and length LEN of the respective request are indicated for one cycle.

Note

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

Error display at the FB P_RCV_RK

ERROR indicates whether an error has occurred. In STATUS the event number is displayed in the event of an error. If no error has occurred, STATUS has the value 0. NDR and ERROR/STATUS are also output in response to a RESET of the FB P_RCV_RK (LEN parameter == 16#00). In the event of an error, the binary result BR is reset. If the block has been completed without errors, the binary result has the status "1".

Note

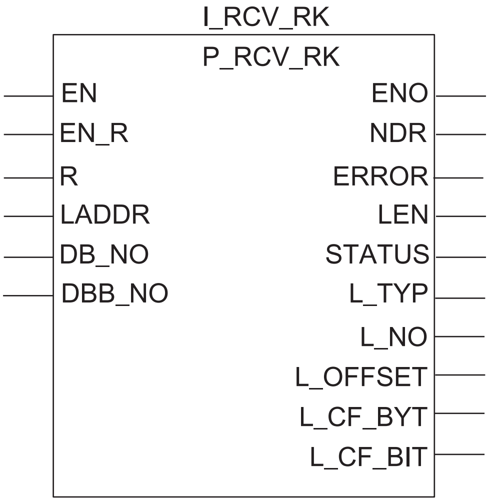
The P_RCV_RK function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode.

Use of interprocessor communication flags

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

The user program can evaluate the interprocessor communication flag in order to find out if the provided data can be accessed again. As soon as the data have been processed, the user must reset the interprocessor communication flag to "0". The communication partner can now issue a FETCH request again.

Block call

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

Note

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

Assignment in the data area

The FB P_RCV_RK works together with an I_RCV_RK instance DB. The DB number is passed on with the call. The instance data block is 60 bytes long. Access to the data in the instance DB is not permitted.

Note

Exception: If the error STATUS == W#16#1E0E occurred, you can consult the SFCERR variable for additional details. This error variable can only be loaded via symbolic access to the instance DB.

FB P_RCV_RK parameters

The table below lists the parameters of the FB 7 P_RCV_RK for the "Provide data" request.

Table 6- 7 FB 7 P_RCV_RK parameters for "Provide data" request

Name	Sort	Data type	Comment	Permitted values, remark
EN_R	INPUT	BOOL	Enables data provision	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Disables providing. Default: 0
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	irrelevant	
DBB_NO	INPUT	INT	irrelevant	
L_TYP [1]	OUTPUT	CHAR	Type of area on local CPU (source)	'D': Data block 'M' Memory bit 'I': Inputs 'O': Outputs 'C': Counters 'T': Timers
L_NO [1]	OUTPUT	INT	Data block number on local CPU (source)	0 ≤ L_NO ≤ 255 (only relevant, if L_TYP = D)
L_OFFSET [1]	OUTPUT	INT	Data byte number on local CPU (source)	0 ≤ L_OFFSET ≤ 510 (dependent on the area type)
L_CF_BYT [1]	OUTPUT	INT	Interprocessor communication flag byte on local CPU	0 ≤ CF_BYTE ≤ 255 255 means: without interprocessor communication flag
L_CF_BIT [1]	OUTPUT	INT	Interprocessor communication flag bit on local CPU	0 ≤ CF_BIT ≤ 7
NDR [1]	OUTPUT	BOOL	Request completed without errors, data accepted	STATUS parameter == 16#00;

Name	Sort	Data type	Comment	Permitted values, remark
ERROR [1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
LEN [1]	OUTPUT	INT	Length of the message frame received	0 ≤ LEN ≤ 4096, specified in number of bytes
STATUS [1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter contains the error information.
[1] The parameter is available until the next time the FB is called.				

Time sequence chart of FB P_RCV_RK

The figure below illustrates the behavior of the NDR, LEN, and ERROR parameters depending on the input circuit of EN_R and R.

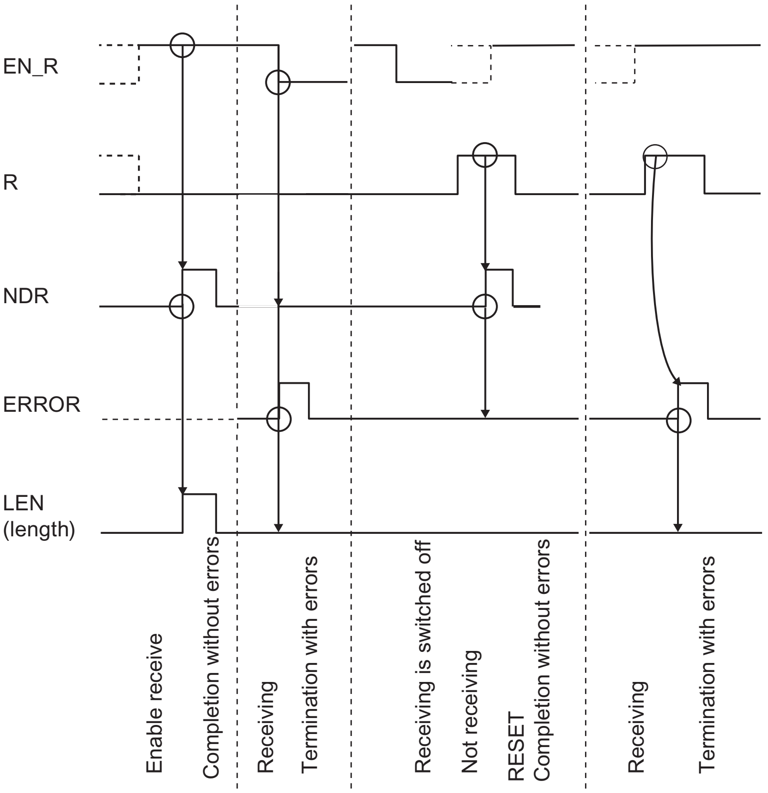


Figure 6-5 Time sequence chart of FB 7 P_RCV_RK for "Provide data" request

Note

Input EN_R must be set to static "1" state. The EN_R parameter must be supplied with result of logic operation "1" during the entire receive request.

6.4.4 Fetching Data with P_SND_RK FB (Active Request)

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

Note

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

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

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

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

The interprocessor communication flag byte and bit on the partner CPU are specified in R_CF_BYT and R_CF_BIT.

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

The FB P_SND_RK function block can be called cyclically when the signal state at the R parameter input is "1". This cancels the transmission from the CP 341 and places the FB P_SND_RK in its initial state. If the signal state at the R input remains static at "1", this means that fetching is deactivated.

Error display at the FB P_SND_RK

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. In STATUS the event number is displayed in the event of an error. If no error has occurred, the value of STATUS is 0. DONE and ERROR/STATUS are also output in response to a RESET of FB P_SND_RK. In the event of an error, the binary result BR is reset. If the block has been completed without errors, the binary result has the status "1".

Note

The FB P_SND_RK function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode.

Special features for (expanded) data blocks

Note the following special features with regard to fetching data from data blocks and expanded data blocks:

- RK 512 allows only an even-numbered amount of data to be fetched. If you specify an odd-numbered amount for the length (LEN), an extra byte is always transmitted. In the destination DB, however, the correct amount of data is entered.
- RK 512 allows only an even-numbered offset. If you specify an odd-numbered offset, the data is fetched from the partner starting from the next lower even-numbered offset.

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

Special features for timers and counters

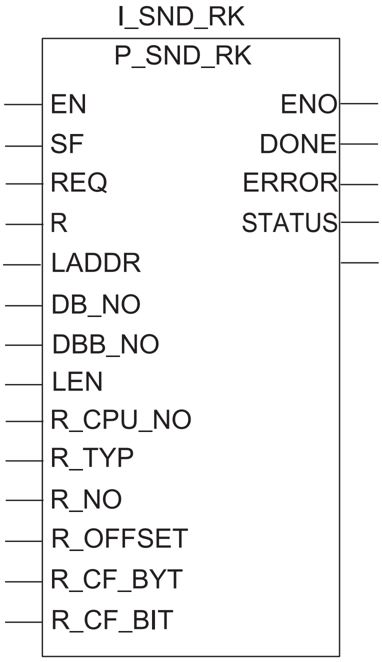
If you fetch timers or counters from the communication partner, remember that 2 bytes are fetched for each timer or counter. For example, if you want to fetch 10 counters, you must enter 20 as the length.

Block call

STL representation

```
CALL P_SND_RK, I_SND_RK
 SF: =
 REQ: =
 R: =
 LADDR: =
 DB_NO: =
 DBB_NO: =
 LEN: =
 R_CPU_NO: =
 R_TYP: =
 R_NO: =
 R_OFFSET: =
 R_CF_BYT =
 R_CF_BIT =
 DONE: =
 ERROR: =
 STATUS: =
```

LAD representation



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 FB P_SND_RK works together with an I_SND_RK instance DB. The DB number is passed on with the call. The instance data block is 62 bytes long. Access to the data in the instance DB is not permitted.

FB P_SND_RK parameters

The table below lists the parameters of the FB 8 P_SND_RK for the "Fetch data" request.

Table 6- 8 FB 8 P_SND_RK parameters for "Fetch data" request

Name	Sort	Data type	Comment	Permitted values, remark
SF	INPUT	CHAR	Selection for Send data or Fetch data	SF = 'F' (Fetch)default value: 'S' (Send)
REQ	INPUT	BOOL	Initiates request on positive edge	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Fetching is disabled. Default: 0
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number of destination	Send DB no.: CPU-specific, zero is not permitted
DBB_NO	INPUT	INT	Data byte number of destination	0 ≤ DBB_NO ≤ 8190 Send data starting from data byte
LEN	INPUT	INT	Data length of the message frame to be fetched	1 ≤ LEN ≤ 4096, specified in number of bytes[1]
R_CPU_NO	INPUT	INT	CPU number of the partner CPU	0 ≤ R_CPU_NO ≤ 4, only with multiprocessor mode; default value: 1
R_TYP	INPUT	CHAR	Address type on the partner CPU	'D': Data block 'X': Expanded data block 'M': Memory bit 'I': Inputs 'O': Outputs 'C': Counters 'T': Timers
R_NO	INPUT	INT	Data block number on the partner CPU	0 ≤ R_NO ≤ 255
R_OFFSET	INPUT	INT	Data byte number on the partner CPU	

Name	Sort	Data type	Comment	Permitted values, remark
R_CF_BYT	INPUT	INT	Interprocessor communication flag byte on partner CPU	0 ≤ CF_BYTE ≤ 255 Default value: 255 (means: without interprocessor communication flag)
R_CF_BIT	INPUT	INT	Interprocessor communication flag bit on the partner CPU	0 ≤ CF_BIT ≤ 7
DONE [1]	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR [1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
STATUS [1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter contains the error information.
[1] The parameter is available until the next time the FB is called.				

Parameters in the FB for data source (Partner CPU)

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

Table 6- 9 Transmittable data types for "Fetch data" request

Source on the partner CPU	R_TYP	R_NO	R_OFFSET (in bytes) (This value is specified by the partner CPU.)
Data block	'D'	0 - 255	0 - 510 (only even-numbered values are appropriate!)
Expanded data block	'X'	0 - 255	0 - 510 (only even-numbered values are appropriate!)
Bit memory	'M'	irrelevant	0 - 255
Inputs	'I'	irrelevant	0 - 255
Outputs	'O'	irrelevant	0 - 255
Counters	'C'	irrelevant	0 - 255
Timers	'T'	irrelevant	0 - 255

Information in the message frame header

The table below shows the information in the message frame header of the RK 512 message frame.

Table 6- 10 Information in the RK 512 message frame header for "Fetch data" request

Source on the partner CPU	To the destination, your S7 automation system (local CPU)	Message frame header, bytes		
		3/4 command type	5/6 S-DBNR/S-offset	7/8 number in
Data block	Data block	ED	DB/DW	Words
Expanded data block	Data block	EX	DB/DW	Words
Bit memory	Data block	EM	Byte address	Bytes
Inputs	Data block	EI	Byte address	Bytes
Outputs	Data block	EO	Byte address	Bytes
Counters	Data block	EC	Counter no.	Words
Timers	Data block	ET	Timer number	Words

Explanation of the abbreviations used:

S-DBNO: Source data block number

S-Offset: Source start address

Time sequence chart of FB P_SND_RK

The figure below illustrates the behavior of the DONE and ERROR parameters depending on the input circuit of REQ and R.

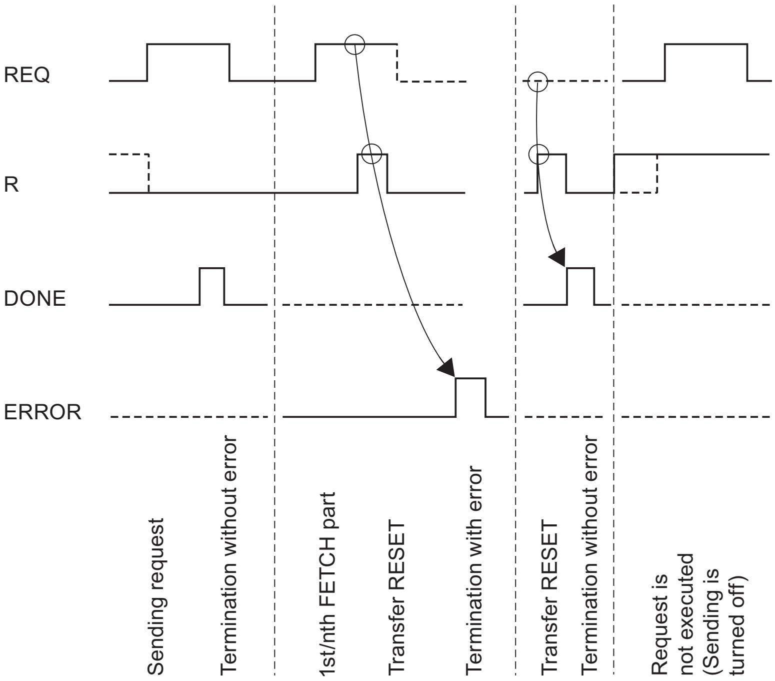

REQ
R
DONE
ERROR
Sending request
Termination without error
1st/nth FETCH part
Transfer RESET
Termination with error
Transfer RESET
Termination without error
Request is not executed (Sending is turned off)

Figure 6-6 Time sequence chart of FB 8 P_SND_RK for "Fetch data" request

Note

The REQ input is edge-triggered. A positive edge at the REQ input is sufficient. The result of the logic operation must not be at "1" at any point during transfer.

6.5 Using the System Function Blocks with the ASCII Driver

6.5.1 Function blocks with ASCII driver

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

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

The following function blocks are available for "reading and controlling" the RS 232C accompanying signals:

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

Note

For the CP 341, you can use only the FC 5 V24_STAT and FC 6 V24_SET functions with **version ≥ 2.0**. Use of Version 1.0 can result in data corruption.

6.5.2 Interface status of the CP 341, checking

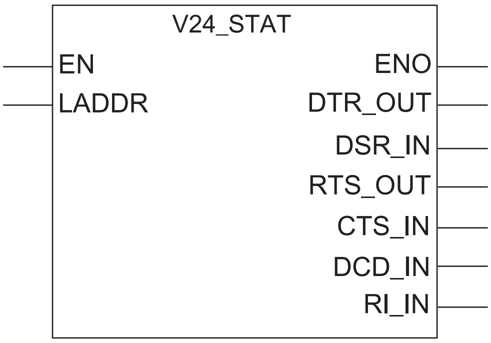
The FC V24_STAT reads the RS 232C accompanying signals from the CP 341 and makes them available to the user in the block parameters. The FC V24_STAT is called cyclically or, alternatively, statically (without conditions) in a time-controlled program.

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

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

The LADDR parameter defines the CP 341 to be addressed.

Block call

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

Note

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

Assignment in the data area

The FC V24_STAT function does not occupy any data areas.

Note

A minimum pulse duration is necessary to detect a signal change. Determining factors are the CPU cycle time, the update time on the CP 341, and the response time of the communication partner.

FC 5 V24_STAT parameters

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

Table 6- 11 FC 5 V24_STAT parameters

Name	Sort	Data type	Comment	Permitted values, remark
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DTR_OUT	OUTPUT	BOOL	Data terminal ready, CP 341 ready for operation	(CP 341 output)
DSR_IN	OUTPUT	BOOL	Data set ready, Communication partner ready for operation	(CP 341 input)
RTS_OUT	OUTPUT	BOOL	Request to send, CP 341 ready to send	(CP 341 output)
CTS_IN	OUTPUT	BOOL	Clear to send, Communication partner can receive data from the CP 341 (response to RTS = ON of the CP 341)	(CP 341 input)
DCD_IN	OUTPUT	BOOL	Data Carrier detect, receive signal level	(CP 341 input)
RI_IN	OUTPUT	BOOL	Ring Indicator, Indication of incoming call	(CP 341 input)

6.5.3 Interface outputs of the CP 341, setting/resetting

FC V24_SET

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

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

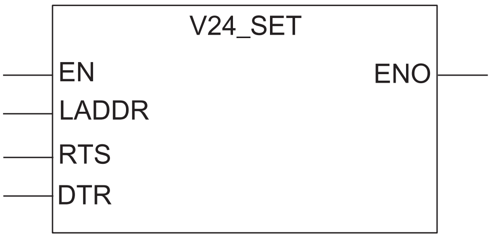
The LADDR parameter defines the CP 341 to be addressed.

