ZIEHL industrie – elektronik GmbH + Co KG
Daimlerstraße 13, D – 74523 Schwäbisch Hall
+ 49 791 504-0, info@ziehl.de, www.ziehl.de

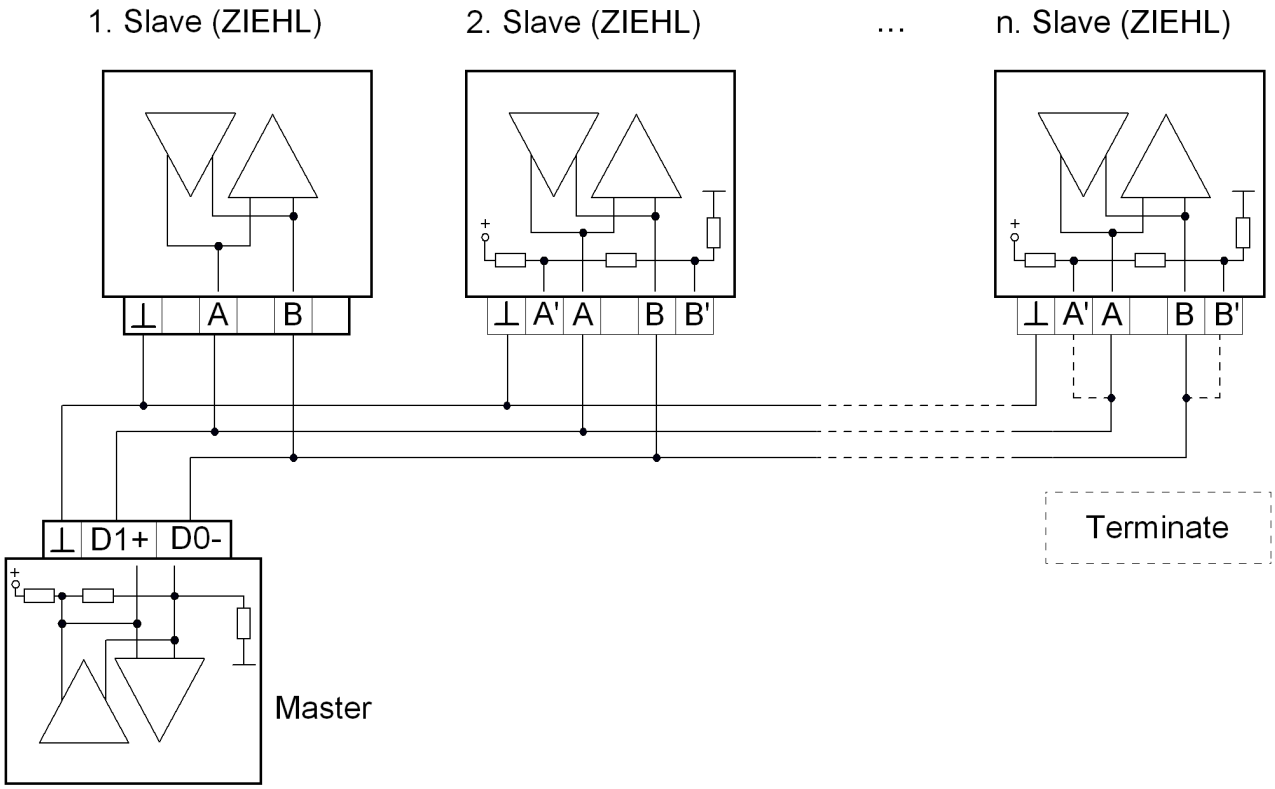
TR600 with RS485 – Appendix 1

updated: 140606 Sc

- RS485 interface with Modbus communication protocol

Table of contents

1 Connection diagram



Connection name	Modbus	Ziehl	EIA/TIA-485
- wire	D0	B (B')	A
+ wire	D1	A (A')	B

2 Important information

Please carefully read the general TR600 operating instructions and comply with the safety instructions.

3 Interface parameter

Baud rate	Data bits	Parity	Stop bit
4800, 9600, 19200, 57600	8	even, odd, none	1 (at Parity none: 2) ➢ from firmware -04: adjustable

The interface parameters are factory set to 9600 baud, 8 bits, even parity, 1 stop bit.

