MANUFACTURER DOCUMENT PACKAGE

Manufacturer's Manual

Selected official pages for procurement and maintenance review

PRODUCT BRAND

Siemens

PRODUCT MODEL

6ES7138-4DF01-0AB0

ET 200S Serial Interface. Source PDF pages: manual pp. 1-2,7,9-11,16-25,31-59,64-73,84-89,98-109; datasheet pp. 1-3. Adjacent model text on shared pages is outside this product reference.

OFFICIAL SOURCE

SIMATIC ET 200S Serial interface modules; 03/2009, A5E00124881-05

Siemens original documentation and product reference

Siemens Data sheet 6ES7138-4DF01-0AB0; 03/10/2017; appended source pp. 1-3

Protocol and CPU STOP precautions remain in the original pages. Source p44 lists 115 baud versus 110 on pp54/58/107; p70 has an FB-name inconsistency. Neither is corrected here. Use full ET 200S station instructions.

AMIKON PRODUCT SUPPORT

+86-18020776786 INFO@AMIKON.CN

AMIKONG.COM

Amikon Limited prepared this independent reference package. Selected manufacturer pages follow with their original content and pagination visible.

SIEMENS

Preface 1

Serial interface module 2

Modbus/USS 3

SIMATIC

ET 200S
Serial interface modules

Operating Instructions

03/2009
A5E00124881-05

Legal information

Warning notice system

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

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

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

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

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

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

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

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

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 adhered to. The information in the relevant documentation must be observed.

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E00124881-05
Ⓟ 03/2009

Copyright © Siemens AG 2009.
Technical data subject to change

Preface

1

How the Manual is Structured

This manual volume is supplementary to the *ET 200S Distributed I/O System* manual.

The *ET 200S Distributed I/O System* manual provides comprehensive information pertaining to the hardware configuration, installation, wiring, commissioning, diagnostics and technical data of the ET 200S distributed I/O system.

This manual provides a description of functions and the technical data of the ET 200S 1SI and ET 200S Modbus/USS series interface modules.

How to Find Your Way Around

At the beginning of each chapter you will find a **Product Overview**, which lists the features and applications of the module described. You will also find the order number of the module and the name and release of the software required. For the current GSD file, go to:

http://support.automation.siemens.com

In each chapter you will then find a section with the heading **Brief Instructions on Commissioning** followed by the name of the relevant module. These brief instructions tell you in a series of short steps how to install and configure the module, how to integrate it in your use program, and how to test it in your user program.

Standards and approvals

For information about standards and approvals, refer to chapter "General technical data" in the *ET 200S Distributed I/O System* manual. This manual is available at:

http://www.siemens.com/simatic-tech-doku-portal

Recycling and disposal

The ET 200S 1SI 3964/ASCII and ET 200S 1SI Modbus/USS series interface modules can be recycled thanks to their low-pollutant equipment. For environmentally sustainable recycling and disposal of your old device, contact a certified disposal service for electronic scrap.

Index

The index contains keywords that come up in the manual.

Serial interface module 2

2.1 Product overview

Order number

6ES7 138-4DF01-0AB0

Product description

The ET 200S 1SI serial interface module is a plug-in module belonging to the ET 200S product range. It provides access to serial communication by means of three hardware interfaces (RS 232C, RS 422 and RS 485) and two software protocols (ASCII and 3964(R)).

You can use the ET 200S 1SI interface module to exchange data between automation systems or computers by means of a point-to-point connection. All communication is based on serial asynchronous transmission.

You select the communication mode when you parameterize the module in the STEP 7 hardware configuration or some other configuration application. Six versions of the module appear in the hardware catalog:

- ASCII (4B)
- ASCII (8B)
- ASCII (32B)
- 3964R (4B)
- 3964R (8B)
- 3964R (32B)

8-byte or 32-byte data transfers increase the throughput rate, but require more I/O memory on the ET 200S rack, whereas 4-byte data transfers require less I/O memory on the ET 200S rack, but provide a lower throughput rate. The module variant you choose depends on your application requirements.

Functionality of the ET 200S 1SI serial interface module

The ET 200S 1SI serial interface module offers the following functions:

- Integrated interface in accordance with RS 232C, RS 422 or RS 485
- Transmission rate up to 115.2 Kbaud, half duplex
- Integration of the following transmission protocols in the module firmware:
  - 3964(R) procedure
  - ASCII driver

The parameterization of the module determines the functionality of the drivers.

The table below lists the functions of the individual driver interfaces.

Table 2- 1 Functions of the module drivers for serial interface module ET 200S 1SI

Function	RS 232C	RS 422	RS 485
ASCII driver	**Yes**	**Yes**	**Yes**
Use of RS 232C secondary signals	Yes	No	No
Controlling/reading of RS 232C secondary signals with FBs	Yes	No	No
RTS/CTS flow control	Yes	No	No
XON/XOFF flow control	Yes	Yes	No
3964(R) procedure	**Yes**	**Yes**	**No**

Communications

The ET 200S 1SI serial interface module permits point-to-point communication with different Siemens modules and non-Siemens products, including:

- SIMATIC S5 via the 3964(R) driver with corresponding interface module on S5 side
- Siemens PDA terminals from the ES 2 family via 3964(R) driver
- MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) and ES 030K data acquisition terminal via 3964(R) driver
- SIMOVERT and SIMOREG (USS protocol) via the ASCII driver (ET 200S SI RS 422/485), with appropriate adaptation of the protocol using a STEP 7 program
- PCs via the 3964(R) procedure (the following development tools are available for programming on PCs: PRODAVE DOS 64R (6ES5 897–2UD11) for MS–DOS, PRODAVE WIN 64R (6ES5 897-2VD01) for Windows or the ASCII driver)
- Barcode readers via the 3964(R) or ASCII driver
- Non-Siemens PLCs via the 3964(R) or ASCII driver
- Other devices with simple protocol structures by means of appropriate protocol adaptation with the ASCII driver
- Other devices that also have a 3964(R) driver

Note

The ET 200S 1SI module cannot be operated with standard FBs downstream of the external communication processors CP 342-5 (Profibus DP) and CP 343-1 (Profinet IO)!

For operation of the module downstream of the communication processors CP 342-5 (Profibus DP) or CP 343-1 (Profinet IO), corresponding special function blocks are available on the Internet pages of Customer Support at:

http://support.automation.siemens.com/WW/view/en/26263724

LED displays

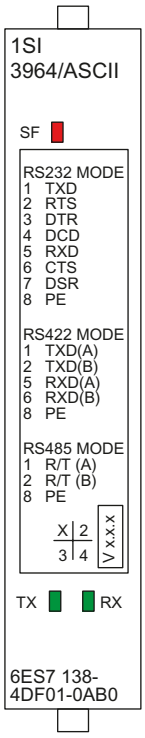
The following status LEDs are installed on the front panel of the interface module:

LED	Color	Description
SF	Red	Group error display
TX	Green	Interface is sending
RX	Green	Interface is receiving

The operating states and errors indicated by these LEDs are described in section Diagnostics (Page 100).

Front panel

The figure below shows the labeling on the front panel of the ET 200S 1SI serial interface module.



2.3 Circuit diagram with terminal assignment

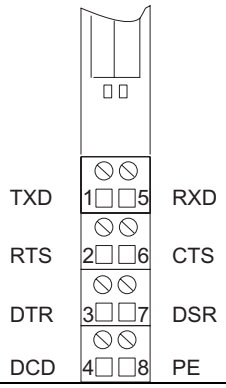
Wiring rules

The cables (terminals 1 to 8) must be shielded. The shield must be terminated at both ends. Use shielding contact elements for this purpose see the (*ET 200S 1SI Distributed I/O Device*) manual.

Terminal assignment for RS 232C communication

The table below shows the terminal assignment for the ET 200S 1SI serial interface module when the RS 232C communication protocol is set.

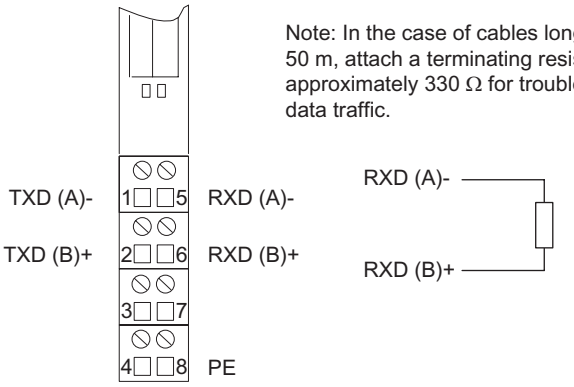
Table 2- 3 Terminal assignment for the ET 200S 1SI serial interface module for RS 232C communication

View	Remarks		
	Mode: Half duplex and full duplex		
	Terminals		
	1	TXD	Transmitted data
	5	RXD	Received data
	2	RTS	Request to send
	6	CTS	Clear to send
	3	DTR	Data terminal ready
	7	DSR	Data set ready
	4	DCD	Data carrier detected
	8	PE	Ground

Terminal assignment for RS 422 communication

The table below shows the terminal assignment for the ET 200S 1SI serial interface module when the RS 422 communication protocol is set.

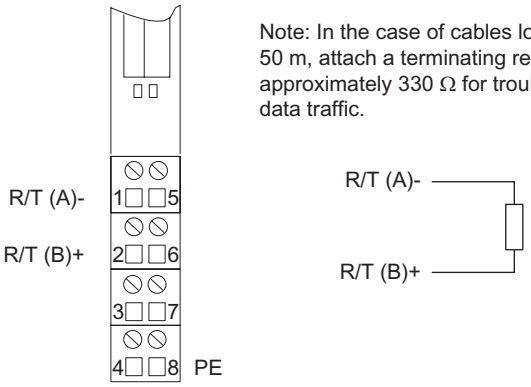
Table 2- 4 Terminal assignment for the ET 200S 1SI serial interface module for RS 422 communication

View	Terminal assignment	Remarks	
	Note: In the case of cables longer than 50 m, attach a terminating resistor of approximately 330 Ω for trouble-free data traffic.	Mode: Full duplex Terminals	
		1	TXD (A)-
		5	RXD (A)-
		2	TXD (B)+
		6	RXD (B)+
		8	PE ground

Terminal assignment for RS 485 communication

The table below shows the terminal assignment for the ET 200S 1SI serial interface module when the RS 485 communication protocol is set.

Table 2- 5 Terminal assignment for the ET 200S 1SI serial interface module for RS 485 communication

View	Terminal assignment	Remarks	
	Note: In the case of cables longer than 50 m, attach a terminating resistor of approximately 330 Ω for trouble-free data traffic.	Mode: Half duplex Terminals	
		1	R/T (A)-
		2	R/T (B)+
		8	PE ground

Terminal assignment of the RS 232C connecting cable for a 9-pin cable connector

The figure below shows the cable connections for RS 232C point-to-point communication between the ET 200S 1SI serial interface module and a communication partner with a 9-pin D connection socket.

- On the ET 200S 1SI end, the signal wires are connected to the correspondingly numbered terminals.
- Use a 9-pin sub D connection socket on the communication partner.

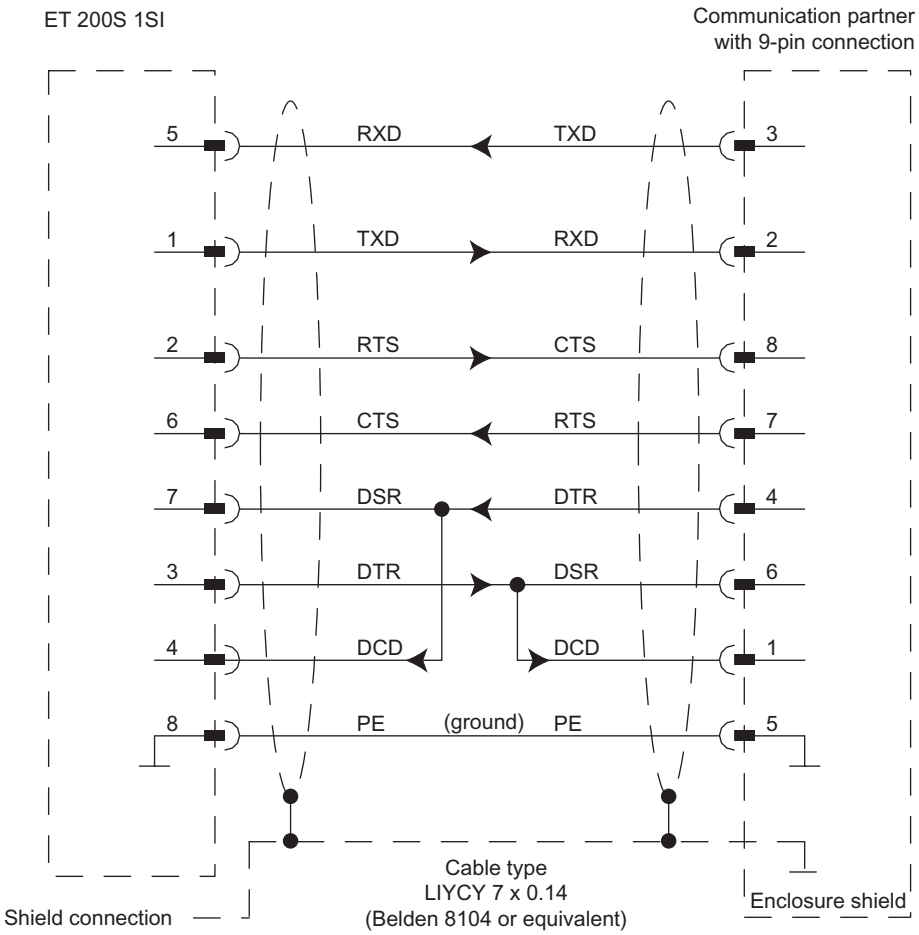


Figure 2-2 Terminal assignment of the RS 232C connecting cable for a 9-pin cable connector

Terminal assignment of the RS 232C connecting cable for a 25-pin cable connector

The figure below shows the cable connections for RS 232C point-to-point communication between the ET 200S 1SI serial interface module and a communication partner with a 25-pin D cable connector.

- On the ET 200S 1SI end, the signal wires are connected to the correspondingly numbered terminals.
- Use a 25-pin sub D connector on the communication partner.