Block call

STL representation

```
CALL
 V24_SET
 LADDR: =
 RTS: =
 DTR: =
```

LAD representation



Note

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

Assignment in the data area

The V24_SET function does not occupy any data areas.

FC 6 V24_SET parameters

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

Table 6- 12 FC 6 V24_SET parameters

Name	Sort	Data type	Comment	Permitted values, remark
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
RTS	INPUT	BOOL	Request to send, CP 341 ready to send	(Control CP 341 output)
DTR	INPUT	BOOL	Data terminal ready, CP 341 ready for operation	(Control CP 341 output)

See also

Principle of the Data Transmission with the ASCII Driver (Page 59)

6.6 Use of function blocks to output message texts to a printer

Introduction

The FB 13 P_PRINT_RK function block is available for printing message texts. The FB 13 P_PRINT_RK transmits a process message to the CP 341, for example. The CP 341 logs the process message on the connected printer.

Outputting message texts

The FB P_PRINT_RK transmits a message text with up to four variables to the CP 341. Configure the message texts with the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface. For the purpose of data transmission, the FB P_PRINT_RK is called cyclically, or alternatively, statically (without conditions) in a time-controlled program.

The DB_NO and DBB_NO parameters enable the pointers (pointers to data blocks) for the format string and the four variables to be accessed. The pointers must be stored in a certain order without gaps in the assigned data block (pointer DB) (see Figure "Pointer DB").

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

A data transmission operation can run over several calls (program cycles), depending on the amount of data involved.

The FB P_PRINT_RK function block can be called cyclically when the signal state at the R parameter input is "1". This cancels the transmission to the CP 341 and places the FB P_PRINT_RK in its initial state. Data that has already been received by the CP 341 is still sent to the communication partner. If the signal state at the R input remains static at "1", this means that sending of print requests is deactivated.

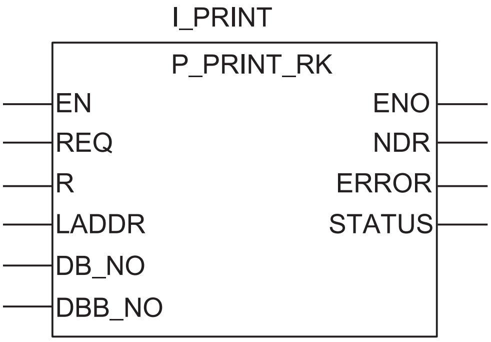
The LADDR parameter specifies the address of the CP 341 to be addressed.

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. If an event has occurred, the corresponding event number is displayed in STATUS. If no error has occurred, STATUS has the value 0. DONE and ERROR/STATUS are also output at RESET of the FB P_PRINT_RK. The binary result BR is reset after an error has occurred. If the block has been completed without errors, the binary result has the status "1".

Note

The P_PRINT function block does not have a parameter check. If the parameter assignment is incorrect, the CPU can go to STOP mode. Before the CP 341 can process an activated request after the CPU has changed from STOP to RUN mode, the CP-CPU startup mechanism of the FB P_PRINT_RK must have been completed. Any requests initiated in the meantime are not lost. They are transmitted once the start-up coordination with the CP 341 is finished.

Block call

STL representation	**LAD representation**
CALL P_PRINT_RK, I_PRINT REQ: = R: = LADDR: = DB_NO: = DBB_NO: = DONE: = ERROR: = STATUS: =	 I_PRINT P_PRINT_RK EN REQ R LADDR DB_NO DBB_NO ENO NDR ERROR 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, instance DB

The FB P_PRINT_RK works together with an I_PRINT instance DB. The DB number is passed on with the call. The instance data block is 40 bytes long. Access to the data in the instance DB is not allowed.

Note

Exception: If the error STATUS == W#16#1E0F occurred, you can consult the SFCERR or SFCSTATUS variable for additional details. See "Calling the SFCERR or SFCSTATUS variable" for more information.

Assignment in the data area, pointer DB

The FB P_PRINT_RK 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 sequence. You have to create the pointer DB.

The figure shows the structure of the pointer DB that is addressed by the DB_NO and DBB_NO parameters of the FB P_PRINT_RK.

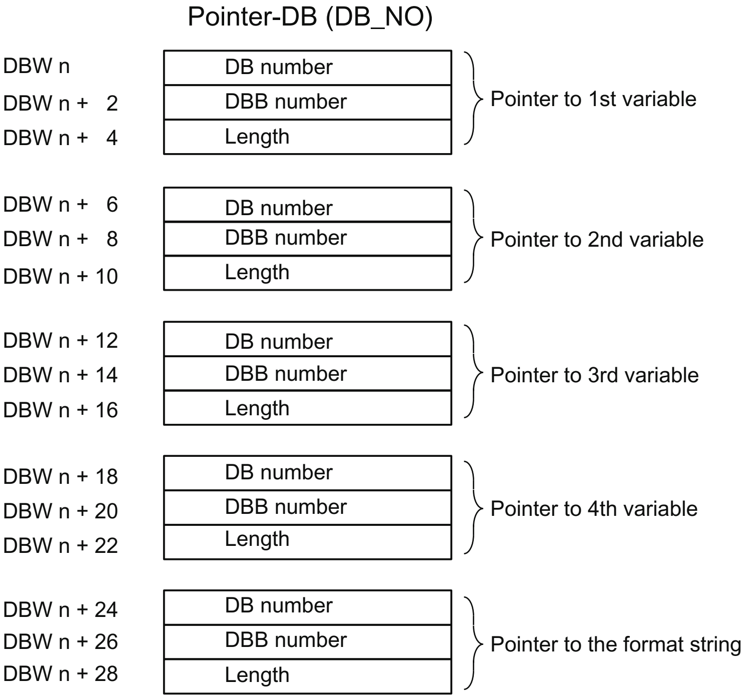


Figure 6-7 Structure of the POINTER DB for the FB P_PRINT_RK function block

Permissible DB numbers

The permissible DB numbers are CPU-specific. If the value 16#00 is specified as the DB number for "Pointer to variable", this variable is interpreted as not present and the pointer is set to the next variable or the format string.

If the DB number is equal to the value 16#00 in "Pointer to format string", the print request is canceled and event number 16#1E43 is indicated in the STATUS parameter output of the FB P_PRINT_RK.

Permissible DBB numbers

The variable or format string is located starting at the assigned DBB number. The maximum length of the variables is 32 bytes and the maximum length of the format string is 150 bytes.

If the maximum length is exceeded, the print request is canceled and event number 16#1E41 is indicated in the STATUS parameter output of the FB P_PRINT_RK.

Permissible length

The length information in the pointer DB must be adapted for the particular display type (data types) and according to the precision used.

FB 13 P_PRINT_RK parameters

The following table lists the parameters of the FB 13 P_PRINT_RK.

Table 6- 13 FB 13 P_PRINT_RK parameters

Name	Sort	Data type	Comment	Permitted values, remark
REQ	INPUT	BOOL	Initiates request on positive edge	
R	INPUT	BOOL	Cancels request	Cancels the request in progress. Printer output is disabled.
LADDR	INPUT	INT	CP 341 start address	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Pointer to pointer DB: CPU-specific, zero is not allowed (The pointers to variables and format string are stored in the pointer DB in a fixed order.)
DBB_NO	INPUT	INT	Data byte number	0 ≤ DBB_NO ≤ 8162 Pointer starting from data byte
DONE [1]	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00;
ERROR	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
STATUS	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter will contain error information.
[1] The DONE parameter is available for one CPU cycle following a successful send request.				

Time sequence chart for FB 13 P_PRINT_RK

The figure below illustrates the behavior of the DONE and ERROR parameters depending on the input circuit of REQ and R.

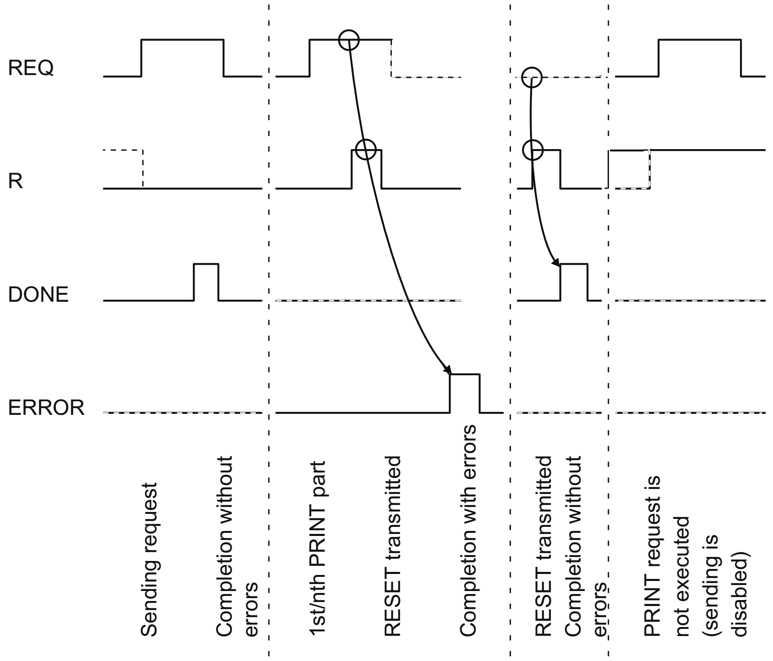


Figure 6-8 Time sequence chart for FB 13 P_PRINT_RK

Note

The REQ input is edge-triggered. A positive edge at the REQ input is sufficient. It does not have to have a signal state of "1" during the entire transmission operation.

6.7 General Information on Program Processing

Start-up Characteristics of the CP 341 Programmable Controller

The configuration data is created with the aid of the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment and transferred to the CPU with the STEP 7 software. Each time the CPU is started up, the current parameters are transferred to the CP 341 by the system service of the CPU.

Start-up Characteristics, FB-CP 341

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

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

Disabling alarms

In the function blocks the interrupts are not disabled.

Addressing the module

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

6.8 Technical Specifications of the Function Blocks

Memory requirements

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

Table 6- 14 Memory requirements of the function blocks/functions in bytes

Block	Name	Version	Load memory	Working memory	Local data
FC 5	V24_STAT	2.0	182	108	2
FC 6	V24_SET	2.0	150	84	2
FB 7	P_RCV_RK	3.2	3756	3144	114
FB 8	P_SND_RK	3.2	3490	2900	40

Minimum number of CPU cycles

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

Table 6- 15 Minimum number of CPU cycles

	Number of CPU cycles for processing ...		
	Completion without errors	Termination with errors	RESET/RESTART
P_RCV_RK	≥ 3	≥ 3	≥ 4
P_SND_RK	≥ 3	≥ 3	≥ 4
V24_STAT	1	-	-
V24_SET	2	>> 2	-

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

Before the CP 341 can receive a message frame in the user program after the CPU has changed from STOP to RUN mode, the CP-CPU startup mechanism P_RCV_RK must have been completed.

System functions used

The following system functions are used in the blocks:

- SFB 52 (RDREC), Read data set
- SFB 53 (WRREC), Write data set

NOTICE
The new standard function blocks of the CP341 • FB7 ≥ V3.0 (P_RCV_RK) • FB8 ≥ V3.0 (P_SND_RK) and • and FB13 ≥ V1.0 (P_PRT341) operate with the new system function blocks SFB52 (RDREC) or SFB53 (WRREC) that are supported by the latest CPU versions (*) only. This conversion was necessary because the old system calls SFC58 (WR_REC) and SFC59 (RD_REC) are not suitable for operation following an IE/PB link or a PROFINET header. Users with older CPU versions that do not support SFB52 (RDREC) or SFB53 (WRREC) yet will find the older versions of FB7 and FB8 with the designations FB107 (P_RCV_RK_OLD) and FB108 (P_SND_RK_OLD) in their regular location. (*) All S7-300 CPUs with MMC and S7-400-CPUs as of firmware status V3.0.0 support the new system function blocks SFB52 and SFB53. CPU318 will permit the use of SFB52/SFB53 with decentralized periphery only.

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

7

7.1 Operating Modes of the CP 341

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

STOP

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

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

Parameter reassignment

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

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

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

RUN

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

7.2 Startup Characteristics of the CP 341

CP 341 start-up

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

- Initialization (CP 341 in POWER ON mode)
- Parameter assignment

Initialization

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

Parameter assignment

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

A parameter reassignment check is executed. The CP 341 is now ready to operate.

Startup behavior when power supply for the module is turned on

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

To reestablish communication between CPU and CP 341, set the CPU to STOP and then to RUN.

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

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

General procedure

CPU-STOP

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

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

CPU START-UP

On start-up, the CP sends the parameters created with the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface. The CP 341 carries out a parameter reassignment only if parameters have changed.

CPU RUN

When the CPU is in RUN mode, send and receive operations are unrestricted. In the first FB cycles following the CPU restart, the CP 341 and the corresponding FBs are synchronized. No new FBs or FCs can be executed until this process has been completed.

Special features for sending message frames and for printout

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

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

Note

The CP 341 sends data to the communication partner only after it has received all the data from the CPU.

Diagnostics with the CP 341 8

8.1 Diagnostics Functions of the CP 341

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

- Diagnostics via Display Elements of the CP 341
- Diagnosis via the STATUS output of the function blocks
- Diagnostics via the diagnostic buffer of the CP 341
- Diagnosis via the diagnostic alarm

Display elements (LEDs)

The display elements show the operating mode or possible error states of the CP 341. The display elements give you an initial overview of any internal or external errors as well as interface-specific errors (see chapter "Diagnostics via the display elements of the CP 341 (Page 177)").

In section "Subsequent Loading of Drivers (Transmission Protocols) (Page 117)" provides information on other LED indicators that occur when loading a firmware update.

STATUS outputs of the FBs

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

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

Error numbers in the response message frame

If you are working with the RK 512 computer link and an error occurs at the communication partner in a SEND or FETCH message frame, the communication partner sends a response message frame with an error number in the 4th byte (see section "Error Numbers in the Response Message Frame (Page 194)").

Diagnostic buffer of the CP 341

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

In the same way as with the diagnostic buffer of the CPU, you can also use the STEP 7 information functions on the programming device to display the user-relevant information of the CP diagnostic buffer (see section "Diagnostics via the diagnostic buffer of the CP 341 (Page 195)").

Diagnostic interrupt

The CP 341 can trigger a diagnostic interrupt on the CPU assigned to it. CP 341 provides 4 bytes of diagnostics information at 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 diagnostic buffer.

The CP 341 also writes diagnostic events which have triggered a diagnostic interrupt to its diagnostic buffer.

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

8.1.1 Diagnostics via the display elements of the CP 341

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

- Group error displays
  - **SF (red)** Error has occurred or parameter reassignment is in progress.
- Special displays
  - **TXD (green)** Sending active; lights up when the CP 341 is sending user data via the interface
  - **RXD (green)** Receiving active; lights up when the CP 341 is receiving user data via the interface

Note

Section "Subsequent Loading of Firmware Updates (Page 118)" provides information on other LED indicators that occur when loading a firmware update.

Group error display SF

The group error display SF always lights up after POWER ON and goes out after initialization. If configuration data has been generated for the CP 341, the SF LED again lights up briefly during parameter reassignment.

The group error display SF lights up whenever the following occur:

- Hardware fault
- Firmware error
- Parameter assignment error or no parameter assignment exists
- BREAKs (receive cable between CP 341 and communications partner becomes disconnected)

8.1.2 Diagnostics Messages of the Function Blocks

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

Numbering scheme for event class/event number

The figure below illustrates the structure of the STATUS parameter.