The RTU mode is used.
The TR600 acts in the BUS system as a slave with an adjustable address from 1 to 247.
The TR600 operating instructions describes how to set the parameters.

4 Telegram structure

Slave-address (1 .. 247)	Function	Data	CRC-16 Checksum
1 Byte	1 Byte	n- Bytes	2 Byte

5 Supported function codes

Function code	Name	Utilization
3 (03H)	Read Holding Registers	Read data from registers
16 (10H)	Write Multiple Registers	Write data into the register

5.1 Function code 3 (03H)

➔ Read data from the registers

Query from master

Byte no.	Meaning		1st example	2nd example
1	Slave address		0x01	0x0A
2	function		0x03	0x03
3	Start address	Hi-byte	0x00	0x00
4		Lo-byte	0x01	0x11
5	Number of words (bytes / 2)	Hi-byte	0x00	0x00
6		Lo-byte	0x04	0x02
7	Checksum CRC-16	Lo-byte	0x15	0x95
8		Hi-byte	0xC9	0x75

Reply from slave (TR600)

Byte no.	Meaning		1st example	2nd example
1	Slave address		0x01	0x0A
2	Function		0x03	0x03
3	Number of bytes (n) (Words x 2)		0x08	0x04
4	1st word (2 bytes)	Hi-byte	0x00	0x02
5		Lo-byte	0x32	0x5A
6	2nd word (2 bytes)	Hi-byte	0x00	0xFF
7		Lo-byte	0x3C	0xFB
8	3rd word (2 bytes)	Hi-byte	0x00	
9		Lo-byte	0x46	
10	n- words (2 bytes)	Hi-byte	0x00	
11		Lo-byte	0x50	

: :	: :			
3 + (n + 1)	Checksum CRC-16	Lo-byte	0x37	0x61
3 + (n + 2)		Hi-byte	0xF8	0x2B

5.2 Function code 16 (10H)

➔ Write data in register

Query from master				
Byte no.	**Meaning**		**1st example**	**2nd example**
1	Slave address		0x01	0x0A
2	Function		0x10	0x10
3	Start address	Hi-byte	0x00	0x00
4		Lo-byte	0x07	0x10
5	Number of words (Bytes / 2)	Hi-byte	0x00	0x00
6		Lo-byte	0x04	0x02
7	Number of Bytes (n)		0x08	0x04
8	1st register	Hi-byte	0x00	0x00
9		Lo-byte	0x5A	0x00
10	2nd register	Hi-byte	0xFF	0x00
11		Lo-byte	0xFB	0x64
12	3rd register	Hi-byte	0x00	
13		Lo-byte	0x0A	
14	4th register	Hi-byte	0x00	
15		Lo-byte	0x14	
: :	: :			
7 + (n + 1)	Checksum CRC-16	Lo-byte	0x68	0xD6
7 + (n + 2)		Hi-byte	0x62	0x6C

Reply from slave (TR600)				
Byte no.	**Meaning**		**1st example**	**2nd example**
1	Slave address		0x01	0x0A
2	Function		0x10	0x10
3	Start address	Hi-byte	0x00	0x00
4		Lo-byte	0x07	0x10
5	number of words (n) (Bytes / 2)	Hi-byte	0x00	0x02
6		Lo-byte	0x04	0x02
7	Checksum CRC-16	Lo-byte	0x70	0x40
8		Hi-byte	0x0B	0x16

