

# TR600 mit RS485 – Anlage 1

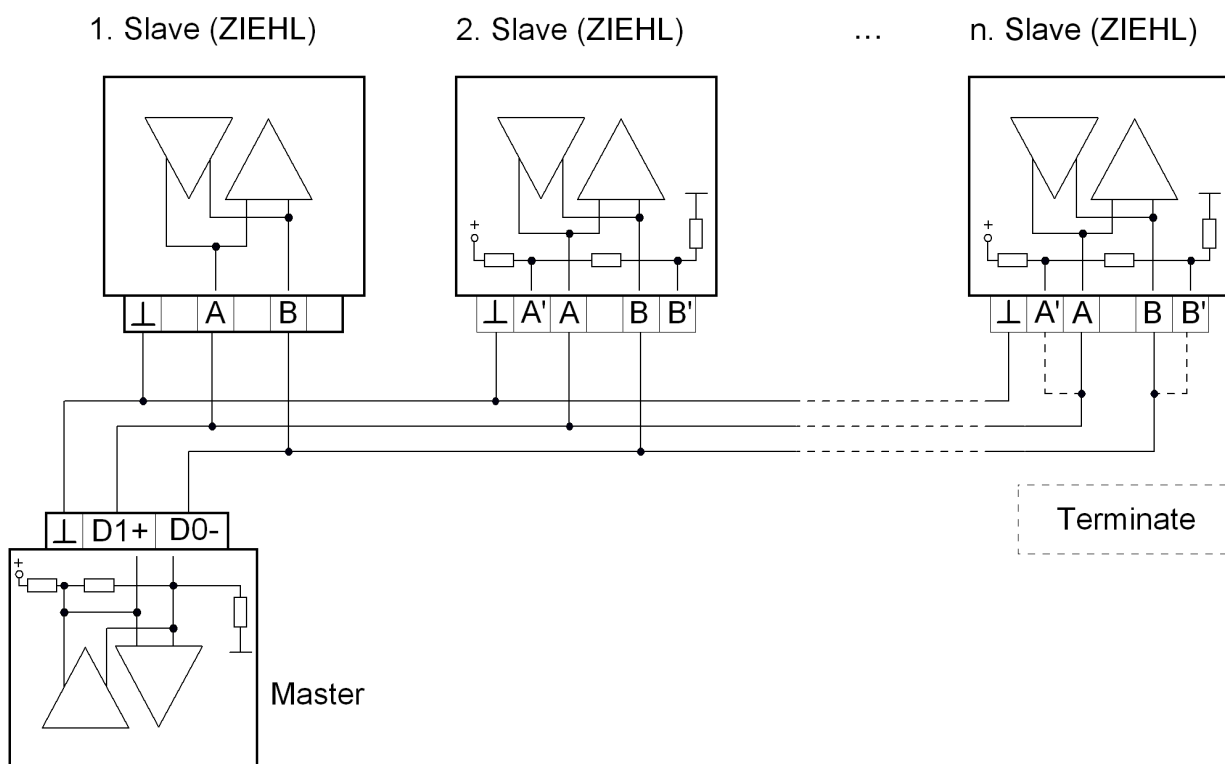
Stand: 140520 Sc

## - RS485 Schnittstelle mit Modbus Kommunikationsprotokoll

### Inhaltsverzeichnis

<b>1 Anschlussplan</b> .....	<b>1</b>
<b>2 Wichtige Hinweise</b> .....	<b>2</b>
<b>3 Schnittstellenparameter</b> .....	<b>2</b>
<b>4 Telegramm Aufbau</b> .....	<b>2</b>
<b>5 Unterstützte Funktionscodes</b> .....	<b>2</b>
5.1 Funktionscode 3 (03H).....	2
5.2 Funktionscode 16 (10H).....	3
5.3 Modbus Registertabellen.....	4
<b>6 Fehlermeldungen</b> .....	<b>6</b>
<b>7 Prüfsumme CRC-16</b> .....	<b>6</b>

## 1 Anschlussplan



Anschluss-Bezeichnung	Modbus	Ziehl	EIA/TIA-485
- Leitung	D0	B (B')	A
+ Leitung	D1	A (A')	B

## 2 Wichtige Hinweise

Bitte lesen Sie auch die allgemeine Betriebsanleitung des TR600 sorgfältig durch und beachten Sie die Sicherheitshinweise.

## 3 Schnittstellenparameter

Baudrate	Datenbits	Parität	Stoppsbit
4800, 9600, 19200, 57600	8	even, odd, none	1 (bei Parität none: 2) ➤ ab Firmware -04: einstellbar

Die Schnittstellenparameter sind werkseitig eingestellt auf 9600 Baud, 8 Bits, even Parität, 1 Stoppsbit.

### Verwendet wird der RTU-Mode.

Das TR600 agiert im BUS-System als Slave mit einer einstellbaren Adresse von 1 bis 247.