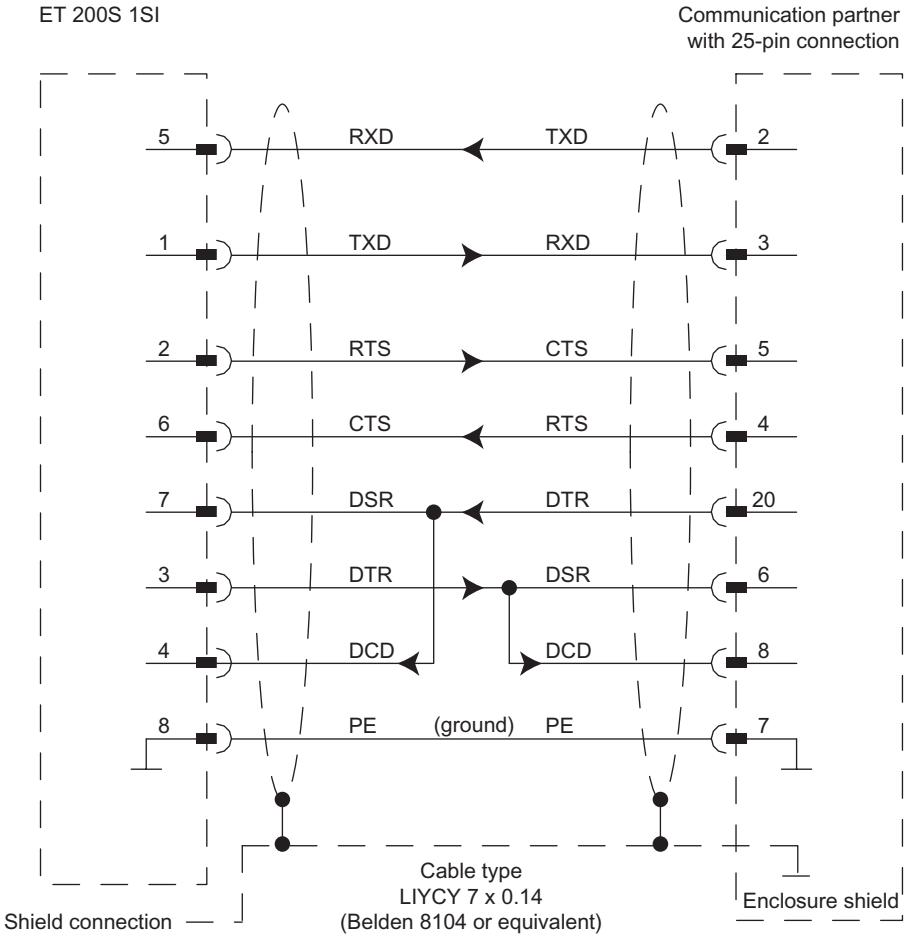


Figure 2-3 Terminal assignment of the RS 232C connecting cable for a 25-pin cable connector

Terminal assignment of the RS 422 connecting cable for a 15-pin cable connector

The figure below shows the cable connections for RS 422 point-to-point communication between the ET 200S 1SI serial interface module and a communication partner with a 15-pin D cable connector.

- On the ET 200S 1SI end, the signal wires are connected to the correspondingly numbered terminals.
- Use a 15-pin sub D connector on the communication partner.

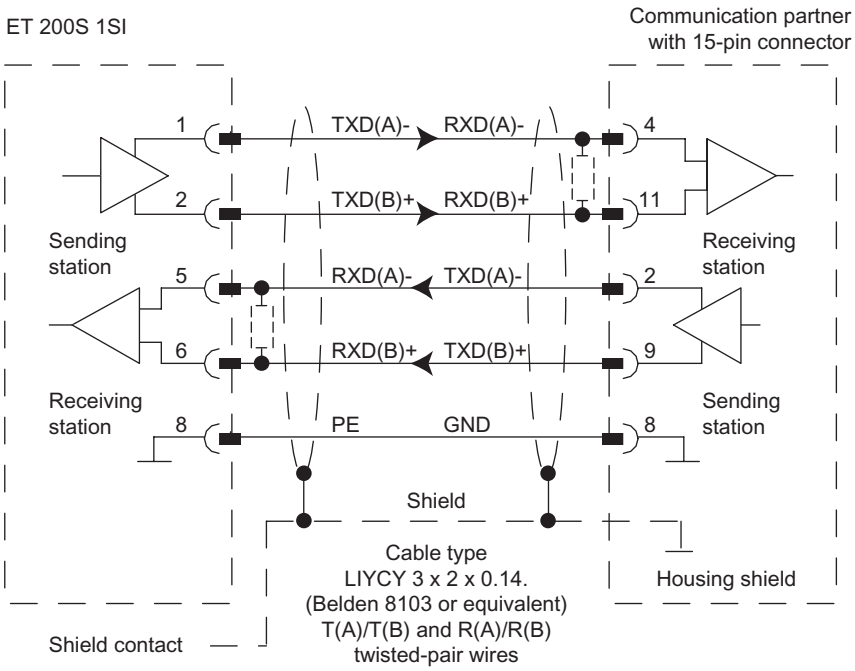


Figure 2-4 Terminal assignment of the RS 422 connecting cable for a 15-pin cable connector

---

Note

If the cables exceed a length of 50 m, attach a terminating resistor of approx. 330 Ω (see figure above) to ensure unimpeded data traffic.

With the type of cable used in this case, the following lengths are supported for an ET 200S 1SI module as the communication partner:

- Max. 1200 m at 19,200 baud
- Max. 500 m at 38,400 baud
- Max. 250 m at 76,800 baud

Terminal assignment of the RS 485 connecting cable for a 15-pin cable connector

The figure below shows the cable connections for RS 485 point-to-point communication between the ET 200S 1SI serial interface module and a communication partner with a 15-pin D cable connector.

- On the ET 200S 1SI end, the signal wires are connected to the correspondingly numbered terminals.
- Use a 15-pin sub D connector on the communication partner.

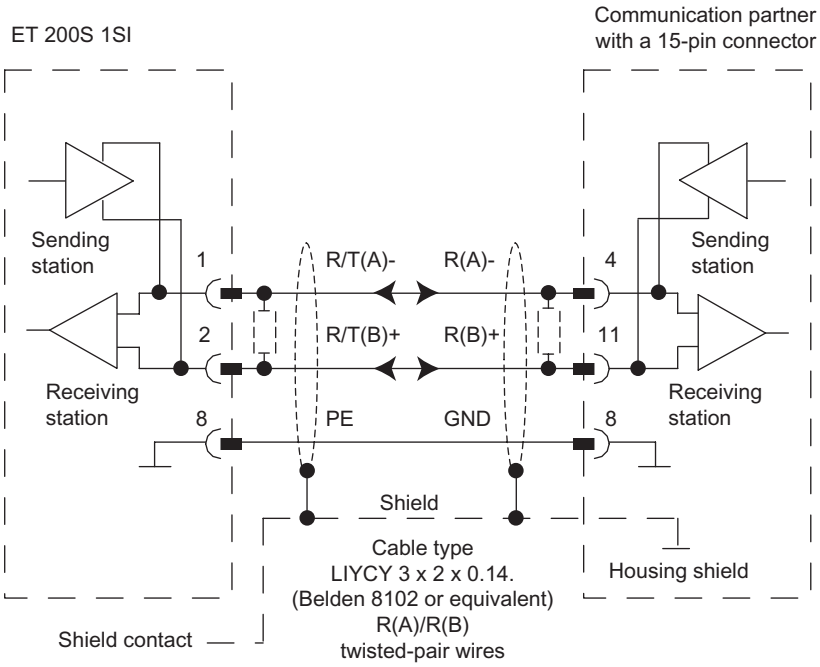


Figure 2-5 Terminal assignment of the RS 485 connecting cable for a 15-pin cable connector

---

Note

If the cables exceed a length of 50 m, attach a terminating resistor of approx. 330 Ω (see figure above) to ensure unimpeded data traffic.

With the type of cable used in this case, the following lengths are supported for an ET 200S 1SI module as the communication partner:

- Max. 1200 m at 19,200 baud
- Max. 500 m at 38,400 baud
- Max. 250 m at 76,800 baud
- Max. 200 m at 115,200 baud

2.4 RS-232C Interface

Definition

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

Features

The RS 232C interface has the following features:

Type:	Voltage interface
Front connector:	8-pin standard ET 200S terminal connector
RS 232C signals:	TXD, RXD, RTS, CTS, DTR, DSR, DCD, GND
Transmission rate:	Up to 115.2 Kbaud (3964(R) procedure) Up to 115.2 Kbaud (ASCII driver)
Cable length:	Up to 15 m, cable type LIYCY 7 x 0.14
Relevant standards:	DIN 66020, DIN 66259, EIA RS 232C, CCITT V.24/V.28
Degree of protection:	IP20

RS 232C signals

The table below describes the RS 232C signals:

Signal	Description	Significance
TXD	**T**ransmitted **D**ata	Transmission line is maintained at logic "1" in idle state.
RXD	**R**eceived **D**ata	Receive line must be maintained at logic "1" by communication partner.
RTS	**R**equest **T**o **S**end	ON: ET 200S 1SI is clear to send. OFF: ET 200S 1SI is not in send mode.
CTS	**C**lear **T**o **S**end	The communication partner can receive data from the ET 200S. The interface module expects this as a response to RTS = ON.
DTR	**D**ata **T**erminal **R**eady	ON: The ET 200S SI is switched on and ready for operation. OFF: The ET 200S SI is not switched on and is not ready for operation.
DSR	**D**ata **S**et **R**eady	ON: The communication partner is switched on and ready for operation. OFF: Communication partner is not switched on and is not ready for operation.
DCD	**D**ata **C**arrier **D**etect	Carrier signal on connection of a modem.

2.5 RS-422/485 Interface

Definition

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

Features

The RS 422/485 interface has the following features:

Type:	Differential voltage interface
Front connector:	8-pin standard ET 200S terminal connector
RS 422 signals:	TXD (A)-, RXD (A)-, TXD (B)+, RXD (B)+, GND
RS 485 signals:	R/T (A)-, R/T (B)+, GND
Transmission rate:	Up to 115.2 Kbaud (3964(R) procedure) Up to 115.2 Kbaud (ASCII driver)
Cable length:	Up to 1,200 m, cable type LIYCY 7 x 0.14
Relevant standards:	EIA RS 422/485, CCITT V.11/V.27
Degree of protection:	IP20

2.6 Basic Principles of Serial Data Transmission

2.6.1 Serial Data Transmission

Point-to-point connection

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

In point-to-point communication, the serial interface module forms the interface between a programmable controller and a communication partner. In point-to-point communication with the ET 200S 1SI serial interface module, data is transmitted via serial interface.

Serial data transmission

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

The ET 200S 1SI serial interface module handles data transmission with the communication partner autonomously via its serial interface. That is why the module has two different drivers for bidirectional data traffic.

- ASCII driver
- 3964(R) procedure

Bidirectional data traffic - operating modes

The ET 200S 1SI has two operating modes for bidirectional data traffic:

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

  The communication partners take it in turns to exchange data in both directions. With half-duplex operation, this means that data is either being sent or received at any given moment. The exception to this may be individual control characters for data flow control (e.g., XON/XOFF), which can also be sent during a receive operation or received during a send operation.

- Full duplex operation (ASCII driver)

  The communication partners exchange data in both directions at the same time. Full duplex operation means you can send and receive data at the same time. Every communication partner must be able to handle send and receive operations simultaneously.

The table below lists the operating modes for data traffic in the case of interfaces with ASCII drivers.

Table 2- 6 Data traffic operating modes for the ET 200S 1SI serial interface module

Data traffic	RS 232C	RS 422	RS 485
Half duplex	Yes	Yes	Yes
Full duplex	Yes	Yes	Not possible

Declarations

Declarations must be made between the two communication partners before serial data transmission can take place. These include:

- Transmission rate (baud rate)
- Character delay time and acknowledgement delay time
- Parity
- Number of data bits
- Number of stop bits
- Number of connection setup and transmission attempts

Sections Basics of data transmission with the 3964(R) procedure (Page 33) and Basic Information on Data Transmission with ASCII Driver (Page 42) describe the role these declarations play in the various transmission procedures, and how they are parameterized.

2.6.4 Transmission integrity

Principle

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

Protocols supported

The figure below shows how the ASCII and 3964(R) protocols supported by the ET 200S 1SI interface module fit into the ISO reference model.

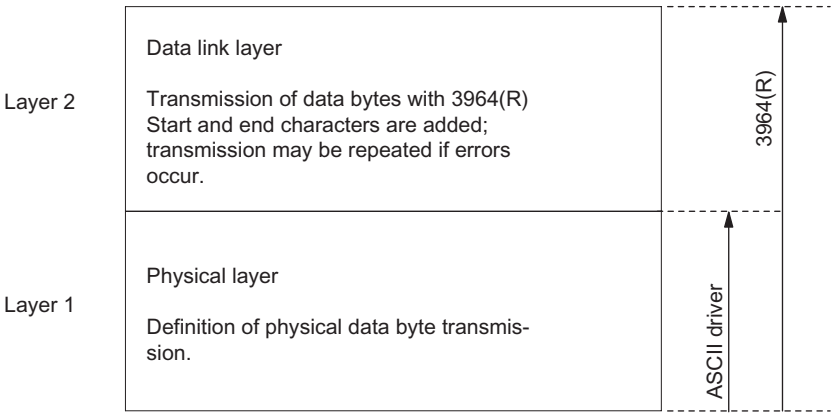


Figure 2-9 How the supported protocols fit into the reference model

Transmission integrity with the ASCII driver

Follow the guidelines below to increase data integrity when working with the ASCII driver:

- Apart from using the parity bit (which can also be deselected, depending on the character frame setting), there are no other measures for ensuring data integrity in the case of data transfer with the ASCII driver. This means that data transmission involving the ASCII driver is very efficient as far as data throughput is concerned, but complete data integrity cannot be guaranteed.
- Using the parity bit makes it possible to detect an inverted bit in a character that is to be transmitted. If two or more bits of a character are inverted, this error can no longer be detected.
- To increase transmission integrity, a checksum and length specification for a message frame can be employed. These measures must be implemented by the user.
- A further increase in data integrity can be achieved by means of acknowledgment message frames in response to send or receive message frames. This is also the case with high-level protocols for data communication (see ISO 7-layer reference model).

Transmission integrity with 3964(R)

The 3964(R) procedure offers increased data integrity:

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

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

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

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

Performance limits with 3964(R)

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

  Although the loss of an entire character (this has to be a zero) is highly unlikely, it may occur under very poor transmission conditions, for example.

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

2.7 Data Transmission With the 3964(R) Procedure

2.7.1 Basics of data transmission with the 3964(R) procedure

Principle

The 3964(R) procedure controls data transmission via a point-to-point connection between the ET 200S module and a communication partner. In addition to the physical layer (layer 1), the 3964(R) procedure incorporates the data link layer (layer 2).

Control characters

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

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

- STX: Start of Text;

  Start of character string to be transmitted
- DLE: Data Link Escape

  Data transmission changeover
- ETX: End of Text;

  End of character string to be transmitted
- BCC: Block Check Character (3964R only)

  Block check character
- NAK: Negative Acknowledge

  Negative feedback

Note

If DLE is transmitted as an information character, it is sent twice on the transmission line (DLE duplication) to distinguish it from the DLE control character used within the context of connection setup and cleardown. The recipient then reverses the DLE duplication.

Priority

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

Block check character

In the case of the 3964R transmission protocol, data integrity is increased by sending an additional block check character (BCC) (see figure below).

Message frame:

STX	Data				DLE ETX		BCC
02_H →	30_H →	31_H →	32_H →		10_H →	03_H →	20_H

30	=	0011	0000
31	=	0011	0001
XOR	=	0000	0001
32	=	0011	0010
XOR	=	0011	0011
10	=	0001	0000
XOR	=	0010	0011
03	=	0000	0011
XOR	=	0010	0000
BCC	→	2	0

Figure 2-10 Block check character

The block check character is equivalent to the even longitudinal parity (EXOR logic operation of all data bytes) of a sent or received block. It starts with the first user data byte (first byte of the message frame) after connection setup and ends after DLE ETX at connection cleardown.

