SIEMENS

SIMATIC

S7-1500 / ET 200MP / ET 200SP CM PtP in operation without SIMATIC system instructions

Programming and Operating Manual

Preface

Documentation guide 1

Overview 2

Basic information 3

Basic communication mechanisms 4

Data transmission with Freeport 5

Data transmission with 3964(R) 6

Configuration data records 7

Error status 8

Sending data 9

Receiving data 10

Clear receive buffer 11

RS232 signals 12

Error messages 13

05/2021

A5E03908199-AB

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

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

NOTICE

indicates that property damage can result if proper precautions are not taken.

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
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E03908199-AB
Ⓟ 04/2021 Subject to change

Copyright © Siemens AG 2021.
All rights reserved

Preface

Purpose of the documentation

You have two general options to use point-to-point communication modules:

Use of the communication modules with	Procedure
SIMATIC S7 CPU and Step 7 (TIA Portal) or Step 7 configuration software	Use the available specific instructions (http://support.automation.siemens.com/WW/view/en/59057093).
Third-party CPU and configuration software	You must implement the functionality according to the documentation in the configuration software of the third-party manufacturer.

Requirements

To use point-to-point communication modules with third-party CPU and configuration software, the CPU must support communication via data records and have a suitable fieldbus controller. Information is available in the system manuals ET 200SP distributed I/O system (http://support.automation.siemens.com/WW/view/en/58649293) and S7-1500, ET 200MP automation system (https://support.industry.siemens.com/cs/ww/en/view/59191792).

Basic knowledge required

The following knowledge is required in order to understand the documentation:

- General knowledge of automation technology
- Knowledge of the industrial automation system SIMATIC
- Knowledge about the use of Windows-based computers

Validity of the documentation

This documentation is valid for the following point-to-point communication modules:

- CM PtP RS232 BA (6ES7540-1AD0x-0AA0) (https://support.industry.siemens.com/cs/ww/en/view/59057152) communication module
- CM PtP RS422/485 BA (6ES7540-1AB0x-0AA0) (https://support.industry.siemens.com/cs/ww/it/view/59057390) communication module
- CM PtP RS232 HF (6ES7541-1AD0x-0AB0) (https://support.industry.siemens.com/cs/ww/en/view/59057160) communication module
- CM PtP RS422/485 HF (6ES7541-1AB0x-0AB0) (https://support.industry.siemens.com/cs/ww/en/view/59061372) communication module
- CM PtP (6ES7137-6AA0x-0BA0) (https://support.industry.siemens.com/cs/ww/en/view/59061378) communication module

Conventions

Please also observe notes labeled as follows:

Note

The notes contain important information on the product described in the documentation, on the handling of the product or on part of the documentation to which particular attention should be paid.

Recycling and disposal

For environmentally sustainable recycling and disposal of your old equipment, contact a certified electronic waste disposal company and dispose of the equipment according to the applicable regulations in your country.

Additional assistance

The range of technical documentation for the individual SIMATIC products and systems can be found on the Internet (http://www.siemens.com/simatic-tech-doku-portal).

Siemens Industry Online Support

You can find current information on the following topics quickly and easily here:

- **Product support**

  All the information and extensive know-how on your product, technical specifications, FAQs, certificates, downloads, and manuals.

- **Application examples**

  Tools and examples to solve your automation tasks – as well as function blocks, performance information and videos.

- **Services**

  Information about Industry Services, Field Services, Technical Support, spare parts and training offers.

- **Forums**

  For answers and solutions concerning automation technology.

- **mySupport**

  Your personal working area in Industry Online Support for messages, support queries, and configurable documents.

This information is provided by the Siemens Industry Online Support in the Internet (https://support.industry.siemens.com).

Industry Mall

The Industry Mall is the catalog and order system of Siemens AG for automation and drive solutions on the basis of Totally Integrated Automation (TIA) and Totally Integrated Power (TIP).

You can find catalogs for all automation and drive products on the Internet (https://mall.industry.siemens.com) and in the Information and Download Center (https://www.siemens.com/automation/infocenter).

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions form one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit (http://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under (http://www.siemens.com/industrialsecurity).

Table of contents

Preface ... **3**

1 Documentation guide ... **8**

2 Overview ... **13**

3 Basic information ... **17**

3.1 Serial transmission of a character ... 17

3.2 Transmission security ... 19

4 Basic communication mechanisms ... **21**

4.1 Data exchange types ... 21

4.2 Communication using data record (Universal) ... 23

4.3 Communication via user data (Performance option) ... 25

5 Data transmission with Freeport ... **26**

5.1 Sending data with Freeport ... 26

5.2 Receiving data with Freeport ... 28

5.3 Code transparency ... 32

5.4 Receive buffer ... 32

5.5 RS232 mode ... 33

5.6 RS422 mode ... 34

5.7 RS485 mode ... 35

5.8 Handshaking ... 36

5.9 Communication via DMX512 ... 39

6 Data transmission with 3964(R) ... **40**

6.1 Data transmission with 3964(R) protocol ... 40

6.2 Control characters ... 40

6.3 Block checksum ... 41

6.4 Sending data with 3964(R) ... 41

6.5 Receiving data with 3964(R) ... 43

7 Configuration data records ... 44

7.1 Freeport communication ... 44
7.1.1 Port configuration record - data record 57 ... 44
7.1.2 Send configuration record - data record 59 ... 47
7.1.3 Receive configuration record - data record 60 ... 49

7.2 3964 communication ... 52
7.2.1 Port configuration record - data record 57 ... 52
7.2.2 3964 protocol configuration - data record 61 ... 52

7.3 Activate special function - data record 58 ... 53

8 Error status ... 55

8.1 Read error status - data record 55 ... 55

9 Sending data ... 56

9.1 Sending data using data record ... 56
9.1.1 Flow chart for sending using data record ... 56
9.1.2 Send data - data record 48 ... 57
9.1.3 Request acknowledgment - Input data ... 59
9.1.4 Example ... 60

9.2 Sending data via output data ... 61
9.2.1 Flow chart for sending using output data ... 61
9.2.2 Output data when using the Performance option ... 62
9.2.3 Request acknowledgment - Input data ... 63

10 Receiving data ... 64

10.1 Receiving data using data record ... 64
10.1.1 Flow chart for receiving using data record ... 64
10.1.2 Data received - data record 49/50 ... 65
10.1.3 Query: New receive frame - Input data ... 66
10.1.4 Example ... 67

10.2 Receiving data via input data ... 68
10.2.1 Flow chart for receiving via input data ... 68
10.2.2 Input data when using the Performance option ... 69

11 Clear receive buffer ... 71

11.1 Clear receive buffer - data record 54 ... 71

12 RS232 signals ... 72

12.1 Set RS232 secondary signals - data record 53 ... 72

12.2 Read RS232 secondary signals - Input data ... 73

13 Error messages ... 74

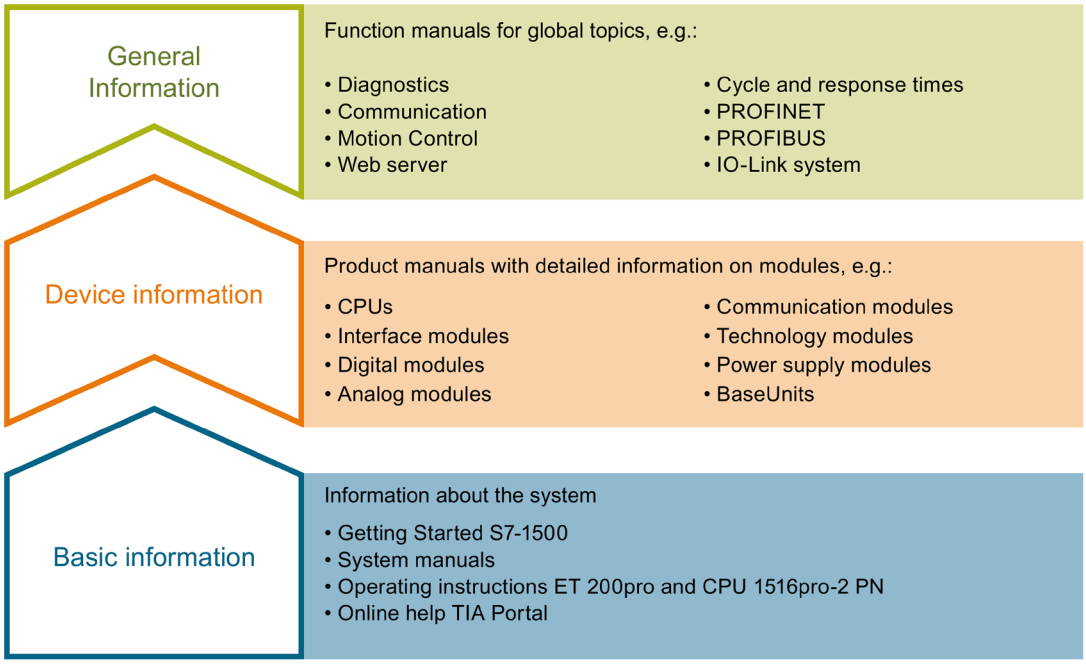
Glossary ... 81

Index ... 85

Documentation guide

1

The documentation for the SIMATIC S7-1500 automation system, the CPUs 1513/1516pro-2 PN based on SIMATIC S7-1500, and the distributed I/O systems SIMATIC ET 200MP, ET 200SP and ET 200AL is divided into three areas.
This division allows you easier access to the specific information you require.



Basic information

System manuals and Getting Started manuals describe in detail the configuration, installation, wiring and commissioning of the SIMATIC S7-1500, ET 200MP, ET 200SP and ET 200AL systems. Use the corresponding operating instructions for the CPUs 1513/1516pro-2 PN. The STEP 7 online help supports you in the configuration and programming.

Device information

Product manuals contain a compact description of the module-specific information, such as properties, terminal diagrams, characteristics and technical specifications.

General information

The function manuals contain detailed descriptions on general topics such as diagnostics, communication, Motion Control, Web server, OPC UA.

You can download the documentation free of charge from the Internet (https://support.industry.siemens.com/cs/ww/en/view/109742705).

Changes and additions to the manuals are documented in product information sheets.

You will find the product information on the Internet:

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/us/en/view/68052815)
- ET 200SP (https://support.industry.siemens.com/cs/us/en/view/73021864)

Manual Collections

The Manual Collections contain the complete documentation of the systems put together in one file.

You will find the Manual Collections on the Internet:

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/ww/en/view/86140384)
- ET 200SP (https://support.industry.siemens.com/cs/ww/en/view/84133942)
- ET 200AL (https://support.industry.siemens.com/cs/ww/en/view/95242965)

"mySupport"

With "mySupport", your personal workspace, you make the best out of your Industry Online Support.

In "mySupport", you can save filters, favorites and tags, request CAx data and compile your personal library in the Documentation area. In addition, your data is already filled out in support requests and you can get an overview of your current requests at any time.

You must register once to use the full functionality of "mySupport".

You can find "mySupport" on the Internet (https://support.industry.siemens.com/My/ww/en).

"mySupport" - Documentation

With "mySupport", your personal workspace, you make the best out of your Industry Online Support.

In "mySupport", you can save filters, favorites and tags, request CAx data and compile your personal library in the Documentation area. In addition, your data is already filled out in support requests and you can get an overview of your current requests at any time.

You must register once to use the full functionality of "mySupport".

You can find "mySupport" on the Internet (http://support.industry.siemens.com/My/ww/en/documentation).

"mySupport" - CAx data

In the CAx data area of "mySupport", you can access the latest product data for your CAx or CAe system.

You configure your own download package with a few clicks.

In doing so you can select:

- Product images, 2D dimension drawings, 3D models, internal circuit diagrams, EPLAN macro files
- Manuals, characteristics, operating manuals, certificates
- Product master data

You can find "mySupport" - CAx data on the Internet (http://support.industry.siemens.com/my/ww/en/CAxOnline).

Application examples

The application examples support you with various tools and examples for solving your automation tasks. Solutions are shown in interplay with multiple components in the system - separated from the focus on individual products.

You will find the application examples on the Internet (https://support.industry.siemens.com/sc/ww/en/sc/2054).

TIA Selection Tool

With the TIA Selection Tool, you can select, configure and order devices for Totally Integrated Automation (TIA).
This tool is the successor of the SIMATIC Selection Tool and combines the known configurators for automation technology into one tool.
With the TIA Selection Tool, you can generate a complete order list from your product selection or product configuration.

You can find the TIA Selection Tool on the Internet (http://w3.siemens.com/mcms/topics/en/simatic/tia-selection-tool).

SIMATIC Automation Tool

You can use the SIMATIC Automation Tool to perform commissioning and maintenance activities simultaneously on various SIMATIC S7 stations as a bulk operation independent of TIA Portal.

The SIMATIC Automation Tool provides a multitude of functions:

- Scanning of a PROFINET/Ethernet system network and identification of all connected CPUs
- Address assignment (IP, subnet, gateway) and station name (PROFINET device) to a CPU
- Transfer of the date and the programming device/PC time converted to UTC time to the module
- Program download to CPU
- RUN/STOP mode switchover
- CPU localization by means of LED flashing
- Reading out of CPU error information
- Reading of the CPU diagnostics buffer
- Reset to factory settings
- Firmware update of the CPU and connected modules

You can find the SIMATIC Automation Tool on the Internet (https://support.industry.siemens.com/cs/ww/en/view/98161300).

PRONETA

SIEMENS PRONETA (PROFINET network analysis) allows you to analyze the plant network during commissioning. PRONETA features two core functions:

- The topology overview automatically scans the PROFINET and all connected components.
- The IO check is a fast test of the wiring and the module configuration of a plant.

You can find SIEMENS PRONETA on the Internet (https://support.industry.siemens.com/cs/ww/en/view/67460624).