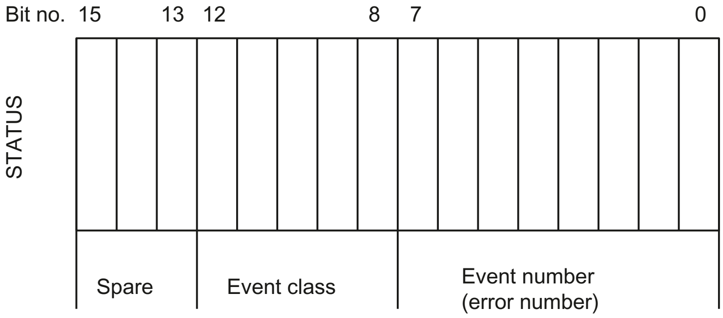


Figure 8-1 Structure of the STATUS parameter

Example

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

Event: "Request canceled due to complete restart, restart, or reset"

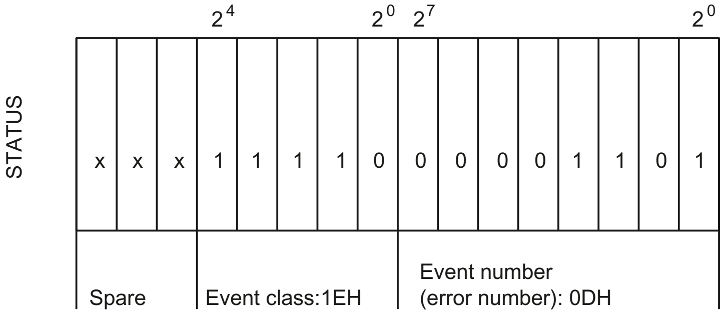


Figure 8-2 Example: Structure of the STATUS parameter for the the "Request canceled due to complete restart, restart or reset" event

Event classes

The table below describes the various event classes and numbers.

Table 8- 1 Event classes and event numbers

Event class 0 (00H): "CP start-up"		
Event no.	**Event text**	**Remedy**
(00)03H	PtP parameters received	-
(00)04H	Parameters already on CP (time versions match)	-
(00)07H	Status transition CPU to STOP	-
(00)08H	Status transition CPU to RUN/STARTUP	-

Event Class 1 (01H): "Hardware fault on CP"		
Event no.	**Event text**	**Remedy**
(01)01H	Fault while testing operating system EPROM of CP	CP defective; replace CP.
(01)02H	RAM test of CP errored	
(01)03H	Request interface of CP defective	
(01)10H	Fault in CP firmware	Switch module off and on again. If necessary, replace module.

Event Class 2 (02H): "Initialization error"		
Event no.	**Event text**	**Remedy**
(02)0FH	Invalid parameter assignment detected at start of assigned communication. Interface parameters could not be assigned.	Correct invalid parameters and restart.

Event class 3(03H): "Error during parameter assignment of FBs" (not displayed in the diagnostic buffer)		
Event no.	**Event text**	**Remedy**
(03)01H	Illegal or missing source/destination data type Area (start address, length) not permitted The DB is not permitted or does not exist (e.g., DB 0) or Other data type invalid or missing Interprocessor communication byte number invalid or Interprocessor communication bit number invalid or neither 'S' nor 'F' selected (for FB P_SND_RK)	Check parameter assignment on CPU and CP, and correct if necessary. **RK 512 only:** Partner returns invalid parameters in message frame header. Check parameter assignment on CPU and CP; create block, if necessary. See request tables for valid data types. **RK 512 only:** Partner returns incorrect parameters in message frame header.

Event Class 4 (04H): "CP detected error in data traffic CP - CPU"		
Event no.	**Event text**	**Remedy**
(04)03H	Incorrect, unknown or illegal data type	Check program, e.g. for incorrect parameter assignment of FB.
(04)07H	An error has occurred during data transmission between the CPU and CP	If the error is signaled continuously, you must check whether the parameters of the FBs called in the user program are correctly assigned. If the error is signaled immediately after POWER ON, a connection to the CPU is not set up at this time. With the 3964(R) procedure and the ASCII driver, the data transmission of the receiving CP 341 is repeated until the data are transferred to the CPU. With RK 512, the request is acknowledged negatively and must be repeated in the user program. If the message occurs sporadically during active data transmission, the CPU can not accept the data at times. With the 3964(R) procedure and the ASCII driver, the data transmission of the receiving CP 341 is repeated until the data are transferred to the CPU. With RK 512, the request is acknowledged negatively and must be repeated in the user program. As a remedy, you should call the FB P_RCV_RK more frequently in your user program.

Event Class 4 (04H): "CP detected error in data traffic CP - CPU"		
Event no.	**Event text**	**Remedy**
(04)08H	Error during data transmission between the CPU and the CP (receipt) • CPU is temporarily overloaded, request is repeated • The data area of the CPU cannot be accessed for the time being, e.g., because the receive block is not called often enough. • The data area of the CPU cannot be accessed for the time being, e.g., because the receive block is disabled in the interim (EN=false).	• Reduce the number of communication calls • Call the receive block more often • Check whether the receive block has been disabled too long
(04)09H	Data cannot be received. Error during data transmission between the CPU and the CP (receipt). Data cannot be received. After multiple attempts, the request was cancelled after 10 s, because • Receive block is not called • Receive block is disabled • Data area of the CPU cannot be accessed • Data area of the CPU is too small	• Check your user program to determine whether the receive block is being executed. • Check whether the receive block is disabled. • Check whether the data area to which the data is to be transferred is available. • Check the length of the data area
(04)0AH	An error has occurred during data transmission between the CPU and CP. The data transmission was canceled due to a RESET, because: • Destination DB is not available • Destination DB is too short • RESET bit set at FB	Create destination DB in the user program or increase the length of the existing destination DB.

Event class 5 (05H): "Error while processing CPU request"		
Event no.	Event	Remedy
(05)01H	Current request canceled as a result of CP restart.	No remedy is possible in the case of a POWER ON. In the case of parameter reassignment of the CP by the programming device, before writing an interface you should ensure there are no more requests running from the CPU.
(05)02H	Request is not permitted in this CP operating mode (e.g., device interface parameters not assigned).	Assign parameters for the device interface.
(05)05H	Printer driver only: System data block with message texts not available on the CP	Use the parameter assignment software to configure the message texts, and then carry out a complete restart.
(05)06H	Printer driver only: Message text not available	Use the parameter assignment software to configure the message texts, and then carry out a complete restart.
(05)07H	Printer driver only: Message text too long	Edit the message text to reduce it to a length of fewer than 150 characters (or no more than 250 characters if it contains variables)
(05)08H	Printer driver only: Too many conversion statements	You have configured more conversion statements than variables. Conversion statements without associated variables will be ignored.
(05)09H	Printer driver only: Too many variables	You have configured more variables than conversion statements. Variables without associated conversion statements will not be output.
(05)0AH	Printer driver only: Unknown conversion statement	Check the conversion statement. Undefined or unsupported conversion statements are replaced in the printout with ******.
(05)0BH	Printer driver only: Unknown control statement	Check the control statement. Undefined or unsupported control statements will be ignored. The control statement will not be output as text either.
(05)0CH	Printer driver only: Conversion statement not executable	Check the conversion statement. Conversion statements that cannot be executed are output in the printout in accordance with the defined width and the valid remainder of the conversion statement or the default display is output with "*" characters.
(05)0DH	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 possible number of characters in text-based display 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	Only for ASCII drivers: An error occurred while sending. The defined end-of-text characters did not occur within the maximum allowed length or in the case of automatic appending, the maximum allowed transmission length was exceeded.	Extend the end-of-text characters in the transmission buffer at the desired point or select a shorter message frame length for automatic appending.

Event class 5 (05H): "Error while processing CPU request"		
Event no.	**Event**	**Remedy**
(05)14H	Specified start addresses too high for desired data type, or start address or DB/DX number too low.	Refer to the request tables for the permissible start addresses and DB/DX numbers that can be specified in the program.
(05)15H	RK 512 only: Incorrect bit number specified in the interprocessor communication flag.	Permissible bit numbers: 0 to 7
(05)16H	RK 512 only: Specified CPU number too high.	Permissible CPU no.: none, 0, 1, 2, 3, or 4
(05)17H	Transmission length > 1 kbyte is too long for CP, or length is too short for interface parameter.	Split the request up into several shorter requests.
(05)1AH	RK 512 only: Error sending a command message frame An associated procedure number has just been entered in STATUS.	See the remedies for the previous error number.
(05)1BH	Printer driver only: Precision invalid	Correct the specified precision in the conversion statement. The precision is always preceded by a period to identify and limit the width (for example, ".2" to output the decimal point and two decimals.) Precision is only relevant to display types A, D, F, and R. It is ignored otherwise.
(05)1CH	Printer driver only: Variable invalid (variable length incorrect/incorrect type)	Correct the specified variable. The corresponding table indicates the possible data types for each display type.
(05)1EH	Printer driver only: The "line end sequences" sent with this request (i.e.: $R / $L / $N) do not fit (any longer) on the (initial) page	Increase the length of your page, reduce the number of lines (or line feeds) or spread your printout over a number of pages

Event class 6 (06H): "Error when processing a partner job" only for RK 512		
Event no.	**Event text**	**Remedy**
(06)01H	Error in 1st command byte (not 00 or FFH)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)02H	Error in 3rd command byte (not A, 0 or E)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)03H	Error in 3rd command byte in the case of continuation message frames (command not as for 1st message frame)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)04H	Error in 4th command byte (command letter incorrect)	Header layout error at partner or a command combination has been requested that is not permitted at the CP. Check the permissible commands. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)05H	Error in 4th command byte in the case of continuation message frames (command not as for 1st message frame)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)06H	Error in 5th command byte (DB number not permissible)	Refer to the request tables for the permissible DB numbers, start addresses, or lengths.
(06)07H	Error in the 5th or 6th command byte (start address too high)	Refer to the request tables for the permissible DB numbers, start addresses, or lengths.
(06)08H	Error in 7th or 8th command byte (impermissible length)	Obtain from the request tables the permissible DB/DX numbers, start addresses or lengths.
(06)09H	Error in 9th and 10th command byte (coordination flag for this data type impermissible or bit number too high)	Basic header structure error on partner. Refer to the request tables for when a coordination flag is permitted.
(06)0AH	Error in the 10th command byte (illegal CPU number)	Header layout error at partner
(06)0BH	SEND message frame was longer/shorter than expected (more/less data received than announced in message frame header).	Correction required at the partner
(06)0CH	FETCH command message frame with user data received.	Correction required at the partner
(06)0DH	The CP received a message frame during an invalid operating mode: • Receive connection between the CPU and CP is not set up or not yet correctly set up • CP startup is not fully completed • The receiving CPU is in STOP mode • Parameters of the addressed interface are currently being reassigned	• Check whether the addressed connection has been assigned the correct parameters • This error message can only occur only during CP startup. Repeat the request. • Place the CPU back in RUN mode and repeat the request. • This is a temporary error. Repeat the request.

Event class 6 (06H): "Error when processing a partner job" only for RK 512		
Event no.	**Event text**	**Remedy**
(06)0EH	Synchronous fault of partner • New (continuation) command message frame received before response message frame was sent. • 1. command message frame expected and continuation message frame came. • Continuation command message frame expected and 1st message frame came.	This error may be reported after your own automation system is restarted in the case of long message frames or when the partner is restarted. These cases represent normal system start-up behavior. The error can also occur during operation as a consequence of error statuses only recognized by the partner. Otherwise, you have to assume an error on the part of the partner device. The error may not occur in the case of requests < 128 bytes.
(06)0FH	DB locked by coordination function	In local program: Reset the interprocessor communication flag after the last transmission data was processed. In the partner program: Repeat the request
(06)10H	Message frame received too short (length < 4 bytes in the case of continuation or response message frames or <10 bytes in the case of command message frames)	Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)11H	Message frame length and length specified in message frame header are not the same.	Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)12H	Error while sending the (continuation) response message frame. An associated procedure error number has been entered in STATUS immediately beforehand.	See remedies for the error number entered immediately beforehand in STATUS.

Event class 7 (07H): "Send error"		
Event no.	Event	Remedy
(07)01H	Sending of the first repetition: • An error was detected when transmitting the message frame, or • The partner requested a repetition by means of a negative acknowledgment character (NAK).	A repetition is not an error, but it can indicate that there is interference on the data link or the partner device has malfunctioned. If the message frame still has not been transmitted after the maximum number of repetitions, an error number describing the first error that occurred is output.
(07)02H	3964(R) only: Error establishing connection: After STX was sent, NAK or any other character (except for DLE or STX) was received.	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(07)03H	3964(R) only: Acknowledgment delay time (QVZ) exceeded: After STX was sent, partner did not respond within the acknowledgment delay time.	The partner device is too slow or not ready to receive, or there is a break in the transmission line, for example. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(07)04H	3964(R) only: Cancellation by partner: One or more characters were received from the partner during sending.	Check whether the partner also indicates an error; possibly it has not received all of the transmitted data (for example, due to an interrupted data link), or because fatal errors are pending, or the behavior of the partner device is faulty. If necessary, use an interface test device switched into the transmission line to check.
(07)06H	3964(R) only: End-of-connection error: • Partner rejected message frame at end of connection with NAK or a random string (except for DLE), or • Acknowledgment characters (DLE) received too early.	Check whether the partner also indicates an error; possibly it has not received all of the transmitted data (for example, due to an interrupted data link), or because fatal errors are pending, or the behavior of the partner device is faulty. If necessary, use an interface test device switched into the transmission line to check.
(07)07H	3964(R) only: Acknowledgment delay time exceeded at end of connection or response monitoring time exceeded after a send message frame: After connection termination with DLE ETX, no response received from partner within acknowledgment delay time.	Partner device too slow or faulty. If necessary, use an interface test device switched into the transmission line to check.

Event class 7 (07H): "Send error"		
Event no.	**Event**	**Remedy**
(07)08H	**ASCII driver and printer driver only:** The waiting time for XON or CTS = ON has expired.	The communication partner is faulty, too slow, or has been taken offline. Check the communication partner; you may need to change the parameter assignment.
(07)09H	Not possible to connect. Number of permitted setup attempts exceeded.	Check the interface cable or the transmission parameters. Also check whether the receive function between the CPU and CP is assigned correctly in the partner device.
(07)0AH	Data could not be transmitted. The permitted number of transmission attempts was exceeded.	Check the interface cable or the transmission parameters.

Event class 8 (08H): "Receive error"		
Event no.	**Event**	**Remedy**
(08)01H	Expectation of the first repetition: An error was detected on receiving a message frame and the CP requested repetition from the partner via a negative acknowledgment (NAK).	A repetition is not an error, but it can indicate that there is interference on the data link or the partner device has malfunctioned. If the message frame still has not been transmitted after the maximum number of repetitions, an error number describing the first error that occurred is output.
(08)02H	3964(R) only: Error establishing connection: • In idle mode, one or more random characters (other than NAK or STX) were received, or • After an STX was received, the partner sent more characters without waiting for the response DLE. After partner power ON: • While partner is being switched on, the CP receives an undefined character.	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(08)05H	3964(R) only: Logical error while receiving: After DLE was received, a further random character (other than DLE or ETX) was received.	Check whether the partner always duplicates the DLE in the message frame header and data string or the connection is released with DLE ETX. Check for malfunction at partner device, possibly by using interface test device switched into the transmission line.
(08)06H	Character delay time (ZVZ) exceeded: • Two successive characters were not received within the character delay time, or 3964(R) only: • 1. character after sending of DLE while setting up connection was not received within the character delay time.	Partner device too slow or faulty. Prove this using an interface test device interconnected in the data link.
(08)08H	3964(R) only: Error in block check character (BCC): The value of the BCC calculated internally does not match the BCC received by the partner when the connection was terminated.	Check whether the connection is seriously disrupted; in this case you may also occasionally see error codes. Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(08)0AH	There is no free receive buffer available: No receive buffer space available for receiving data.	The FB P_RCV_RK must be called more frequently.

Event class 8 (08H): "Receive error"		
Event no.	**Event**	**Remedy**
(08)0CH	Transmission error: • A transmission error (parity error, stop bit error, overflow error) was detected. 3964(R) only: • If a corrupted character is received in idle mode, the error is reported immediately so that disturbances on the transmission line can be detected early. RK 512 and 3964(R) only: • If this happens during send or receive operations, repetition is started.	Disturbances on the data link cause message frame repetitions, thus lowering user data throughput. The risk of an undetected error increases. Change your system setup or cable wiring. Check the connecting cable of the communication partners or check whether both devices have the same setting for baud rate, parity and number of stop bits.
(08)0DH	BREAK: Break in receive line to partner.	Reconnect or switch on partner.
(08)15H	Discrepancy between settings for transmission attempts of the CP and the communication partner.	Set the same number of transmission attempts for the communication partner as for the CP. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(08)16H	• The length of a received message frame exceeded the maximum specified length.	Correction at the partner is necessary.
(08)18H	ASCII driver only: DSR = OFF or CTS = OFF	The partner has switched the DSR or CTS signal to "OFF" before or during a transmission. Check the partner's control of the RS 232C accompanying signals.