Note

With DLE duplication, the DLE character is accounted for twice when generating the BCC.

2.7.2 Sending data with the 3964(R) procedure

Sending data with 3964(R)

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

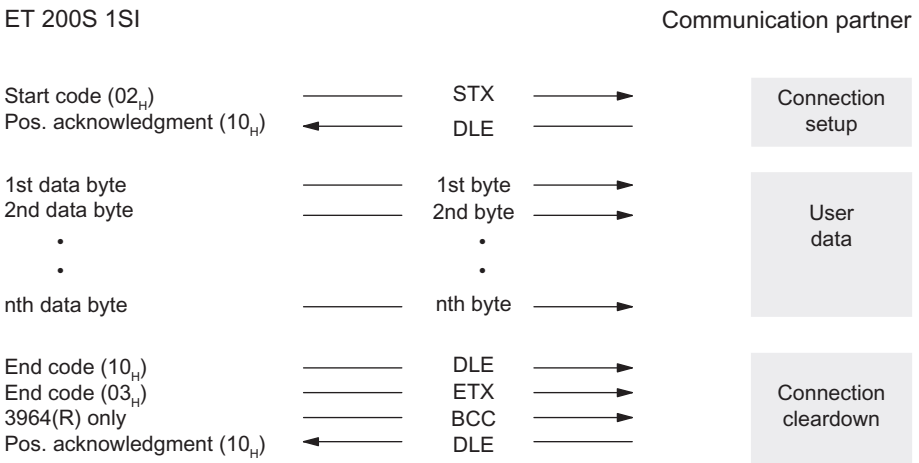


Figure 2-11 Data traffic when sending with the 3964(R) procedure

Establishing a send connection

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

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

Sending data

If a connection has been successfully set up, the user data contained in the output buffer of the ET 200S module is sent to the communication partner with the selected transmission parameters. The partner monitors the times between incoming characters. The interval between two characters must not exceed the character delay time.

Send connection cleardown

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

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

After the defined number of attempts to send the data block, the procedure stops trying and sends a NAK to the communication partner. The system program reports the error to the S_SEND function block (STATUS output parameter).

2.7.3 Receiving data with the 3964(R) procedure

Receiving data with 3964(R)

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

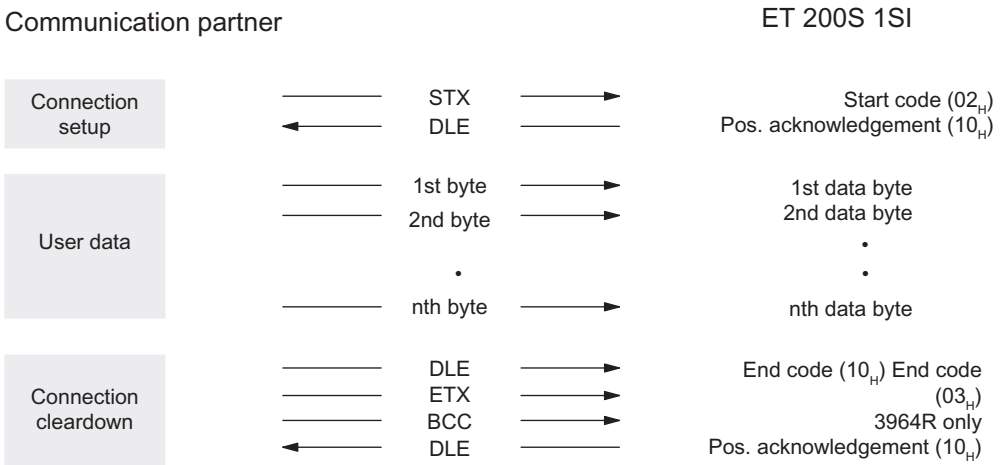


Figure 2-12 Data traffic when receiving with the 3964(R) procedure

Establishing a receive connection

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

If the idle procedure receives any control character (except for STX or NAK), it waits for the character delay time to expire and then sends the NAK character.

Receiving data

If the procedure receives the STX character and a cleared receive buffer available, it responds with DLE. Incoming receive characters are then stored in the receive buffer. If two consecutive DLE characters are received, only one of these is stored in the receive buffer.

After each receive character, the procedure waits out the character delay time for the next character. If this period expires before another character is received, a NAK character is sent to the communication partner. The system program reports the error to the S_RCV function block (STATUS output parameter).

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

Receive connection cleardown

If transmission errors occur during receiving (lost character, frame error, parity error, etc.), the procedure continues to receive until the connection is shut down, then a NAK is sent to the communication partner. A repetition is then expected. If the block still cannot be received without errors after the number of repetition attempts specified during parameterization, or if the communication partner does not start the repetition within a block wait time of 4 seconds, the procedure cancels the receive operation. The system program reports the error to the S_RCV function block (STATUS output parameter).

When the 3964(R) procedure detects a DLE ETX character string, it stops receiving and confirms that the block has been successfully received by sending a DLE character to the communication partner. If errors are found in the received data, it outputs a NAK character to the communication partner. A repetition is then expected.

If the 3964(R) procedure detects the character string DLE ETX BCC, it stops receiving. It compares the received BCC block check character with the internally calculated longitudinal parity. If the block check character is correct and no other receive errors have occurred, the 3964(R) procedure sends a DLE character and returns to the idle state. If the BCC is faulty or a different receive error occurs, a NAK character is sent to the communication partner. A repetition is then expected.

Note

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

Procedure parameters

Select the following identical procedure parameters for both connection partners of a 3964(R) communication route:

- Character delay time
- Acknowledgment delay time
- Connection attempts
- Transmission attempts

Exception:

If operating the ET 200S 1SI module with low priority, you must reduce its number of attempts to connect at least by the count of one compared to those of the connection partner in order to minimize the time required to resolve initialization conflicts (refer to the initialization conflicts section in chapter Data Transmission With the 3964(R) Procedure (Page 39)).

2.7.4 Data Transmission With the 3964(R) Procedure

Handling erroneous data

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

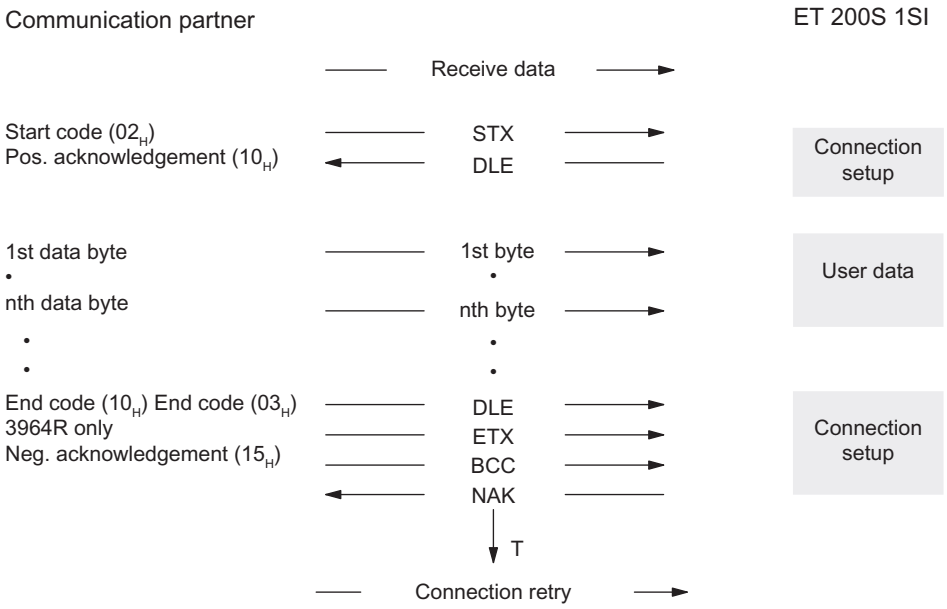


Figure 2-13 Data traffic when erroneous data is received

When DLE, ETX, BCC is received, the ET 200S 1SI module compares the BCC of the communication partner with its own internally calculated value. If the BCC is correct and no other receive errors occur, the ET 200S 1SI module responds with DLE.

Otherwise, it responds with a NAK character and waits out the block wait time (T) of 4 seconds for a new attempt. If, after the defined number of transmission attempts, the block cannot be received, or if no further attempt is made within the block wait time, the ET 200S 1SI module cancels the receive operation.

Initialization conflict

The figure below illustrates the data transmission sequence in the event of an initialization conflict.

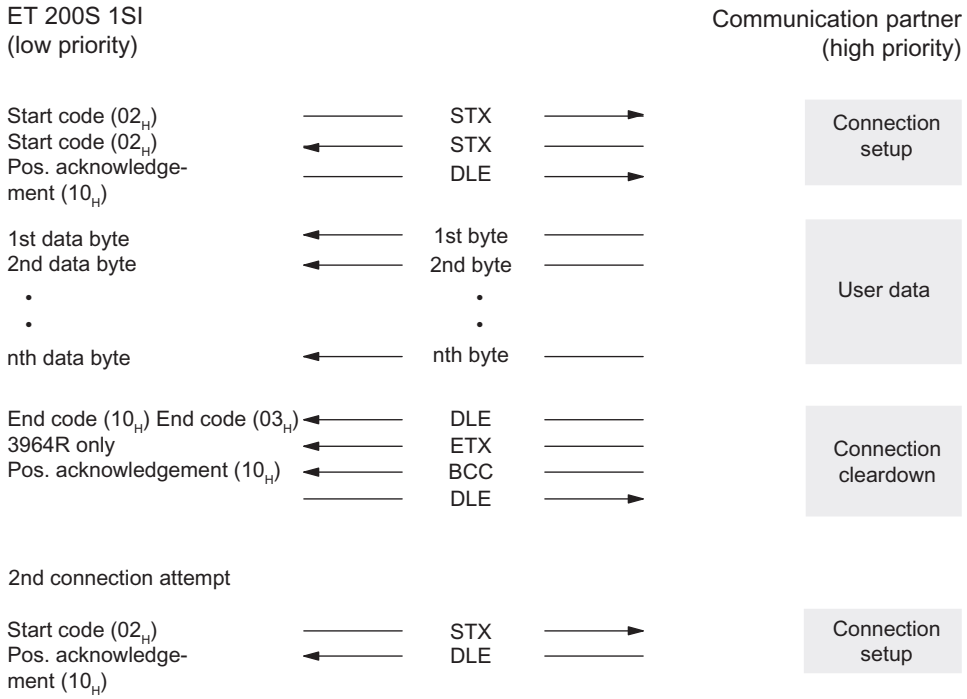


Figure 2-14 Data traffic in the event of an initialization conflict

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

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

Procedure errors

The procedure recognizes both errors caused by malfunctioning of the communication partner and errors caused by faults on the line.

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

If the system program frequently reports an error number at the STATUS output of the FB for send and receive repetitions, this implies occasional disturbances in data traffic. The high repetition frequency balances this out, however. In this case, you are advised to check the transmission link for possible sources of interference, because frequent repetitions reduce the user-data rate and integrity of the transmission. However, the disturbance could also be the result of a malfunction on the part of the communication partner.

If the receive line is interrupted, the system program reports a BREAK status (a break is displayed via the diagnostic interrupt on the ET 200S module) (see Section Diagnostics (Page 100)). No repetition is started. The BREAK status is automatically reset as soon as the connection is restored on the line.

For every detected transmission error (lost character, frame or parity error), a standard number is reported, regardless of whether the error was detected during sending or receiving of a data block. The error is only reported, however, if previous repetition attempts have failed.

2.8 Data Transmission with the ASCII Driver

2.8.1 Basic Information on Data Transmission with ASCII Driver

Introduction

The ASCII driver controls data transmission via a point-to-point connection between the ET 200S 1SI module and a communication partner. The ASCII driver contains the physical layer (layer 1).

The structure of the message frames is left open because it is the S7 user who transfers the complete send message frame to the ET 200S 1SI module. For the receive direction, the end criterion of a message frame must be parameterized. The structure of the send message frames may differ from that of the receive message frames.

The ASCII driver allows data with any structure to be sent and received (all printable ASCII characters as well as all other characters from 00 to FF_H (in the case of character frames with 8 data bits) or from 00 to $7F_H$ (in the case of character frames with 7 data bits)).

See also

Basics on communication via function blocks (Page 64)

Startup properties and operating modes (Page 84)

2.8.2 Transmitting data using the ASCII driver

Transmitting data using the ASCII driver

When transmitting data, specify the number of bytes of user data to be transmitted as the LEN parameter at the call of function block S_SEND. The user data must contain start-of-text and end-of-text characters as required.

If working with the end criterion "character delay time expired" when receiving data, the ASCII driver will pause between two frames, even when sending. You can call FB S_SEND at any time, however, the ASCII driver does not start output until a period longer than the parameterized character delay time has elapsed after the last frame was transmitted.

Note

With an XON/XOFF flow control parameterization, the user data must not contain any of the parameterized XON or XOFF characters. Defaults are DC1 = 11_H for XON and DC3 = 13_H for XOFF.

Transmitting data

The following diagram shows the transmission sequences:

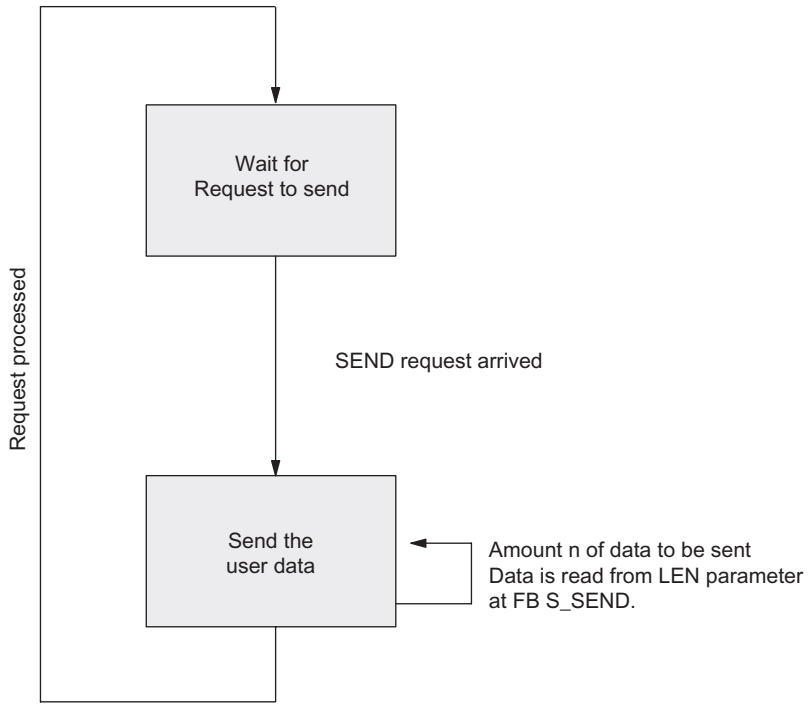


Figure 2-15 Sequence of a Send Operation

2.8.3 Receiving data with the ASCII driver

Receiving data with the ASCII driver

For data transmission with the ASCII driver, you can choose between three different end criteria. The end criterion defines the point at which a message frame has been received completely. Configurable end criteria are:

- **Expiration of character delay time**

  The message frame has neither a fixed length nor a defined end character; the end of the message frame is defined by a pause on the line (expiration of character delay time). The minimum values of the individual baud rates are listed later.