SINETPLAN

SINETPLAN, the Siemens Network Planner, supports you in planning automation systems and networks based on PROFINET. The tool facilitates professional and predictive dimensioning of your PROFINET installation as early as in the planning stage. In addition, SINETPLAN supports you during network optimization and helps you to exploit network resources optimally and to plan reserves. This helps to prevent problems in commissioning or failures during productive operation even in advance of a planned operation. This increases the availability of the production plant and helps improve operational safety.

The advantages at a glance

- Network optimization thanks to port-specific calculation of the network load
- Increased production availability thanks to online scan and verification of existing systems
- Transparency before commissioning through importing and simulation of existing STEP 7 projects
- Efficiency through securing existing investments in the long term and the optimal use of resources

You can find SINETPLAN on the Internet (https://www.siemens.com/sinetplan).

Overview

2

Overview of components and article numbers

Tabular overview of communication modules.

Communication module	Article number
CM PtP RS232 BA	6ES7540-1AD0x-0AA0
CM PtP RS422/485 BA	6ES7540-1AB0x-0AA0
CM PtP RS232 HF	6ES7541-1AD0x-0AB0
CM PtP RS422/485 HF	6ES7541-1AB0x-0AB0
CM PtP (ET 200SP)	6ES7137-6AA0x-0BA0

Overview of components and interfaces

Tabular overview of communication modules and their functions.

Communication module	Interface	Protocols		Connection of the communication cable	
		Freeport	3964(R)	D-Sub 9-pin	D-Sub 15-pin
CM PtP RS232 BA	RS232	X	X	X	-
CM PtP RS422/485 BA	RS422	X	X	-	X
	RS485	X	-	-	X
CM PtP RS232 HF	RS232	X	X	X	-
CM PtP RS422/485 HF	RS422	X	X	-	X
	RS485	X	-	-	X
CM PtP (ET 200SP)	RS232	X	X	ET 200SP BaseUnit with 18 terminals[1]	
	RS422	X	X		
	RS485	X	-		

[1] ET 200SP BaseUnit instead of sub D; same for all; pin assignment depending on physical transmission properties

Overview of components and data transmission rates

Communication module	Data transmission rate in bps											
	300	600	1200	2400	4800	9600	19200	38400	57600	76800	115200	250000[1]
CM PtP RS232 BA	X	X	X	X	X	X	X	-	-	-	-	-
CM PtP RS422/485 BA	X	X	X	X	X	X	X	-	-	-	-	-
CM PtP RS232 HF	X	X	X	X	X	X	X	X	X	X	X	-
CM PtP RS422/485 HF	X	X	X	X	X	X	X	X	X	X	X	-
CM PtP (ET 200SP)	X	X	X	X	X	X	X	X	X	X	X	X

[1] Especially for using the DMX512 protocol with RS485 (Page 39)

Overview of components and receive buffer size

Module	Receive buffer size in KB	Max. frame length in KB	Bufferable frames
CM PtP RS232 BA	2	1	255
CM PtP RS422/485 BA	2	1	255
CM PtP RS232 HF	8	4	255
CM PtP RS422/485 HF	8	4	255
CM PtP (ET 200SP)	4	2	255

Secondary signals and data flow control

- Software data flow control with XON/XOFF

  The Freeport protocol supports data flow control with XON/XOFF via the RS232 and RS422 interfaces.

- Hardware data flow control with RTS/CTS

  The Freeport protocol supports data flow control with RTS/CTS via the RS232 interface.

- Automatic operation of secondary signals

  The Freeport protocol supports operation of the RS232 secondary signals by means of the RS232 interface. Only available when hardware data flow control is activated.

Application options for the communication modules

You may set up a communication connection with different partners, depending on the communication modules used:

- Freeport: Transmission of ASCII character strings without specified protocol format
- 3964(R): Communication between automation systems (master/master communication)

GSD file

Engineering must be provided with information regarding functionality and parameters of the communication modules for the higher-level application.

The properties of the point-to-point communication modules are defined in a GSD file (Generic Station Description) that is defined based on the GSD schema (ISO 15745: GSD for PROFIBUS and GSDML for PROFINET IEC 61804: EDDL IEC 62453: FDT/DTM). The Engineering System must provide a suitable interpreter that makes the data of the file available to the system.

The GSD file contains:

- Device and identification data
- Parameter assignment data
- Diagnostics data

For parameter assignment of the communication modules, the GSD file must be installed/imported in the Engineering System.

ET 200MP

The respective GSD file for the ET 200MP distributed I/O system is available for download on the Internet:

- GSD file PROFINET IO (https://support.industry.siemens.com/cs/ww/en/view/68189683)
- GSD file PROFIBUS DP (https://support.industry.siemens.com/cs/ww/en/view/80206700)

ET 200SP

The respective GSD file for the ET 200SP distributed I/O system is available for download on the Internet:

- GSD file PROFINET IO (http://support.automation.siemens.com/WW/view/en/57138621)
- GSD file PROFIBUS DP (http://support.automation.siemens.com/WW/view/en/73016883)

Parameter assignment in the application when using a GSD file for PROFIBUS slaves

Important parameters are missing in the GSD file for PROFIBUS slaves. The parameter assignment of the communication module must therefore be performed from the user program using the following data records:

Freeport communication:

- Port configuration record - data record 57 (Page 44) (Port_Config)
- Send configuration record - data record 59 (Page 47) (Send_Config)
- Receive configuration record - data record 60 (Page 49) (Receive_Config)
- Activate special function - data record 58 (Page 53) (Set_Features)

3964 communication:

- Port configuration record - data record 57 (Page 44) (Port_Config)
- 3964 protocol configuration - data record 61 (Page 52) (P3964_Config)
- Activate special function - data record 58 (Page 53) (Set_Features)

Note

Activate special function - data record 58 (Page 53) must always be called as the last data record.

Note

To use the communication module, the CPU must support communication by means of data records.

For a SIMATIC S7 CPU and Step 7 (TIA Portal) / Step 7, the parameter assignment can be performed with the instructions given in parentheses. You can find more information on configuring with GSD file in the FAQ with the entry ID 109487470 (https://support.industry.siemens.com/cs/ww/en/view/109487470) in Siemens Industry Online Support.

Basic information 3

3.1 Serial transmission of a character

Introduction

Different networking options are available for the exchange of data between two or more communication partners. The PtP connection between two communication partners is the simplest means of information exchange.

Point-to-point connection

In point-to-point connection, the communication module forms the interface between an automation system and a communication partner. Data is sent in serial mode via PtP connection with the communication module.

Serial data transmission

During serial data transmission, the individual bits of a character of information to be transferred are sent successively in a defined sequence.

Bidirectional data traffic - operating mode

In the context of bidirectional data traffic, we distinguish between two operating modes for the communication module:

- Half duplex mode

  The data is transferred between one or several communication partners in both directions alternately. Half duplex mode means that data is either sent or received at a given point in time.

- Full duplex mode

  The data is exchanged simultaneously between one or several communication partners, i.e. it is possible to send and receive data simultaneously at a given point in time.

Data transmission

The so-called time base synchronism (fixed time base for transmission of a fixed character string) is only upheld during transmission of a character. Each character to be sent is preceded by a synchronization pulse, which is also known as start bit. The length of the start bit transmission determines the clock cycle. The end of character transmission is formed by one or two stop bits.

Handshakes

In addition to the start and stop bits, both communication partners need further handshake procedures for serial data transmission. These include:

- Data transmission rate
- Character and acknowledgment delay time
- Parity
- Number of data bits (7 or 8 bits/characters)
- Number of stop bits (1 or 2)

3.2 Transmission security

Transmission security plays an important role in the transmission of data and in the selection of the transmission procedure. Generally speaking, the more layers of the ISO/OSI reference model are applied, the higher the transmission security.

Classification of existing protocols

The figure below illustrates how the protocols of the communication module fit into the reference model.

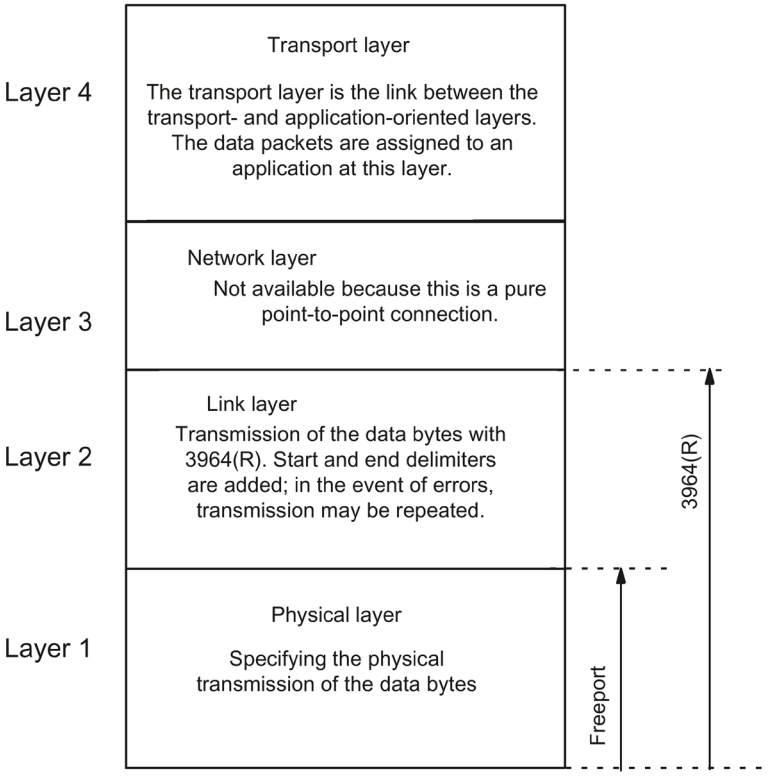


Figure 3-1 Classification of the existing protocols of the communication module in the ISO/OSI reference model

Transmission security with Freeport

Data security when using Freeport:

- Despite the efficiency of data transmission with the Freeport protocol, it is not possible to ensure secure data transmission with Freeport because no data security measures other than the use of a parity bit (can also be disabled, depending on the character frame setting) are applied. A certain degree of data security can be achieved through parameter assignment of the frame start and frame end conditions.
- The parity bit ensures that the inversion of a bit in a character to be sent can be recognized. If two or more bits of a character are inverted, this error can no longer be detected.
- To increase transmission security, you can, for example, implement a checksum, a frame length specification, or configurable end conditions. These measures must be implemented by the user.
- A further increase in data security can be achieved by means of acknowledgment message frames in response to send or receive message frames. This is the case with high-grade protocols for data communication (ISO 7-layer reference model).

Transmission security with 3964(R)

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

- Data transmission without block check character: **3964**

  Transmission security is achieved by means of a specified frame structure, frame breakdown, and frame repetitions.

- Data transmission with block check character: **3964R**

  The high degree of transmission security is achieved by means of a specified frame structure and breakdown, frame repetitions, as well as inclusion of a block check character (BCC). The hamming distance in 3964R mode is 3.

The hamming distance represents a measure for data transmission security.

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

Basic communication mechanisms

4

4.1 Data exchange types

Data communication

Two types of data exchange between the CPU and the communication module are possible with the communication modules, depending on whether you are using the Performance option:

Acyclic data exchange (Universal)	**Cyclic data exchange (with performance optimization)**
The CPU communicates with the communication module asynchronously by reading or writing data records (Page 23).	The CPU communicates with the communication module synchronously with the application cycle via the net data of the communication module (Page 25).
Limiting the telegram length depending on the communication module to 1, 2, or 4 KB	Limiting the telegram length to 24 bytes for receive frames and 30 bytes for send frames. Longer frames are rejected.
Transmitting a frame requires several cycles of the CPU. The number of cycles increases with the number of communication modules that communicate via data record.	Transmission of a frame requires an application cycle of the CPU and several communication modules can be served in parallel (reaction time optimized, timing behavior improved).
Allocation of an address range of 8 bytes of input data and 0 bytes of output data	Allocation of an address range of 32 bytes of input data and 32 bytes of output data
Available from firmware version V1.0 of the communication module	Available from firmware version V2.0 of the communication modules
Configuration and parameter assignment with GSD file. The option for performance optimization cannot be changed with the configuration data records 57 to 61.	

Example

In this example, 18 communication modules are used in distributed operation, but the measured values are received by one digital scale each. The scales send their measured values cyclically at such a high speed that the update time of the measured values in the CPU is limited by the communication speed between CPU and communication module.

The following comparison shows the difference of the update time depending on whether performance optimization is used:

	Number of communication modules read in parallel by the CPU	**Update time (interval between two value updates in the CPU) per communication module**
Without Performance option	1	Approx. 30 ms
	18	Approx. 250 ms
With Performance option	1	Approx. 20 ms[1]
	18	Approx. 20 ms[1]

[1] Depending on the application, faster update times are also possible; for example, for sensors that only send short frames cyclically.

4.2 Communication using data record (Universal)

Write/read data record

Note

Reading and writing data records

For information about the functions for reading/writing data records, refer to the documentation of your controller.

The "Write data record" function can be used to configure the devices by means of configuration data records. The "Write data record" function also transmits the send data to the devices. Accordingly, "Read data record" can be used to read diagnostics data and receive data from the devices.