5.3 Modbus register tables

Register of function code 3 (03H) - Read data from the registers				for function code	
Adr.	Data type	Description / Value range		3 (03H)	16 (10H)
0000	Signed Int	Sensor 1 Preferences	-2 = Pt100 3-wire, -1 = nc (not connected), 0 .. 999 = Pt100 2-wire (line resistance)	x	x
0001	Signed Int	Sensor 2 Preferences		x	x
0002	Signed Int	Sensor 3 Preferences		x	x
0003	Signed Int	Sensor 4 Preferences		x	x
0004	Signed Int	Sensor 5 Preferences		x	x
0005	Signed Int	Sensor 6 Preferences		x	x
0006	Signed Int	Alarm 1: Limit	-199 ... 860 -> -199 … 860 °C	x	x
0007	Signed Int	Alarm 2: Limit		x	x
0008	Signed Int	Alarm 3: Limit		x	x
0009	Signed Int	Alarm 4: Limit		x	x
000A	Signed Int	Alarm 5: Limit		x	x
000B	Signed Int	Alarm 6: Limit		x	x
000C	Signed Int	Alarm 1: Sensor allocation	0 … 5 = Sensor 1 … 6, 6 = S1+2+3, 7 = S4+5, 8 = S4+5+6, 9 = S1+2+3+4+5+6, 10 = S1+2, 11 = S3+4, 12 = S5+6	x	x
000D	Signed Int	Alarm 2: Sensor allocation		x	x
000E	Signed Int	Alarm 3: Sensor allocation		x	x
000F	Signed Int	Alarm 4: Sensor allocation		x	x
0010	Signed Int	Alarm 5: Sensor allocation		x	x
0011	Signed Int	Alarm 6: Sensor allocation		x	x
0012	Signed Int	Alarm 1: Hysteresis	1 ... 99 -> 1 … 99°C	x	x
0013	Signed Int	Alarm 2: Hysteresis		x	x
0014	Signed Int	Alarm 3: Hysteresis		x	x
0015	Signed Int	Alarm 4: Hysteresis		x	x
0016	Signed Int	Alarm 5: Hysteresis		x	x
0017	Signed Int	Alarm 6: Hysteresis		x	x
0018	Signed Int	Alarm 1: Delay Alarm on	1 ... 999 -> 0,1 … 99,9s	x	x
0019	Signed Int	Alarm 2: Delay Alarm on		x	x
001A	Signed Int	Alarm 3: Delay Alarm on		x	x
001B	Signed Int	Alarm 4: Delay Alarm on		x	x
001C	Signed Int	Alarm 5: Delay Alarm on		x	x
001D	Signed Int	Alarm 6: Delay Alarm on		x	x
001E	Signed Int	Alarm 1: Delay Alarm off	0 ... 999 -> 0 … 999s	x	x
001F	Signed Int	Alarm 2: Delay Alarm off		x	x
0020	Signed Int	Alarm 3: Delay Alarm off		x	x
0021	Signed Int	Alarm 4: Delay Alarm off		x	x
0022	Signed Int	Alarm 5: Delay Alarm off		x	x
0023	Signed Int	Alarm 6: Delay Alarm off		x	x
0024	Signed Int	Alarm 1: Auto reset	0 = auto reset, 1 = locked	x	x
0025	Signed Int	Alarm 2: Auto reset		x	x
0026	Signed Int	Alarm 3: Auto reset		x	x
0027	Signed Int	Alarm 4: Auto reset		x	x
0028	Signed Int	Alarm 5: Auto reset		x	x
0029	Signed Int	Alarm 6: Auto reset		x	x
002A	Signed Int	Alarm 1: Function	1 = Relays on when exceeded limit 2 = Relays off when exceeded limit 3 = Relay on when undershot limit 4 = Relay off when undershot limit	x	x
002B	Signed Int	Alarm 2: Function		x	x
002C	Signed Int	Alarm 3: Function		x	x
002D	Signed Int	Alarm 4: Function		x	x
002E	Signed Int	Alarm 5: Function		x	x
002F	Signed Int	Alarm 6: Function		x	x
0030	Signed Int	Error function	1 = On error Relay on 2 = On error Relay off	x	x
0031	Signed Int	Code save	0 = inactive, 1 = active	x	x