- **Receipt of end character(s)**

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

- **Receipt of a fixed number of characters**

  The length of the receive message frames is always identical.

Code transparency

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

- With one or two end characters
  - Not code-transparent
- End criterion "Expiration of character delay time" or "Receipt of a fixed number of characters"
  - Code-transparent
- Code-transparent operation is not possible with XON/XOFF flow control operation.

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

Minimum character delay time according to baud rate

The minimum value for the character delay time depends on the baud rate. The table below lists the minimum character delay time in ms for the individual baud rates.

Table 2- 7 Minimum character delay time

Baud rate	Minimum character delay time
115	365 ms
300	130 ms
600	65 ms
1.200	32 ms
2.400	16 ms
4.800	8 ms
9.600	4 ms

Baud rate	Minimum character delay time
19.200	2 ms
38.400	1 ms
57.600	1 ms
76.800	1 ms
115.200	1 ms

Reception buffer of the ET 200S module

The receive buffer of the ET 200S 1SI interface module has a length of 4096 bytes. During parameterization, you can specify whether the receive buffer should be cleared on startup or whether the data in the receive buffer should be retained by preventing overwriting. Additionally, you can activate or deactivate buffering of the message frames received.

The receive buffer of the ET 200S 1SI serial interface module is a ring buffer.

- If multiple message frames are written to the receive buffer of the ET 200S 1SI module: The ET 200S 1SI module always transmits the oldest message frame to the CPU.
- If you only ever want to transmit the most recent message frame to the CPU, you must deactivate dynamic message frames **and** switch off overwrite protection.

Note

If continuous reading out of the receive data in the user program is interrupted for a while, you may find that when the receive data is requested again, the CPU receives an old message frame from the ET 200S 1SI module before it receives the most recent one.

The old message frame is the one that was en route between the ET 200S 1SI and the CPU at the time of the interruption, or the one that had already been received by the FB.

2.8.4 End criterion for the data transmission with the ASCII Driver

End criterion "Expiration of character delay time"

When receiving data, the end of the message frame is recognized on expiration of the character delay time. The received data is accepted by the CPU with the S_RCV function block.

In this case, the character delay time must be short enough to expire prior to the second of two consecutive message frames. However, it should also be long enough to prevent a pause within the sending of a message frame (initiated by the communication partner) from being mistaken for the end of the message frame.

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

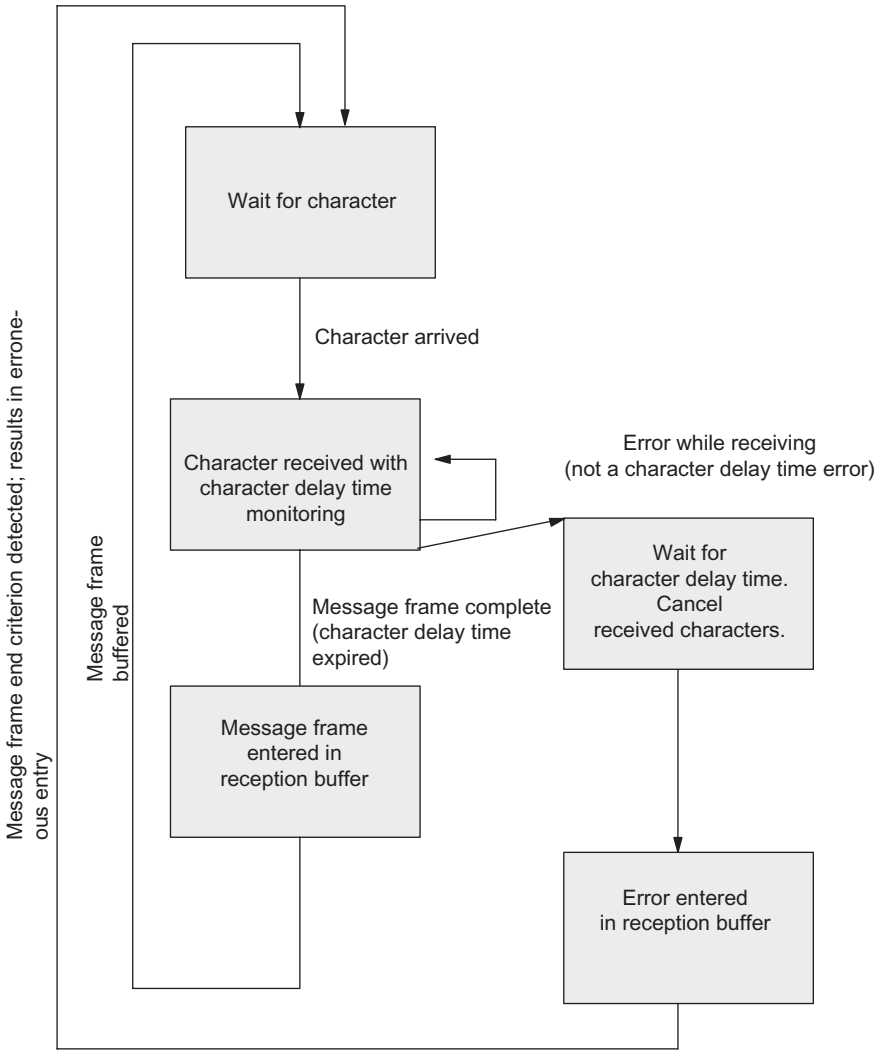


Figure 2-16 Flow diagram for receiving with end criterion "Expiration of character delay time"

End criterion "Receipt of end character(s)"

When receiving data, the end of the message frame is recognized when the defined end character(s) is/are received. The received data, including the end character, is accepted by the CPU with the S_RCV function block.

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

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

The figure below illustrates a receive operation with the end criterion "Receipt of end character(s)".

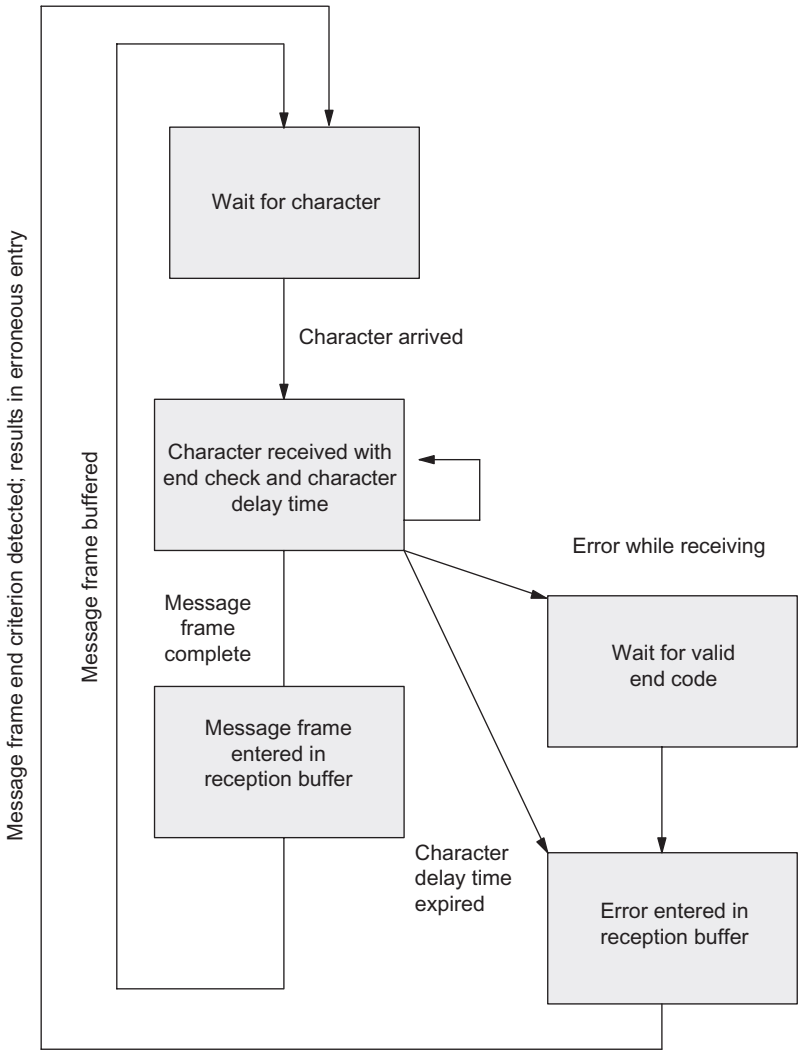


Figure 2-17 Flow diagram for receiving with end criterion "Receipt of end character(s)"

End criterion "Receipt of a fixed number of characters"

When receiving data, the end of the message frame is detected after the defined number of characters has been received. The received data is accepted by the CPU with the S_RCV function block.

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

The figure below illustrates a receive operation with the end criterion "Receipt of a fixed number of characters".

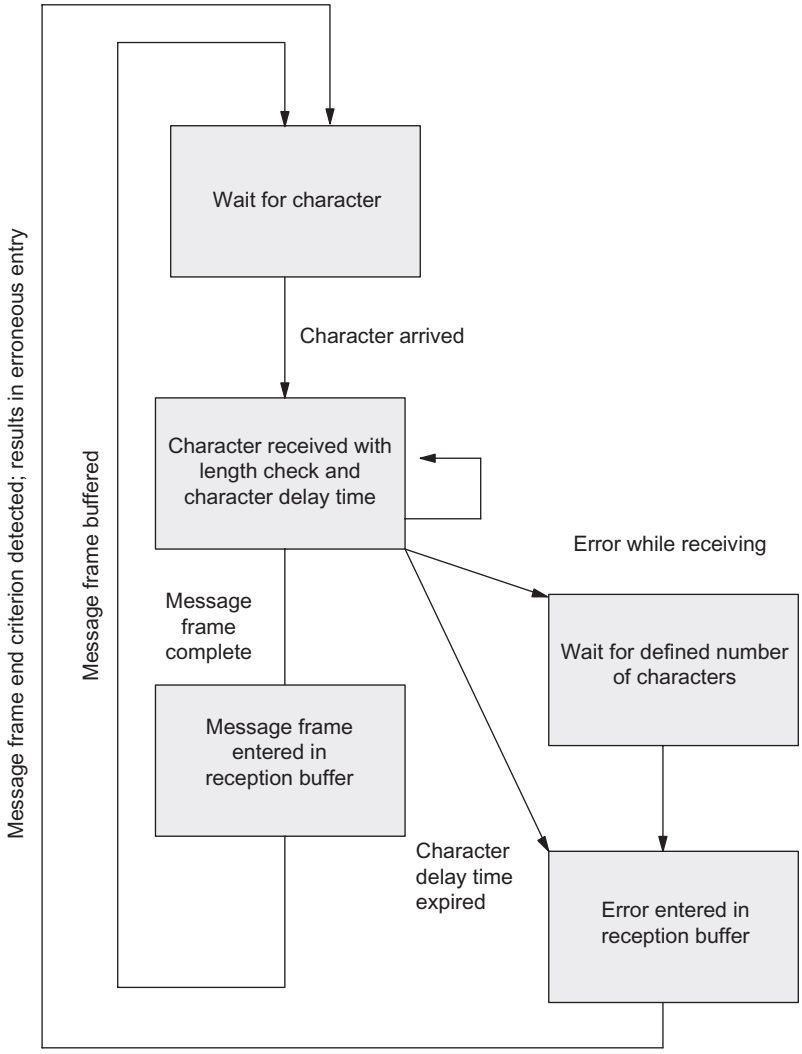


Figure 2-18 Flow diagram for receiving with end criterion "Receipt of a fixed number of characters"

2.8.5 RS-232C secondary signals for data transmission with the ASCII driver

RS 232C secondary signals

The ET 200S 1SI module supports the following RS 232C secondary signals:

- DCD (Input) Data carrier detect
- DTR (Output) Data terminal ready; ET 200S 1SI ready for operation
- DSR (Input) Data set ready; communication partner ready for operation
- RTS (Output) Request to send; ET 200S 1SI clear to send
- CTS (Input) Clear to send; communication partner can receive data from the ET 200S 1SI module (response to RTS = ON of the ET 200S 1SI)

When the ET 200S 1SI module is switched on, the output signals are in the OFF state (deactivated).

You can configure the use of the DTR/DSR and RTS/CTS control signals by means of the parameterization interface, or control their use via functions (FCs) in the user program:

Using RS 232C secondary signals

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

- When automatic control of all RS 232C secondary signals is configured
- When data flow control (RTS/CTS) is configured
- By means of the S_VSTAT and S_VSET function blocks (FBs)

Note

When automatic control of the RS 232C secondary signals is configured, neither RTS/CTS data flow control nor RTS and DTR control by means of the S_VSET FB are possible.

When RTS/CTS data flow control is configured, RTS control by means of the S_VSET FB is not possible.

However, it is always possible to read all RS 232C secondary signals by means of FB S_VSTAT.

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

Automatic control of the RS 232C secondary signals

Automatic control of the RS 232C secondary signals is implemented as follows on the ET 200S 1SI module:

- As soon as the ET 200S 1SI module is configured for a mode with automatic operation of the RS 232C secondary signals, it sets the RTS line to OFF and the DTR line to ON (ET 200S 1SI ready for operation).

  This prevents the transfer of message frames until the DTR line is set to ON. No data can be received via the RS 232C interface as long as DTR = OFF. Any send requests will be canceled with an appropriate error message.
- When a **send request** is queued, the module sets RTS=ON, and triggers the configured data output wait time. After the data output time has elapsed and CTS = ON, the data is sent via the RS 232C interface.
- If the CTS line is not set to ON within the data output wait time or CTS changes to OFF during the transmission process, the module cancels the send request and generates an error message.
- Once the data has been sent and the configured clear RTS time has elapsed, the RTS line is set to OFF. The ET 200S 1SI does not wait for CTS to change to OFF.
- Data can be **received** via the RS 232C interface as soon as the DSR line is set to ON. If the receive buffer of the ET 200S 1SI module is on the verge of overflowing, the ET 200S SI module will not respond.
- An active send request or data receiving operation will be canceled and an error message output if DSR changes from ON to OFF.

Note

When automatic control of the RS 232C secondary signals is configured, neither RTS/CTS data flow control nor RTS and DTR control by means of the S_VSET FB are possible.

Timing diagram

The figure below illustrates the chronological sequence of a send request.