Overview of data records

Configuration data records are available for the following tasks:

- Freeport communication
  Configuration of the communication module for data transmission using the open Freeport frame
  - Port configuration record - data record 57 (Page 44)
  - Activate special function - data record 58 (Page 53)
  - Send configuration record - data record 59 (Page 47)
  - Receive configuration record - data record 60 (Page 49)
- 3964 communication
  Configuration of the communication module for data transmission with 3964(R) protocol
  - Port configuration record - data record 57 (Page 44)
  - 3964 protocol configuration - data record 61 (Page 52)
- Special functions - Activation
  Activating or verifying the active state of diagnostic interrupts.
  - Activate special function - data record 58 (Page 53)

The following data records are available for operating the communication modules:

- Query error status
  Acknowledgment of the communication module in response to the writing of a configuration data record
  - Read error status - data record 55 (Page 55)
- Send data
  Transmitting send data to the communication module and querying transmission acknowledgment in the user data
  - Send data - data record 48 (Page 57)
  - Request acknowledgment - Input data (Page 59)
- Receive data
  Display of a receive frame in the net data
  - Data received - data record 49/50 (Page 65)
  - Query: New receive frame - Input data (Page 66)
- Delete receive buffer
  Deleting the receive buffer of the communication module
  - Clear receive buffer - data record 54 (Page 71)
- RS232 secondary signals
  Manipulation of the RS232 secondary signals of the communication module and reading the RS232 secondary signals in the net data
  - Set RS232 secondary signals - data record 53 (Page 72)
  - Read RS232 secondary signals - Input data (Page 73)

Sending

Sending data via data records is described in the section Sending data using data record (Page 56).

Receiving

Receiving data via data records is described in the section Receiving data using data record (Page 64).

4.3 Communication via user data (Performance option)

From firmware version V2.0 of the communication module, the option for performance optimization is available. This option is suitable if you are exclusively sending and receiving short frames. Unlike in communication via data records (Universal), the number of communication modules in a project does not affect the update time.

In this case, sending and receiving of user data takes place via the net data (cyclic IO data of the automation system) of the communication module. Communication via data records is used for configuration and diagnostics.

Sending

Sending data via net data is described in the section Sending data via output data (Page 61).

Receiving

Receiving data via net data is described in the section Receiving data via input data (Page 68).

Data transmission with Freeport 5

Introduction

Freeport is a freely programmable, frame-based protocol. Freeport enables users to map almost all protocols available in the market in applications.

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

The Freeport driver supports sending and receiving of messages with any structure (all characters from 00_H through FF_H (for character frames with 8 data bits) or from 00_H through $7F_H$ (for character frames with 7 data bits).

The frame start and end criteria must be configured both for the send and the receive direction. The structure of the send frames may differ from that of the receive frames.

5.1 Sending data with Freeport

Sending data

Note

During parameter assignment of the XON/XOFF data flow control, user data may not contain any of the configured XON or XOFF characters. The default settings are DC1 = 11_H for XON and DC3 = 13_H for XOFF.

Specifying settings for sending

To send a message, the partner must be informed of the start and end of the message (see Send configuration record - data record 59 (Page 47)). You can select one of the following options:

- Send the Break signal at the start of the message

  You can specify that an additional Break is sent at the beginning of each message transmission on expiration of the RTS ON delay time.
  The duration of the "Break" signal is specified in bit times (refer to Send configuration record - data record 59 (Page 47) bytes 12 and 13).

  Compliance with the send break can be deactivated if other mechanisms are used for synchronization.

- Send Idle Line

  You can specify that at the beginning of each message transmission the send line is held in Idle state for a specific time.
  The duration of the "Idle Line" is specified in bit times (refer to Send configuration record - data record 59 (Page 47) bytes 14 and 15).

  Compliance with the send break can be deactivated if other mechanisms are used for synchronization.

- RTS ON delay

  You can configure the time that has to expire after the RTS (Request to send) before the actual data transmission starts (see Send configuration record - data record 59 (Page 47) bytes 8 and 9).

- RTS OFF delay

  You can configure the time that has to expire after transmission has been completed before the RTS signal is deactivated (see Send configuration record - data record 59 (Page 47) bytes 10 and 11).

- Sending including the end delimiter

  You can configure the number of end delimiters (1 or 2) and their value.

  All data up to the end delimiter(s) is sent, independent of the selected frame length. The end delimiter must be included in the data to be sent. Data is sent only up to and including the end delimiter, even if the data length specified is longer (refer to Send configuration record - data record 59 (Page 47) bytes 18 and 21).

- Number of appended characters

  Input of the number of appended characters. All data transferred to the communication module will be sent. The end delimiter(s) is/are appended automatically. Depending on the number of end delimiters, one to five characters more than the specified number are sent to the partner (refer to Send configuration record - data record 59 (Page 47) bytes 22 and 28).

Note

If you do not configure either the "End delimiter" (0) or the "Number of appended characters" (0), all data transferred to the communication module is sent to the partner in a transparent operation.

5.2 Receiving data with Freeport

Introduction

You need to specify message start and end criteria in the Freeport driver to enable detection of a message in the incoming data flow.

Note

The start and end criteria described below can be combined as required. Such combinations may lead to unwanted results if the start and end criteria influence or even exclude each other.

Specifying the message start

For data transmission with Freeport, you can choose between several different start criteria. The start criterion defines when a frame starts. Once a criterion that indicates the start of the message is met, the data stream is scanned for message end criteria. (See Receive configuration record - data record 60 (Page 49) byte 9)

Two different methods are available for detecting the message start:

- **Start on any character**
- **Start on special condition**

Start on any character

- **Start on any character**

  Any character can be used to define the start of the message (default).

  This means that the first character sent at the start of communication, or after the frame end has been detected, will be identified as the first character of a message. (refer to Receive configuration record - data record 60 (Page 49) byte 12)

Start on special condition

The start of the message is detected based on the following specified conditions:

- **After detection of a line break**

  The frame start is not accepted unless a break has been received beforehand, in other words, it is compulsory for the partner to send a break before sending a frame.

- **After detection of an idle line**

  The frame start is not accepted until the configured idle line duration has expired. This procedure requires a minimum interval between two frames. (See Receive configuration record - data record 60 (Page 49) bytes 10 and 11)

- **After receipt of a start character**

  The frame start is detected when the configured start character is identified. (See Receive configuration record - data record 60 (Page 49) byte 12)

- **After detection of one or several start sequences**

  The frame start is detected when the configured string with a length of up to five characters is identified. You can configure up to 4 start sequences. (See Receive configuration record - data record 60 (Page 49) bytes 13 to 37)

Specifying the message end

You can choose from several different end criteria for data transmission using the Freeport driver. The end criterion defines the point at which a message frame has been received completely. (See Receive configuration record - data record 60 (Page 49) byte 39)

Configurable end criteria are:

- **Recognize message end by message timeout**
- **Recognize message end by response timeout**
- **After character delay time elapses (default)**
- **After receipt of a fixed frame length**
- **After receipt of a maximum number of characters**
- **Read message length from message**
- **After receipt of an end sequence**

Message timeout

When data is received, the end of frame is detected on expiration of the configured time for transferring a frame. Time measurement starts after the start criterion has been met. (See Receive configuration record - data record 60 (Page 49) bytes 52 and 53)

An error message is output and the frame is discarded if a message end condition is met before the fixed frame length has been reached.

Response timeout

When data is received, the end of frame is detected on expiration of the configured time between the end of the send operation and the receipt of a frame start. (refer to Receive configuration record - data record 60 (Page 49) bytes 50 and 51)

An error message is output and the frame is discarded if a message end condition is met before the fixed frame length has been reached.

Expiration of character delay time

When data is received, the frame end is detected when the configured maximum time (bit times) between successive characters is exceeded (character delay time). (See Receive configuration record - data record 60 (Page 49) bytes 54 and 55)

In this case, the character delay time must be set in such a way as to ensure that it expires between two consecutive frames. However, it should be of sufficient length to exclude incorrect identification of the end of the frame whenever the communication partner performs a transmission pause within a frame.

Fixed frame length

When data is received, the end of the frame is identified after the configured frame length has been reached. (See Receive configuration record - data record 60 (Page 49) bytes 40 and 41)

An error message is output and the frame is discarded if the character delay time expires (if activated) before the fixed frame length has been reached.

Please note the following if the frame length of the received characters does not match the fixed configured frame length:

- All characters received after the configured maximum number of characters has been reached will be discarded until a new start criterion (e.g. "Idle Line") is detected.
- Receiving is terminated if the character delay time (if activated) expires before the fixed configured frame length has been reached. An error message is generated and the message frame fragment is discarded.
- If a different (activated) end criterion is met before the fixed configured frame length has been reached, this frame part is assessed as a valid frame and the partner waits for a new start criterion. All characters received prior to fulfillment of a new start criterion are discarded.

Maximum number of characters

When receiving data, the end of the message frame is recognized after the declared number of characters have arrived. (See Receive configuration record - data record 60 (Page 49) bytes 42 and 43)

This setting can be combined with the "Character delay time" settings. The frame received is also assessed as free of error if another end condition occurs, regardless of whether the maximum number of characters has been reached.

Please note the following if the frame length of the received characters does not match the configured maximum frame length:

- All characters received after the configured maximum number of characters has been reached will be discarded until a new start criterion (e.g. "Idle Line") is detected.
- If a different (activated) end criterion is met before the configured maximum number of characters has been reached, this frame part is assessed as a valid frame and the partner waits for a new start criterion. All characters received prior to fulfillment of a new start criterion are discarded.

Note

If no further end criterion is activated, the fixed frame length and maximum number of characters will respond in the same way.

Message length in the message

The length of the receive frame is also sent. When data is received, the frame end is identified after the frame length also sent has been reached. If "After character delay time elapses" is activated as an additional end criterion, receiving may be terminated prematurely. An error message is generated and the message frame fragment is discarded.

The following parameters define the characters to be used for evaluation of the message length:

- **Offset of the length field in the message**

  In the message, the value defines the position of the character that is to be used to determine the message length.

  You can set values from 1 to 4096 characters, depending on the buffer size.

  (See Receive configuration record - data record 60 (Page 49) bytes 44 to 45)

- **Size of the length field**

  This value specifies the number of characters as of the first evaluation position to be used to determine the message length.

  You can set values of 0, 1, 2 and 4 characters.

  (See Receive configuration record - data record 60 (Page 49) bytes 46 to 47)

- **Number of characters not counted in length specification**

  Number of characters appended to the frame without counting towards the frame length. This value defines the number of bytes at the end of the frame which should not be included in the evaluation of the message length.

  You can set values from 0 to 255 characters.

  (See Receive configuration record - data record 60 (Page 49) bytes 48 to 49)

End sequence

When data is received, the end of the frame is identified when the configured end sequence (max. 5 characters) is received. The received data is transferred to the CPU including the end sequence. (See Receive configuration record - data record 60 (Page 49) bytes 56 to 61)

If you are working with the end sequence, transmission is not code-transparent and you must exclude the presence of end code in the user data.

5.3 Code transparency

Code transparency

Code transparency means that user data can contain any character combinations, without recognition of the end criterion.

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

- With up to five end delimiters or using XON/XOFF flow control
  - Not code-transparent
- End criterion character delay time, fixed frame length, maximum frame length, message timeout, or response timeout and message length in the message
  - Code-transparent

5.4 Receive buffer

Receive buffer of the module

The receive buffer of the module may have a size of up to 8 KB, depending on the communication module used (see chapter Overview (Page 13)). The maximum frame length always equals half the length of the receive buffer. This means that each communication module is capable of buffering at least two frames.

During parameter assignment, you can specify whether to clear the receive buffer at startup and whether to prevent overwriting of data in the receive buffer. You can also specify the range of values (1 to 255) for the number of buffered receive frames. (See Receive configuration record - data record 60 (Page 49) bytes 62 to 64)

The receive buffer on the communication module is organized as FIFO:

- If several frames are entered in the receive buffer of the communication module, the communication module always sends the oldest frame to the CPU.
- If you always want to transfer the last frame received to the CPU, you must set the value "1" for the number of buffered frames and deactivate overwrite protection.

Note

If continuous reading of the receive data in the user program is interrupted for a certain time and the receive data is then requested again, the communication module will possibly transmit an older frame before the CPU receives the most recent one. At the time of interruption, the old frame had already been transferred from the receive buffer of the communication module and prepared for transmission to the CPU.

5.5 RS232 mode

The following communication modules support RS232 mode:

- CM PtP RS232 BA
- CM PtP RS232 HF
- CM PtP (ET 200SP)

In RS232 mode, data is sent via two lines. A separate line is available for the send direction and the receive direction. Simultaneous sending and receiving is possible (full duplex).

RS232 secondary signals

In addition to the TXD (transmit data), RXD (receive data) and GND signals, the communication module provides the following RS232 secondary signals when using RS232 hardware:

DCD	(input)	Data Carrier detect; Data carrier signal detected
DTR	(output)	Data terminal ready; Module ready for operation
DSR	(input)	Data set ready; Communication partner ready for operation
RTS	(output)	Request to send; Module is ready to send
CTS	(input)	Clear to send; Communication partner can receive data from the module (response to RTS = ON of the communication module)
RI	(input)	Ring Indicator; Ring indicator

After power on of the communication module, the output signals are in the OFF state (inactive).

You can parameterize the operation of the DTR/DSR and RTS/CTS control signals in the configuration data record of the module (Port configuration record - data record 57 (Page 44)).

The RS232 secondary signals cannot be manipulated by data record communication (Set RS232 secondary signals - data record 53 (Page 72)) in case of:

- parameterized data flow control "Hardware RTS always ON"

  (corresponds to automatic operation of the secondary signals)
- parameterized data flow control "Hardware RTS always ON"

  (corresponds to hardware flow control with RTS/CTS)
- parameterized data flow control "Hardware RTS always ON, ignore DTR/DSR "

An error message is generated if an attempt is made to manipulate RS232 secondary signals by means of data record communication.

For more information, refer to chapter Handshaking (Page 36).

5.6 RS422 mode

The following communication modules support RS422 mode:

- CM PtP RS422/485 BA
- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

In RS422 mode, data is transmitted via two line pairs (four-wire mode). A separate line pair is available for the send direction and the receive direction. Simultaneous sending and receiving is possible (full duplex).