Event class 9 (09H): "Response message frame received from interconnection partner with errors or error message frame"		
Event no.	**Event text**	**Remedy**
(09)02H	**Only for RK 512:** Memory access error in partner (memory does not exist) **With SIMATIC S5 as partner:** • Incorrect area for indicator word, or • Data area does not exist (except DB/DX), or • Data area too short (except DB/DX)	Check that the partner has the desired data area and that it is big enough, or check the parameters of the called system function block. Check the length specified in the system function block.
(09)03H	**Only for RK 512:** DB/DX access error in the partner (DB/DX does not exist or is too short) **With SIMATIC S5 as partner:** • DB/DX does not exist, or • DB/DX too short, or • DB/DX number impermissible Permissible source area exceeded with FETCH request	Check that the partner has the desired data area and that it is big enough, or check the parameters of the called system function block. Check the length specified in the system function block.
(09)04H	**Only for RK 512:** Partner reports "Illegal request type".	Partner malfunction, because the CP never outputs a system command.
(09)05H	**RK 512 only:**Partner error or SIMATIC S5 partner error: • Source/destination type not permissible, or • Memory error in partner programmable controller, or • Error notifying CP/CPU at the partner, or • Partner programmable controller is in STOP mode	Check if the partner can transmit the desired data type. Check the configuration of the partner hardware. Set the mode selector switch of the partner programmable controller to RUN.
(09)08H	**Only for RK 512:** Partner detects synchronous error: Message frame sequence error.	This error occurs at the restart of your own programmable controller or of the partner. This represents normal system startup behavior. You do not need to correct anything. The error is also conceivable during operation as a consequence of previous errors. Otherwise, you can assume a malfunction of the partner device.
(09)09H	**Only for RK 512:** DB/DX locked at the partner by coordination flag	In the partner program: Reset the coordination memory after the last transmission data was processed! In the program: Repeat the request!

Event class 9 (09H): "Response message frame received from interconnection partner with errors or error message frame"		
Event no.	Event text	Remedy
(09)0AH	Only for RK 512: Errors in message frame header that are detected by the partner: 3rd command byte in header is incorrect	Check whether the error was caused by faults or by malfunction of the partner station. Prove this using an interface test device interconnected in the data link.
(09)0BH	Only for RK 512: Error in message frame header: 1. or 4th command byte in header is incorrect	Check whether the error was caused by faults or by malfunction of the partner station. Prove this using an interface test device interconnected in the data link.
(09)0CH	Only for RK 512: Partner detects incorrect message frame length (total length).	Check whether the error was caused by faults or by malfunction of the partner station. Prove this using an interface test device interconnected in the data link.
(09)0DH	Only for RK 512: Partner was not yet restarted.	Restart the partner programmable controller or set the mode selector switch on the CP to RUN.
(09)0EH	Only for RK 512: Unknown error number received in the response message frame.	Check whether the error was caused by faults or by malfunction of the partner station. Prove this using an interface test device interconnected in the data link.

Event class 10 (0AH): **"Errors in response message frame of the partner detected by the CP"**		
(0A)01H	**Only for RK 512:** Synchronization error of partner, because: • Response message frame without request • Response message frame received before continuation message frame sent • Continuation response message frame received after an initial message frame was sent • A first response message frame was received after a continuation message frame was sent	This error is reported after your own programming device is restarted in the case of long message frames or when the partner is restarted. This represents normal system start-up behavior. You do not have to correct anything. The error can also occur during operation as a consequence of error statuses only recognized by the partner. Otherwise, you can assume an error on the part of the partner device. The error may not occur in the case of requests < 128 bytes.
(0A)02H	RK 512 only: Error in the structure of the received response message frame (1st byte not 00 or FF)	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(0A)03H	RK 512 only: Received response message frame has too many or too few data.	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(0A)04H	RK 512 only: Response message frame for SEND request arrived with data.	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(0A)05H	RK 512 only: No response message frame received from partner within monitoring time.	Is the partner a slow device? This error is also often displayed as a consequence of a previous error. For example, procedure receive errors (event class 8) can be displayed after a FETCH message frame was sent. Reason: The response message frame could not be received because of disturbances. The monitoring time expires. This error also possibly occurs if the partner is restarted before it could respond to the last received FETCH message frame.

Displaying and evaluating the STATUS output

You can display and evaluate the actual addresses in the STATUS output of the function block.

Note

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

Event class 30

Event class 30 contains error messages that can be output during communication via the S7 backplane bus between the CP 341 and the CPU.

The table below contains a description of event class 30.

Table 8- 2 Event class 30

Event class 30 (1EH): **"Communication error between CP and CPU"**		
Event no.	Event	Additional information/Remedy
(1E)0DH	Request canceled due to complete restart, restart, or reset	
(1E)0EH	Static error when the RD_REC SFC was called. Return value RET_VAL of SFC is available for evaluation in SFCERR variable in instance DB.	Load the SFCERR variable from the 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 the SFCERR variable from the instance DB.
(1E)41H	The number of bytes specified in the LEN parameter of the FBs is not permissible	You must stay within a range of values of 1 to 4096 bytes.
(1E)42H	FB P_PRINT_RK: The number of bytes specified for the variable or format string in the pointer DB under length is not permissible.	You must specify a permissible length: 32 bytes for variables, 150 bytes for format string
(1E)43H	FB P_PRINT_RK: No pointer available for format string.	Enter the data block no. and data word no. for the format string in the pointer DB.

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 of the corresponding function block. The program example in section "Programming Example for Standard Function Blocks (Page 199)" demonstrates how to load the SFCERR variable.

The error messages written to the SFCERR variable are listed in the System Software for S7 300/400, System and Standard Functions Reference Manual under the SFC 58 "WR_REC" and SFC 59 "RD_REC" system functions.

8.1.3 Error Numbers in the Response Message Frame

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

Error numbers in the response message frame

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

Table 8- 3 Error messages in the response message frame with RK 512

REATEL	Error Messages
	Event Class/Event Number
0AH	0905H
0CH	0301H 0607H 0609H 060AH 0902H
10H	0301H 0601H 0604H 0605H 090BH
12H	0904H
14H	0606H 0903H
16H	0602H 0603H 090AH
2AH	060DH 090DH
32H	060FH 0909H
34H	0608H 060BH 060CH 0611H 090CH
36H	060EH 0908H

8.1.4 Diagnostics via the diagnostic buffer of the CP 341

Diagnostic buffer of the CP 341

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

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

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

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

Note

The diagnostic buffer is a ring buffer designed 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. As a result, the latest entry is always at the top. In the event of a POWER OFF or parameter reassignment of the CP 341, the contents of the diagnostic buffer are lost!

Note

To view the time of day of the individual diagnostic entries, you must select the CPU in "HW Config" and perform the time-of-day synchronization in the "Diagnostics/Clock" folder ("master" synchronization type, time interval of, for example, 10 seconds). When the CP 341 is used in distributed operation (ET 200M), the time of day cannot be displayed.

Outputting the diagnostic buffer on the programming device

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

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

Requirements: In order to obtain module information, there must be an online connection from the programming device to the programmable controller (online view in the project window).

Use the following procedure:

1. Open the relevant SIMATIC 300 station (double-click or select menu command Edit > Open).
2. Then open the "Hardware" object (double-click or select menu command Edit > Open).

   Result: The window containing the configuration table appears.
3. Select the CP 341 in the configuration table.
4. Select PLC > Module Information.

   Result: The "Module Information" dialog box for the CP 341 appears. The first time the dialog is called, the "General" tab appears (default setting).
5. Go to the "Diagnostic Buffer" tab.

   Result: The latest diagnostic events of the CP 341 are displayed in plain text on the "Diagnostic Buffer" tab. Additional information about error causes may appear in the "Event details" output field.

   The "Event ID" field displays the number code for the the event. The first part is fixed. The second part corresponds to the event class and event numbers of the event.

   Select the "Help on Event" button to display the remedy for the particular event text.

Click "Update" to read the latest data from the CP 341. Click "Help on Event" to show a help text for the selected diagnostic event containing information about troubleshooting.

8.1.5 Diagnostic Interrupt

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

"Diagnostics alarm = NO" is the default.

Diagnostic interrupt

In the event of a fault the CP 341 provides diagnostic information on the S7–300 backplane bus. In response to a diagnostic interrupt, the CPU reads the system-specific diagnostic data and enters it in its diagnostic buffer. You can read the contents of the diagnostic buffer on the CPU using a programming device.

When a diagnostics event occurs, the SF LED (red) lights up. In addition, the OB 82 is called with this diagnostic 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 diagnostic interrupt.

Diagnostic information (as bit pattern)

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

2nd byte:

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

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

1st, 3rd and 4th bytes:

The 1st, 3rd, and 4th bytes of the diagnostic data represent the error that 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 diagnostic data.

Event	1st byte								3rd byte								4th byte							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Wire break	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Incorrect parameter	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Diagnostic information (hexadecimal)

The table below shows the 4 bytes of diagnostic data of the CP 341 in hexadecimal notation.

Event	1st byte	2nd byte	3rd byte	4th Byte
Wire break	25H	0CH	02H	00H
Incorrect parameter	83H	0CH	00H	00H

Relationship of the diagnostic interrupt and the CPU operating mode

A diagnostic interrupt is generated via the I/O bus on an incoming event (rising edge) and outgoing event (falling edge).

When the CPU switches from STOP to RUN mode, the following applies:

- Events (either incoming or outgoing) that occurred when the CPU was in STOP mode are not stored
- Events that are still present when the CPU switches back to RUN mode are signaled via a diagnostic interrupt.

Programming Example for Standard Function Blocks 9

9.1 General Information

The programming example given here and included in the zXX21_01_PtP_Com_CP34x project describes standard functions for operating the CP 341 communications processor.

Objective

The programming example

- aims to show examples of the most important functions
- enables the correct functioning of the connected hardware to be checked (and is therefore simple and easy to follow)
- can easily be extended for your own purposes.

The example shows how an 3964(R)/ASCII connection or an RK 512 computer link to a communications partner can be configured using the standard function blocks P_SND_RK and P_RCV_RK (to send and receive data respectively).

The example also shows how the inputs and outputs of the CP 341 can be controlled and monitored using the V24_STAT and V24_SET standard functions.

There are three SIMATIC stations in the example because the CP 341 has to be assigned parameters differently for data transfer:

- CP341 protocol 3964: Coupling with FB P_SND_RK and FB P_RCV_RK
- CP341 protocol RK512: Coupling with FB P_SND_RK and FB P_RCV_RK
- CP341 V24: Read and control RS 232C secondary signals with FC V24_STAT and FC V24_SET

Note that the "CP340 PTP Connection" (point-to-point) and "CP340 Printing and V24" stations contain the examples for the CP 340.

The CP 341 is assigned parameters by the CPU when the latter is started up (system service).

Requirement

The example can be executed with the minimum hardware equipment. The STEP 7 function monitor/modify variables is also used (e.g. to modify transmitted data).

Program example

The programming example of the CP 341, together with the programming interface and the function blocks, is supplied on the installation CD which come with this manual.

It is available both compiled and as an ASCII source file. A list of all the symbols used in the example is also included.

9.2 Device Configuration

Application

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

- An S7-300 automation system (rack, power supply, CPU)
- A CP 341 module with a communications partner (e.g., a second CP), or insertion of a "short-circuit connector", i.e. the send line is bridged to the receive line

  The use of the "short-circuit connector" is only supported by the ASCII protocols.
- A programming device.

9.3 Settings

Settings in the CPU by means of STEP 7

Use STEP 7 to configure your controller as follows.

- Slot 1: Power supply
- Slot 2: CPU
- Slot 4: CP 341, start address 256
- Slot 5: CP 341, start address 272

Settings on the CP 341

You cannot make any hardware settings on the CP 341.

Use STEP 7 to configure all relevant data, including the parameters for the CP 341 with the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment and upload them to the CPU.

You can run the "CP341 protocol 3964" program example without making changes in the application program with:

- 3964(R) Procedure
- ASCII driver with "on expiry of character delay time" end criterion
- ASCII driver with "on receipt of fixed message frame length" end criterion.

For the ASCII driver with the "on receipt of the end character(s)" end criterion, you must also program the end codes.

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

9.4 Blocks Used

Blocks used

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

Block	Symbol	Description
OB 1	CYCLE	Cyclic program processing
OB 100	RESTART	Cold restart processing
DB 21	SEND IDB	Instance DB for P_SND_RK FB
DB 22	RECV IDB	Instance DB for P_RCV_RK FB
DB 40	SEND WORK DB	Work DB for the standard FB 8
DB 41	RECV WORK DB	Work DB for the standard FB 7
DB 42	SEND SRC DB	Send data block
DB 43	RECV DST DB	Receive data block
FB 7	P_RCV_RK	Standard FB for receiving data (RK 512)
FB 8	P_SND_RK	Standard FB for sending data (RK 512)
FC 5	V24_STAT	Standard FC for reading CP outputs
FC 6	V24_SET	Standard FC for writing to CP outputs
FC 14	V24_CYC	Control RS 232C secondary signals
FC 21	SEND	Sending Data
FC 22	RECEIVE	receiving data

9.5 Installation, Error Messages

Scope of supply and installation

The programming example of the CP 341, together with the **CP 341: Point-to-Point Communication, Parameter Assignment** parameter assignment interface and the function blocks is located on the CD supplied with the module.

The program example is installed together with the parameter assignment interface. Following installation, you will find the program example in the project: "zXX21_01_PtP_Com_CP34x"

Open the project in the STEP 7 SIMATIC Manager by selecting **File > Open > Project**.

The sample program is available as a compiled program and as an ASCII source file. A list of all the symbols used in the example is also included.

If there is no second CP 341 available to serve as a communication partner, you have to delete the CP 341 in HW Config by selecting **Edit > Delete**. In addition, in OB 1 the FC 22 call (FC for receive) must be commented out.

Loading to the CPU

The hardware for the example is completely set up and the programming device is connected.

After resetting the CPU memory (STOP operating mode), transfer the entire example to the user memory. Then switch the mode selector from STOP to RUN.

Error behavior

If an error occurs during startup, the cyclically processed block call commands will not be executed and the error LED will be set.

In the event of an error message, the ERROR parameter output of the blocks is set. A more detailed description of the error is then stored in the STATUS parameter of the blocks. If the STATUS parameter contains either the 16#1E0E or the 16#1E0F error message, the more detailed description will be stored in the SFCERR variable in the instance DB.

9.6 Activation, Start-Up Program and Cyclic Program

Activation, start-up program

The start-up program is located in OB 100.

The control bits and the counters are reset in the start-up procedure

Cyclic program

The cyclic program is in OB 1.

In the example, the function blocks 7 P_RCV_RK FB and 8 P_SND_RK FB work with the functions FC 21 and FC 22 and with data blocks DB 21 and DB 22 as instance DBs and DB 42 and DB 43 as send and receive DBs.

The functions FC 5 V24_STAT and FC 6 V24_SET work with the function FC 14.

In the example the function blocks are assigned parameters partly via constants and partly via symbolically addressed actual operands.

Description, "CP341 Protocol 3964", "CP341 Protocol RK512"

Data is transferred from the CP 341 in slot 4 to the CP 341 in slot 5. If you are using some other communication partner, the FC 22 call (RECEIVE) is omitted.

Description of FC 21 (SEND)

Program section "Generate edge P_SND_REQ":

The P_SND_RK is initially executed once with P_SND_RK REQ=0. P_SND_RK REQ is then set to 1. If a signal state change from 0 to 1 is detected at the P_SND_RK REQ control parameter, the P_SND_RK request is started.

If P_SND_RK DONE=1 or P_SND_RK ERROR=1, P_SND_RK REQ is reset to 0.

Program section "P_SND_RK DONE=1":

If a transfer has been successful, the P_SND_RK DONE parameter is set to 1 at the parameter output of P_SND_RK. To distinguish between consecutive transfers, a send counter (P_SND_RK COUNTER_OK) is included in data word 0 of source data block DB 42.

Program section "P_SND_RK ERROR=1":

If P_SND_RK is executed with P_SND_RK ERROR=1, the error counter P_SND_RK COUNTER_ERR is incremented in data word 2. In addition, P_SND_RK STATUS is copied, since it will be overwritten with 0 during the next cycle, making it impossible to read it out.

Description of FC 22 (RECEIVE)

Program section "Enable receive data":

In order to receive data, the P_RCV_RK EN_R receive enable at the P_RCV_RK block must be set to 1.

Program section "P_RCV_RK NDR=1":

If P_RCV_RK NDR is set, it means that new data has been received and the receive counter P_RCV_RK COUNTER_OK is incremented.

Program section "P_RCV_RK ERROR=1":

If an error occurs, i.e., the error bit at the parameter output of P_RCV_RK is set, the P_RCV_RK COUNTER_ERR error counter is incremented. In addition, P_RCV_RK STATUS is copied, since it will be overwritten with 0 during the next cycle, making it impossible to read it out.