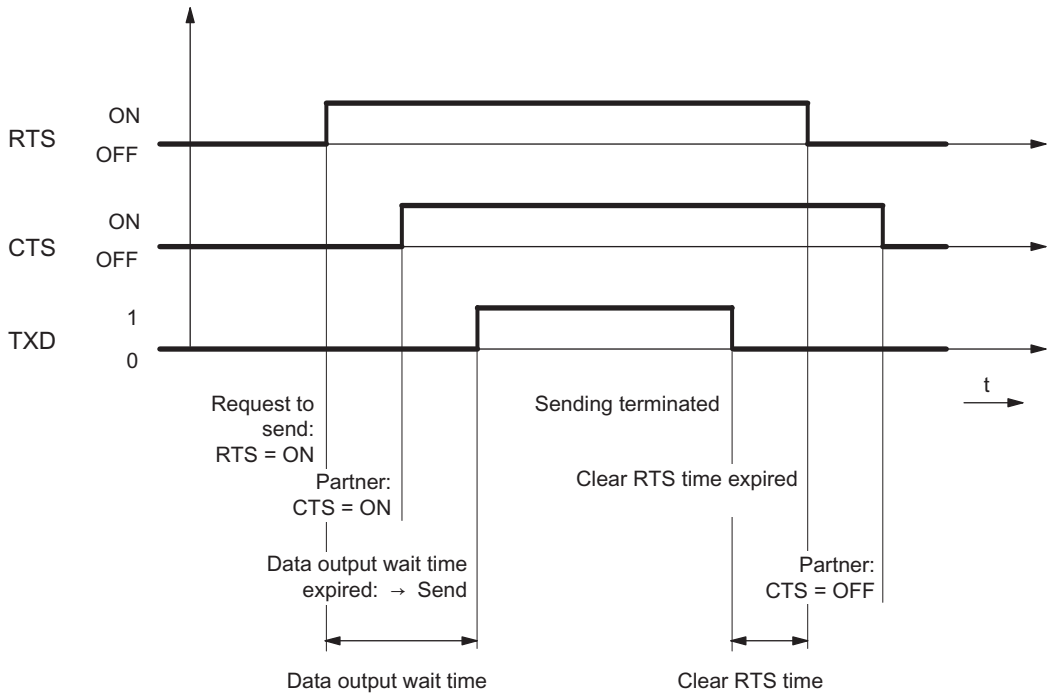


Figure 2-19 Timing diagram for automatic operation of RS 232C secondary signals

Data flow control/Handshaking

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

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

Data flow control of the ET 200S 1SI module is implemented as follows:

- As soon as the ET 200S 1SI module is configured for operation in a mode with flow control, it sends the XON character or sets the RTS line to ON.
- If the defined number of message frames or 50 characters are reached before the receive buffer overflows (size of the receive buffer: 4096 bytes), the ET 200S 1SI module transmits the XOFF character, or sets the RTS channel to OFF. If the communication partner ignores this state and continues transmission, an error message is generated if the receive buffer overflows. The data received in the last message frame will be discarded.

- As soon as a message frame is fetched by the S7 CPU and the receive buffer is ready to receive, the ET 200S 1SI module sends the XON character or sets the RTS line to ON.
- If the ET 200S 1SI module receives the XOFF character or the CTS control signal is set to OFF, the ET 200S 1SI module interrupts the send process. If neither an XON character is received nor CTS is set to ON once a configured time has elapsed, transmission is canceled and an appropriate error message (0708_H) is generated at the STATUS output of the function blocks.

Reading/controlling via FB S_VSTAT and FB S_VSET

The S_VSTAT function block can be used to determine the status of each RS 232C secondary signal. The S_VSET function block can be used to control the DTR and RTS output signals. Section Basics on communication via function blocks (Page 64) provides information on how to use the function blocks to interface the CPU and ET 200S 1SI module.

2.9 Configuring and Parameterizing the Serial Interface Module

2.9.1 Configuring the Serial Interface Module

Principle

If you wish to communicate with the ET 200S 1SI interface module via a PROFIBUS network using an S7 master, you need to use the STEP 7 hardware configuration to set the module up on the PROFIBUS network and set its communication parameters.

If you select the ET 200S 1SI module in the hardware catalog and add it to the ET 200S basic module in the network configuration, the order number of the module, the number of the slot and the addresses of inputs and outputs are automatically transferred to the configuration table. You can then call up the properties dialog box of the ET 200S 1SI module and set the type of communication and other parameters.

2.9.2 Configuring the ASCII driver

Principle

The table below lists the parameters that can be set for the serial interface module's ASCII driver.

Table 2- 8 Parameters for the ASCII driver

Parameters	Description	Value range	Default value
Diagnostics interrupt	Specify whether the module should generate a diagnostic interrupt in the event of a serious error.	• No • Yes	No
Activate BREAK detection	If there is a line break or if the interface cable is not connected, the module generates the error message "Break".	• No • Yes	No
Type of interface	Specify the electrical interface to be used (see sections RS-232C Interface (Page 22) and RS-422/485 Interface (Page 23).	• RS 232C • RS 422 (full duplex) • RS 485 (half duplex)	RS 232C
Half-duplex and full-duplex initial state of the receive line.	Specify the initial state of the receive line in RS 422 and RS 485 operating modes. Not used in RS 232C operating mode. The "Inverted signal levels" setting is only required if compatibility needs to be ensured when a part is replaced.	RS 422: R(A) 5 V/R(B) 0 V (BREAK) R(A) 0 V/R(B) 5 V Inverted signal levels RS 485: None R(A) 0 V/R(B) 5 V	RS 422: R(A) 5 V/R(B) 0 V (BREAK) RS 485: R(A) 0 V/R(B) 5 V

Parameters	Description	Value range	Default value
Data flow control (with default parameters; change default values in the user program)	You can send and receive data with data flow control. Data transmission is synchronized by means of data flow control if one communication partner works faster than the other. Select the type of data flow control and set the relevant parameters (see Section Basic Information on Data Transmission with ASCII Driver (Page 42)). Note: Data flow control is not possible with the RS 485 interface. Data flow control with "RTS/CTS" and "Automatic control of V24 signals" is only supported on RS 232C interfaces.	• None • XON/OFF • RTS/CTS • Automatic control of V.24 signals	None
Baud rate	Select the rate of data transmission in bits per second.	• 110 • 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800 • 115200	9600
Data bits	Select the number of bits to which a character is mapped.	• 7 • 8	8
Stop bits	Select the number of stop bits that are appended to each character during data transmission to signal the end of a character.	• 1 • 2	1
Parity	The data bit sequence can be extended by one character to include the parity bit. The additional value (0 or 1) sets the value of all bits (data bits and parity bits) to a defined state. None: Data is sent without a parity bit. Odd: The parity bit is set so that the total number of all data bits (including the parity bit) returns an odd value with signal state "1". Even: The parity bit is set so that the total number of all data bits (including the parity bit) returns an even value with signal state "1". Any: The signal state of the parity bit is not relevant. Parity is not checked when data is received, but is always set to "0" when data is sent.	• None • Odd • Even • Any	Even

Parameters	Description	Value range	Default value
Indication for end of receive message frame	If data is transmitted by means of the ASCII driver, the end of the receive message frame can be detected in three different ways. Here you can select one of the three transmission modes and enter the specific parameters. Note: If the character delay time expires while data is being received, the receive operation is prematurely canceled in all three modes. The message frame is discarded, except in "Expiration of character delay time" mode. • Expiration of character delay time: The message frame is detected on expiration of the character delay time. • Receipt of end character(s): The end of the message frame is detected when the defined end character(s) is/are received. • Receipt of a fixed number of characters: The end of a message frame is detected on the basis of the defined message frame length. All message frames to be received have the same length.	• Expiration of character delay time • Receipt of end character(s) • Receipt of a fixed number of characters	Expiration of character delay time
Expiration of character delay time, ms	The maximum permissible interval between the receipt of two characters. [1]	1 ms to 65535 ms	4 ms
End character 1 [2]	You can define a maximum of two end characters for the purpose of receiving data with end characters. The selected end characters limit the length of the message frame.	• With 7 data bits:[3] 1 to $7F_H$ • With 8 data bits:[3] 1 to FF_H	3
End character 2 [2]	You can define a maximum of two end characters for the purpose of receiving data with end characters. The selected end characters limit the length of the message frame. Second end code, if specified	• With 7 data bits:[3] 0 to $7F_H$ • With 8 data bits:[3] 0 to FF_H	0
Message frame length on receipt [4]	Specify the message frame length if data with a fixed number of characters is to be received. The message frame length must correspond exactly to the number of data bytes to be received by the communication partner.	1 to 224 bytes	100

Parameters	Description	Value range	Default value
Dynamic message frames	For the purpose of receiving messages, you can specify whether only one message is to be buffered or whether the messages are to be buffered dynamically. If the dynamic message frame option is activated, the module can buffer several messages of different lengths. The buffer concerned is a ring buffer. If the buffer is full, the oldest message is overwritten unless the parameter "Prevent overwriting of buffer" is activated. In this case, the most recent message is discarded. In both cases, a diagnostic interrupt indicates that data has been lost.	• Activated • Deactivated	Activated
Prevent overwriting of buffer	This parameter prevents buffered message frames from being overwritten when the module receives a new message frame but the receive buffer has not yet been emptied. This prevents previously received message frames from being lost.	• No • Yes	Yes
Clear ET 200S 1SI receive buffer on startup	Specify whether the module's receive buffer should be cleared automatically when the CPU changes from STOP to RUN operating mode (CPU startup). In this way, you can ensure that the module's receive buffer only contains message frames that were received after CPU startup.	• No • Yes	Yes

[1] The minimum character delay time depends on the baud rate.

[2] Can only be set if the end criterion is "Receipt of end character(s)".

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

[4] Can only be set if the end criterion is "Receipt of a fixed number of characters".

2.9.3 Configuring the driver for the 3964(R) protocol

Principle

The table below lists the parameters that can be set for the serial interface module's 3964(R) protocol.

Table 2- 9 Driver parameters for the 3964(R) protocol

Parameters	Description	Value range	Default value
Diagnostics interrupt	Specify whether the module should generate a diagnostic interrupt in the event of a serious error.	• No • Yes	No
Activate BREAK detection	If there is a line break or the interface cable is not connected, the module generates a "Break" error message.	• No • Yes	No
Type of interface	Specify the electrical interface to be used.	• RS 232C • RS 422	RS 232C
Receive line initial state	Specify the initial state of the receive line in RS 422 operating mode. Not used in RS 232C operating mode. The "Inverted signal levels" setting is only required to ensure the compatibility of the replacement part.	R(A) 5 V/R(B) 0 V (BREAK) R(A) 0 V/R(B) 5 V Inverted signal levels	R(A) 5 V/R(B) 0 V (BREAK)
Protocol operating mode	Specify whether the data should be sent with a block check character (BCC) to increase data integrity. The block check character is equivalent to the even longitudinal parity (EXOR logic operation of all data bytes) of a sent/received block. If a communication partner detects a block check character when receiving data, it compares the BCC with the internally calculated longitudinal parity. If the block check character is invalid, there is a wait of 4 seconds (block wait time) and then data transmission is repeated. The receive operation is canceled if the block cannot be received after a defined number of attempts have been made, or no further attempt is made within the block wait time.	• No block check • Block check	Block check

Parameters	Description	Value range	Default value
Baud rate	Select the rate of data transmission in bits per second.	• 110 • 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 76800 • 115200	9600
Data bits	Select the number of bits to which a character is mapped.	• 7 8	8
Stop bits	Select the number of stop bits that are appended to each character during data transmission to signal the end of a character.	• 1 • 2	1
Parity	The data bit sequence can be extended by one character to include the parity bit. The additional value (0 or 1) sets the value of all bits (data bits and parity bits) to a defined state. • None: Data is sent without a parity bit. • Odd: The parity bit is set so that the total number of all data bits (including the parity bit) returns an odd value with signal state "1". • Even: The parity bit is set so that the total number of all data bits (including the parity bit) returns an even value with signal state "1". • Any order: The signal state of the parity bit is not relevant. Parity is not checked when data is received, but is always set to "0" when data is sent.	• None • Odd • Even • Any	Even
Character delay time (ms)	The maximum permissible interval between the receipt of two characters. Set the minimum character delay time for your application. Remember that the character delay time must have a specific minimum value, depending on the baud rate.	20 to 655,350 ms in 10 ms increments	220 ms
Acknowledgement delay time (ms)	Specify the maximum time that can expire before an acknowledgement is received from the communication partner when the connection is set up and cleared down. Remember that the acknowledgement delay time must have a specific minimum value, depending on the baud rate.	10 to 655,350 ms in 10 ms increments	2,000 ms (550 ms without block check)

Parameters	Description	Value range	Default value
Connection attempts	Define the number (*n*) of attempts to set up a connection. (After *n* failed attempts, the function is canceled and the error is displayed at the STATUS output of the S_SEND function block.)	1 to 255	6
Transmission attempts	Define the number (*n*) of attempts to transmit a message frame. (After *n* failed attempts at sending the message frame without errors, the function is canceled and the error is displayed at the STATUS output of the S_SEND function block.) Possible causes of cancellation: • Parity error • BBC error; parity error • Different parameterization of the communication partners (e.g., baud rate, parity, character frame, block check character, different protocols)	1 to 255	6
Priority	If both partners request a send request at the same time, the partner taking lower priority will defer its send request. For data transmission, you must assign one communication partner a higher priority and the other a lower priority.	• High • Low	Low
Clear ET 200S 1SI receive buffer on startup	Specify whether the module's receive buffer should be cleared automatically when the CPU changes from STOP to RUN operating mode (CPU startup). In this way, you can ensure that the module's receive buffer only contains message frames that were received after CPU startup.	• No • Yes	Yes

2.10 Communication Using Function Blocks

2.10.1 Basics on communication via function blocks

Overview

Communication between the CPU, ET 200S 1SI and a communication partner is based on the function blocks and the protocols of the ET 200S 1SI module. For information about communication with third party CPUs (non-S7), refer to section Basics of reference data (Page 86).

The function blocks form the software interface between the CPU and ET 200S 1SI serial interface module. They must be called from the user program in cycles.

Setting up communication with the CPU

Whenever the CPU is started up, the ET 200S 1SI module is assigned the current parameters by the CPU's system service. After the connection between the CPU and ET 200S 1SI module has been set up, the ET 200S 1SI module must be initialized.

Each function block has its own startup mechanism. Before active requests can be processed, the associated startup mechanism must be complete.

The ET 200S 1SI module can trigger a diagnostic interrupt in the CPU. The operating system makes 2 bytes of interrupt information available to the user. The user has to program the evaluation of interrupt information (OB82). It is not permissible to call the function blocks in the process or diagnostic interrupt program. The interrupts are not deactivated in the function blocks.

Protocol changeover takes place in the ET 200S 1SI module. In accordance with the selected protocol (3964(R) procedure or ASCII driver), the interface of the ET 200S 1SI module is adapted to suit the interface of the communication partner.

Function blocks of the ET 200S 1SI module

The S7-300 automation system provides you with a number of function blocks that initiate and control communication between the CPU and ET 200S 1SI serial interface module in the user program. The table below lists the FBs used by the ET 200S 1SI module.