The data can be exchanged simultaneously between one or several communication partners. In RS422 multipoint mode, only one multipoint slave may send data at any given time.

Interface operating modes

The following table is a summary of the interface operating modes for the various communication modules and protocols.

The communication module can be used in the following topologies in RS422 mode:

- Link between two nodes: Point-to-point connection
- Link between several nodes: Multipoint coupling
  (only available with CM PtP (ET 200SP))

Operating mode	Description
Full duplex (RS 422) four-wire mode (point-to-point connection)	Both devices have the same priority in this operating mode.
Full duplex (RS422) four-wire mode (multipoint master)	The communication module can be used as multipoint master.
Full duplex (RS 422) four-wire mode (multipoint slave)	The communication module can be used as multipoint slave.

In the case of a multipoint master/slave topology in the RS422 mode:

- The master's sender is connected to the receivers of all slaves.
- The slaves' senders are connected to the master's receiver.
- The receiver of the master and the receiver of one slave are assigned default settings of the receive line (refer to chapter Port configuration record - data record 57 (Page 44) byte 25). All other slaves operate without default settings.

5.7 RS485 mode

The following communication modules support RS485 mode:

- CM PtP RS422/485 BA
- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

In RS485 mode, data is transferred via one line pair (two-wire mode). The line pair is available alternately for the send and receive directions. It is possible to either send or receive (half duplex). On completion of a send operation, operation is immediately switched to receive mode (ready to receive). Send mode is reset again as soon as a new send job is received.

RS485 mode

The following table is a summary of the interface operating modes for the various communication modules and protocols.

Operating mode	Description
Half duplex (RS485) two-wire mode	Operating mode for point-to-point connection or multipoint connection (multipoint) in two-wire mode. The communication module can be the master as well as the slave.

When you operate the Freeport driver in RS485 mode (half duplex, two-wire mode), you must make provisions in the user program to ensure that only one device sends data at any given time. If more than one device sends data at the same time, the frame is corrupted.

Changeover times for RS485 communication module in half duplex mode

A maximum time of 0.1 ms is set for the changeover between sending and receiving.

5.8 Handshaking

Introduction

Handshaking controls the data flow between two communication partners. The use of the handshaking method prevents data loss during transmission if the devices are operating at different speeds.

We can basically distinguish between the following methods:

Table 5- 1 Overview of methods and interfaces

Method	RS232	RS422	RS485
Software data flow control XON/XOFF	X	X	-
Hardware data flow control (RTS/CTS)	X	-	-
Automatic operation of secondary signals	X	-	-

Software data flow control

Software data flow control is implemented as follows on the communication module:

- **XON/XOFF**
  - As soon as the communication module has been set to the "XON/XOFF" operating mode by means of parameter assignment, it sends the XON character, thereby allowing the communication partner to send data.
  - On reaching the configured maximum number of frames or 16 characters ahead of receive buffer overflow, the communication module sends the XOFF character, thereby requesting that the communication partner stop sending. If the communication partner nonetheless continues to send data, an error message is generated if the receive buffer overflows. Data received in the last frame is discarded.
  - As soon as a frame has been fetched by the CPU and the receive buffer is ready to receive data again, the communication module sends the XON character.
  - If the communication module receives the XOFF character during sending, it cancels the current send operation until it receives a XON again from its communication partner. If no XON is received within a specific configurable time, send operation is canceled and a corresponding error message is output.

Note

You can configure the characters for XON and XOFF (any ASCII character).

During parameter assignment of the XON/XOFF software data flow control, user data may not contain any of the configured XON or XOFF characters.

Hardware data flow control

Note

The DTR/DSR signals do not have to be wired for "Hardware RTS always ON, ignore DTR/DSR" parameter assignment.

If "Hardware RTS always ON" is configured, it is imperative that you fully wire the interface signals used. Make sure that the local RTS (out) is connected with the CTS (in) of the communication partner and the local CTS is connected with the RTS of the communication partner. Accordingly, the local DTR must be connected with the DSR of the communication partner and the local DSR with the DTR of the communication partner.

- **Hardware RTS always ON, ignore DTR/DSR**
  - As soon as the communication module has been set to an operating mode with "Hardware RTS always ON" through parameter assignment, it outputs the RTS = ON signal to the communication partner to indicate its ready state.
  - RTS is set to OFF as soon as the configured maximum number of frames or 16 characters before buffer overflow is reached.
    If the communication partner nonetheless continues to send data, an error message is generated on overflow of the receive buffer. Data received in the last frame is discarded.
  - RTS is reset to ON as soon as the frame has been fetched by the CPU and the receive buffer is ready to receive data again.
  - If CTS switches to OFF during the send operation, the communication module interrupts the send operation until CTS is reset to ON. If CTS is not reset to ON within a specific configurable time, the send operation is canceled and a corresponding error message is output.
- **Hardware RTS always ON**

  The "Hardware RTS always ON" mode corresponds to the "Hardware RTS always ON, ignore DTR/DSR" mode. However, you also need to wire DTR and DSR.
  - As soon as the communication module has been set set to an operating mode with "Hardware RTS always ON" through parameter assignment, it sets DTR = ON and RTS = ON to signal its general ready state to the communication partner.
  - RTS is set to OFF as soon as the configured maximum number of frames or 16 characters before buffer overflow is reached.
    If the communication partner nonetheless continues to send data, an error message is generated on overflow of the receive buffer. Data received in the last frame is discarded.
  - RTS is reset to ON as soon as the frame has been fetched by the CPU and the receive buffer is ready to receive data again.
  - If CTS switches to OFF during the send operation, the communication module interrupts the send operation until CTS is reset to ON. If CTS is not reset to ON within a specific configurable time, the send operation is canceled and a corresponding error message is output.
  - A switch from DSR = ON to DSR = OFF cancels an active send job and triggers an error message.

Automatic operation of secondary signals

- **Hardware RTS always switched**

  "Hardware RTS always switched" is implemented as follows on the communication module:

  – As soon as the communication module is set to the operating mode with "Hardware RTS always switched" through parameter assignment, it sets the line RTS to OFF and DTR to ON (communication module ready for operation).

    It is not possible to send frames until the DSR line is set to ON. No data is sent via the RS232C interface as long as DSR is set to OFF. A send job is canceled and a corresponding error message is generated.

  – If a send job is pending, RTS is set to ON and the configured data output wait time starts. On expiration of the data output time, the system checks whether the communication partner has set CTS to ON. If so, the data is sent via the RS232 interface.

  – If the CTS line is not set to ON within the data output time, or changes to OFF during the send operation, the send job is aborted and an error message generated.

  – Once the data has been sent and the configured clear RTS time has elapsed, the RTS line is set to OFF. The system does not wait for CTS to change to OFF.

  – It is always possible to receive data via the RS232 interface. There will be no reaction if there is a danger of the receive buffer of the communication module overflowing.

  – A switch from DSR = ON to DSR = OFF cancels an active send job and triggers an error message.

  Note

  Set the "data output wait time" in such a way that the communication partner is able to enter the ready to receive state before the time elapses.

  Set the "Clear RTS time" in such a way that the communication partner is able to receive the last characters of the frame completely before RTS is set to OFF and the send request is canceled.

Additional information

For information on the settings, refer to chapter Port configuration record - data record 57 (Page 52).

Note

Operation of DTR/DSR or RTS/CTS is accepted by the communication module with the following settings. Operation by means of Set RS232 secondary signals - data record 53 (Page 72) is impossible (any attempt made will trigger an error message):

- Hardware RTS always ON, ignore DTR/DSR
- Hardware RTS always ON
- Hardware RTS always switched

5.9 Communication via DMX512

You can use the ET 200SP CM PtP (from firmware version V1.0.5) communication module for communication via DMX512 (Digital Multiplex). For communication via DMX512, the use of the performance optimization option (Page 25) is also possible, provided you use the max. value 29_D as the highest address.

You can find more information on setting up a DMX512 connection in the FAQ with the entry ID 109778975 (https://support.industry.siemens.com/cs/ww/en/view/109778975) in Siemens Industry Online Support.

Data transmission with 3964(R) 6

6.1 Data transmission with 3964(R) protocol

Introduction

The 3964(R) protocol controls PtP data exchange between the communication module and a communication partner. The 3964(R) protocol contains both the physical layer (layer 1) and the link layer (layer 2).

6.2 Control characters

Introduction

During data transmission, the 3964(R) protocol adds control characters to the information data (link layer). The communication partner can use these control characters to check whether it has received all data completely and without errors.

Control characters of the 3964(R) protocol

The 3964(R) protocol evaluates the following control characters:

- **STX** Start of Text; start of the string to be sent
- **DLE** Data Link Escape
- **ETX** End of Text; end of the string to be sent
- **BCC** Block Check Character (only with 3964R)
- **NAK** Negative Acknowledge; negative response

Note

If the DLE character is transferred as an information character, it is sent twice during connection setup and shutdown (DLE duplication) in order to distinguish it from the DLE control character. The receiver reverses the DLE duplication.

Priority

For operation with 3964(R) protocol, one communication partner must be assigned a higher priority and one communication partner a lower priority. If both partners simultaneously start the connection setup, the partner having lower priority will postpone its send job.

6.3 Block checksum

Block checksum

With the 3964R transfer protocol, data security is enhanced by sending an additional block check character (BCC = Block Check Character).

The block checksum represents the even longitudinal parity (EXOR logic operation of all data bytes) of a sent or received block. Its calculation begins with the first byte of user data (first byte of the frame) after the connection establishment, and ends after the DLE ETX character at connection termination.

Note

With DLE duplication, the DLE character is included twice in the BCC calculation.

6.4 Sending data with 3964(R)

Connection setup for sending

The 3964(R) protocol sends the STX control character to set up the connection. The transition to send mode is initiated when the communication partner responds with the DLE character before the ACK delay time has expired.

The protocol retries to connect if the communication partner responds with NAK or any other character (except DLE or STX), or if inactivity triggers an acknowledgment timeout. After the parameterized number of unsuccessful retries to connect, the connection setup is canceled and the NAK character is transmitted to the communication partner. The communication module outputs a corresponding error message.

Sending data

On successful connection setup, the user data from the output buffer of the communication module and the selected transmission parameters are sent to the communication partner. The partner monitors the time intervals between incoming characters. The interval between two characters must not exceed the character delay time. Monitoring of the character delay time starts immediately after the connection was set up.

If the communication partner transmits an NAK while a transmission is busy, the block transmission is aborted and repeated as shown above starting with the connection setup. If any other character is received, a wait time for expiration of the character delay time is triggered and an NAK is then transmitted to set the communication partner to idle state. Transmission is then restarted with STX connection setup.

Connection shutdown during transmission

After the buffer content has been transmitted, the routine inserts the DLE, ETX characters, appends the BCC block checksum (only for 3964R) and waits for an ACK character. If the communication partner sends the DLE characters within the acknowledgment delay time, the data block has been received without errors. If the communication partner responds with NAK, or any other character (except DLE), or with a corrupt character, or if the acknowledgment delay time expires without response, the routine restarts transmission with STX connection setup.

After the parameterized number of retries to connect, the routine is aborted and NAK is transmitted to the communication partner. The communication module outputs a corresponding error message.

6.5 Receiving data with 3964(R)

Connection setup for receiving

While in idle state and not processing a send job, the module waits for the communication partner to set up the connection.

A wait time is started (wait time = acknowledgment delay time - 10 ms, however, maximum of 400 ms) if no free receive buffer is available during the connection setup with STX. An error message is generated if no free receive buffer is available on expiration of this time. An NAK is transmitted and initiates the transition to idle state. Otherwise, the DLE character is transmitted and the data is received as described above.

The acknowledgment delay time should be set to the same value at both communication partners.

If any character (except for STX or NAK) is received in idle state, the routine waits for expiration of the character delay time (CDT) and then transmits the NAK character. The communication module outputs a corresponding error message.

Receiving data

After successful connection setup, the incoming receive characters are saved to the receive buffer. If two consecutive DLE characters are received, only one of these is saved to the receive buffer.

After connection setup and after each receive character, the routine waits for the next character within the character delay time. If this period expires before another character is received, an NAK is sent to the communication partner. The communication module outputs a corresponding error message. A retry is then expected.

If transfer errors occur during receiving (lost character, frame error, parity error, etc.), the procedure continues to receive data until the connection is terminated and then sends an NAK to the communication partner. A retry is then expected. Receiving is aborted if the block cannot be received without errors after the specified number of transmission retries, or if the communication partner fails to retry transmission within a block delay time of 4 seconds. The communication module reports the first corrupted transfer and the final cancelation.

Connection setup for receiving

Once the DLE ETX string is detected, receiving is terminated and DLE is transmitted to the communication partner for a block that was received without error. In the case of a receive error, an NAK is sent to the communication partner. A retry is then expected.

Receiving is stopped once the DLE ETX BCC string is detected. The BCC is compared with the internally generated longitudinal parity. DLE is transmitted (return to idle state) if the BCC is correct and no other receive error is pending. If the BCC is faulty or a different receive error occurs, an NAK is sent to the communication partner. A retry is then expected.

Configuration data records

7

7.1 Freeport communication

The following three configuration data records must be sent to the communication module for Freeport communication.

7.1.1 Port configuration record - data record 57

Note

Parameter assignment error

If the communication module detects a parameter assignment error when writing the data record 57, the data record is rejected with error status 80E1H.

The corresponding cause of error can be read by means of data record "Read Error Status - data record 55" (error codes of the port configuration).