All relevant values can be monitored in the VAT for testing purposes.

Description "CP341 V24"

- The "read and control RS 232C accompanying signals" functions can only be carried out with the ASCII driver. Control is only possible if you have not set the “Automat. control of V24 signals" parameter on the "Transfer" tab.
- The V24 signals can be read and written using the variable table. The signal states SET_DTR and SET_RTS can be preselected by means of memory bits M 1.6 and M 1.7. When the signal at memory bit I 0.7 changes from "0" to "1", this state is transferred to the CP by the FC V24_SET function.
- The FC V24_STAT function is called cyclically. The status of the CP 341 V24 signals can be read out via memory bits 3.0 to 3.5.

Technical Specifications A

A.1 Technical Data of the CP 341

General technical specifications

The following table contains the general technical specifications of the CP 341.

You will find more general technical specifications for the SIMATIC S7-300 in the reference manual entitled *S7-300 Programmable Controllers, Module Specifications*.

Table A- 1 General technical specifications

Technical specifications	
Dimensions W x H x D	40 x 125 x 120 mm
Weight	0.3 kg
Current consumption from 24 V (24 V DC infeed via front panel connector)	• CP 341-RS 232C: Typ. 100 mA • CP 341-20mA-TTY typ. 100 mA • CP 341-RS 422/485: Typ. 100 mA
Area, static Area, dynamic Polarity reversal protection Electrical isolation	20.4 to 28.8 V 18.5 to 30.2 V Yes Yes, relative to all other voltages
Current consumption from backplane bus	Approx. 70 mA
Power loss	• CP 341-RS 232C: Typ. 2.4 W • CP 341-20mA-TTY: Typ. 2.4 W • CP 341-RS 422/485: Typ. 2.4 W
Indicators	LEDs for transmitting (TXD), receiving (RXD) and group faults (SF)
Interrupts Diagnostic interrupt	Programmable
Diagnostic functions • Diagnostic information can be read out	 Yes
Supplied protocol drivers	• ASCII driver • 3964(R) procedure • RK 512 computer link • Printer driver
Transmission rate with 3964(R) protocol	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps (half-duplex)

Technical specifications	
Transmission rate with RK 512 computer link	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps (half-duplex)
Transmission rate with ASCII driver	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps
Transmission rate with printer driver	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps
Character frame	• No. of bits per character (7 or 8) • No. of start/stop bits (1 or 2) • Parity (none, even, odd)
Quantity of user data transported per program cycle	Send/receive: 32 bytes
Memory requirement of FB P_SND_RK and FB P_RCV_RK (RAM)	Approx. 5500 bytes

Technical specifications RS232C interface

The table below shows the technical specifications for the RS 232C interface of the CP 341–RS 232C:

Table A- 2 Technical specifications of the RS 232C interface

Technical specifications	
Interface	RS232C, 9-pin sub D male connector
RS232C signals	TXD, RXD, RTS, CTS, DTR, DSR, RI, DCD, GND All electrically isolated relative to the S7-internal supply (backplane bus) and the external 24 V DC supply
Max. transmission distance	15 m
Max. baud rate	115200 bps

Technical specifications of the 20mA-TTY interface

The table below shows the technical specifications for the 20-mA-TTY interface of the CP 341-20mA-TTY:

Table A- 3 Technical specifications of the 20mA-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 electrically isolated relative to the S7-internal supply (backplane bus) and the external 24 V DC supply
Max. transmission distance	1000 m active, 1000 m passive
Max. baud rate	19200 bps

Technical specifications of the X27 (RS 422/485) interface

The table below shows the technical specifications for the X27 (RS 422/485) interface of the CP 341–RS 422/485.

Table A- 4 Technical specifications of the X27 (RS 422/485) interface

Technical specifications	
Interface	RS 422 or RS 485, 15-pin sub D female connector
RS 422 signals RS 485 signals	TXD (A), RXD (A), TXD (B), RXD (B), GND R/T (A), R/T (B), GND All electrically isolated relative to the S7-internal supply (backplane bus) and the external 24 V DC supply
Max. transmission distance	1200 m
Max. baud rate	115200 bps

Technical specifications of the 3964(R) procedure

The table below shows the technical specifications for the 3964(R) procedure.

Table A- 5 Technical specifications of the 3964(R) procedure

3964(R) procedure with default values	
Max. message frame length	4096 bytes
Parameter	The following can be assigned: • With/without block check character • Priority: low/high • Transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps • Character frame: 9, 10, 11 or 12 bits • Receive line initial state: none, R(A)5V/R(B)0V, R(A)0V/R(B)5V
3964(R) procedure with parameter assignment	
Max. message frame length	4096 bytes
Parameter	The following can be assigned: • With/without block check character • Priority: low/high • Transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps • Character frame: 9, 10, 11 or 12 bits • Character delay time: 20 ms to 65530 ms in 10 ms increments • Acknowledgment delay time: 20 ms to 65530 ms in 10 ms increments • Number of connection attempts: 1 to 255 • Number of transmission attempts: 1 to 255 • Receive line initial state: none, R(A)5V/R(B)0V, R(A)0V/R(B)5V

Technical specifications RK512 computer link

The table below shows the technical specifications for the RK512 computer link.

Table A- 6 Technical specifications of the RK512 computer link

RK 512 computer link	
Max. message frame length	4096 bytes
Parameter	The following can be assigned: • Transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps • Character frame: 10, 11, or 12 bits • Character delay time: 20 ms to 65530 ms in 10 ms increments • Acknowledgment delay time: 20 ms to 65530 ms in 10 ms increments • Number of connection attempts: 1 to 255 • Number of transmission attempts: 1 to 255 • Initial state of receive line: none, R(A)5V/R(B)0V, R(A)0V/R(B)5V

Technical specifications of the ASCII driver

The table below shows the technical specifications for the ASCII driver.

Table A- 7 Technical specifications of the ASCII driver

ASCII driver	
Max. message frame length	4096 bytes
Parameter	The following can be assigned: • Transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps • Character frame: 9, 10, 11 or 12 bits • Character delay time: 2 ms to 65535 ms in 1 ms increments • Flow control: none, XON/XOFF, RTS/CTS; automat. control of V.24 signals • XON/XOFF character (only when "flow control" = "XON/XOFF") • Wait for XON after XOFF (wait time for CTS = ON): 20 ms to 65530 ms in 10-ms increments • Time to RTS OFF: 0 ms to 65530 ms in 10 ms increments (only for "automat. control of V.24 signals") • Data output waiting time: 0 ms to 65530 ms in 10 ms increments (only for "automat. control of V.24 signals") • Number of message frames to be buffered: 1 to 250 • Prevent overwriting: yes/no (only when "buffered received frames" = "1") • Indicator for end of received message frame: – After character delay time expires – On receipt of end-of-text character(s) – On receipt of fixed number of characters

Table A- 8 Additional parameters of end-of-frame detection

ASCII driver with end-of-frame detection after expiration of character delay time	
Parameter	No other parameter assignments need to be made. The end of a frame is detected when the programmed character delay time elapses.
ASCII driver with end-of-frame detection using assignable end-of-text characters	
Parameter	The following can be assigned: Number of end-of-text characters: 1 or 2 Hex code for first/second end-of-text character
ASCII driver with end-of-frame detection using configured frame length	
Parameter	The following can be assigned: Message frame length: 1 to 4096 bytes

Technical specifications of the printer driver

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

Table A- 9 Technical specifications of the printer driver

Printer driver	
Length of the text SDB	15 kbytes
Parameter	The following can be assigned: • Transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bps • Character frame: 10 or 11 bits • Flow control: None, XON/XOFF, RTS/CTS • XON/XOFF characters (with "flow control" = "XON/XOFF" only) • Wait for XON after XOFF (wait time for CTS = ON): 20 ms to 65530 ms in 10 ms increments • Message text: max. 150 characters (max. 250 characters when variables are displayed) • 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, CR LF, 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

See also

Technical Specifications of the Function Blocks (Page 168)

A.2 Transmission Rates

Transmission Rates

The tables below indicate the transmission times required depending on the transmission protocol selected.

Two S7-300s, each with a CPU 319-3 PN/DP (6ES7 318-3EL00-0AB0) and a CP 341-V2 (6ES7 341-1AH02-0AE0), were used to measure the time. An P_SND_RK (V3.2) function block was programmed in the user program of the active CPU, and an P_RCV_RK (V3.2) function block was programmed in the user program of the passive CPU. The time that elapsed between the initiation and completion of the request was measured.

ASCII driver

Table A- 10 Transmission rates (in s) for ASCII driver (at minimum character delay time)

User data	Baud rate (bd)										
	115200	76800	57600	38400	19200	9600	4800	2400	1200	600	300
1 byte	0.003	0.003	0.003	0.003	0.003	0.006	0.011	0.021	0.040	0.081	0.160
10 bytes	0.004	0.004	0.005	0.005	0.008	0.015	0.029	0.058	0.115	0.231	0.455
20 bytes	0.005	0.005	0.006	0.008	0.014	0.027	0.052	0.104	0.207	0.416	0.822
50 bytes	0.007	0.010	0.012	0.017	0.031	0.061	0.121	0.242	0.483	0.966	1.935
100 bytes	0.012	0.017	0.022	0.031	0.059	0.118	0.227	0.472	0.942	1.884	3.727
200 bytes	0.022	0.031	0.041	0.060	0.117	0.234	0.415	0.931	1.861	3.722	7.353
500 bytes	0.051	0.076	0.099	0.148	0.291	0.577	1.076	2.309	4.616	9.233	18.458
1000 bytes	0.101	0.151	0.198	0.294	0.581	1.154	2.146	4.134	9.212	18.424	36.817
4000 bytes	0.331	0.596	0.784	1.170	2.317	4.613	9.205	18.393	36.776	73.553	147.080

3964(R) procedure

Table A- 11 Transmission times (in s) with the 3964(R) procedure

User data	Baud rate (bd)										
	115200	76800	57600	38400	19200	9600	4800	2400	1200	600	300
1 byte	0.003	0.003	0.003	0.004	0.006	0.010	0.018	0.034	0.066	0.130	0.258
10 bytes	0.004	0.004	0.005	0.006	0.011	0.020	0.039	0.075	0.149	0.296	0.590
20 bytes	0.005	0.006	0.007	0.009	0.017	0.032	0.062	0.121	0.241	0.480	0.957
50 bytes	0.007	0.010	0.013	0.018	0.034	0.066	0.131	0.259	0.517	1.031	2.057
100 bytes	0.012	0.018	0.023	0.033	0.063	0.124	0.246	0.489	0.976	1.948	3.896
200 bytes	0.022	0.033	0.042	0.062	0.121	0.240	0.475	0.949	1.895	3.786	7.570
500 bytes	0.053	0.078	0.102	0.150	0.295	0.586	1.167	2.329	4.653	9.301	18.599
1000 bytes	0.103	0.152	0.200	0.297	0.585	1.163	2.317	4.629	9.249	18.491	36.976
4000 bytes	0.332	0.597	0.786	1.172	2.321	4.621	9.216	18.424	36.824	73.635	147.276

RK 512 computer link

Table A- 12 Transmission times (in s) with the RK 512 computer link

User data	Baud rate (bd)										
	115200	76800	57600	38400	19200	9600	4800	2400	1200	600	300
1 byte	0.007	0.009	0.010	0.012	0.020	0.035	0.066	0.130	0.258	0.514	1.029
10 bytes	0.008	0.010	0.012	0.015	0.025	0.044	0.084	0.167	0.332	0.663	1.323
20 bytes	0.009	0.012	0.014	0.018	0.030	0.056	0.107	0.213	0.424	0.844	1.694
50 bytes	0.013	0.016	0.020	0.027	0.048	0.091	0.176	0.350	0.699	1.396	2.790
100 bytes	0.018	0.024	0.030	0.042	0.077	0.149	0.291	0.580	1.159	2.315	4.627
200 bytes	0.032	0.043	0.055	0.079	0.149	0.289	0.569	1.133	2.262	4.520	9.034
500 bytes	0.071	0.100	0.128	0.184	0.351	0.687	1.357	2.699	5.389	10.776	21.537
1000 bytes	0.139	0.196	0.250	0.363	0.697	1.364	2.700	5.375	10.723	21.438	42.853
4000 bytes	0.540	0.769	0.983	1.431	2.765	5.427	10.758	21.430	42.770	85.443	170.795

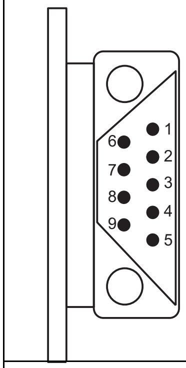
Connecting cables B

B.1 RS 232C Interface of the CP 341-RS 232C

Pin assignment

The table below shows the pin assignment for the 9-pin sub D male connector in the front panel of the CP 341-RS 232C (compatible with the 9-pin COM port (PC/programming device).

Table B- 1 Pin assignment for the sub D male connector of the integrated interface of the CP 341-RS 232C

Male Connector on CP 341-RS 232C*	Pin	Designation	Input/output	Meaning
	1	DCD1 Received Detector	Input	Received signal level
	2	RXD Received Data	Input	Receive data
	3	TXD Transmitted Data	Output	Transmit data
	4	DTR Data Terminal Ready	Output	Data terminal ready
	5	GND Ground	-	Functional ground (GND$_{int}$)
	6	DSR Data Set Ready	Input	Data set ready
	7	RTS Request To Send	Output	Request to send
	8	CTS Clear To Send	Input	Clear to send
	9	RI Ring Indicator	Input	Incoming call
* 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 interfaces. GND must always be connected on both sides (pin 5), otherwise the interface modules could be destroyed.

On the following pages you will find examples of connecting cables for a point-to-point connection between the CP 341-RS 232C and S7 modules or SIMATIC S5.

RS 232C connecting cables (S7 (CP 341) - S7 (CP 340/ CP 341/CP 441))

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441.

For the connecting cables you will require the following female connectors:

- On the CP 341 side: 9-pin sub D female connector with screw-locking
- On the communication partner: 9-pin sub D female connector with screw-locking

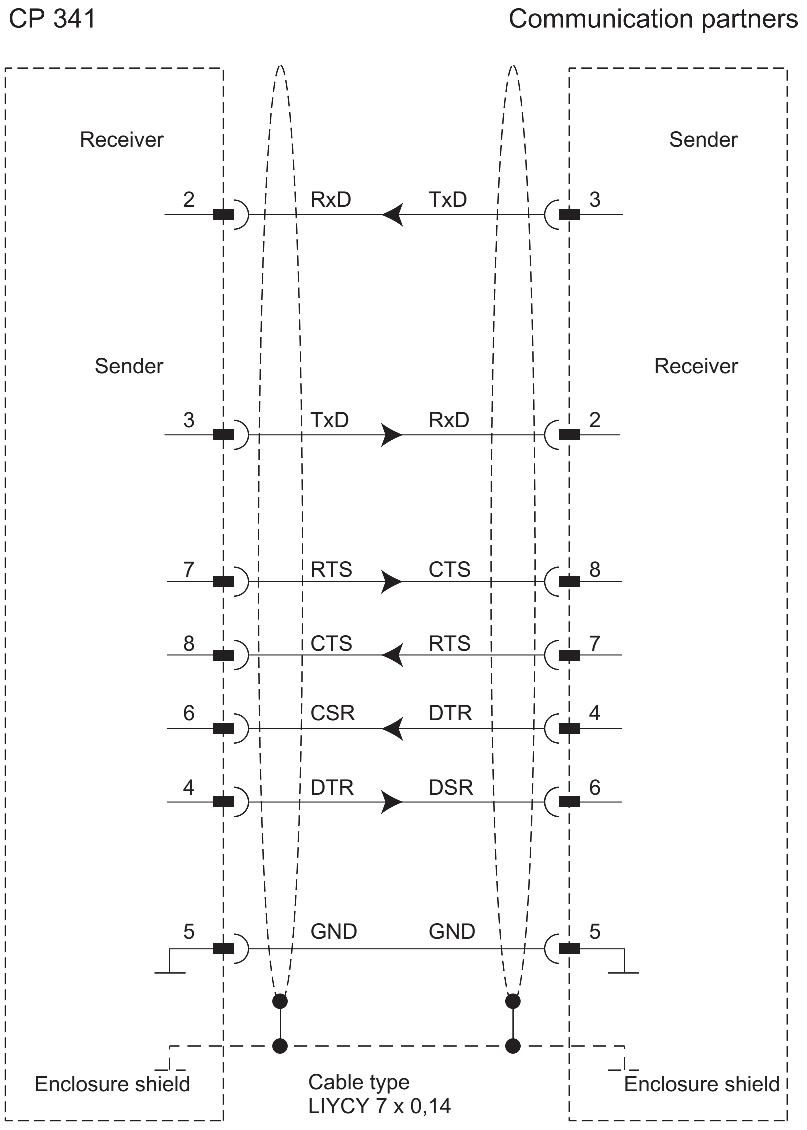


Figure B-1 RS 232C connecting cable CP 341 - CP 340/CP 341/CP 441

The cable (max. 15 m) is available under the order number (6ES7 902-1...) specified in appendix Accessories and order numbers (Page 235).