Table 2- 13 Function blocks of the ET 200S 1SI module

FB	Name	Significance
FB 2	S_RCV	The S_RCV function block allows you to receive data from a communication partner and store it in a data block.
FB 3	S_SEND	The S_SEND function block allows you to send all or part of a data block to a communication partner.
FB 4	S_VSTAT	The S_VSTAT function block allows you to read the signal states at the RS 232C interface of the ET 200S 1SI module.
FB 5	S_VSET	The S_VSET function block allows you to set/reset the outputs of the ET 200S 1SI module's RS 232C interface.
FB 6	S_XON	The S_XON function block allows you to set additional parameters if the module has been parameterized for XON/XOFF flow control.
FB 7	S_RTS	The S_RTS function block allows you to set additional parameters if the module has been parameterized for RTS/CTS flow control.
FB 8	S_V24	The S_V24 function block allows you to set additional parameters if the module has been configured for automatic control of V.24 signals.

Note

These instance data blocks may not be loaded on the CPU when SEND/RECEIVE block communication is active.

See also

Technical data (Page 107)

2.10.2 FB 3 S_SEND function block

FB 3 S_SEND: Send data to a communication partner

The S_SEND FB transmits a field from a data block, specified by the DB_NO, DBB_NO and LEN parameters, to the ET 200S 1SI module. For the purpose of data transmission, the S_SEND FB is called statically (without conditions) in a cycle, or **alternatively**, in a time-controlled program.

Data transmission is initiated by a positive edge at the REQ input. Depending on the quantity of data, data may be transferred over several calls (program cycles).

The S_SEND FB can be called cyclically by setting the signal state at parameter input R to "1". This cancels transmission to the ET 200S 1SI module and resets the S_SEND FB to its initial state. Data that has already been received by the ET 200S 1SI module is still sent to the communication partner. If the signal state remains static at "1" at input R, it means that sending has been deactivated.

The address of the ET 200S 1SI module to be addressed is specified in the LADDR parameter.

The DONE output shows "request completed without errors". ERROR indicates whether an error has occurred. If an error has occurred, the corresponding event number is displayed in STATUS (see the chapter titled Diagnostics (Page 100)). If no error has occurred, STATUS has the value 0. DONE and ERROR/STATUS are also output at RESET of the S_SEND FB (see time sequence chart). The binary result is reset if an error has occurred. If the block is terminated without errors, the binary result has the status "1".

Startup

The COM_RST parameter of the S_SEND FB notifies the FB of a startup.

Set the COM_RST parameter in the startup OB to 1.

Call the FB in cyclic operation without setting or resetting the COM_RST parameter.

If the COM_RST parameter is set:

- The FB obtains information about the ET 200S 1SI module (number of bytes in the I/O area, whether or not in the distributed I/O).
- The FB resets and terminates any request previously started (before the CPU's last transition to STOP).

When the FB has obtained information about the ET 200S 1SI module, it resets the COM_RST parameter itself.

Note

The S_SEND function block has no parameter check. If it is not parameterized correctly, the CPU may switch to STOP mode.

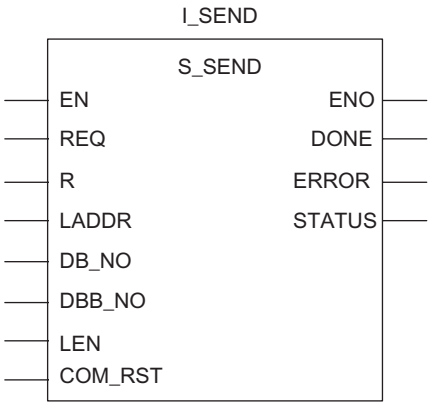
The CPU startup mechanism of the ET 200S module for the FB S_SEND must have been complete (see above) before an initiated request can be processed by the ET 200S 1SI module after the CPU transition from STOP to RUN. Any requests initiated in the meantime will not be lost. They are transmitted to the ET 200S 1SI module once the startup coordination is complete.

FB 3 call:

STL representation

```
CALL              S_SEND, I_SEND
   REQ:            =
   R:              =
   LADDR:          =
   DB_NO:          =
   DBB_NO:         =
   LEN:            =
   DONE:           =
   ERROR:          =
   STATUS:         =
   COM_RST:        =
```

LAD representation



Note

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

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

Allocation in the data storage area

The S_SEND FB works together with an I_SEND instance DB. The DB number is supplied with the call. The data in the instance DB cannot be accessed.

Note

Exception: If error STATUS == W#16#1Exx occurs, you can view the SFCERR variable for additional details (see the chapter titled Diagnostics (Page 100)). This error variable can only be loaded via symbolic access to the instance DB.

FB 3 S_SEND parameters

The table below lists the S_SEND (FB 3) parameters.

Table 2- 14 FB 3: S_SEND parameters

Name	Type	Data type	Description	Permitted values, remark
REQ	INPUT	BOOL	Initiates request on positive edge	
R	INPUT	BOOL	Cancels request	Cancels the active request. Sending is blocked.
LADDR	INPUT	INT	Start address of the ET 200S 1SI module	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Transmitted data block no.: CPU-specific (zero is not permissible).
DBB_NO	INPUT	INT	Data byte number	0 ≤ DBB_NO ≤ 8190 Transmit data from data word
LEN	INPUT	INT	Data length	1 ≤ LEN ≤ 224, specified in number of bytes
DONE[1]	OUTPUT	BOOL	Request completed without errors	STATUS parameter == 16#00
ERROR[1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
STATUS[1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter will contain error information.
COM_RST	IN_OUT	BOOL	Restarts the FB	
[1] The DONE, ERROR and STATUS parameters are available for one CPU cycle following a successful send request.				

Timing chart for FB 3 S_SEND

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

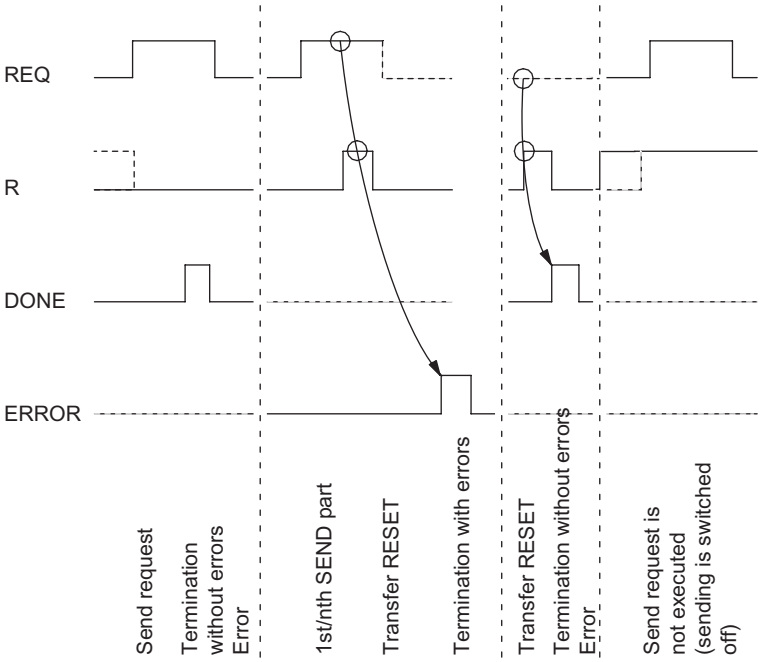


Figure 2-20 Timing chart for FB 3 S_SEND

Note

The REQ input is edge-triggered. A positive edge at the REQ input is sufficient. The result of the logic operation does not have to be at "1" during transfer.

2.10.3 FB 2 S_RCV function block

FB S_RCV: Data received from a communication partner

The FB S_RCV transmits data from the ET 200S 1SI module to an S7 data area as specified by the DB_NO and DBB_NO parameters. For data transmission, FB P_SND_RK is called cyclically or, alternatively, statically in a time-controlled program (unconditional).

A (static) signal state "1" at the EN_R parameter enables a check to determine whether data is to be read from the ET 200S 1SI module. An active transmission event can be canceled with signal state "0" at the EN_R parameter. The canceled receive request is terminated with an error message (STATUS output). Receiving is disabled as long as the signal state at the EN_R parameter is "0". Depending on the quantity of data, data may be transferred over several calls (program cycles).

If the function block detects signal state "1" at the "R" parameter, the current transmission request is canceled and the S_RCV FB is reset to its initial state. Receiving is disabled as long as the signal state at the R parameter is "1". If the signal state returns to "0", the canceled message frame is received again from the beginning.

The address of the ET 200S 1SI module to be addressed is specified in the LADDR parameter.

The NDR output indicates "Request completed without errors/data activated" (all data read) ERROR indicates whether an error has occurred. In STATUS, the error number is displayed in the event of an error. If the receive buffer is occupied by more than 2/3 of its capacity, STATUS returns a warning after each S_RCV call. If no errors or warnings have occurred, status has a value of "0".

NDR and ERROR/STATUS are also output when the S_RCV FB is reset (LEN parameter ==16#00) (see time sequence chart). The binary result is reset if an error has occurred. If the block is terminated without errors, the binary result has the status "1".

Startup

The COM_RST parameter of the S_RCV FB notifies the FB of a startup.

Set the COM_RST parameter in the startup OB to 1.

Call the FB in cyclic operation without setting or resetting the COM_RST parameter.

If the COM_RST parameter is set:

- The FB obtains information about the ET 200S 1SI module (number of bytes in the I/O area, whether or not in the distributed I/O).
- The FB resets and terminates any request previously started (before the CPU's last transition to STOP).

When the FB has obtained information about the ET 200S 1SI module, it resets the COM_RST parameter itself.

Note

The S_RCV function block has no parameter check. If it is not parameterized correctly, the CPU may switch to STOP mode.

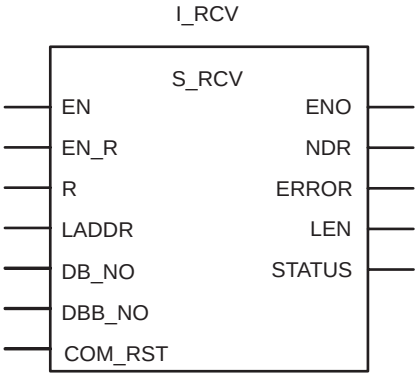
Before an initiated request can be received by the ET 200S 1SI module after the CPU has switched from STOP to RUN mode, the ET 200S module's CPU startup mechanism for the S_RCV FB must be complete.

FB 2 call

STL representation

```
CALL            S_RCV, I_RCV
      EN_R:      =
      R:         =
      LADDR:     =
      DB_NO:     =
      DBB_NO:    =
      NDR:       =
      ERROR:     =
      LEN:       =
      STATUS:    =
      COM_RST:   =
```

LAD representation



Note

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

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

Allocation in the data storage area

The S_RCV FB works together with an I_RCV instance DB. The DB number is supplied with the call. The data in the instance DB cannot be accessed.

Note

Exception: If error STATUS == W#16#1Exx occurs, you can view the SFCERR variable for more precise error information. This error variable can only be loaded via symbolic access to the instance DB.

FB 2 S_RCV parameters

The table below lists the S_RCV (FBs) parameters.

Table 2- 15 FB 2: S_RCV parameters

Name	Type	Data type	Description	Permitted values, remark
EN_R	INPUT	BOOL	Enables data reading	
R	INPUT	BOOL	Cancels request	Cancels the active request. Receiving disabled.
LADDR	INPUT	INT	Start address of the ET 200S 1SI module	The start address is taken from STEP 7.
DB_NO	INPUT	INT	Data block number	Receive data block no.: CPU-specific, zero is not allowed
DBB_NO	INPUT	INT	Data byte number	0 ≤ DBB_NO ≤ 8190 Receive data from data word
NDR[1]	OUTPUT	BOOL	Request completed without errors, data activated	STATUS parameter == 16#00
ERROR[1]	OUTPUT	BOOL	Request completed with errors	Error information is written to the STATUS parameter.
LEN[1]	OUTPUT	INT	Length of the message frame received	1 ≤ LEN ≤ 224, specified in number of bytes
STATUS[1]	OUTPUT	WORD	Specification of error	If ERROR == 1, the STATUS parameter will contain error information.
COM_RST	IN_OUT	BOOL	Restarts the FB	
[1] The NDR, ERROR, LEN and STATUS parameters are available for **one** CPU cycle following a successful receive request.				

Timing chart for FB 2 S_RCV

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

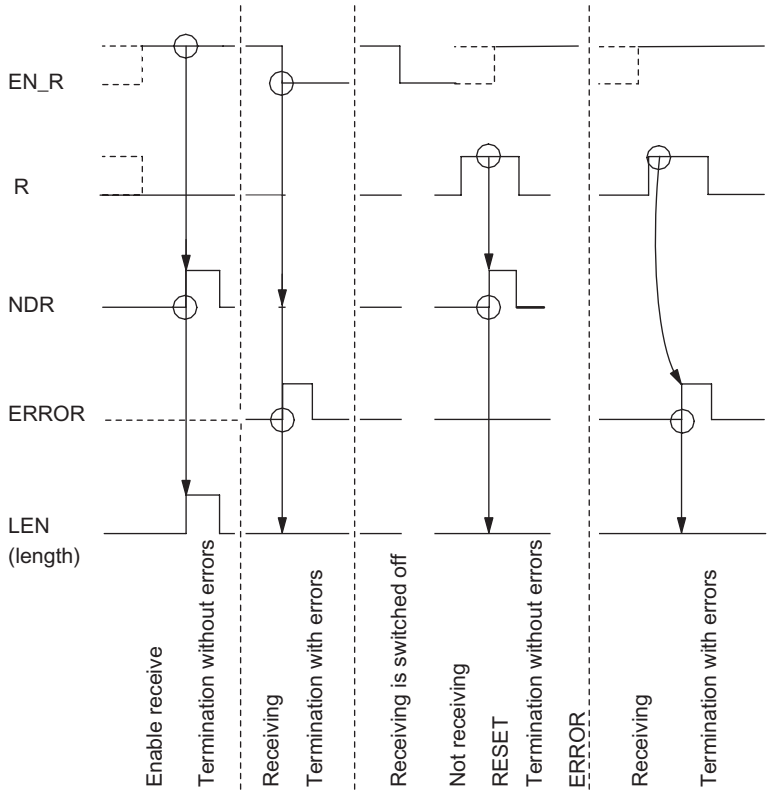


Figure 2-21 Timing chart for FB 2 S_RCV

Note

The EN_R input must be set statically to "1". The EN_R parameter must be supplied with logic operation result "1" throughout the entire receive request.

2.11 Startup properties and operating modes

Operating modes of the ET 200S 1SI serial interface module

The ET 200S 1SI module has the following operating modes:

- STOP: When the ET 200S 1SI module is in STOP mode, no protocol driver is active and all send and receive requests from the CPU are given a negative acknowledgement. The ET 200S 1SI module remains in STOP mode until the cause of the STOP has been eliminated (e.g., a wire break or invalid parameter).
- New parameterization: When you assign new parameters to the ET 200S 1SI module, the protocol driver is initialized. The SF LED comes on during new parameterization.

  Sending and receiving operations are not possible, and send and receive message frames stored in the ET 200S 1SI module are lost when the driver is restarted. Communication between the ET 200S 1SI module and the CPU is restarted (current message frames are canceled).

  Once new parameterization is complete, the ET 200S 1SI module is in RUN mode and is ready to send and receive.
- RUN: The ET 200S 1SI module processes the CPU send requests. The message frames received from the communication partner will be prepared for their transfer to the CPU.