Port configuration: PORT_CFG (Write Record 0039H)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 0039H – PORT_CFG							
1								
2	Block length: 0018H							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00H							
6	Reserved[1]							
7								
°	Protocol:							
8	0000H – Freeport protocol 0001H – 3964 protocol 0002H through FFFFH – Reserved							
9								

Port configuration: PORT_CFG (Write Record 0039H)			
◦	Data transmission rate:		
10	0000H - Reserved	0007H - 19.2 kbit/s	
11	0001H - 300 bit/s	0008H - 38.4 kbit/s	
	0002H - 600 bit/s	0009H - 57.6 kbit/s	
	0003H - 1.2 kbit/s	000AH - 76.8 kbit/s	
	0004H - 2.4 kbit/s	000BH - 115.2 kbit/s	
	0005H - 4.8 kbit/s	000CH – 250 kbit/s[2]	
	0006H - 9.6 kbit/s	000DH through FFFFH – Reserved	
◦	Parity:		
12	0000H - Reserved		
13	0001H - None		
	0002H - Even		
	0003H - Odd		
	0004H - Mark (set parity bit to 1)		
	0005H - Space (set parity bit to 0)		
	0006H - Any (The parity is not checked during receiving. The parity is always set to "0" during sending.)		
	0007H through FFFFH - Reserved		
◦	Data bits:		
14	0000H - Reserved		
15	0001H - 8 data bits per character		
	0002H - 7 data bits per character		
	0003H through FFFFH - Reserved		
◦	Stop bits:		
16	0000H - Reserved		
17	0001H - 1 stop bit		
	0002H - 2 stop bits		
	0003H through FFFFH - Reserved		
◦	Data flow control:		
18	RS232 settings:	RS485 settings:	RS422 settings:
19	0000H - Reserved	0000H - Reserved	0000H - Reserved
	0001H - None	0001H - None	0001H - None
	0002H - XON/XOFF	0002H through FFFFH - Reserved	0002H - XON/XOFF
	0003H - Hardware RTS always ON, DSR required		0003H through FFFFH - Reserved
	0004H - Hardware RTS always switched, DSR required		
	0005H - Hardware RTS always ON, ignore DTR/DSR		
	0006H through FFFFH - Reserved		
20	XON: 00H through FFH - XON character		
21	XOFF: 00H through FFH - XOFF character		
22	Wait for XON after XOFF in ms: (0000H through FFFFH)		
23			

Port configuration: PORT_CFG (Write Record 0039H)	
24	Operating mode:
	00H - RS232 01H - RS422 full duplex four-wire mode (point-to-point) 02H - RS422 full duplex (RS422) four-wire mode (multipoint master) (CM PtP (ET 200SP)) 03H - Full duplex (RS422) four-wire mode (multipoint slave) (CM PtP (ET 200SP)) 04H - Half duplex (RS485) two-wire operation 05H through FFH - Reserved
25	Receive line default:
	00H - "No" default setting 01H -Signal R(A)=5 V, signal R(B)=0 V (break detection): Break detection is possible with this default setting. Can only be selected with: "Full duplex (RS 422) four-wire mode (point-to-point connection)" and "Full duplex (RS 422) four-wire mode (multipoint slave)". 02H - Signal R(A)=0 V, Signal R(B)=5 V: This default corresponds to the idle state (no active send operation). No break detection is possible with this default setting. 03H through FFH - Reserved
26	Break detection:
	00H - Break detection deactivated 01H - Break detection activated 02H through FFH - Reserved
27	Reserved[1]

[1] Reserved bits must be set to 0.

[2] Available as of firmware version V1.0.5 of the communication module and only for RS485 mode

7.1.2 Send configuration record - data record 59

Note

Parameter assignment error

If the communication module detects a parameter assignment error when writing the data record 59, the data record is rejected with error status 80E1H.

The corresponding cause of error can be read by means of data record "Read Error Status - data record 55" (error codes of the send configuration).

Send configuration: SEND_CFG (Write Record 003BH)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type 003BH – SEND_CFG							
1								
2	Block length: 001CH							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00H							
6	Reserved[1]							
7								
8	RTS ON delay in ms: (0000H through FFFFH)							
9	(not available for RS422/485 modules)							
10	RTS OFF delay in ms: (0000H through FFFFH)							
11	(not available for RS422/485 modules)							
12	Break duration in bit times: (0000H through FFFFH)							
13								
14	Idle Line duration in bit times: (0000H through FFFFH)							
15								
16	Reserved[1]							
17								
°	Number of end delimiters: USR_END							
18	02H							
19	Number of end delimiters (0, 1 or 2)							
20	First end delimiter (for number of end delimiters = 1 or 2)							
21	Second end delimiter (for number of end delimiters = 2)							

Send configuration: SEND_CFG (Write Record 003B$_H$)	
°	Number of appended characters: APP_END
22	05$_H$
23	Number of appended characters (0, 1, 2, 3, 4 or 5)
24	First appended character (for number of appended characters = 1, 2, 3, 4 or 5)
25	Second appended character (for number of appended characters = 2, 3, 4 or 5)
26	Third appended character (for number of appended characters = 3, 4 or 5)
27	Fourth appended character (for number of appended characters = 4 or 5)
28	Fifth appended character (for number of appended characters = 5)
29	Reserved[1]
30	
31	

[1] Reserved bits must be set to 0.

7.1.3 Receive configuration record - data record 60

Note

Parameter assignment error

If the communication module detects a parameter assignment error when writing the data record 60, the data record is rejected with error status 80E1H.

The corresponding cause of error can be read by means of data record "Read Error Status - data record 55" (error codes of the receive configuration).

Note

End sequence #1 character check (bytes 56 to 61) - Note sequence

If **one** end delimiter is to be used, the entry **must** be made in "Character #5" (byte 61) and only this character must be activated in "Character #5" (byte 56).

If **two** end delimiters are to be used, the entry **must** be made in "Character #5" (byte 61) and in "Character #4" (byte 60) and only these characters must be activated in "Character #5" and "Character #4" (byte 56).

The same applies to the use of additional characters.

Receive configuration: RCV_CFG (Write Record 003CH)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 003CH – RCV_CFG							
1								
2	Block length: 0040H							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00H							
6	Reserved[1]							
7								
°	Start conditions:							
8	Reserved[1]							
9	Sequence #4[2]	Sequence #3[2]	Sequence #2[2]	Sequence #1[2]	Idle Line[2]	Break detection[2]	Any character[2]	Start character[2]
10	Idle Line duration in bit times: 0000H through FFFFH							
11								
12	Start characters: 00H through FFH							

Receive configuration: RCV_CFG (Write Record 003C$_H$)						
°	Frame start sequence #1 character check					
13	Reserved[1]	Character #5[2]	Character #4[2]	Character #3[2]	Character #2[2]	Character #1[2]
14	Character #1: 00$_H$ through FF$_H$					
15	Character #2: 00$_H$ through FF$_H$					
16	Character #3: 00$_H$ through FF$_H$					
17	Character #4: 00$_H$ through FF$_H$					
18	Character #5: 00$_H$ through FF$_H$					
°	Frame start sequence #2 character check					
19	Reserved[1]	Character #5[2]	Character #4[2]	Character #3[2]	Character #2[2]	Character #1[2]
20	Character #1: 00$_H$ through FF$_H$					
21	Character #2: 00$_H$ through FF$_H$					
22	Character #3: 00$_H$ through FF$_H$					
23	Character #4: 00$_H$ through FF$_H$					
24	Character #5: 00$_H$ through FF$_H$					
°	Frame start sequence #3 character check					
25	Reserved[1]	Character #5[2]	Character #4[2]	Character #3[2]	Character #2[2]	Character #1[2]
26	Character #1: 00$_H$ through FF$_H$					
27	Character #2: 00$_H$ through FF$_H$					
28	Character #3: 00$_H$ through FF$_H$					
29	Character #4: 00$_H$ through FF$_H$					
30	Character #5: 00$_H$ through FF$_H$					
°	Frame start sequence #4 character check					
31	Reserved[1]	Character #5[2]	Character #4[2]	Character #3[2]	Character #2[2]	Character #1[2]
32	Character #1: 00$_H$ through FF$_H$					
33	Character #2: 00$_H$ through FF$_H$					
34	Character #3: 00$_H$ through FF$_H$					
35	Character #4: 00$_H$ through FF$_H$					
36	Character #5: 00$_H$ through FF$_H$					
37	Reserved[1]					

Receive configuration: RCV_CFG (Write Record 003C$_H$)								
°	End conditions:							
38	Reserved[1]							
39	Reserved[1]	Fixed frame length[2]	Sequence #1[2]	Read length[2]	Maximum frame length	Character delay time[2]	Message timeout[2]	Response timeout[2]
40	Fixed frame length: 0001$_H$ through 0xxx$_H$[3]							
41								
42	Maximum frame length: 0001$_H$ through 0xxx$_H$[3]							
43								
44	Offset of the length field in the message: 0000$_H$ through 0xxx$_H$[3]							
45								
46	Size of the length field: 01$_H$, 02$_H$, 04$_H$ - all other values are not permitted							
47								
48	Number of characters not counted in length specification: 00$_H$ through FF$_H$							
49								
50	Response timeout in ms: 0000$_H$ through FFFF$_H$							
51								
52	Message timeout in ms: 0000$_H$ through FFFF$_H$							
53								
54	Character delay time in bit times: 0000$_H$ through FFFF$_H$							
55								
°	End sequence #1 character check							
56	Reserved[1]			Character #5[2]	Character #4[2]	Character #3[2]	Character #2[2]	Character #1[2]
57	Character #1: 00$_H$ through FF$_H$							
58	Character #2: 00$_H$ through FF$_H$							
59	Character #3: 00$_H$ through FF$_H$							
60	Character #4: 00$_H$ through FF$_H$							
61	Character #5: 00$_H$ through FF$_H$							
62	Maximum number of buffered received frames:							
	00$_H$ - Reserved 01$_H$ through FF$_H$ - 1 through 255							
63	Prevent overwriting:							
	00$_H$ - Deactivated (on receive buffer overflow, a new message overwrites the oldest message)01H - Activated (on receive buffer overflow, the new message is discarded) 02$_H$ through FF$_H$ - Reserved							
64	Clear receive buffer at startup:							
	00$_H$ - Deactivated 01$_H$ - Activated 02$_H$ through FF$_H$ - Reserved							
65	Reserved[1]							
66								
67								

1 Reserved bits must be set to 0.

2 You enable the respective parameter by setting the associated bit to 1.

3 Buffer size depends on the communication module used.

7.2 3964 communication

The following two configuration data records must be transferred to the communication module for 3964 communication:

7.2.1 Port configuration record - data record 57

Note

The "Port configuration record" is described in chapter Port configuration record - data record 57 (Page 44).

7.2.2 3964 protocol configuration - data record 61

Note

Parameter assignment error

If the communication module detects a parameter assignment error when writing the data record 61, the data record is rejected with error status 80E1H.

The corresponding cause of error can be read by means of data record "Read Error Status - data record 55" (error codes of the 3964 protocol).

3964 protocol configuration: P3964 (Write Record 003DH)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 003DH – P3964							
1								
2	Block length: 000CH							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00H							
6	Reserved[1]							
7								
8	3964(R) - With block check:							
	00H - Deactivated 01H - Activated 02H through FFH - Reserved							
9	3964(R) priority:							
	00H - Low priority 01H - High priority 02H through FFH - Reserved							

3964 protocol configuration: P3964 (Write Record 003D$_H$)	
°	3964(R) character delay time:
10	0000$_H$ through 0013$_H$ - Reserved
11	0014$_H$ through FFFF$_H$ - 20 to 65535 ms Default: 00DC$_H$ - 220 ms
°	3964(R) acknowledgment delay time:
12	0000$_H$ through 0013$_H$ - Reserved
13	0014$_H$ through FFFF$_H$ - 20 to 65535 ms Default: 07D0$_H$ - 2000 ms
14	3964(R) connection attempts:
	00$_H$ - Reserved 01$_H$ through FF$_H$ - 1 to 255 connection attempts Default: 06$_H$ - 6 connection attempts
15	3964(R) transmission attempts:
	00$_H$ - Reserved 01$_H$ through FF$_H$ - 1 to 255 transmission attempts Default: 06$_H$ - 6 transmission attempts

[1] Reserved bits must be set to 0.

7.3 Activate special function - data record 58

The following data record can be used to activate diagnostic interrupts (write) and check whether diagnostic interrupts are activated (read).

Note

Parameter assignment error

If the communication module detects a parameter assignment error when writing the data record 58, the data record is rejected with error status 80E1$_H$.

The corresponding cause of error can be read by means of data record "Read Error Status - data record 55" (error codes of the special functions).

Activate special function: PtP SET_FTRS (Write Record 003AH)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 003A$_H$ – SET_FTRS							
1								
2	Block length: 0008$_H$							
3								
4	Block version high byte: 01$_H$							
5	Block version low byte: 00$_H$							
6	Reserved[1]							
7								

Activate special function: PtP SET_FTRS (Write Record 003AH)	
8	
9	Diagnostics interrupt:
	00H - Deactivated 01H - Activated 02H through FFH - Reserved
10	Reserved[1]
11	

[1] Reserved bits must be set to 0.

Error status

8

The following data record contains the acknowledgment of the communication module to the writing a configuration data record.

8.1 Read error status - data record 55

Read Error Status: PtP RD_ESTAT (Read Record 0037H)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 0037H – RD_ESTAT							
1								
2	Block length: 0010H							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00							
6	Reserved[1]							
7								
8	Error codes of the special functions							
9								
10	Error codes of the port configuration							
11								
12	Error codes of the send configuration							
13								
14	Error codes of the receive configuration							
15								
16	Error codes of the RS32 secondary signals							
17								
18	Error codes of the 3964 protocol							
19								

[1] Reserved bits must be set to 0.

A list of all error codes is available in chapter Error messages (Page 74).