RS 232C connecting cables (S7 (CP 341) - CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 544, CP 524, CPU 928B, CPU 945 or CPU 948.

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

- On the CP 341 side: 9-pin sub D female connector with screw-locking
- On the communication partner: 25-pin sub D male connector with screw-locking

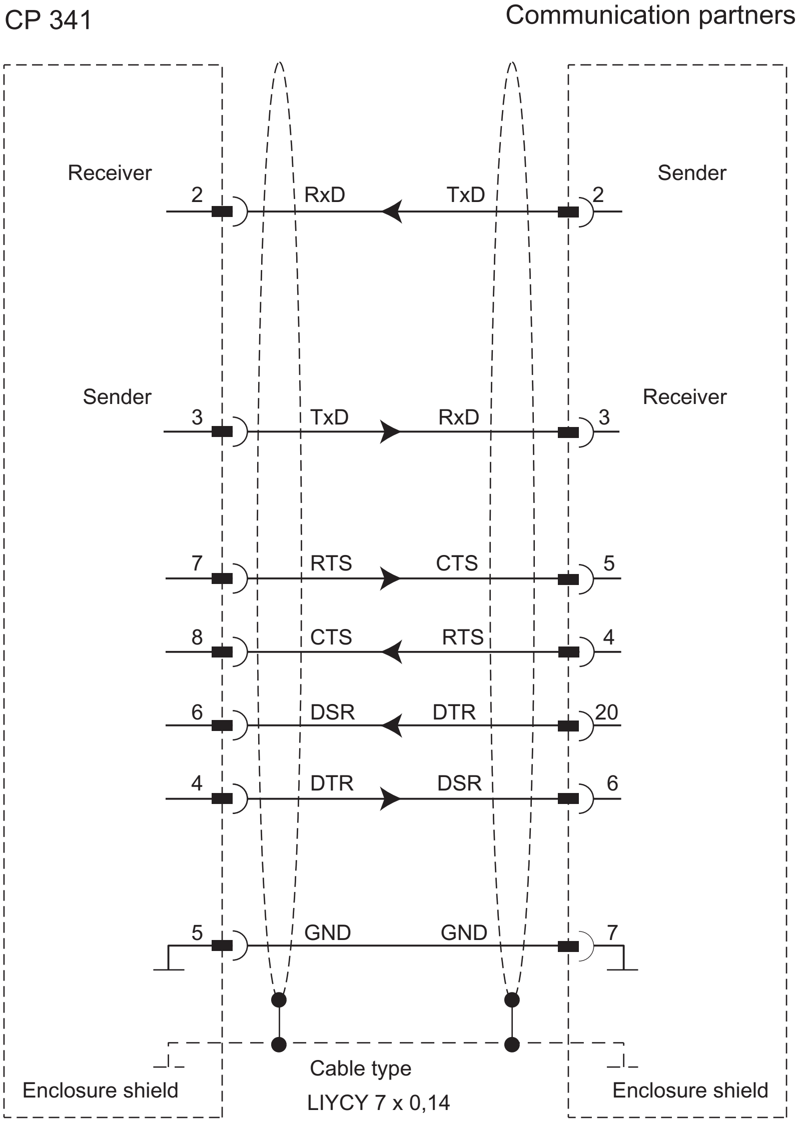


Figure B-2 RS 232C connecting cable (CP 341 - CP 544, CP 524, CPU 928B, CPU 945, CPU 948

RS 232C connecting cables (S7 (CP 341) - CP 521 SI/CP 521 BASIC)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 521 SI/CP 521 BASIC.

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

- On the CP 341 side: 9-pin sub D female connector with screw-locking
- On the communication partner: 25-pin sub D male connector with screw-locking

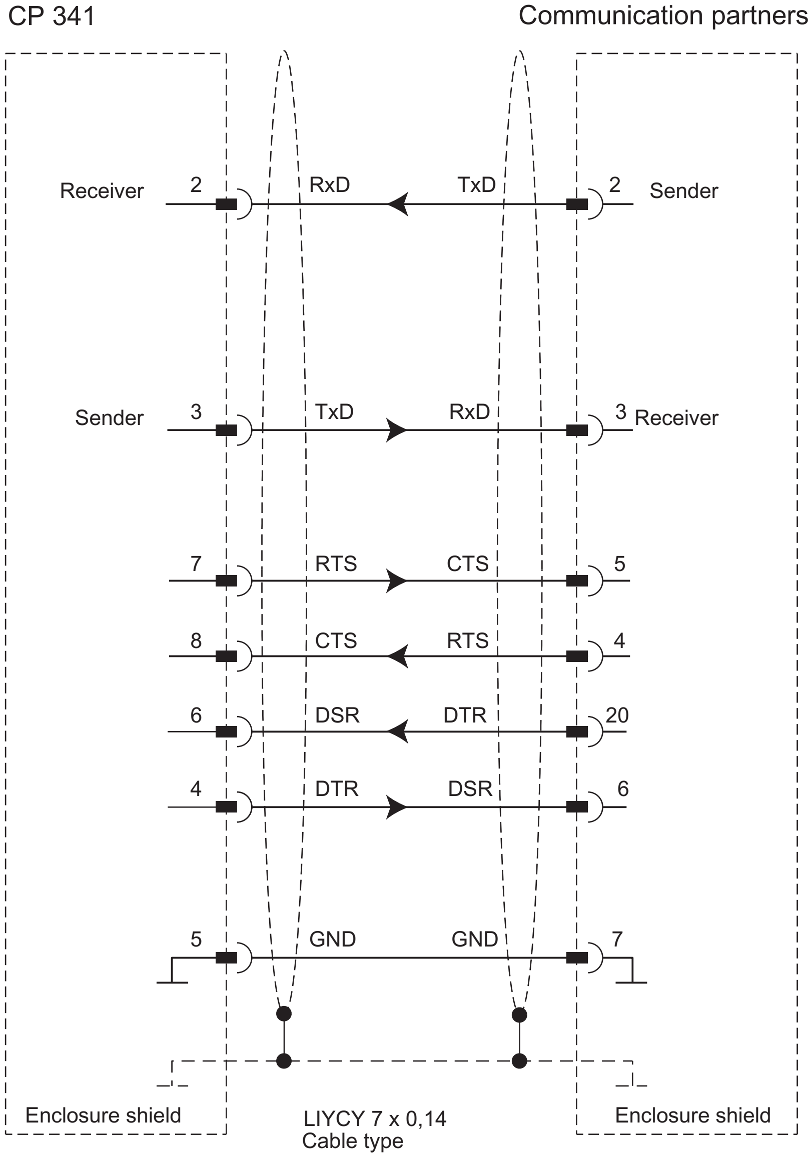


Figure B-3 RS 232C connecting cable CP 341 - CP 521 SI/CP 521 BASIC

RS 232C connecting cables (S7 (CP 341) - CP 523)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 523.

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

- On the CP 341 side: 9-pin sub D female connector with screw-locking
- On the communication partner: 25-pin sub D male connector with screw-locking

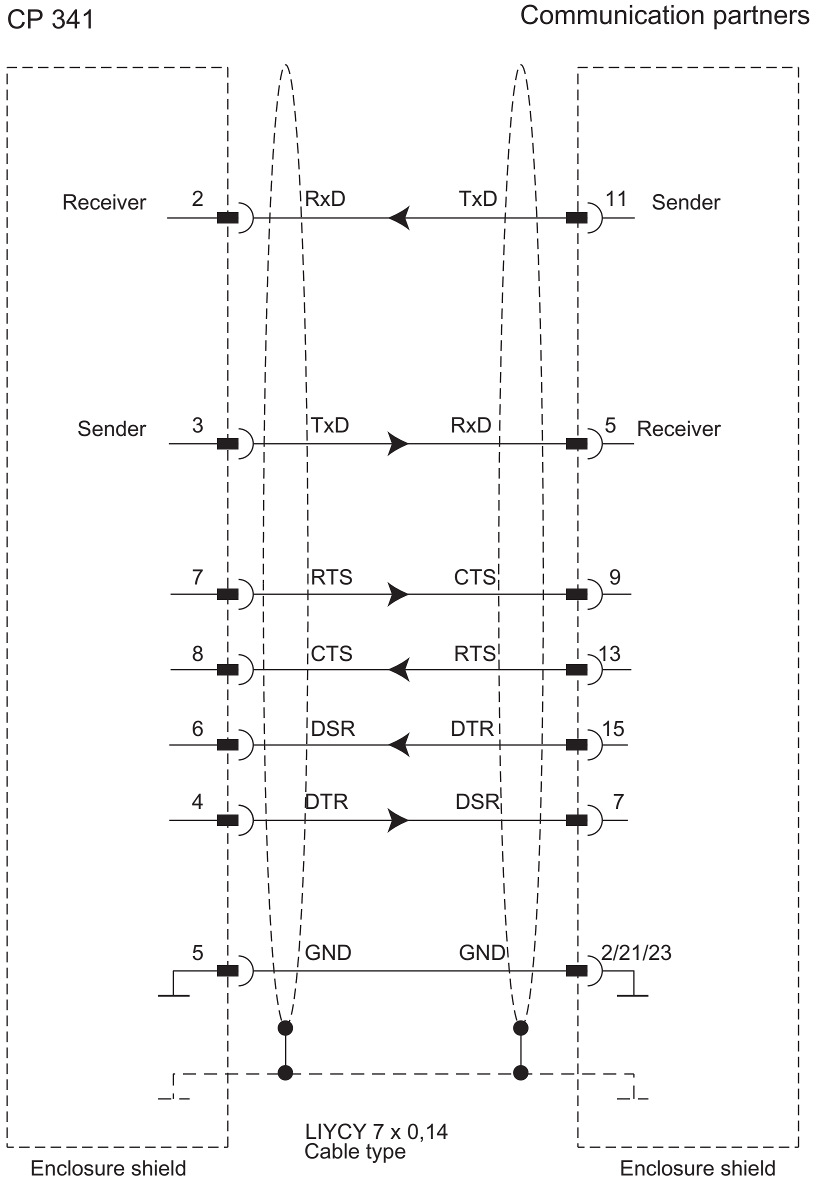


Figure B-4 RS 232C connecting cable CP 341 - CP 523

RS 232C connecting cable (S7 (CP 341) - IBM-Proprinter (PT 88), DR 230)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and an IBM Proprinter with serial interface (PT 88 or IBM-compatible printer).

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

- On the CP 341 side: 9-pin sub D female
- For the IBM Proprinter: 25-pin sub D male

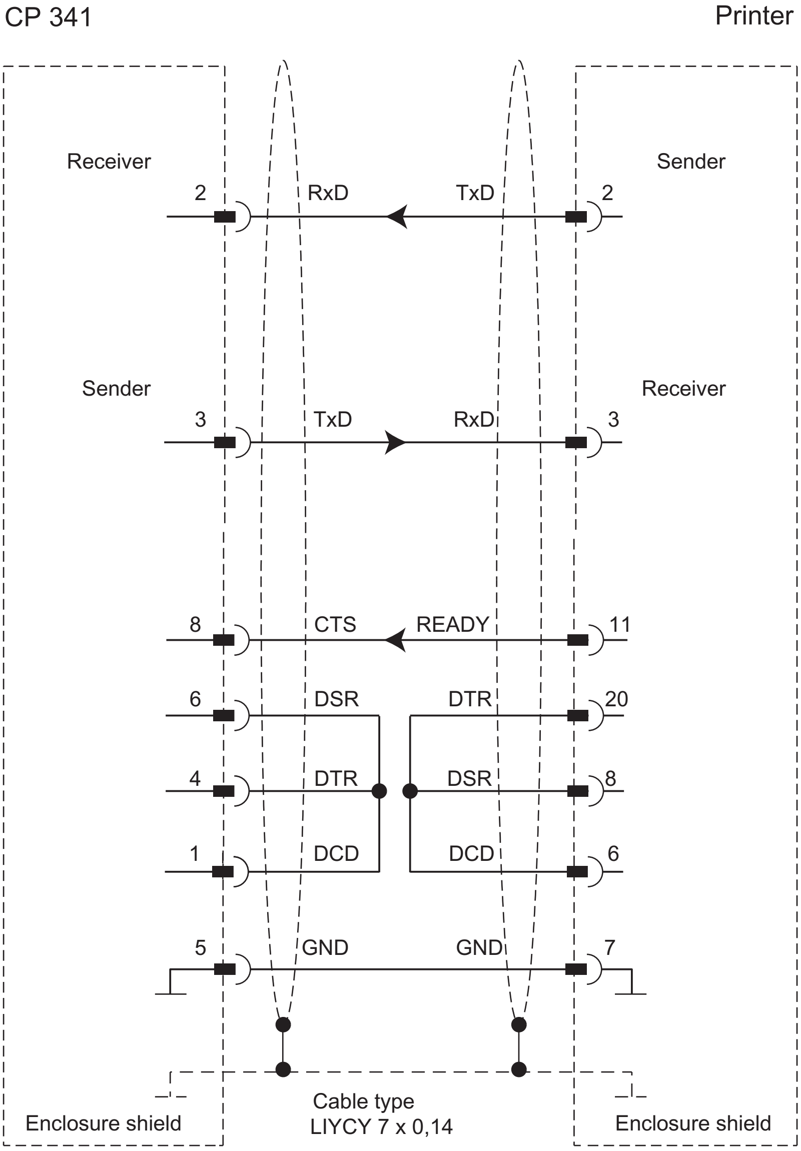


Figure B-5 RS 232C connecting cable CP 341 - IBM Proprinter

RS 232C connecting cable (S7 (CP 341) - laser printer)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a laser printer with serial interface (PT 10 or Laserjet series II).

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

- On the CP 341 side: 9-pin sub D female
- For the IBM Proprinter: 25-pin sub D male

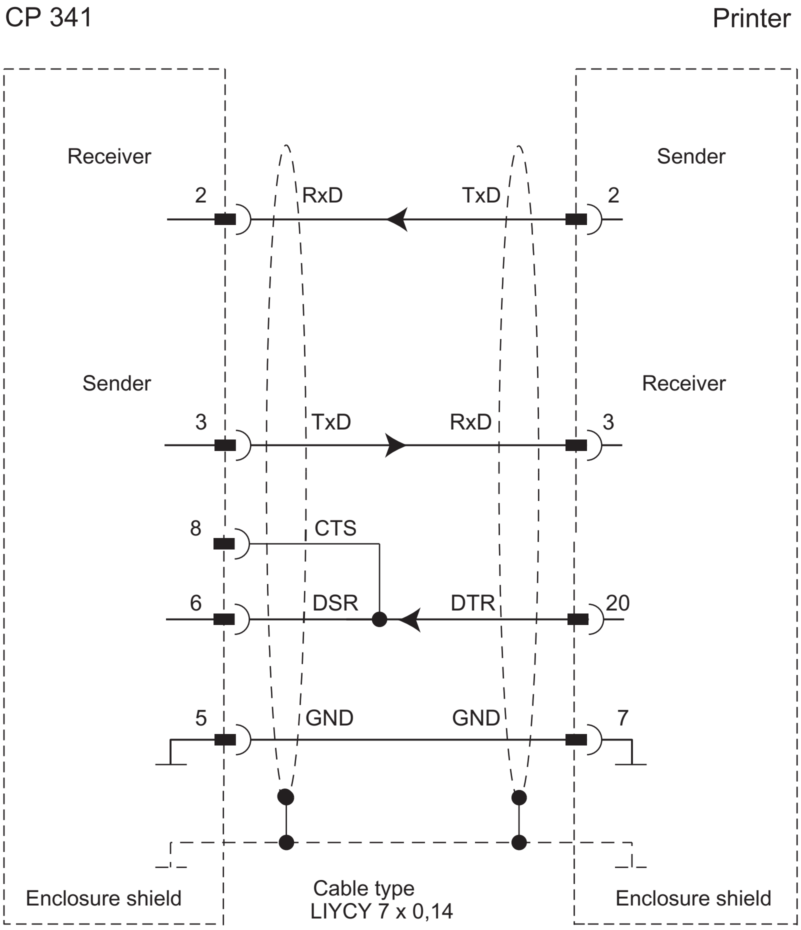


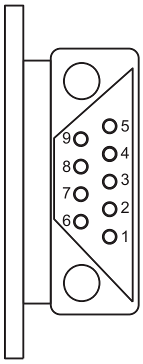
Figure B-6 RS 232C connecting cable CP 341 - laser printer

B.2 20 mA TTY interface on the CP 341-20mA-TTY

Pin assignment

The table below shows the pin assignment for the 9-pin sub D female connector in the front panel of the CP 341-20mA-TTY.