Startup properties of the ET 200S 1SI module

Startup consists of two phases:

- Initialization: As soon as voltage is applied to the ET 200S 1SI module, the serial interface is initialized and waits for parameterization data from the CPU.
- Parameterization: During parameterization, the ET 200S 1SI module receives the module parameters that have been assigned to the current slot with STEP 7.

Behavior of the ET 200S 1SI module when the CPU operating mode changes

After the ET 200S 1SI module has started up, all data between the CPU and ET 200S 1SI is exchanged via the function blocks.

- CPU STOP: When the CPU is in STOP mode, communication via PROFIBUS is not possible. Any active data transmission between the module and the CPU, including both send and receive requests, is canceled and the connection is reestablished.

  If the ASCII driver is parameterized without flow control, data traffic at the RS 232C interface of the ET 200S 1SI module continues. In other words, the current send request is still completed. With the ASCII driver receive message frames are received until the receive buffer is filled.
- CPU startup: During startup, the CPU transfers parameters to the ET 200S 1SI module.

  You can automatically clear the receive buffer of ET 200S 1SI at CPU startup by setting corresponding parameters.
- CPU RUN: When the CPU is in RUN mode, send and receive operations are unrestricted. In the first FB cycles following the CPU restart, the ET 200S 1SI module and the corresponding FBs are synchronized. A new FB S_SEND or S_RCV is executed after this operation was completed.

Points to note when sending message frames

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

If the CPU changes to STOP while data is being transferred from the CPU to the module, the FB S_SEND outputs error (05) 02_H after restart. To prevent this behavior, the user program can call FB S_SEND with the RESET input from the startup OB.

Note

The ET 200S 1SI module only sends data to the communication partner once it has received all the data from the CPU.

Points to note when receiving message frames

You can use STEP 7 to parameterize "Clear module receive buffer during startup = yes/no".

- If you have set the "yes" parameter, the ET 200S 1SI module receive buffer is automatically cleared when the CPU switches from STOP to RUN mode.
- If you have set the "no" parameter, the number of message frames you have specified will be buffered in the receive buffer of the ET 200S 1SI module.

If the CPU switches to STOP mode while data is being transferred from the CPU to the ET 200S 1SI module, the FB outputs error (05) 02_H following a warm restart. To prevent this behavior, the user program can call FB S_SEND with the RESET input from the startup OB. With "Clear module receive buffer during startup = no", the module transfers the message frame to the CPU again.

Dynamic message frame buffer

Select dynamic or single frame buffering. If you select the "activate" check box, the module can buffer different message frames of different length. The buffer is a ring buffer. The oldest message is overwritten on buffer overflow unless the "Prevent overwriting of buffer" parameter is activated. In this case, the most recent message is discarded. Overwriting of an alarm triggers a diagnostics interrupt that reports the loss of data.

2.12 Reference Data for Masters Other than S7-PROFIBUS

2.12.1 Basics of reference data

Data exchange between the master and the ET 200S 1SI module

The ET 200S 1SI module is configured for the transmission of 4, 8 or 32 bytes, input or output, with consistency over the whole length. The ET 200S 1SI module uses the 4-, 8- or 32-byte input/output memory for data transmission to and from the CPU via PROFIBUS DP transmission media.

The CPU can write data to the inputs and outputs and read it from the inputs and outputs at any time, in the following way:

- The CPU transmits a request to the ET 200S 1SI module in the first byte of the module's output memory.
- The ET 200S 1SI module accepts the request by transferring the request code to the input memory.
- The CPU exchanges data using segments of 3, 7 or 31 bytes (as many segments as are required, depending on the I/O size), until all the request data is transferred.

The first byte of the segment is a coordination byte that synchronizes transmission of the segment between the CPU and ET 200S 1SI module (see figure below). The remaining bytes of the I/O memory contain the request data.

The CPU transmits data to the ET 200S SI module as follows:

Byte	Content
0	Coordination byte
1	Data byte 0
2	Data byte 1
⋮	⋮
N	Data byte n

The ET 200S SI module transmits data to the CPU as follows:

Byte	Content
0	Coordination byte
1	Data byte 0
2	Data byte 1
⋮	⋮
N	Data byte n

n = 3, 7 or 31, depending on the module version selected in the configuration

Figure 2-22 Data exchange between the CPU and ET 200S 1SI module.

Description of the coordination byte

The table below describes the contents of the coordination byte (byte 0), which synchronizes data transmission between the CPU and the ET 200S 1SI serial interface module.

Table 2- 21 Contents of coordination byte 0 for data transmission

Byte segment	Description
Request byte written by the CPU	Bit 7 6 5 4 3 2 1 0 Res. | Job code | Error | Execution number
Bit 7	Reserved for special applications of FB S_SEND. To evaluate the coordination byte you must hide this bit.
Request code	Set by the CPU to initiate a request.
Execution number	**Send request:** Incremented by the CPU by 1 when the CPU sends another segment to the ET 200S 1SI module...*or* **Receive request:** Accepted by input byte 0 of the CPU every time the CPU receives a new segment in the correct order from the interface module. Indicates the last valid execution number when the error bit is set. (Value range: 1 to 7).
Error	Set by the CPU to indicate that a segment has not been received in the correct order. The execution number field indicates the last valid execution number.
Request byte written by the ET 200S 1SI module	Bit 7 6 5 4 3 2 1 0 Res. | Job code | Error | Execution number
Bit 7	Reserved for special applications of FB S_SEND. To evaluate the coordination byte you must hide this bit.
Request code	Accepted by the ET 200S 1SI module to acknowledge that the request has been accepted.
Execution number	**Send request:** Accepted by output byte 0 of the module every time the module receives a new segment in the correct order from the CPU. Indicates the last valid execution number when the error bit is set. **Receive request:** Incremented by the module by 1 when the module sends another segment to the CPU. (Value range: 1 to 7).
Error	The sender monitors the error bit of the receiver for a segmented transaction. If the error bit has been set: • **CPU = sender** (send request): The CPU sends the segments again, starting with the next segment after the number reported by the receiver. • **Module = sender** (receive request): The 1SI module cancels further transmission of the Rx message frame to the user and error message 0x0551 is output in the status word. The module waits for this error message to be acknowledged (idle). After the active error sequence is complete, the canceled Rx message frame is reported to the user again, or is made available for transfer.

Request code definitions

The table below lists the requests on the basis of how bits 4 to 6 are assigned in coordination byte 0.

Table 2- 22 Request codes

Bits 6 5 4	Hex. value	Definition
0 0 0	0_H	Idle state
0 0 1	1_H	Send
0 1 0	2_H	Receive
0 1 1	3_H	Read V.24 signal state
1 0 0	4_H	Write V.24 signals
1 0 1	5_H	Transfer parameters: With this request, you can set additional parameters that are not specified in the GSD file.
1 1 0	6_H	Reserved
1 1 1	7_H	Request end acknowledgement

Rules for writing request codes

The following rules apply to writing request codes in the coordination byte so that the CPU and ET 200S 1SI module can synchronize data transmission:

- Before the CPU user program can write a request code to the output coordination byte, it must see an idle code from the ET 200S 1SI module's input coordination byte.
- Then, before the CPU user program can write the first segment to output byte 1...n, it must see the request acknowledgement code (i.e., the accepted request code) in the module's input coordination byte.
- If the user program sees any request acknowledgement code other than the one sent by the program, it must not write to output byte 0...n until it has again seen an idle code from the ET 200S 1SI module's input coordination byte.

  This situation can arise if, for example, two separate requests are executed in the same cycle, both requests see the idle code, and each writes a different request code to the output byte. Because the CPU cycle and the PROFIBUS DP cycle are not synchronous, it is not certain which request will reach the module first. For this reason, each request must be able to wait for the end of the other request before it is processed.

Receive status of the 1SI module

The 1SI module always indicates is to receive status when it is in idle state (request acknowledgment byte 0 = 00H). The receive status is written to the bytes 1 and 2.

Status	Meaning
0000_H	No received message available
0001_H	Received message/receive message frame available
$0B01_H$	Receive buffer loaded by more than 2/3 of its capacity

Status word definitions

In the examples of data transmission shown on the pages that follow, the ET 200S 1SI module uses bytes 1 and 2 for the status message in some of the responses to the CPU. The table titled "Diagnostic messages in the STATUS parameter" lists the status words and their definitions.

Order of the bytes in the word

In the case of all 16-bit words (e.g., status and length), the most significant byte is sent first during data transmission between the CPU and the ET 200S 1SI module.

Receive status of the 1SI module

The user is always shown the status of the 1SI module's receive buffer when the module is in an idle state (request acknowledgement byte 0 = 00_H. The status is then stored in bytes 1 + 2.

Status	Significance
0000H	No received message available
0001H	Received message/receive message frame available
0B01H	Reception buffer is more than 2/3 full.

2.12.7 Troubleshooting

Error conditions

The serial interface module issues an error in response to the following conditions:

- If the send request is longer than 224 bytes, the module responds with a request end acknowledgement, and the status word contains the error code. The CPU then writes an idle code to the request and terminates the request.
- If a receive request has been sent to the module and the received message contains an error, the module accepts the receive request code with execution number zero, and the status word contains the error code. The CPU then writes an idle code to the request and terminates the request.
- If a receive request has been sent to the module and no received message is available, the module accepts the receive request code with execution number zero, and the status word contains the value 0101_H. This is not an error condition, but it does prevent the module from being deactivated in receive request mode and from waiting for a receive message. As a result, send requests can still be executed. The CPU writes an idle code to the request and terminates the request.

Exceptions

As mentioned previously, a particular operation (e.g., a send request) cannot be initiated in the user program until the module is in an idle state. After a request has been sent, the operation must wait until the module has accepted the request code before the operation in question can be executed. In the case of operations with segmentation during startup, the following exceptions may occur:

Note

In the following descriptions of a send or parameterize operation, the sender is the CPU and the receiver is the serial interface module. In the case of a receive operation, the sender is the serial interface module and the receiver is the CPU.

- **Error:** The sender monitors the error bit of the receiver for a segmented transaction. The following actions are triggered after the error bit was set:
  - **CPU = sender** (send request): The CPU retransmits the segments, starting at the next segment after the number reported by the receiver.
  - **Module = sender** (receive request): The 1SI module cancels further transmission of the Rx message frame to the user and error message 0x0551 is output in the status word. The module waits for acknowledgment of this error message. After this current receive sequence was completed, the canceled Rx message frame is reported to the user again, or is made available for transfer.

- **Execution number not in the correct order:** If, during a segmented operation, the receiver receives a segment with an execution number that is not the previous execution number + 1, it must report an error and respond with the last execution number.
  - **CPU = recipient** (receive request): After having received segment with set error bit and are message 0x0551 in the status word, the CPU must cancel their receive request and discard the previously activated data.
- **Modified request code:**
  - If the receiver receives a segment with a request code that differs from the code with which the segmented operation began, and if it is not 000 or 111, the receiver ignores the other code and discards the associated data.
  - If the receiver receives a segment with an idle state request code during a segmented operation, the operation is canceled and the idle state is adopted without an error bit being set.
  - If the receiver receives a segment with a request end acknowledgement request code during a segmented operation, the operation is canceled and the idle state is adopted without an error bit being set.
  - If, during a segmented operation, the sender receives a response with another request code, the message must be canceled. The idle code is then sent again, the module must enter an idle state, and the operation must be executed again.

2.13 Diagnostics

Overview

The diagnostics functions of the ET 200S 1SI module permit you to locate errors quickly, even during operation. The following diagnostics options are available:

- Diagnostics via the status LEDs on the front panel of the ET 200S 1SI module.
- Diagnostics via the STATUS output of the function blocks
- Diagnostics via PROFIBUS slave diagnostics

Diagnostics information using status LEDs

The following status LEDs are located on the front panel of the ET 200S 1SI interface module:

- TX (green): Lights up when the module sends data via the interface.
- RX (green): Lights up when the module receives data via the interface.
- SF (red): Indicates one of the following possible errors/faults:
  - Hardware fault
  - Parameter assignment error
  - Wire break or loose cable between the module and the communication partner:

    Only detected with RS 422 interface connections if the initial state of the receive line is set to R(A) 5 V/R(B) 0 V.
  - Communication errors (parity, frame errors, buffer overflow)

Structure of the function block diagnostic messages

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

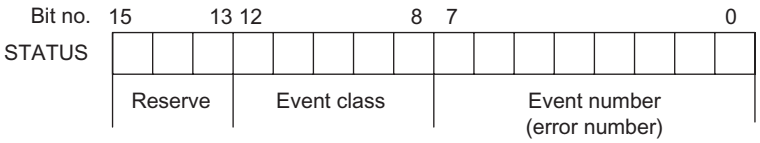


Figure 2-23 Structure of the STATUS parameter

Example: The figure below illustrates the content of the STATUS parameter for the "Request canceled due to restart, warm restart or reset" event (event class $1E_H$, event number $0D_H$).

Event: Job aborted due to complete restart, restart, or reset.

				2^4				2^0	2^7							2^0
STATUS	x	x	x	1	1	1	1	0	0	0	0	0	1	1	0	1
	Reserve			Event class: $1E_H$					Event number: $0D_H$							

Figure 2-24 Example: STATUS parameter for event class 1EH, event 0DH

Function block diagnostic messages

The table below describes the event classes, the definitions of the event numbers, and the recommended remedy for each error condition.

Table 2- 29 Diagnostic messages in STATUS parameter

Event number	Event	Remedy
Event class 2 ($0x02_H$): "Initialization error"		
(02) 01_H	No (valid) parameterization	Assign the module valid parameters. If necessary, check that the system is correctly installed.
Event class 5 (05_H): "Error while processing CPU request"		
(05) 02_H	Request not permitted in this operating mode of the ET 200S 1SI module (e.g., device interface not parameterized).	The send message frame is greater than 224 bytes in length. The ET 200S 1SI module has canceled the send request. Select a shorter message frame length.
(05) $0E_H$	Invalid message frame length	The length of the send message frame exceeds 224 bytes. The ET 200S 1SI module has canceled the send request. Select a shorter frame length.
(05) 50_H	The parameter update request is invalid for the currently selected form of ET 200S 1SI module data flow control.	Either change the parameters of the function block (FB 6 S_XON, FB 7 S_RTS, FB 8 S_V24) in the AS program, or change the form of data flow control for the ET 200S 1SI module in the hardware configuration so that they correspond.
(05) 51_H	Frame execution error during communication between the ET 200S 1SI module and the automation system. The error occurred in the automation system during transmission of a received message frame from the ET 200S 1SI module.	The module and the automation system have canceled the transmission. Repeat the receive request; the ET 200S 1SI module sends the received message again.
Event class 7 (07H): "Send error"		
(07) 02_H	Only for 3964(R): Error establishing connection: After STX was sent, NAK or any other code (except for DLE or STX) was received.	Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is switched into the transmission line.