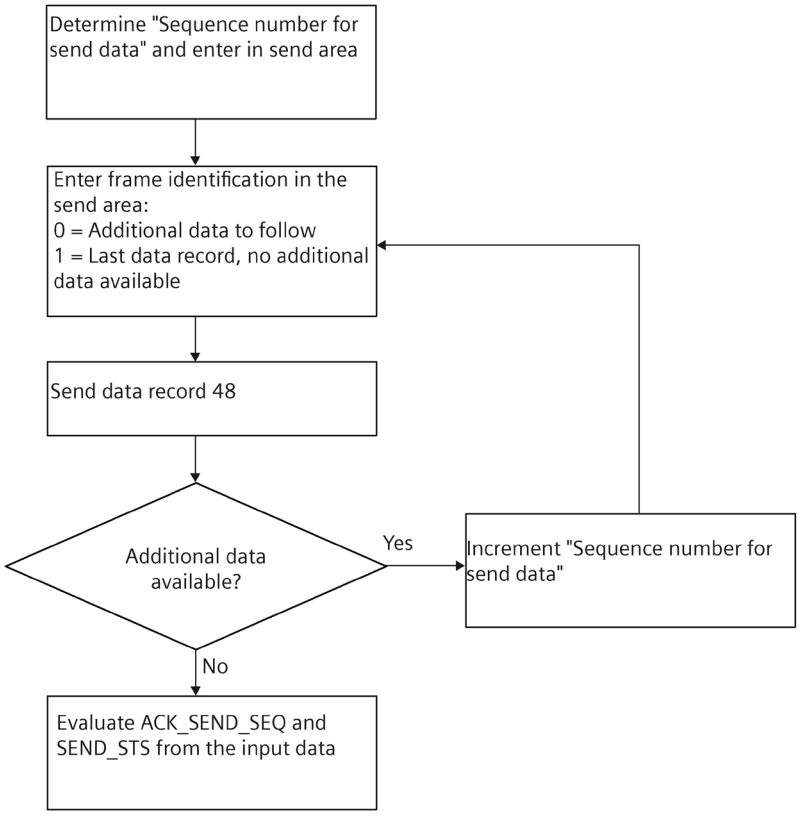
Sending data

9

9.1 Sending data using data record

9.1.1 Flow chart for sending using data record

Users can send data based on the sequence outlined below. Management data and send data is transmitted to the communication module by means of a special data record. The communication module provides the acknowledgment in the input data.



9.1.2 Send data - data record 48

The following data record is used to write send data to the communication module:

Sending data: SEND_PTP Record (Write Record 0030H)								
Byte ↓	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Sequence number for send data (1 to 255)							
1	Frame end identifier (0 or 1)							
2	Data to transmit byte 0							
:	:							
:	:							

Sequence number for send data

The sequence number for send data identifies a new send frame or a new data record of a frame. The acknowledgment by the module in the input data (Page 59) must be read as first sequence number and incremented by the count of one. For the first data record, the sequence number therefore starts with 1 after a STOP-RUN transition of the CPU. It is then incremented by one for each data record with new content.

The value "0" may not be used as sequence number for send data. as it serves to detect the start of the module. The maximum value 255 is therefore followed by 1.

Frame end identifier

The frame end identifier serves to identify the last data record belonging to a frame.

- 0 means: At least one more data record follows for the current frame.
- 1 means: This data record is the last one for the current frame.

Data to transmit

The data must be stored in the data record as of "Data to transmit byte 0". The size and therefore the number of data records for transmitting a frame is defined by the transmission medium:

- PROFINET: 1, 2, or 4 KB, depending on the communication module
- PROFIBUS: 240 bytes

If the length of the data to transmit is not a multiple of the data record length, the length of the last data record corresponds to the length of the remaining data to transmit.

Acknowledgment from module

The module writes the sequence number received in the data record as acknowledgment to the net data (Page 59).

For the last data record belonging to a frame,the acknowledgment is not written until the module has sent the frame.

This acknowledgment (ACK_SEND_SEQ) must therefore be evaluated along with SEND_STS after the last data record was written.

Startup characteristics

During startup, the module writes to the input data "ACK_SEND_SEQ = 0" (Page 59).

A busy job must be aborted when ACK_SEND_SEQ = 0.

9.1.3 Request acknowledgment - Input data

The acknowledgment is entered by the communication module in ACK_SEND_SEQ in the input data. The input data has the following structure:

Byte ↓	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	: :	MODE[1]	: :					
1	ACK_SEND_SEQ							
2	SEND_STS							
3								
4	: :							
5								
6								
7								

[1] This bit is used for the plausibility check:
0B: Universal
1B: Performance option

Before sending a new frame, the feedback of the communication module must be received in the input data.

When the following feedback is received in the input data, the last send frame was sent without errors and the module is ready to accept the next frame:

- ACK_SEND_SEQ = Last "sequence number for send data"
- SEND_STS = 0

When SEND_STS ≠ 0, the last send frame was not sent without errors. In this case, you can find the meaning of the SEND_STS value in the section Error messages (Page 74).

Reaction of the module to an unexpected sequence number

The module rejects a data record with an unexpected sequence number for send data when writing the data record and returns the error message 80B8H. In SEND_STS, the value 81D7H can be found in the input data. In this case, active communication must be canceled and transmission with the first data record (Page 57) must be repeated.

The send request is discarded, the incorrect sequence number is applied (if it is not 0) and incremented from this value.

9.1.4 Example

A frame with a length of 500 bytes is to be transmitted. The data record has a length of 240 bytes.

- First data record, length of 240 bytes:

  "Sequence number for send data" = 1

  "Frame end identifier" = 0

  238 bytes user data

- Second data record, length of 240 bytes:

  "Sequence number for send data" = 2

  "Frame end identifier" = 0

  238 bytes user data

- Third data record, length of 26 bytes

  "Sequence number for send data" = 3

  "Frame end identifier" = 1

  24 bytes user data

- Wait for feedback from the communication module; see Request acknowledgment - Input data (Page 59).

Afterwards, a frame with a length of 100 bytes is to be transmitted.

- First = last data record, length of 102 bytes:

  "Sequence number for send data" = 4

  "Frame end identifier" = 1

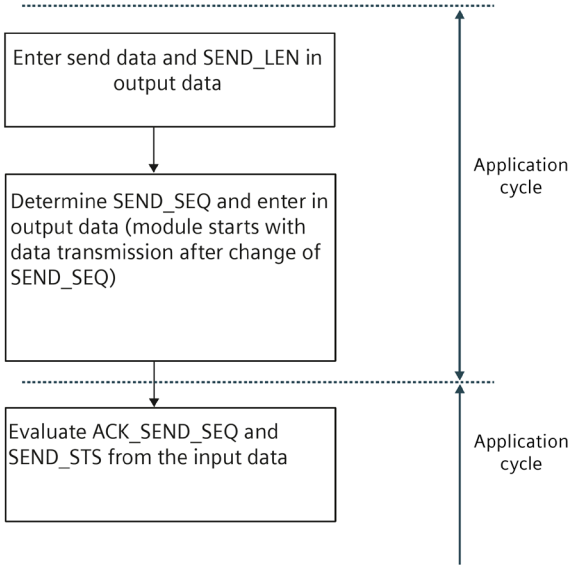
  100 bytes user data

- Wait for feedback from the communication module; see Request acknowledgment - Input data (Page 59).

9.2 Sending data via output data

9.2.1 Flow chart for sending using output data

Users can send data based on the sequence outlined below. Management data and send data is transmitted to the communication module using output data. The communication module provides the acknowledgment in the input data.



9.2.2 Output data when using the Performance option

When using the Performance option, the send data is written to the communication module via the output data.

Output data

The following table shows the assignment of the output data:

Byte ↓	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SEND_SEQ							
1	ACK_RCV_ RECORD	SEND_LEN						
2	Data to transmit byte 0							
3	Data to transmit byte 1							
: :	: :							
30	Data to transmit byte 28							
31	Data to transmit byte 29							

SEND_SEQ

SEND_SEQ is the sequence number for send data and identifies a new send frame. In case of a change of SEND_SEQ, the communication module adopts the SEND_LEN value and the data to be transmitted and starts with the serial data transmission.

The SEND_SEQ value for the first send job after a STOP-RUN transition of the CPU is "1". For each additional send frame, the acknowledgment (Page 63) in the input data must be read and this value must then be incremented for SEND_SEQ by the count of one.

The value "0" may not be used. As it serves to detect the STOP-RUN transition of the CPU. The maximum value 255 is therefore followed by 1.

ACK_RCV_RECORD

After receiving a new sequence number for receive data (RCV_SEQ), the ACK_RCV_RECORD bit must be toggled in the output data to acknowledge receipt of the data. Afterwards, the communication module can once again send new input data.

SEND_LEN

SEND_LEN specifies the length of the data to be transmitted (1 to 30 bytes).

Data to transmit

The data to be transmitted contains the send frame whose length is limited to 30 bytes. The data must be stored as of "Data to transmit byte 0".

Acknowledgment from module

The module writes the received sequence number to ACK_SEND_SEQ in the input data (Page 63) as acknowledgment and confirmation that it is ready to accept the next send frame.

9.2.3 Request acknowledgment - Input data

The acknowledgment is entered by the communication module in ACK_SEND_SEQ in the input data. The input data has the following structure:

Byte ↓	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	: :	MODE[1]	: :					
1	ACK_SEND_SEQ							
2	SEND_STS							
3								
4	: :							

[1] This bit is used for the plausibility check:
0_B: Universal
1_B: Performance option

Before sending a new frame, the feedback of the communication module (acknowledgment) must be received in the input data.

When the following feedback is received, the last send frame was sent without errors and the module is ready to accept the next frame:

- ACK_SEND_SEQ = Last sequence number for SEND_SEQ
- SEND_STS = 0

When SEND_STS ≠ 0, the last send frame was not sent without errors. In this case, you can find the meaning of the SEND_STS value in the section Error messages (Page 74).

Reaction of the module to an early send job

If you do not wait for the send acknowledgment, the module writes the value $81D0_H$ in SEND_STS as long as the current send job is active. The communication module initially postpones the new send job and continues with it as soon as the current send job is completed.

Reaction of the module to an unexpected sequence number

The module rejects an unexpected sequence number for send data. In SEND_STS, the value $81D7_H$ can be found.

The send request is discarded, the incorrect sequence number is applied (if it is not 0) and incremented from this value.

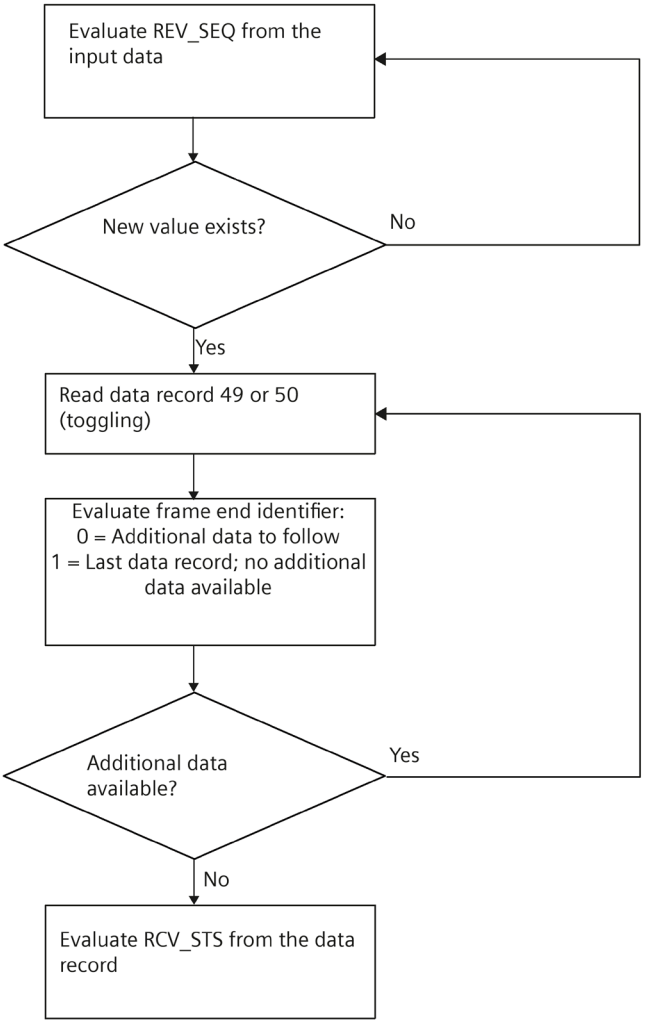
Receiving data

10

10.1 Receiving data using data record

10.1.1 Flow chart for receiving using data record

Users can receive data based on the sequence outlined below. The communication module signals the availability of new received data in the input data. The communication module provides management and receive data in a special data record.



10.1.2 Data received - data record 49/50

A new value for RCV_SEQ in the input data (Page 66) indicates a new frame is ready for reading in the communication module.

The following data records are used to read receive data from the communication module:

Receiving data: RCV_PTP Record (Read Record 0031H/0032H)								
Byte ↓	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Sequence number for this data record (1 to 255)							
1	Frame end identifier (0 or 1)							
2	RCV_STS							
3								
4	Receive data byte 0							
:	:							
:	:							

Change of the data record number when receiving data

After startup of the communication module, reading of the frame content must start with the data record number 31H. You must then toggle the data record number with each read process.

Sequence number of this data record

The "Sequence number of this data record" is equal to RCV_SEQ for the first data record of a frame in the input data (Page 66) that were displayed by the new frame. This value is incremented by the count of one for each data record read subsequently. An overflow to 2 is triggered at 255. The values 0 and 1 have a special significance.

Frame end identifier

The frame end identifier is equal to 1 for the last data record that has to be read for this frame, while it is 0 for all other data records.

The size of the last data record that belongs to a frame is determined by the frame length and limited accordingly by the communication module.

RCV_STS

RCV_STS displays the result of the data reception.