Table B- 2 Pin assignment for the 9-pin sub D female connector of the integrated interface of the CP 341-20mA-TTY

Female connector on CP 341-20mA-TTY*	Pin	Designation	Input/output	Meaning
	1	TxD-	Output	Transmit data
	2	20 mA -	Input	24 V ground
	3	20 mA + (I_1)	Output	20mA current generator 1
	4	20 mA + (I_2)	Output	20mA current generator 2
	5	RxD +	Input	Receive data +
	6	-		
	7	-		
	8	RxD -	Output	Receive data -
	9	TxD +	Input	Transmit data +
* View from the front				

Block diagram

The figure below shows the block diagram for a 20mA-TTY interface.

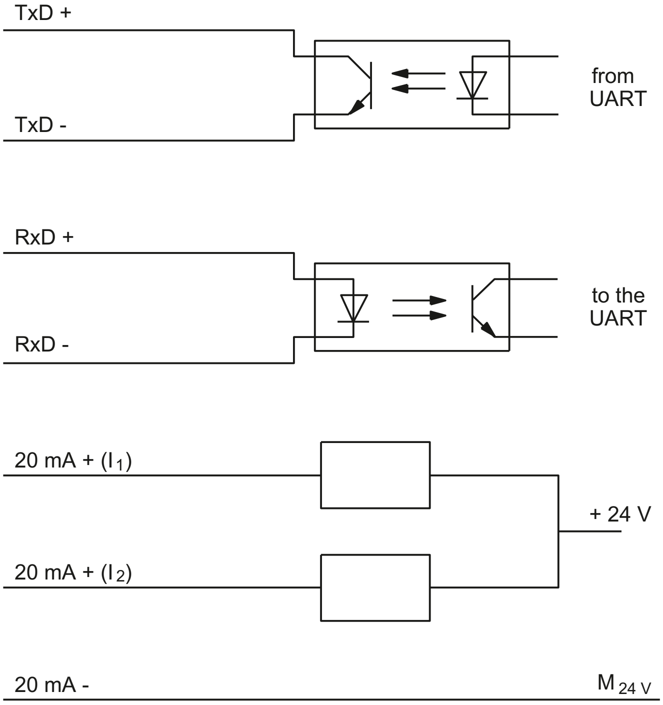


Figure B-7 Block diagram for the 20mA-TTY interface

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 interface modules.

In the following

On the following pages you will find examples of connecting cables for a point-to-point connection between the CP 341-20mA-TTY and S7 modules or SIMATIC S5.

20mA-TTY connecting cable (S7 (CP 341) - S7 (CP 340/ CP 341/CP 441))

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441.

For the connecting cables you will require the following male connectors:

- On the CP 341 side: 9-pin sub D male connector with screw-locking
- On the communication partner: 9-pin sub D male connector with screw-locking

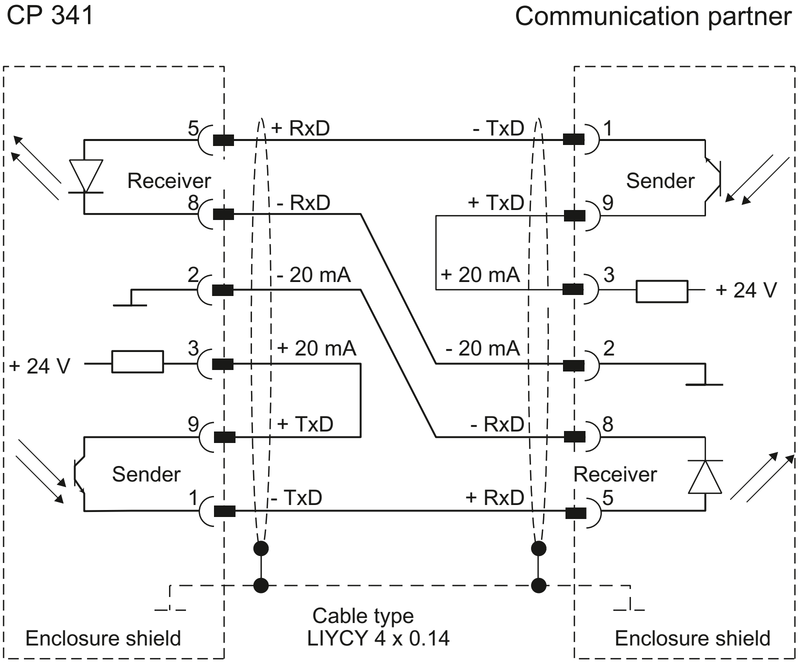


Figure B-8 20mA-TTY connecting cable CP 341 - CP 340/CP 341/CP 441

The cable is available under the order number (6ES7 902-2...) specified in appendix Accessories and order numbers (Page 235).

Note

This cable type (LIYCY 4 x 0.14) can be used in the following lengths for the CP 341 as communication partner: max. 1000 m at 9600 baud, max. 500 m at 19.2 kbaud.

20mA-TTY connecting cable (S7 (CP 341) - CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 544, CP 524, CPU 928B, CPU 945 or CPU 948.

For the connecting cables you will require the following male connectors:

- On the CP 341 side: 9-pin sub D male connector with screw-locking
- On the communication partner: 25-pin sub D male connector with screw-locking

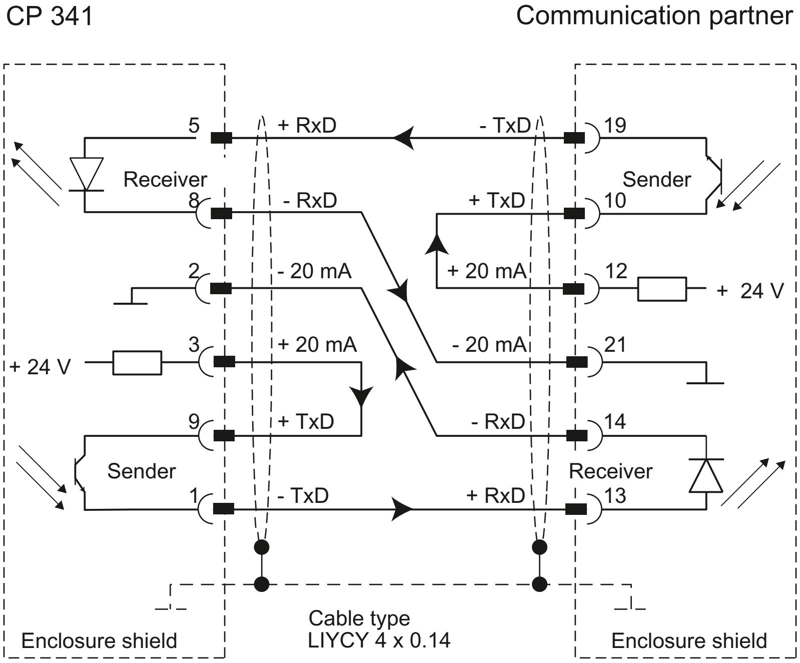


Figure B-9 20mA-TTY connecting cable CP 341 - CP 544, CP 524, CPU 928B, CPU 945, CPU 948

20mA-TTY connecting cable (S7 (CP 341) - CP 523)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 523.

For the connecting cables you will require the following male connectors:

- On the CP 341 side: 9-pin sub D male connector with screw-locking
- On the communication partner: 25-pin sub D male connector with screw-locking

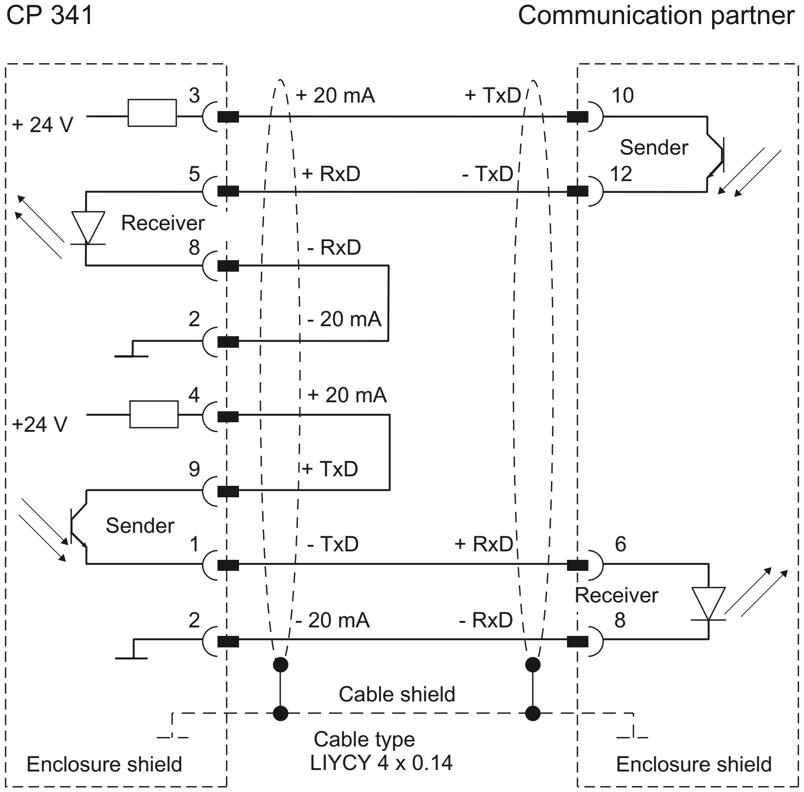


Figure B-10 20mA-TTY connecting cable CP 341 - CP 523

20mA-TTY connecting cable (S7 (CP 341) - CP 521 SI/CP 521 BASIC/ IBM-compatible printer)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 521 SI/CP 521 BASIC.

For the connecting cables you will require the following male connectors:

- On the CP 341 side: 9-pin sub D male connector with screw-locking
- On the communication partner: 25-pin sub D male connector with screw-locking

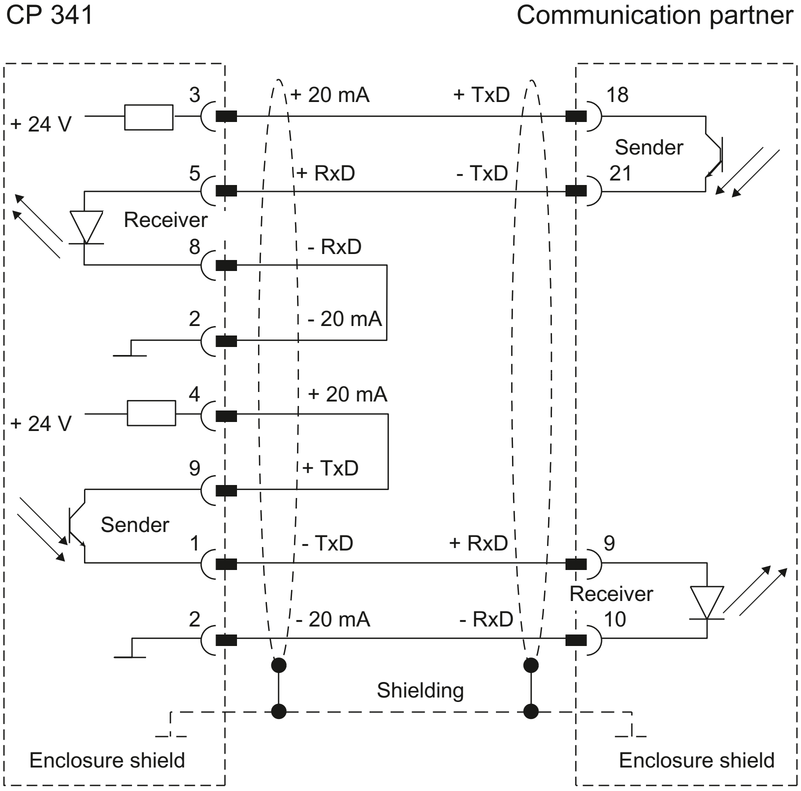


Figure B-11 20mA-TTY connecting cable CP 341 - CP 521 SI/CP 521 BASIC

20mA-TTY connecting cable (S7 (CP 341) - CPU 944/AG 95)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CPU 944/AG 95.

For the connecting cables you will require the following male connectors:

- On the CP 341 side: 9-pin sub D male connector with screw-locking
- On the communication partner: 15-pin sub D male connector with screw-locking

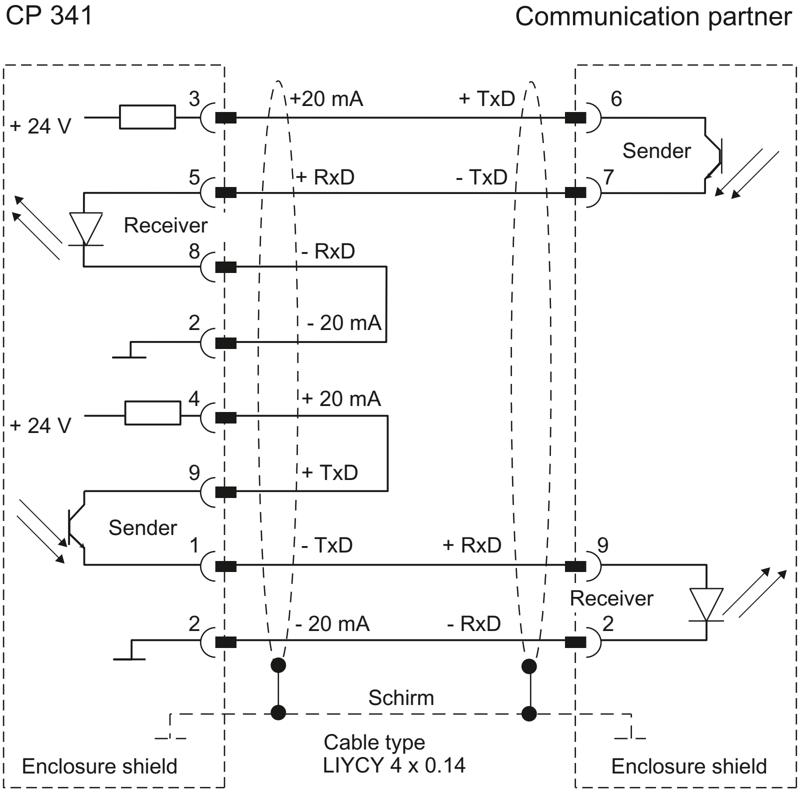


Figure B-12 20mA-TTY connecting cable CP 341 - CPU 944/AG 95

See also

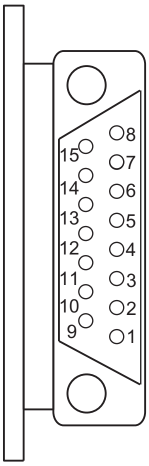
Mounting the CP 341 (Page 107)

B.3 X27 (RS 422/485) Interface of the CP 341-RS 422/485

Pin assignment

The table below shows the pin assignment for the 15-pin sub D females connector in the front panel of the CP 341-RS 422/485.

Table B- 3 Pin assignment for the 15-pin sub D female connector of the integrated interface of the CP 341-RS 422/485

Female connector on CP 341-RS 422/485*	Pin	Designation	Input/output	Meaning
 8 15 7 14 6 13 5 12 4 11 3 10 2 9 1	1	-	-	-
	2	T (A) -	Output	Send data (four-wire mode)
	3	-	-	-
	4	R (A)/T (A) -	Input Input/output	Receive data (four-wire mode) Receive/send data (two-wire mode)
	5	-	-	-
	6	-	-	-
	7	-	-	-
	8	GND	-	Functional ground (isolated)
	9	T (B) +	Output	Send data (four-wire mode)
	10	-	-	-
	11	R (B)/T (B) +	Input Input/output	Receive data (four-wire mode) Receive/send 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 interface modules. GND must always be connected on both sides (pin 8), otherwise the interface modules again be destroyed.

In the following

On the following pages you will find examples of connecting cables for a point-to-point connection between the CP 341-RS 422/485 and S7 modules or SIMATIC S5.

X 27 connecting cable (S7 (CP 341) - CP 340/CP 341/CP 441)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441 for RS 422 mode.