Event number	Event	Remedy
(07) 03_H	Only for 3964(R): Acknowledgment delay time exceeded: After STX was sent, partner did not respond within the acknowledgment delay time.	The partner device is too slow or not ready to receive, or there is a break in the transmission line, for example. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is switched into the transmission line.
(07) 04_H	Only for 3964(R): Cancellation by partner: One or more characters were received from the partner during sending.	Check whether the partner is also indicating errors, possibly because not all transmitted data has arrived (e.g., break in the transmission line), fatal errors are pending or the partner device has malfunctioned. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is interconnected in the transmission line.
(07) 05_H	Only for 3964(R): Negative acknowledgment when sending	Check whether the partner is also indicating errors, possibly because not all transmit data has arrived (e.g., break in the transmission line), fatal errors are pending, or the partner device has malfunctioned. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is interconnected in the transmission line.
(07) 06_H	Only for 3964(R): End-of-connection error: • Partner rejected message frame at end of connection with NAK or a random string (except for DLE), or • Acknowledgment characters (DLE) received too early.	Check whether the partner is also indicating errors, possibly because not all transmit data has arrived (e.g., break in the transmission line), fatal errors are pending, or the partner device has malfunctioned. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is interconnected in the transmission line.
(07) 07_H	Only for 3964(R): Acknowledgment delay time exceeded at end of connection or response monitoring time exceeded after a send message frame: After connection termination with DLE ETX, no response received from partner within acknowledgment delay time.	Partner device too slow or faulty. If necessary, use an interface test device switched into the transmission line to check.
(07) 08_H	Only for ASCII drivers: The wait time for XON or CTS = ON has expired.	The communication partner is faulty, too slow or has been taken offline. Check the communication partner; you may need to change the parameter assignment.
(07) $0B_H$	Only for 3964(R): Initialization conflict cannot be resolved because both partners have high priority.	Change the parameter assignment.
(07) $0C_H$	Only for 3964(R): Initialization conflict cannot be resolved because both partners have low priority.	Change the parameter assignment.

Event number	Event	Remedy
Event class 8 (08_H): "Receive error"		
(08) 02_H	Only for 3964(R): Error establishing connection: • In idle mode, one or more random codes (other than NAK or STX) were received, or • After an STX was received, partner sent more characters without waiting for response DLE. After partner power ON: • The module receives an undefined character while the partner is being switched on.	Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is switched into the transmission line.
(08) 05_H	Only for 3964(R): Logical error while receiving: After DLE was received, a further random code (other than DLE or ETX) was received.	Check whether the partner always duplicates the DLE in the message frame header and data string or the connection is cleared down with DLE ETX. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is interconnected in the transmission line.
(08) 06_H	Character delay time (ZVZ) exceeded: • Two successive characters were not received within character delay time, or Only for 3964(R): • 1. character after sending of DLE while establishing connection was not received within the character delay time.	Partner device too slow or faulty. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is interconnected in the transmission line.
(08) 07_H	Only for 3964(R): Illegal message frame length: A zero-length message frame has been received.	Receipt of a zero-length message frame does not constitute an error. Check why the communication partner is sending message frames without user data.
(08) 08_H	Only for 3964(R): Error in block check character (BCC): The value of BCC calculated internally does not match the BCC received by the partner when the connection was terminated.	Check whether the connection is seriously disrupted; in this case you may also occasionally see error codes. Check for malfunction of the partner device; you may need to use an interface test device (FOXPG) that is interconnected in the transmission line.
(08) 09_H	Only for 3964(R): The number of repetitions must be set to the same value.	Declare a block wait time at the communication partner identical to that in your module. Check for malfunction of the communication partner; you may need to use an interface test device that is switched into the transmission line.
(08) $0A_H$	There is no free receive buffer available: No receive buffer space available for receiving data.	The S_RCV FB must be called more frequently.

Event number	Event	Remedy
Event class 8 (08_H): "Receive error"		
(08) $0C_H$	Transmission error: • Transmission error (parity/stop bit/overflow error) detected. Only for 3964(R): • If this happens during send or receive operations, repetition is started. • If a corrupted character is received in idle mode, the error is reported immediately so that disturbances on the transmission line can be detected early. • If the SF LED (red) lights up, there is a break in the connecting cable between the two communication partners.	Disturbances on the transmission line cause message frame repetitions, thus lowering user data throughput. The risk of undetected error increases. Change your system setup or cable wiring. Check the connecting cables of the communication partners or check whether both devices have the same setting for baud rate, parity and number of stop bits.
(08) $0D_H$	BREAK: Break in receive line to partner.	Reconnect or switch on partner.
(08) 10_H	Only for ASCII drivers: Parity error: • If the SF LED (red) lights up, there is a break in the connecting cable between the two communication partners.	Check the connecting cables of the communication partners or check whether both devices have the same setting for baud rate, parity and number of stop bits. Change your system setup or cable wiring.
(08) 11_H	Only for ASCII drivers: Character frame error: • If the SF LED (red) lights up, there is a break in the connecting cable between the two communication partners.	Check the connecting cables of the communication partners or check whether both devices have the same setting for baud rate, parity and number of stop bits. Change your system setup or cable wiring.
(08) 12_H	Only for ASCII drivers: More characters were received after the module had sent XOFF or set CTS to OFF.	Reconfigure the communication partner or read module data more rapidly.
(08) 18_H	Only for ASCII drivers: DSR = OFF or CTS = OFF	The partner has switched the DSR or CTS signal to "OFF" before or during a transmission. Check the partner's control of the RS 232C accompanying signals.
(08) 50_H	The length of the receive message frame is greater than 224 bytes or the defined message frame length.	Adjust the message frame length of the partner
Event class 11 ($0B_H$): Warning		
(0B) 01_H	Reception buffer is more than 2/3 full	

Event number	Event	Remedy
Event class 30 ($1E_H$): "Communication error between module and CPU"		
(1E) $0D_H$	"Request canceled due to restart, warm restart or reset"	
(1E) $0E_H$	Static error at the call of SFC DPRD_DAT. The RET_VAL return value of the SFC is available for evaluation in the SFCERR variable of the instance DB.	Load the SFCERR variable from the instance DB.
(1E) $0F_H$	Static error at the call of SFC DPWR_DAT. The RET_VAL return value of the SFC is available for evaluation in the SFCERR variable of the instance DB.	Load the SFCERR variable from the instance DB.
(1E) 10_H	Static error during call of the SFC RD_LGADR. The RET_VAL return value for the SFC is made available for evaluation in the SFCERR variable in the instance DB.	Load the SFCERR variable from the instance DB.
(1E) 11_H	Static error during call of the SFC RDSYSST. The RET_VAL return value for the SFC is made available for evaluation in the SFCERR variable in the instance DB.	Load the SFCERR variable from the instance DB.
(1E) 20_H	Parameter outside of the range.	Change the input of the function block so that it is within the valid range.
(1E) 41_H	The number of bytes specified at the FBs' LEN parameter is not permissible.	Maintain the range of values from 1 to 224 bytes.

Evaluating the SFCERR variable

More detailed information on errors (1E) $0E_H$, (1E) $0F_H$, (1E) 10_H and (1E) 11_H belonging to event class 30 can be obtained via the SFCERR variable.

You can load the SFCERR variable from the instance DB of the corresponding function block.

The error messages entered in the SFCERR variable are described in the sections on the "DPRD_DAR" and SFC15 "DPWR_DAT" system functions in the *System Software for S7-300/400, System and Standard Functions* Reference Manual.

PROFIBUS slave diagnostics

The slave diagnostic data is compliant with EN 50170, Volume 2, PROFIBUS. Depending on the DP master, diagnostic data for all DP slaves conforming to this standard can be read with STEP 5 or STEP 7.

PROFIBUS slave diagnostics comprise module diagnostics, module status and channel-specific diagnostics. Detailed information on DP slave diagnostics can be found in the manual titled *ET 200S Distributed I/O Device*.

Channel-specific diagnostics: Channel-specific diagnostics provide information about channel errors in modules and starts after the module status. The table below lists the types of channel-specific error.

Table 2- 30 Types of channel error for the ET 200S 1SI serial interface module

Fault type	Significance	Remedy
00110: Wire break	Wire broken or disconnected.	Check the wiring to the terminals. Check the cable to the partner.
00111: Overflow	Buffer overflow; message length overflow	The S_RCV FB must be called more frequently.
01000: Underflow	3964(R) only: Message with a length of 0 sent	Check why the communication partner is sending message frames without user data.
01001: Error	Internal module error occurred.	Replace the module.
10000: Parameter assignment error	Module is not parameterized.	Correct the parameterization.
10110: Message error	Frame error, parity error	Check the communication settings.

2.14 Technical data

General technical data

The general technical data specified in the chapter "General technical data" of the *ET 200S Distributed I/O System* manual applies to serial interface module ET 200S 1SI 3964/ASCII. This manual is available at:

http://www.siemens.com/simatic-tech-doku-portal

Technical data for protocols and the interface

Table 2- 31 General technical data for the ET 200S 1SI module

General technical data	
Indicator elements:	• LED (green): TX (transmit) • LED (green): RX (receive) • LED (red): SF (group fault)
Protocol drivers supplied	3964(R) driver ASCII driver
Baud rates for 3964(R) protocol Baud rates for ASCII drivers	110, 300, 600, 1200, 2400, 4800, 9600, 19.200, 38.400, 57.600, 76.800, 115.200
Character frames (10 or 11 bits)	Number of bits per character: 7 or 8 No. of start/stop bits: 1 or 2 Parity: None, even, odd, any
Memory requirements of the standard blocks (FBs)	Sending and receiving: approx. 4,300 bytes
Technical data for the RS 232C interface	
Interface	RS 232C, 8 terminals
RS 232C signals	TXD, RXD, RTS, CTS, DTR, DSR, DCD, PE All electrically isolated from the internal power supply of the ET 200S 1SI module.
Maximum transmission distance	15 m
Technical data for the RS 422/485 interface	
Interface	• RS 422, 5 terminals • RS 485, 3 terminals
RS 422 signals RS 485 signals	TXD (A)-, RXD (A)-, TXD (B)+, RXD (B)+, PE R/T (A)-, R/T (B)+, PE All electrically isolated from the internal power supply of the ET 200S 1SI module.
Maximum transmission distance	1,200 m

Technical data

Dimensions and weight	
Dimensions W x H x D (in mm)	15 x 81 x 52
Weight	Approx. 50 g
Module-specific data	
RS 232C • Number of inputs • Number of outputs	 4 3
RS 422 • Number of input pairs • Number of output pairs	 1 1
RS 485 Number of I/O pairs	 1
Cable length • Shielded (RS 232C) • Shielded (RS 422/485)	 Max. 15 m Max. 1200 m
Degree of protection[1]	IEC 801-5
Voltages, currents, potentials	
Rated supply voltage of electronics (L+) • Polarity reversal protection	24 V DC Yes
Potential isolation • Between channels and backplane bus • Between channels and electronics power supply • Between channels • Between channels and PROFIBUS DP	 Yes Yes No Yes
Insulation test at • Channels to backplane bus and load voltage L+ • Load voltage L+ to backplane bus	 500 VDC 500 VAC
Current source • From backplane bus • From power supply L+	 Max. 10 mA Max. 120 mA 50 mA, typical
Power loss of the module	Typically 1.2 W
Status, interrupts, diagnostics	
Status indicator	• Green LED (TX) • Green LED (RX)
Diagnostics functions • Group fault display • Diagnostics information can be displayed	 Red LED (SF) Possible

Dimensions and weight	
Outputs	
Output, RS 232C range • For capacitive load • Short-circuit protection • Short-circuit current • Voltage at the outputs or inputs to PE (ground)	± Max. 10 V Max. 2500 pF Yes Approx. 60 mA Max. 25 V
Output, RS 422/485 • Load resistance • Short-circuit protection • Short-circuit current	 Min. 50 kΩ Yes Approx. 60 mA
[1] External protection equipment required in the user-supply input wire links: • Blitzductor standard mounting rail adapter • Blitzductor protective module KT AD-24V	

SIEMENS

Data sheet 6ES7138-4DF01-0AB0

SIMATIC DP, ELECTRONIC MODULE FOR ET 200S, 1 SI SERIAL INTERFACE ONE CHANNEL, 15MM WIDE RS232/422, 485 ASCII, 3964R

Figure similar

Supply voltage	
Rated value (DC)	24 V
Reverse polarity protection	Yes

Input current	
from supply voltage L+, max.	120 mA
from backplane bus 3.3 V DC, max.	10 mA

Power loss	
Power loss, typ.	1.2 W

Memory	
Standard blocks	5 100 byte; S_SEND 2700, S_RCV 2400, S_XON 2600, S_RTS 2600, S_V24 2700, S_VSTAT 1800, S_VSET 1800

Interfaces	
Number of interfaces	1
Interface physics, RS 232C (V.24)	Yes; RS 232C signals: 8 (TxD, RxD, RTS, CTS, DTR, DSR, DCD, PE)

Subject to change without notice
© Copyright Siemens

Interface (physical) RS 422/485 (X.27)	Yes; RS-422 signals: 5 (TxD(A), RxD(A), TxD(B), RxD(B), PE); RS-485 signals: 3 (R/T(A), R/T(B), PE)
RS 232, cable length shielded, max.	15 m
RS 422/RS 485, cable length shielded, max.	1 200 m
Point-to-point	
Integrated protocol driver	
— 3964 (R)	Yes
— ASCII	Yes
Telegram length, max.	
— 3964 (R)	224 byte
— ASCII	224 byte
Transmission rate, RS 422/485	
— with 3964 (R) protocol, max.	115.2 kbit/s; half duplex: 110, 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 76 800, 115 200 bps
— with ASCII protocol, max.	115.2 kbit/s; Full duplex: 110, 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 76 800, 115 200 bps
Transmission speed, RS 232	
— with 3964 (R) protocol, max.	115.2 kbit/s; half duplex: 110, 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 76 800, 115 200 bps
— with ASCII protocol, max.	115.2 kbit/s; Full duplex: 110, 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 76 800, 115 200 bps
Character frame (adjustable)	
— Bits per character	7 or 8
— Number of start/stop bits	1 or 2
— Bits per character frame	10
— Parity	none, odd, even, any
Data quantity per PLC sampling cycle	
— Send	32 byte; with IM151-1 Standard as of 6ES7 151-1AA04-0AB0; with IM151-1 High Feature as of 6ES7 151-1BA01-0AB0; otherwise 8 bytes
— Receive	32 byte; with IM151-1 Standard as of 6ES7 151-1AA04-0AB0; with IM151-1 High Feature as of 6ES7 151-1BA01-0AB0; otherwise 8 bytes

Interrupts/diagnostics/status information	
Diagnostics indication LED	
• Receive RxD (green)	Yes
• Transmit TxD (green)	Yes
• Group error SF (red)	Yes

Potential separation	
Potential separation interface	
• Between RS 422/485 and internal power supply	Yes
• between RS 232 and internal power supply	Yes

Subject to change without notice
© Copyright Siemens

Ambient conditions	
Ambient temperature during operation	
• min.	0 °C
• max.	60 °C
Ambient temperature during storage/transportation	
• min.	-40 °C
• max.	70 °C

Dimensions	
Width	15 mm
Height	81 mm
Depth	52 mm

Weights	
Weight, approx.	50 g

last modified: 03/07/2017

Subject to change without notice
© Copyright Siemens