- 009xH: Frame received without error
- 8xxxH: Frame received with error

See chapter Error messages (Page 74).

10.1.3 Query: New receive frame - Input data

Sequence number for receive data

A new received frame is indicated by a sequence number for receive data in byte 4 of the input data.

Byte ↓	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	:	MODE[1]	:					
1	:							
2								
3								
4	RCV_SEQ							
5	:							
6	RCV_STS							
7								

[1] This bit is used for the plausibility check:
0B: Universal
1B: Performance option

RCV_SEQ

RCV_SEQ contains the sequence number of the first data record of the frame currently being received by the module.

The values 0 and 1 have special meanings:

RCV_SEQ = 0:

- The communication module has performed a restart after power on and that no frame has been received yet.

RCV_SEQ = 1:

- This is the first frame after power on of the communication module. Reading of the frame content must start with the data record number 31H.

A busy job must be aborted in both cases.

RCV_STS

In RCV_STS, you can read status information and an error number in case of an error (Page 74).

10.1.4 Example

A frame with a length of 500 bytes was received. The data record has a length of 240 bytes

- In the user data, the frame is indicated by "Sequence number for RCV" = 1.
- You read the data record 0031_H which includes:

  "Sequence number of this data record" = 1

  "Frame end identifier" = 0

  "RCV status or error code" = $009x_H$ 1)

  It contains 236 bytes of user data.
- You read the second data record 0032_H which includes:

  "Sequence number of this data record" = 2

  "Frame end identifier" = 0

  "RCV status or error code" = $009x_H$ 1)

  It contains 236 bytes of user data.
- You read the third data record 0031_H which includes:

  "Sequence number of this data record" = 3

  "Frame end identifier" = 1

  "RCV status or error code" = $009x_H$ 1)

  It contains 28 bytes of user data. The communication module has limited the data record length to 32 bytes.

  This completes reading of the first frame from the CM.

A frame with a length of 100 bytes was received subsequently.

The sequence number for RCV changes from 1 to 4 in the user data.

- You read the data record 0032_H which includes:

  "Sequence number of this data record" = 4

  "Frame end identifier" = 1

  "RCV status or error code" = $009x_H$ 1)

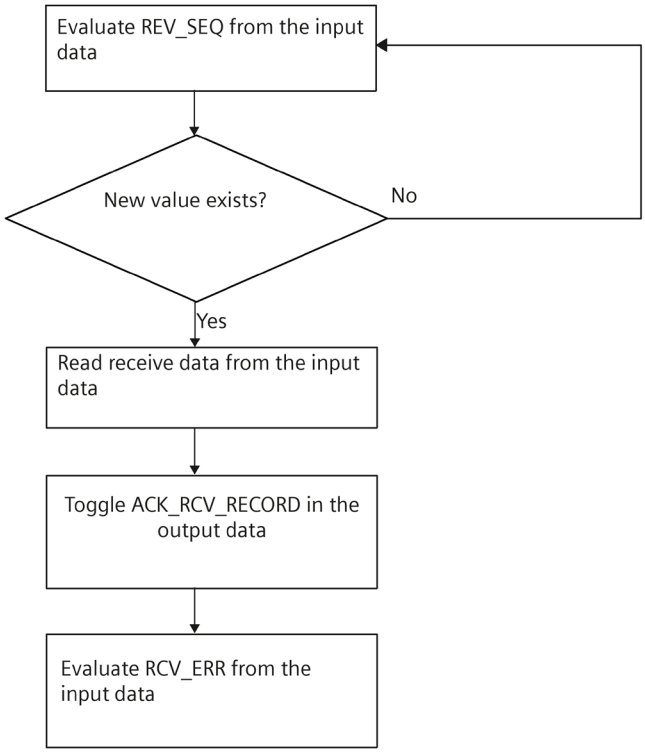
  The data record contains 100 bytes of user data and the communication module has limited the data record length to 104 bytes.

1) 0x009x Frame received without error
   0x8xxx Frame received with error
   (see error codes in chapter Query: New receive frame - Input data (Page 66))

10.2 Receiving data via input data

10.2.1 Flow chart for receiving via input data

Users can receive data based on the sequence outlined below. The communication module provides management and receive data in the input data.



10.2.2 Input data when using the Performance option

When using the Performance option, the receive data is read from the communication module via the input data.

Input data

The following table shows the assignment of the input data:

Byte ↓	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SGN_STAT	MODE	Ring_ Status	DCD_ Status	CTS_ Status	RTS_ Status	DSR_ Status	DTR_ Status
1	ACK_SEND_SEQ							
2	SEND_STS							
3								
4	RCV_SEQ							
5	RCV_ Record	RCV_LEN						
6	RCV_STS							
7								
8	Receive data byte 0							
9	Receive data byte 1							
:	:							
:	:							
30	Receive data byte 22							
31	Receive data byte 23							

SGN_STAT

This bit indicates whether RS232 secondary signals (Page 73) are available in the current operating mode of the communication module.

MODE

This bit is used for the plausibility check of the data exchange type:

MODE = 0: Universal (data communication via data record)

MODE = 1: Performance option (data communication via net data)

ACK_SEND_SEQ

The module acknowledges an executed send job by writing the associated sequence number in ACK_SEND_SEQ.

The module thus confirms that it is ready to execute the next send job.

RCV_SEQ

RCV_SEQ contains the sequence number of the frame currently shown in the input data and received by the module. This value is incremented by the count of one for each frame read subsequently. An overflow to 2 is triggered at 255.

The values 0 and 1 have special meanings:

RCV_SEQ = 0:

- The communication module has performed a restart after power on and that no frame has been received yet.

  or

- The receive buffer was cleared.

RCV_SEQ = 1:

- This is the first frame after power on of the communication module.

  or

- This is the first frame after clearing the receive buffer.

Acknowledge receipt of data

When a new frame with a new sequence number RCV_SEQ was received by the module, the receive data is read out. The ACK_RCV_RECORD bit must then be toggled in the output data to acknowledge receipt of the data. Afterwards, the communication module can write a new frame to the input data again.

RCV_RECORD

RCV_RECORD returns the value that was last received by the communication module in the ACK_RCV_RECORD bit in the output data.

RCV_LEN

RCV_LEN contains the length of the received data (1 to 24 bytes).

SEND_STS

In SEND_STS, you can read status information and an error number in case of an error (Page 74).

RCV_STS

RCV_STS displays the result of the data reception.

- $009x_H$: Frame received without error
- $8xxx_H$: Frame received with error

See chapter Error messages (Page 74).

Clear receive buffer

11

11.1 Clear receive buffer - data record 54

The receive buffer of the communication module is cleared by writing the following data record.

Delete receive buffer: PtP RCV_RST (Write Record 0036H)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 0036H – RCV_RST							
1								
2	Block length: 0008H							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00H							
6	Reserved[1]							
7								
8	0	0	0	0	0	0	0	1
9	Reserved[1]							
10								
11								

[1] Reserved bits must be set to 0.

MODE = Universal:

The communication module does not acknowledge the reset of its receive buffer.

MODE = Performance option:

Clearing of the receive buffer is signaled back with RCV_SEQ = 0.

RS232 signals

12

12.1 Set RS232 secondary signals - data record 53

The RS232 secondary signals of the communication module are manipulated by writing to the following data record, provided the communication module supports this action in the current mode.

Note

Parameter assignment error

If the communication module detects a parameter assignment error when writing the data record 53, the data record is rejected with error status 80E1H.

The corresponding cause of error can be read by means of data record "Read Error Status - data record 55" (error codes of the RS232 secondary signals).

Set RS232 accompanying signals: PtP SGN_SET (Write Record 0035H)								
Bit->	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte								
0	Block type: 0035H - SGN_SET							
1								
2	Block length: 0008H							
3								
4	Block version high byte: 01H							
5	Block version low byte: 00H							
6	Reserved[1]							
7								
8	Reserved[1]					Set_DSR_ State	Set_DTR_ State	Set_RTS_ State
9	Reserved[1]					DSR_State[2]	DTR_State	RTS_State
10	Reserved[1]							
11								

[1] Reserved bits must be set to 0.

[2] Only available with interface type DCE

Note:

Set_xxx_State = 0: the status of the specified control bit is not changed.

Set_xxx_State = 1: the status of the specified control bit must be set to the value that was specified by xxx_State.

For information on reading the acknowledgment to the writing of this data record, refer to chapter "Error status (Page 55)".

12.2 Read RS232 secondary signals - Input data

The RS232 secondary signals of the communication module can be read in the input data.

Byte ↓	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SGN_STAT	Mode	Ring_ Status	DCD_ Status	CTS_ Status	RTS_ Status	DSR_ Status	DTR_ Status
: :	: :							

SGN_STAT

This bit indicates whether RS232 secondary signals are available in the current operating mode of the communication module:

- SGN_STAT = 0: RS232 secondary signals not available
- SGN_STAT = 1: RS232 secondary signals available

Ring_Status

This bit indicates the status of the RS232 secondary signal RI (ring indicator).

DCD_Status

This bit indicates the status of the RS232 secondary signal DCD (data carrier signal detected).

CTS_Status

This bit indicates the status of the RS232 secondary signal CTS (send permission).

RTS_Status

This bit indicates the status of the RS232 secondary signal RTS (send request).

DSR_Status

This bit indicates the status of the RS232 secondary signal DSR (ready state of the communication partner).

DTR_Status

This bit indicates the status of the RS232 secondary signal DTR (ready state of the communication module).

Error messages

13

Overview of the status and error messages

Code	Description	Remedy
0000H	No error	-
RECEIVE status and error codes		
0094H	Frame end identified based on the "Receipt of fixed/maximum frame length"	-
0095H	Frame end identified based on "Message timeout"	-
0096H	Frame end identified based on expiration of the "Character delay time"	-
0097H	No response frame was received within the parameterized maximum response time.	-
0098H	Frame end identified based on the fulfillment of the "Read message length from message" conditions	-
0099H	Frame end identified based on the receipt of the "End sequence"	-
SEND status and error codes		
7001H	Initial call for a new frame: Data record transmission initiated	-
7002H	Interim call: Data record transmission in process	-
8085H	Invalid length specification	Select a suitable frame length. • Communication using data record: Depending on the module, the following are permissible: 1 ... 1024/2048/4096 bytes • Communication using net data: Maximum length is 30 bytes.
8087H	The number of characters received by the module exceeds the number supported for performance optimization.	Select a suitable frame length or use communication via data record. For performance optimization, the maximum length is 24 bytes.
Error codes of the special functions		
8190H	Incorrect setting of the CRC calculation	Select a suitable value for CRC calculation. The following are valid: deactivated or activated. Check whether the module addressed supports CRC calculation.
8191H	Incorrect setting of the diagnostic interrupt	Select a suitable value for "Diagnostic interrupt". The following are valid: Diagnostic interrupt deactivated (0) or diagnostic interrupt activated (1).

Code	Description	Remedy
Error codes of the "Port configuration"		
81A0H	The module does not support this protocol.	Select a valid protocol for the module.
81A1H	The module does not support this data transmission rate.	Select a valid data transmission rate for the module.
81A2H	The module does not support this parity setting.	Select a suitable value for "Parity". The following are valid: • None (1) • Even (2) • Odd (3) • Mark (4) • Space (5) • Any (6)
81A3H	The module does not support this number of data bits.	Select a suitable value for "Number of data bits". The following are valid: • 7 (2) • 8 (1)
81A4H	The module does not support this number of stop bits.	Select a suitable value for "Number of stop bits". The following are valid: • 1 (1) • 2 (2)
81A5H	The module does not support this type of data flow control.	Select a valid data flow control for the module.
81A7H	Invalid value for XON or XOFF	Select suitable values for XON and XOFF. Valid range of values: 0...7F. XON must not be equal to XOFF.
81A8H	Error in frame header	Check the data/values in the frame header.
81AAH	Invalid operating mode	Valid operating modes are: • Full duplex (RS232) (0) • Full duplex (RS422) four-wire mode (point-to-point) (1) • Full duplex (RS422) four-wire mode (multipoint master) (2) / (CM PtP (ET 200SP)) • Full duplex (RS422) four-wire mode (multipoint slave) (3) / (CM PtP (ET 200SP)) • Half duplex (RS485) two-wire mode (4)

Code	Description	Remedy
81ABH	Invalid default setting of the receive line	Valid default settings are: • "No" default setting (0) • Signal R(A)=5 V, signal R(B)=0 V (break detection) (1): Can only be selected with: "Full duplex (RS422) four-wire mode (point-to-point connection)" and "Full duplex (RS422) four-wire mode (multipoint slave)". • Signal R(A)=0 V, signal R(B)=5 V (2): This default corresponds to the idle state (no active send operation). No break detection is possible with this default setting.
81ACH	Invalid value for "Break detection"	Select a suitable value for "Break detection". The following are valid: • "0" (break detection deactivated) • "1" (break detection activated)
81AFH	The module does not support this protocol.	Select a valid protocol for the module.
Error codes of the "Send configuration"		
81B5H	More than two end delimiters or End sequence > 5 characters	Select suitable values for "End delimiter" and "End sequence". The following are valid: • deactivated (0), • 1 (1) or 2 (2) end delimiters or • deactivated (0), • 1 (1) up to 5 (5) characters for the end sequence.
81B6H	Send configuration rejected because the 3964(R) protocol was selected	Verify that no send configuration is transmitted if the 3964R) protocol is set.
Error codes of the "Receive configuration"		
81C0H	Invalid start condition	Select a suitable start condition. The following are valid: • Send break before frame start • Send Idle Line.
81C1H	Invalid end condition or no end condition selected	Select a suitable end condition.
81C3H	Invalid value for "Maximum message length"	Select a suitable value for "Maximum message length". Valid range of values (depending on the module): 1-1024/2048/4096 (Byte)
81C4H	Invalid value for "Offset of the length specification in the message"	Select a suitable value for "Offset of the length specification in the message". Valid range of values (depending on the module): 1-1024/2048/4096 (Byte)