0032	Signed Int	Sensor 1:	Max. value -> write 1 = Reset value	x	x
0033	Signed Int	Sensor 2:		x	x
0034	Signed Int	Sensor 3:		x	x
0035	Signed Int	Sensor 4:		x	x
0036	Signed Int	Sensor 5:		x	x
0037	Signed Int	Sensor 6:		x	x
0038	Signed Int	Sensor group 1, 2, 3:		x	x
0039	Signed Int	Sensor group 4, 5:		x	x
003A	Signed Int	Sensor group 4, 5, 6:		x	x
003B	Signed Int	Sensor group 1 – 6:		x	x
003C	Signed Int	Sensor group 1, 2:		x	x
003D	Signed Int	Sensor group 3, 4:		x	x
003E	Signed Int	Sensor group 5, 6:		x	x
003F	Signed Int	Sensor 1:	Min. value -> write 1 = Reset value	x	x
0040	Signed Int	Sensor 2:		x	x
0041	Signed Int	Sensor 3:		x	x
0042	Signed Int	Sensor 4:		x	x
0043	Signed Int	Sensor 5:		x	x
0044	Signed Int	Sensor 6:		x	x
0045	Signed Int	Sensor group 1, 2, 3:		x	x
0046	Signed Int	Sensor group 4, 5:		x	x
0047	Signed Int	Sensor group 4, 5, 6:		x	x
0048	Signed Int	Sensor group 1 – 6:		x	x
0049	Signed Int	Sensor group 1, 2:		x	x
004A	Signed Int	Sensor group 3, 4:		x	x
004B	Signed Int	Sensor group 5, 6:		x	x
004C	Signed Int	Sensor 1:	Measurement (in °C) -> Sensor error: - 32767 = Short circuit - 32766 = Interrupt - 32748 = Not assigned	x	
004D	Signed Int	Sensor 2:		x	
004E	Signed Int	Sensor 3:		x	
004F	Signed Int	Sensor 4:		x	
0050	Signed Int	Sensor 5:		x	
0051	Signed Int	Sensor 6:		x	
0052	Signed Int	Sensor group 1, 2, 3:	Measurement (in °C) -> Value of warmest sensor of group	x	
0053	Signed Int	Sensor group 4, 5:		x	
0054	Signed Int	Sensor group 4, 5, 6:		x	
0055	Signed Int	Sensor group 1 – 6:		x	
0056	Signed Int	Sensor group 1, 2:		x	
0057	Signed Int	Sensor group 3, 4:		x	
0058	Signed Int	Sensor group 5, 6:		x	
0059	Signed Int	Alarm 1: Status	0 = off, 1 = delay Alarm, 2 = on, 3 = delay Alarm off, 4 = locked	x	
005A	Signed Int	Alarm 2: Status		x	
005B	Signed Int	Alarm 3: Status		x	
005C	Signed Int	Alarm 4: Status		x	
005D	Signed Int	Alarm 5: Status		x	
005E	Signed Int	Alarm 6: Status		x	
005F	Signed Int	Alarm 7: Status	0 = off, 1 = on	x	
0060	Signed Int	Relay K 1: Status	0 = off 1 = on	x	
0061	Signed Int	Relay K 2: Status		x	
0062	Signed Int	Relay K 3: Status		x	
0063	Signed Int	Relay K 4: Status		x	
0064	Signed Int	Relay K 5: Status		x	
0065	Signed Int	Relay K 6: Status		x	
0066	Signed Int	Relay K 7: Status		x	
0067	Unsigned Int	Software version 12360-14xx-yy -> HiByte = xx, LoByte = yy		x	

Parameters

Measurements / status displays

6 Error messages

The telegram sent from the master is checked by the slave (TR600).
During a malfunction, an error message is generated and sent back to the master. While doing so, the 7th bit is set in function byte to "1".

Error telegram:

Byte no.	Meaning		1st example	2nd example
1	Slave address		0x01	0x0A
2	Function		0x81	0x90
3	Error code		0x02	0x03
4	Checksum CRC-16	Lo-byte	0xC1	0x7D
5		Hi-byte	0x91	0xC3

The following error codes are possible:

- 1 (01H) Invalid function
- 2 (02H) Invalid start address
- 3 (03H) Invalid data value
- 4 (04H) Slave device error

Error not detected by slave (telegram will be rejected):

- False checksum CRC-16
- Unknown slave address

7 Checksum CRC-16

The checksum is attached to each Modbus telegram and is used to detect transmission errors. It is 2 bytes long and is calculated from all bytes in a telegram. During this, first the lo-byte and then the hi-byte are transmitted.

Please refer to the original Modbus documentation for more details, which can be found at **http://www.modbus.org**