For the connecting cables you will require the following male connectors:

- At the CP 341 end: 15-pin sub D male connector with screw-locking
- On the communication partner: 15-pin sub D male connector with screw-locking

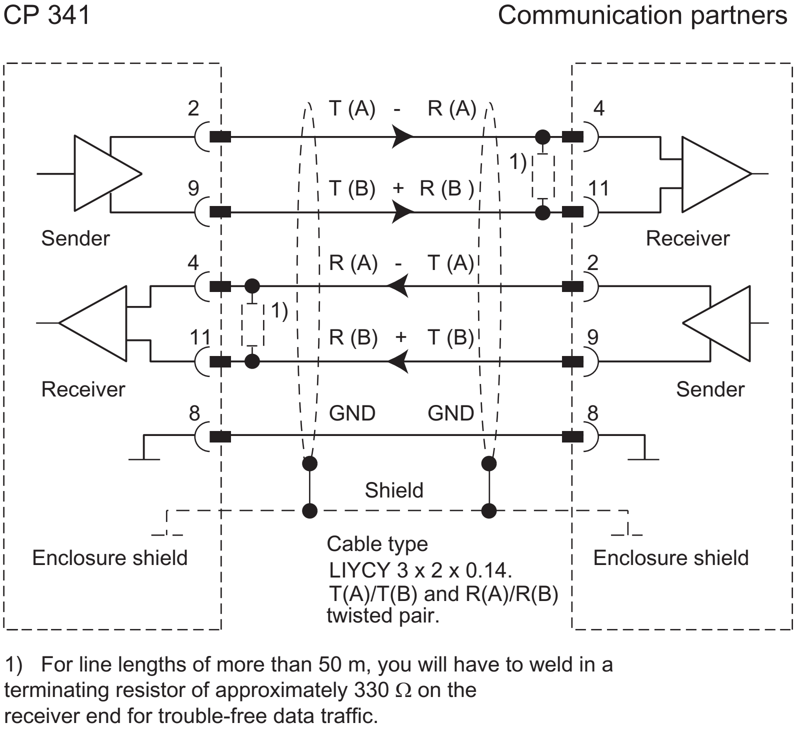


Figure B-13 X27 connecting cable CP 341 - CP 340/CP 341/CP 441 for RS 422 mode (four-wire)

The cable is available under the order number (6ES7 902-3...) specified in appendix Accessories and order numbers (Page 235).

Note

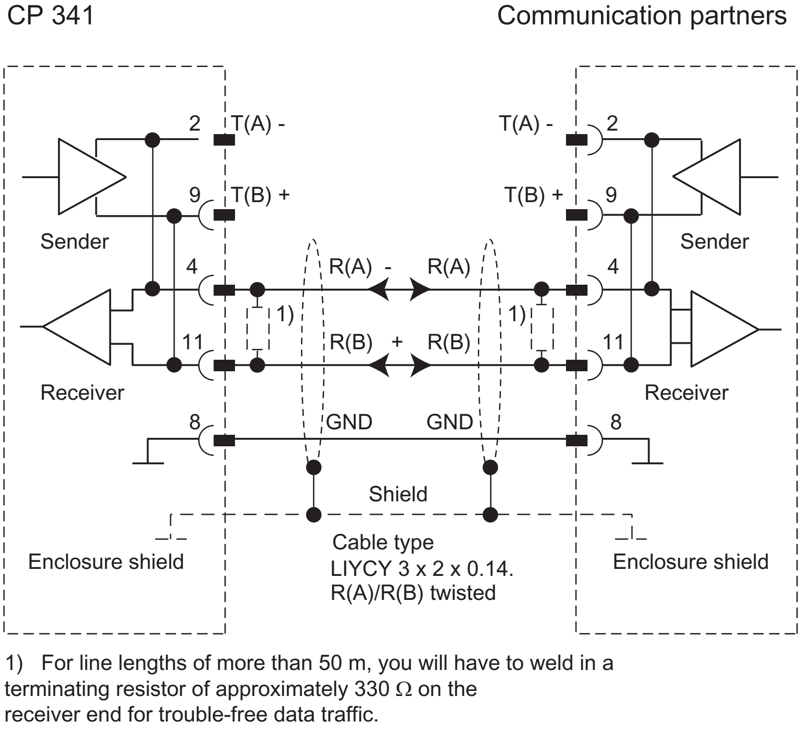
This cable type can be used in the following lengths for the CP 341 as communication partner: max. 1200 m at 19200 bps, max. 500 m at 38400 bps. 250 m at 115200 bps.

X 27 connecting cable (S7 (CP 341) - CP 340/CP 341/CP 441)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 340/CP 341/CP 441 for RS 485 mode.

For the connecting cables you will require the following male connectors:

- At the CP 341 end: 15-pin sub D male connector with screw-locking
- On the communication partner: 15-pin sub D male connector with screw-locking



1) For line lengths of more than 50 m, you will have to weld in a terminating resistor of approximately 330 Ω on the receiver end for trouble-free data traffic.

Figure B-14 X27 connecting cable CP 341 - CP 340/CP 341/CP 441 for RS 485 mode (two-wire)

Note

The previous figure shows the wiring if you want to make the connecting cable yourself. In both RS 485 mode (two wire) and RS 422 mode (four wire) you can also use Siemens connecting cables. The figure below illustrates the internal wiring in the connecting cable.

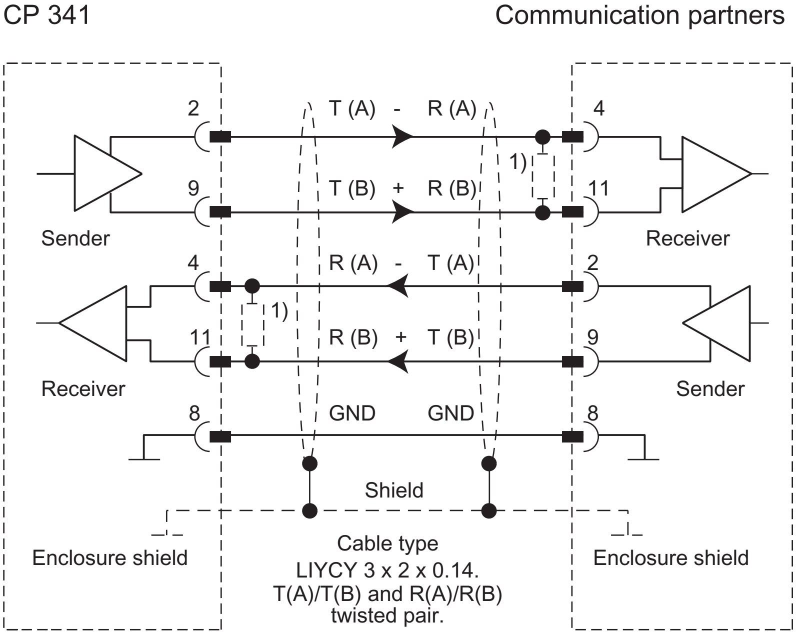
The jumpers 2-4 and 9-11 are "installed" by parameter assignment of the CP.

Connecting cable X 27 (S7 (CP 341) - CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The figure below illustrates the connecting cable for a point-to-point connection between a CP 341 and a CP 544, CP 524, CPU 928B, CPU 945, CPU 948 for RS 422 mode.

For the connecting cables you will require the following male connectors:

- At the CP 341 end: 15-pin sub D male connector with screw-locking
- On the communication partner: 15-pin sub D male connector with clip fixing



1) For line lengths of more than 50 m, you will have to weld in a terminating resistor of approximately 330 Ω on the receiver end for trouble-free data traffic.

Figure B-15 X27 connecting cable CP 341 - CP 544, CP 524, CPU 928B, CPU 945, CPU 948 for RS 422 mode (four-wire)

Accessories and order numbers C

Module variants

The table below contains the different variants of the CP 341.

Table C- 1 Order numbers of the module variants of the CP 341

Product	Order number
CP 341–RS 232C	6ES7 341–1AH02–0AE0
CP 341–20mA–TTY	6ES7 341–1BH02–0AE0
CP 341–RS 422/485	6ES7 341–1CH02–0AE0

Connecting cables

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

Table C- 2 Order numbers of the connecting cables

Connecting cables for CP 341 - CP 340; CP 341 – CP 341; CP 341 - CP 441	Version	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
20mA-TTY interface	20mA-TTY, 5 m	6ES7 902–2AB00–0AA0
	20mA-TTY, 10 m	6ES7 902–2AC00–0AA0
	20mA-TTY, 50 m	6ES7 902–2AG00–0AA0
X27 (RS 422) interface	X27 (RS 422), 5 m	6ES7 902–3AB00–0AA0
	X27 (RS 422), 10 m	6ES7 902–3AC00–0AA0
	X27 (RS 422), 50 m	6ES7 902–3AG00–0AA0

Literature on SIMATIC S7 D

D.1 Literature on SIMATIC S7

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 with which you can find out about the S7–300.

Manuals for configuring and commissioning

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. The following table also provides you with an overview of the documentation to **STEP 7**.

Table D- 1 Manuals for configuring and programming the S7–300

Title	Contents
Manual Programming with STEP 7 (http://support.automation.siemens.com/WW/view/en/18652056)	The programming manual offers basic information on the design of the operating system and a user program of an S7 CPU. It is intended to give first-time users of an S7-300/400 an overview of the programming methodology, thereby providing a basis on which they can build their user programs.
Manual Configuring Hardware and Communication Connections with STEP 7 (http://support.automation.siemens.com/WW/view/en/18652631)	This STEP 7 manual explains the principles behind the use and functions of the STEP 7 automation software. It will provide both first-time users of STEP 7 and those with knowledge of STEP 5 with an overview of the procedures for configuring, programming and starting up an S7-300/400. When working in the software, users can access the relevant sections of the online help where they will find specific support for its application.
Reference Manual Instruction list (IL) for S7-300/400 (http://support.automation.siemens.com/WW/view/en/18653496)	The manuals for the STL, LAD, FBD, and SCL language packages feature both instructions for users and language descriptions. You only need one of the languages to program an S7-300/400, however, you can change the language as required within a project. If you are using the languages for the first time, we recommend that you refer to the manual in order to familiarize yourself with the programming methodology. When working with the software you can use the online help to access detailed information about using the associated editors and compilers.
Reference Manual Ladder Diagram (LAD) for S7-300/400 (http://support.automation.siemens.com/WW/view/en/18654395)	
Reference Manual Function block diagram (FBD) for S7-300/400 (http://support.automation.siemens.com/WW/view/en/18652644)	

Title	Contents
Reference Manual S7-SCL for S7-300/400 (http://support.automation.siemens.com/WW/view/en/5581793) [1]	
Manual S7–GRAPH for S7-300/400 Programming Sequential Control Systems (http://support.automation.siemens.com/WW/view/en/1137630) [1]	The GRAPH, HiGraph and CFC languages support additional options for implementing sequential controls, state controls, or graphic interconnections of blocks. The manuals feature both instructions for users and language descriptions. If you are using the language for the first time, we recommend that you refer to the manual in order to familiarize yourself with the programming methodology. When working with the software you can also use the online help (with the exception of HiGraph) to access detailed information about using editors and compilers.
Manual Programming S7–HiGraph State Graphs (http://support.automation.siemens.com/WW/view/en/1137299) [1]	
Manuals CFC for SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/15236182) [1]	
Reference Manual System and Standard Functions for S7-300/400 (http://support.automation.siemens.com/WW/view/en/1214574)	The S7 CPU operating systems feature integrated system and standard functions, which you can use during programming in any of the supported languages (STL, LAD and SCL). The manual provides an overview of the basic functions supported by S7 and, for reference purposes, detailed interface descriptions for use in your user program.

[1] Option packages for S7–300/400 system software

Glossary

Address

The address indicates the physical storage space and enables direct access to the operand that is stored under this address.

Block

Blocks are parts of the user program that are separated by their function, structure or purpose. STEP 7 has the following blocks

- Code blocks (FB, FC, OB SFB, SFC)
- Data blocks (DB, SDB) and
- user-defined data types (UDT)

Block call

A block call is the branching of the program processing into the called block.

Block parameter

Block parameters are place holders within multiple use blocks, which are supplied with updated valves during the calling up of the corresponding block.

Communication processor

Communication processors are modules for point-to-point connections and bus connections

Configuring

Configuring refers to the configuration of separate modules of a programmable controller in the configuration table.

CP 341 programming interface: Point-to-Point Communication, parameter assignment

Using the CP 341 programming interface: Point-to-Point Communication, Parameter Assignment - assign parameters to the interface of the communication processor.

CPU

Central Processing Unit = Central module of the S7 Programmable Controller with control and computing unit, memory, system program and interfaces to the I/O modules.

Cycle time

The cycle time is the time that the CPU requires to process the user program once.

Cyclic program processing

In cyclic program processing the user program runs in program loop, or cycle, that is constantly repeated.

Data block (DB)

Data blocks are blocks that contain data and parameters with which the user program works. Unlike all other blocks, they do not contain any instructions. There are global data blocks and instance data blocks. The data contained in the data blocks can be accessed absolutely or symbolically. Complex data can be stored in structured form.

Data type

With the help of the data types you can specify how the value of a variable or constant in the user program is to be used. The data types are divided into elementary and structured data types

Default setting

The default setting is a reasonable basic setting that can be used whenever no other value is specified.

Diagnostic buffer

Each CPU has its own diagnostic buffer, in which detailed information on all the diagnostic events are entered in the sequence in which they occur.

The CP 341 has its own diagnostic buffer in which all the diagnostic events of the CP 341 are entered (hardware / firmware errors, initialization / parameterization errors, sending and receiving errors).

Diagnostic events

Diagnostic events are such as module errors, system errors in the CPU which may be caused by a program error or transitions from one operating mode to another.

Diagnostic functions

The diagnostic functions cover the entire system diagnostics and include the recognition, interpretation and reporting of errors within the Programmable Controller.

Downloading from the programming device

Downloading of load objects (e.g. code blocks) from the programming device into the load memory of the central processing unit (CPU).

Downloading to the programming device

Uploading of load objects (e.g. code blocks) from the load memory of the central processing unit into the programming device.

Function blocks (FBs)

Function blocks are components of the user program and are, according to IEC standard, "blocks with memory". The memory for the function blocks is an allocated data block, the "instance data block". Function blocks can be configured, i.e. you can use them with and without parameters.

Hardware

Hardware is the entire physical and technical equipment of a programmable controller.

Instance data block

The instance data block is a block allocated to a function block, which contains data for this special function block.

Interrupt

Interrupt is a term that designates the interruption of the processing of a program in the processor of a programmable controller by an external alarm

Module

Modules are pluggable PCBs for programmable controllers.

Module parameters

Module parameters are values with which the behavior of the module can be set. There are two different types of module parameters: static and dynamic.

Online Help

STEP 7 provides you with the option of having context-dependant help texts displayed on the screen while you are working with the programming software.

Online/Offline

When you are online there is a data connection between the programmable controller and programming device, when you are offline there is no data connection between them.

Operand

An operand is part of a STEP-7 instruction and states with which unit the process should execute something. It can be addressed both absolutely and symbolically.

Operating mode

The SIMATIC S7 programmable controllers have three different

operating modes: STOP, START-UP and RUN. The functionality of the CPU is different in the various operating modes.

Operating system of the CPU

The operating system of the CPU organizes all the functions and process of the CPU that are not connected to a special control task.

Parameter assignment

Parameter assignment refers to the setting of a module's behavior.

Parameters

Parameters are values that can be allocated. There are two different types of parameters: block parameters and module parameters.

Point-to-point communication

In point-to-point communication the communication processor forms the interface between a programmable controller and a communication partner.

Procedure

Procedure refers to the process of a data transmission according to a specific protocol.

Process image

The process image is a special memory area in the programmable controller. At the start of the cyclic program the signal states of the input modules are transmitted to the process image of the inputs. At the end of the cyclic program the process image of the outputs is transmitted as signal state to the output modules.

Programmable controller

A programmable controller is a stored-program control consisting of at least one CPU, various input and output modules, and operating and monitoring devices.

Protocol

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

Rack

The rack is the module rail containing the slots for the modules.

S7–300 backplane bus

The S7 300 backplane bus for module intercommunication, and power distribution to the modules.

Software

Software refers to the entirety of all programs that are used on a computing system. The operating system and user programs belong to this.

Startup

The START-UP operating mode forms the transition from STOP mode to RUN mode.

STEP 7

STEP 7 is the programming software of SIMATIC S7.

System blocks

System blocks are different from other blocks in that they are already integrated into the S7–300 system and are available for already defined system functions. There are system data blocks, system functions and system function blocks.

System function blocks (SFBs)

System functions are blocks without memory that are already integrated into the operating system of the CPU and can be called up by the user whenever necessary.

System functions (SFCs)

System functions are blocks without memory that are already integrated into the operating system of the CPU and can be called up by the user whenever necessary.

User program

The user program contains all instructions and declarations for processing the signals used for controlling a system or a process. In SIMATIC S7 the user program is structured and divided into small units, the blocks.

Variable

A variable is an operand (e.g. I 1.0) which can have a symbolic name and therefore also be addressed symbolically.

Working memory

The working memory is a RAM storage unit in the CPU which the processor draws on when running the user program.

Index