Code	Description	Remedy
81C5H	Invalid value for "Size of length field"	Select a suitable value for "Size of length field". Valid range of values in bytes: • 1 (1) • 2 (2) • 4 (4)
81C6H	Invalid value for "Number of characters not counted in length specification"	Select a suitable value for "Number of characters not counted in length specification". Valid range of values: 0 to 255 (bytes)
81C7H	The total of "Offset in the message + size of the length field + number of character not counted" is greater than the maximum frame length	Select a suitable value for "Offset in message", "Size of the length field" and "Number of characters not counted". Valid range of values: • Offset in the message (depending on the module): 0-1022/2046/4094 (Byte) • Size of the length field: 1, 2, or 4 (bytes) • Number of characters not counted: 0-255 (bytes)
81C8H	Invalid value for "Response timeout"	Select a suitable value for "Response timeout". Valid range of values: 1-65535 (ms)
81C9H	Invalid value for "Character delay time"	Select a suitable value for "Character delay time". Valid range of values: 1-65535 (bit times)
81CBH	Frame end sequence is activated, but no character is activated for the check	Activate one or several characters for the check.
81CCH	Frame start sequence is activated, but no character is activated for the check	Activate one or several characters for the check.
81CDH	Invalid value for "Prevent overwriting"	Select a suitable value for "Prevent overwriting". The following are valid: • Prevent overwriting is deactivated (0) or • Prevent overwriting is activated (1)
81CEH	Invalid value for "Clear receive buffer at startup"	Select a suitable value for "Clear receive buffer at startup". The following are valid: • Clear receive buffer at startup is deactivated (0) • Clear receive buffer at startup is activated (1)
SEND status and error codes		
81D0H	Receiving send requests during runtime of a send job	Make sure that you do not receive send requests during runtime of a send job.
81D1H	The waiting time for XON or CTS = ON has expired.	The communication partner has a fault, is too slow or is offline. Check the communication partner or, if necessary, change the parameter assignment.
81D2H	"Hardware RTS always ON": Send command aborted due to change from DSR = ON to OFF	Check the communication partner. Make sure that DSR is ONfor the entire duration of transmission.
81D3H	Send buffer overflow / send frame too long	Select a shorter frame length. The following are valid (depending on the module): 1-1024/2048/4096 (Byte)
81D5H	Transmission aborted due to parameter changes, detected wire break, or because CPU in STOP	Check the parameter assignment, for wire break, and the CPU status.

Code	Description	Remedy
81D6H	Transmission canceled because end identifier was not received	Check the parameter assignment of the end delimiters and the frame of the communication partner.
81D7H	Communication error between the user program and module	Check communication (e.g. consistency of the sequence number).
81D8H	Transmission attempt rejected because module is not configured	Configure the module.
81DFH	The module has reset the interface for one of the following reasons: • Module was restarted • Module parameters were reassigned • CPU STOP	–
Error codes when receiving messages		
81E0H	Frame canceled: Receive buffer overflow / send frame too long	Increase the call rate for the receive function in the user program or configure communication with data flow control.
81E1H	Frame canceled: Parity error	Check the connection line of the communication partners, or verify that both devices are configured for the same data transmission rate, parity and stop bit number.
81E2H	Frame canceled: Character frame error	Check the settings for the start bit, data bits, parity bit, data transmission rate, and stop bit(s).
81E3H	Frame canceled: Character overflow error	Firmware error: Contact Customer Support.
81E4H	Frame canceled: The total length of "Offset in the message + size of the length field + number of character not counted" is greater than the receive buffer	Select a suitable value for Offset in message, Size of the length field, and Number of characters not counted.
81E5H	Frame canceled: Break	Receive line to partner is interrupted. Reconnect or switch on partner.
81E6H	Maximum number of "Buffered receive frames" exceeded	In the user program, increase the call rate of the instruction, or parameterize communication with data flow control, or increase the number of buffered frames.
81E8H	Frame canceled: The character delay time has expired before the message and criterion was detected	Partner device faulty or too slow. Check this, if required, using an interface tester that is interconnected in the transmission line.
81EBH	Frame canceled: Maximum frame length reached	Select a shorter frame length at the communication partner. The following are valid (depending on the module): 1-1024/2048/4096 (Byte) Check the parameters of the frame end detection.

Code	Description	Remedy
Error codes of RS232 secondary signals		
81F0H	The module does not support RS232 secondary signals	You have made an attempt to set up secondary signals on a module that does not support RS232 secondary signals. Verify that this is an RS232 module or that RS232 mode (ET 200SP) is set.
81F1H	No operation of RS232 secondary signals	RS232 secondary signals cannot be manipulated manually as long as hardware data flow is active.
Error codes of the receive configuration		
82C1H	Invalid value for "Buffered receive frames".	Select a suitable value for "Buffered receive frames". Valid range of values: 1-255
82C2H	Receive configuration rejected because the 3964(R) protocol was selected	Verify that no receive configuration is transmitted if the 3964(R) protocol is set.
3964(R) protocol error codes		
8380H	Parameter assignment error: Invalid value for "Character delay time".	Select a suitable value for "Character delay time". Valid range of values: 1-65535 (ms)
8381H	Parameter assignment error: Invalid value for "Response timeout".	Select a suitable value for "Response timeout". Valid range of values: 1-65535 (ms)
8382H	Parameter assignment error: Invalid value for "Priority".	Select a suitable value for "Priority". The following are valid: • High (1) • Low (0)
8383H	Parameter assignment error: Invalid value for "Block check"	Select a suitable value for "Block check". The following are valid: • with block check (1) • without block check (0)
8384H	Parameter assignment error: Invalid value for "Connection attempts".	Select a suitable value for "Connection attempts". Valid range of values: 1-255
8385H	Parameter assignment error: Invalid value for "Transmission attempts".	Select a suitable value for "Transmission attempts". Valid range of values: 1-255
8386H	Runtime error: Number of connection attempts exceeded	Check the interface cable and the transmission parameters. Also check whether the receive function is configured correctly at the partner device.
8387H	Runtime error: Number of transmission attempts exceeded	Check the interface cable, the transmission parameters and the configuration of the communication partner.
8388H	Runtime error: Error at the "Block check character" The internally calculated value of the block check character does not correspond to the block check character received by the partner at the connection end.	Check whether the connection is seriously disrupted; in this case you may also occasionally see error codes. Check for proper function at the partner device, possibly by using an interface test device that is switched into the transmission line.

Code	Description	Remedy
8389H	Runtime error: Invalid character received while waiting for free receive buffer	The send request of the communication partner (STX, 02H) is not answered with DLE until the receive buffer has been cleared. No additional character may be received before it has been cleared (except STX again). Check for proper function at the partner device, possibly by using an interface test device that is switched into the transmission line.
838AH	Runtime error: Logical error during receiving. Another character (other than DLE, ETX) was received after receipt of DLE.	Check whether the partner DLE in the frame header and in the data string is always in duplicate or the connection is released with DLE ETX. Check for proper function of the partner device, possibly using an interface test device that is switched into the transmission line.
838BH	Runtime error: Character delay time exceeded	Partner device too slow or faulty. Check for proper function of the partner device, possibly using an interface test device that is switched into the transmission line.
838CH	Runtime error: Wait time for free receive buffer has started	Increase the call rate of the instruction in the user program, or parameterize communication with data flow control.
838DH	Runtime error: Frame repetition does not start within 4 s after NAK	Check the communication partner. A received frame that is possibly corrupted must be repeated by the partner within 4 seconds.
838EH	Runtime error: One or several characters (except NAK and STX) were received in idle mode.	Verify the proper function of the partner device, for example, by means an interface tester that is interconnected in the transmission line.
838FH	Runtime error: Initialization conflict - Both partners have set high priority	Set the "Low" priority at one of the partners
8390H	Parameter assignment error: Error in frame header	Check the data/values in the frame header.
8391H	Parameter assignment error: 3964 Parameter data record rejected because Freeport is set	Verify that no 3964 parameter data is transmitted if the Freeport protocol is set.

Glossary

Address

The address indicates the physical storage space and enables direct access to the operand that is stored under this address.

Automation system

An automation system is a programmable logic controller consisting of at least one CPU, various I/O modules, and operator control and monitoring devices.

Bit times

"Bit times" are always specified as a number of bits.

The "time" set with bits depends on the selected data transmission rate that is taken into consideration automatically.

Example:

The frame end is to be detected after a gap of two characters.

The set data transmission rate is 9600 bit/s.

The set character frame is 10 bits.

10 x 2 = 20 bit times

This corresponds to a time of:

20 x 1/9600 ≈ 0.0021 s

Communication module

Communication modules are used for point-to-point connections and bus links.

Configuration

The term configuration refers to the summary of individual modules of an automation system in the configuration table.

CPU

Central Processing Unit = Central module of the automation system that consists of the control and computing units, memory, system program, and interfaces to the I/O modules.

CPU operating system

The CPU operating system organizes all the functions and sequences of the CPU that are not linked to a special control task.

Cycle time

The cycle time denotes the time a CPU requires to execute one user program cycle.

Cyclic program processing

In cyclic program processing, the term cycle denotes execution of the user program in a continuous repetitive program loop.

Default setting

The default is a useful basic setting that is used whenever no other value is specified.

Diagnostic functions

The diagnostic functions cover the entire system diagnostics and the detection, evaluation and reporting of errors in the automation system.

Diagnostics buffer

Memory area in which detailed information on all diagnostics events is entered based on the order of their occurrence.

Diagnostics events

Diagnostics events are, for example, module errors or system errors in the CPU that may be caused by a program error.

Hardware

The term hardware denotes the entire physical and technical equipment of an automation system.

Interrupt

The term interrupt designates the interruption of program execution in the processor of an automation system which is triggered by an external alarm.

Module parameters

Module parameters are values that are used to determine the behavior of the module. We distinguish between static and dynamic module parameters.

Online/Offline

In online mode, there is a data connection between the automation system and programming device, while there is no data connection in offline mode.

Parameter assignment

The term parameter assignment denotes actions that define the response of a module or block.

Parameters

Parameters represent values that can be used to define the behavior of a module or software. We distinguish between module parameters and block parameters.

Performance option

From firmware version V2.0 of the communication module, the option for performance optimization is available. This option is suitable if you are exclusively sending and receiving short frames with several communication modules. The frame length is limited to 24 bytes for receive frames and 30 bytes for send frames. The reaction time is optimized, especially when several CM PtPs are used in parallel.

Point-to-point connection

In point-to-point connection, the communication module forms the interface between an automation system and a communication partner.

Protocol

The communication partners participating in data transmission must conform to fixed rules for the handling and execution of data traffic. These rules are called protocols.

Rack

The rack is a mounting rail with slots that can be used to mount modules.

Software

The term software refers to all programs that are used on a computing system. These include the operating system and user programs.

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, namely the blocks.

Work memory

Work memory is CPU RAM (Random Access Memory) that is accessed by the processor when processing the user program.

Index

3

3964 protocol configuration - data record 61, 52
3964(R)
  Receiving data, 43
  Sending data, 41
3964(R) protocol, 40
  Control characters, 40
  Priority, 40
3964R procedure
  Block check character, 41

A

Acyclic data communication, 21
Application options for the communication modules, 14
Article numbers, 13
Automatic operation of secondary signals, 38

B

BCC, 40
Bidirectional data traffic, 17
Block check character, 41

C

Clear receive buffer - data record 54, 71
Code transparency, 32
Configuration data records, 23
CTS, 33
Cyclic data communication, 21

D

Data flow control, 14, 36
Data record 48, 57
Data record 49/50, 65
Data record 53, 72
Data record 54, 71
Data record 55, 55
Data record 57, 44
Data record 58, 53
Data record 59, 47
Data record 60, 49
Data record 61, 52
Data transmission rates, 14
DCD, 33
DLE, 40
DMX512, 39
DSR, 33
DTR, 33

E

ETX, 40

F

Freeport
  Code transparency, 32
  End criteria, 29
  Message end, 26
  Message start, 26
  Receive buffer, 32
  Receiving data, 28
  RS232 secondary signals, 33
  Sending data, 26
  Start criteria, 28
Full duplex mode, 17

H

Half duplex mode, 17
Handshaking, 36
Hardware data flow control, 37
Hardware RTS always ON, 37
Hardware RTS always ON, ignore DTR/DSR, 37
Hardware RTS always switched, 38

I

Interfaces, 13

N

NAK, 40

P

Performance optimization, 25
Performance option, 21
    Input data, 69
    Output data, 62
Point-to-point connection, 17
Port Configuration Record, 44

Q

Query acknowledgment
    Data record, 59
    Performance option, 63
Query: New receive frame,

R

Read error status - data record 55, 55
Read RS232 secondary signals - input user data, 73
Receive buffer, 32
Receive buffer size, 14
Receive configuration record - data record 60, 49
Receiving using data record
    Flow chart, 64
Receiving using net data
    Flow chart, 68
RI, 33
RS232 mode, 33
RS232 secondary signals, 33
    Automatic use, 38
RS422 mode, 34
RS485 mode, 35
RTS, 33

S

Secondary signals, 14
Send configuration record - data record 59, 47
Send point-to-point - data record 48, 57
Sending using data record
    Flow chart, 56
Sending via user data
    Flow chart, 61
Serial data transmission, 17
Set RS232 secondary signals - data record 53, 72
Software data flow control, 36
STX, 40

T

Transmission security, 19
    with 3964(R), 20
    with Freeport, 20

U

Unidirectional/bidirectional data traffic, 33
Universal, 21

X

XON/XOFF, 36