RS-232/RS-485/RS-UNI

Transmitting and receiving data with devices featuring a serial interface

Application note
110285_en_01

1 Description

This document describes as an example, which sequences are required to transmit or receive data with a device featuring a serial interface.

In case you are using the following devices, you can inform yourself using this document. The example sequences are for 17 byte user data. If your module has more or less user data, you can adjust the designs accordingly.

Type	Order No.	User data (byte)
AXL F RS UNI 1H	2688666	17
AXL SE RS232	1181787	17
AXL SE RS485	1088128	17
AXL SE RS232 EF	1507979	61
AXL SE RS485 EF	1507978	61
IB IL RS 232-ECO	2702795	11
IB IL RS 485-ECO	2702141	11
IB IL RS UNI-PAC	2700893	11, 27, 59 (adjustable via DIP switch)

Observe this note

Make sure you always use the latest documentation.
It can be downloaded at phoenixcontact.net/products.

Table of contents

2 Sequences

The designations follow the AXL SE RS232 data sheet.

Example: Process data assignment for the “Transmit characters” command with 17 characters (Z1 ... Z17)

Word	**0**		**1**		**2**		**...**	**9**	
Byte	**0**	**1**	**2**	**3**	**4**	**5**	**...**	**18**	**19**
OUT	10_{hex}	xx	17_{dec}	Z1	Z2	Z3	...	Z16	Z17
IN	10_{hex}	Status bits	xx	xx	xx	xx	...	xx	xx

ℹ The commands and the process data assignment for the device you are using can be found in the device-specific data sheet. It can be downloaded at phoenixcontact.net/products.

Comments on the tables below

Comment	Content
*1)	OUT byte 0 = 00_{hex} Write “Read number of characters received and fill level of the receive buffer” command This step is not necessarily required. It is listed here to get a defined entry into the sequence.
*2)	This step is not necessarily required since the module has a transmit FIFO in which the characters to be transmitted are buffered.
*3)	OUT byte 0 = xx_{hex} By sending this command, the action is triggered. Always perform the steps in the specified order, that is to say: Always write the output data OUT byte 3 ... xx and OUT byte 2. Then write the command OUT byte 0.

2.1 Transmitting a maximum of 17 characters

Step	Process data	Meaning/note
1	OUT byte 0 = 00_{hex}	*1)
2	Wait until bit Tx_buf_not_empty in IN byte 1 = 0	Wait until the send buffer is empty*2)
3	OUT byte 3 ... 19 = characters to be transmitted	Transfer transmit data
4	OUT byte 2 = number of characters to be transmitted	Transfer number of characters to be transmitted
5	OUT byte 0 = 10_{hex}	Write “Transmit characters” command*3)
6	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
7	Wait until bit Tx_buf_not_empty in IN byte 1 = 0	Wait for confirmation that the transmission process is completed

2.2 Transmitting 20 characters

Step	Process data	Meaning/note
1	OUT byte 0 = 00$_{hex}$	*1)
2	Wait until bit Tx_buf_not_empty in IN byte 1 = 0	Wait until the send buffer is empty*2)
3	OUT byte 3 ... 19 = the first 17 characters	Transfer transmit data
4	OUT byte 2 = 17	Transfer number of characters to be transmitted
5	OUT byte 0 = 10$_{hex}$	Write “Transmit characters” command*3)
6	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
7	OUT byte 3 ... 5 = the remaining 3 characters	Transfer transmit data
8	OUT byte 2 = 3	Transfer number of characters to be transmitted
9	OUT byte 0 = 50$_{hex}$	Toggle “Transmit characters” command
10	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
11	Wait until bit Tx_buf_not_empty in IN byte 1 = 0	Wait for confirmation that the transmission process is completed

2.3 Transmitting 40 characters that are to be transferred without a break

Step	Process data	Meaning/note
1	OUT byte 0 = 00$_{hex}$	*1)
2	Wait until bit Tx_buf_not_empty in IN byte 1 = 0	Wait until the send buffer is empty*2)
3	OUT byte 3 ... 19 = the first 17 characters	Transfer transmit data
4	OUT byte 2 = 17	Transfer number of characters to be transmitted
5	OUT byte 0 = 20$_{hex}$	Write “Store characters temporarily” command
6	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
3	OUT byte 3 ... 19 = the next 17 characters	Transfer transmit data
4	OUT byte 2 = 17	Transfer number of characters to be transmitted
5	OUT byte 0 = 60$_{hex}$	Toggle “Store characters temporarily” command
6	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
7	OUT byte 3 ... 8 = the remaining 6 characters	Transfer transmit data
8	OUT byte 2 = 6	Transfer number of characters to be transmitted
9	OUT byte 0 = 10$_{hex}$	Write “Transmit characters” command
10	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
11	Wait until bit Tx_buf_not_empty in IN byte 1 = 0	Wait for confirmation that the transmission process is completed

2.4 Receiving without a specified number of characters

Step	Process data	Meaning/note
1	OUT byte 0 = 00_{hex}	*1)
2	When bit Rx_buf_not_empty in IN byte 1 = 0	Cancel sequence because there is no data in the receive buffer yet
3	OUT byte 0 = 30_{hex}	Write “Read characters” command
4	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
5	IN byte 2 = number of characters read In byte 3 ... 19 = receive data	Take receive data

2.5 Receiving with a specified number of characters (max. 17 characters)

Step	Process data	Meaning/note
1	OUT byte 0 = 00_{hex}	*1)
2	When bit Rx_buf_not_empty in IN byte 1 = 0	Cancel sequence because there is no data in the receive buffer yet
3	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
4	Wait until the IN word 1 = expected number (specification)	Wait until the expected number of characters was received
5	OUT byte 0 = 30_{hex}	Write “Read characters” command
6	Wait until IN byte 0 = OUT byte 0	Wait for confirmation of the command
7	IN byte 2 = number of characters read In byte 3 ... 19 = receive data	Take receive data