**Das Einstellen der Parameter wird in der Betriebsanleitung des TR600 beschrieben.**

## 4 Telegramm Aufbau

Slave-Adresse (1 .. 247)	Funktion	Daten	CRC-16 Prüfsumme
1 Byte	1 Byte	n- Bytes	2 Byte

## 5 Unterstützte Funktionscodes

Funktionscode	Bezeichnung	Verwendung
3 (03H)	Read Holding Registers	Daten aus den Registern lesen
16 (10H)	Write Multiple Registers	Daten in die Register schreiben

### 5.1 Funktionscode 3 (03H)

➔ Daten aus Registern lesen

Anfrage vom Master				
Byte Nr.	Bedeutung		1. Beispiel	2. Beispiel
1	Slave-Adresse		0x01	0x0A
2	Funktion		0x03	0x03
3	Start-Adresse	Hi-Byte	0x00	0x00
4		Lo-Byte	0x01	0x11
5	Anzahl- Worte (Bytes / 2)	Hi-Byte	0x00	0x00
6		Lo-Byte	0x04	0x02
7	Prüfsumme CRC-16	Lo-Byte	0x15	0x95
8		Hi-Byte	0xC9	0x75

Antwort vom Slave (TR600)				
Byte Nr.	Bedeutung		1. Beispiel	2. Beispiel
1	Slave-Adresse		0x01	0x0A
2	Funktion		0x03	0x03
3	Anzahl Bytes (n) (Worte x 2)		0x08	0x04
4	1. Wort (2 Bytes)	Hi-Byte	0x00	0x02
5		Lo-Byte	0x32	0x5A
6	2. Wort (2 Bytes)	Hi-Byte	0x00	0xFF
7		Lo-Byte	0x3C	0xFB
8	3. Wort (2 Bytes)	Hi-Byte	0x00	
9		Lo-Byte	0x46	
10	n- Worte	Hi-Byte	0x00	

11	(2 Bytes)	Lo-Byte	0x50	
⋮	⋮			
3 + (n + 1)	Prüfsumme CRC-16	Lo-Byte	0x37	0x61
3 + (n + 2)		Hi-Byte	0xF8	0x2B

## 5.2 Funktionscode 16 (10H)

→ Daten in Register schreiben

<b>Anfrage vom Master</b>				
Byte Nr.	Bedeutung		1. Beispiel	2. Beispiel
1	Slave-Adresse		0x01	0x0A
2	Funktion		0x10	0x10
3	Start- Adresse	Hi-Byte	0x00	0x00
4		Lo-Byte	0x07	0x10
5	Anzahl- Worte (Bytes / 2)	Hi-Byte	0x00	0x00
6		Lo-Byte	0x04	0x02
7	Anzahl Bytes (n)		0x08	0x04
8	1. Register	Hi-Byte	0x00	0x00
9		Lo-Byte	0x5A	0x00
10	2. Register	Hi-Byte	0xFF	0x00
11		Lo-Byte	0xFB	0x64
12	3. Register	Hi-Byte	0x00	
13		Lo-Byte	0x0A	
14	4. Register	Hi-Byte	0x00	
15		Lo-Byte	0x14	
⋮	⋮			
7 + (n + 1)	Prüfsumme CRC-16	Lo-Byte	0x68	0xD6
7 + (n + 2)		Hi-Byte	0x62	0x6C

<b>Antwort vom Slave (TR600)</b>				
Byte Nr.	Bedeutung		1. Beispiel	2. Beispiel
1	Slave-Adresse		0x01	0x0A
2	Funktion		0x10	0x10
3	Start- Adresse	Hi-Byte	0x00	0x00
4		Lo-Byte	0x07	0x10
5	Anzahl Worte (n) (Bytes / 2)	Hi-Byte	0x00	0x02
6		Lo-Byte	0x04	0x02
7	Prüfsumme CRC-16	Lo-Byte	0x70	0x40
8		Hi-Byte	0x0B	0x16

### 5.3 Modbus Registertabellen

Register von Funktionscodes 3 (03H) / 16 (10H)				Betrifft Funktionscode	
Adr.	Datentyp	Beschreibung / Wertebereich		3 (03H)	16 (10H)
0000	Signed Int	Sensor 1	Einstellungen	x	x
0001	Signed Int	Sensor 2	Einstellungen	x	x
0002	Signed Int	Sensor 3	Einstellungen	x	x
0003	Signed Int	Sensor 4	Einstellungen	x	x
0004	Signed Int	Sensor 5	Einstellungen	x	x
0005	Signed Int	Sensor 6	Einstellungen	x	x
0006	Signed Int	Alarm 1:	Limit	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:	Sensorzuordnung	x	x
000D	Signed Int	Alarm 2:	Sensorzuordnung	x	x
000E	Signed Int	Alarm 3:	Sensorzuordnung	x	x
000F	Signed Int	Alarm 4:	Sensorzuordnung	x	x
0010	Signed Int	Alarm 5:	Sensorzuordnung	x	x
0011	Signed Int	Alarm 6:	Sensorzuordnung	x	x
0012	Signed Int	Alarm 1:	Hysterese	x	x
0013	Signed Int	Alarm 2:	Hysterese	x	x
0014	Signed Int	Alarm 3:	Hysterese	x	x
0015	Signed Int	Alarm 4:	Hysterese	x	x
0016	Signed Int	Alarm 5:	Hysterese	x	x
0017	Signed Int	Alarm 6:	Hysterese	x	x
0018	Signed Int	Alarm 1:	Verzögerung Alarm ein	x	x
0019	Signed Int	Alarm 2:	Verzögerung Alarm ein	x	x
001A	Signed Int	Alarm 3:	Verzögerung Alarm ein	x	x
001B	Signed Int	Alarm 4:	Verzögerung Alarm ein	x	x
001C	Signed Int	Alarm 5:	Verzögerung Alarm ein	x	x
001D	Signed Int	Alarm 6:	Verzögerung Alarm ein	x	x
001E	Signed Int	Alarm 1:	Verzögerung Alarm aus	x	x
001F	Signed Int	Alarm 2:	Verzögerung Alarm aus	x	x
0020	Signed Int	Alarm 3:	Verzögerung Alarm aus	x	x
0021	Signed Int	Alarm 4:	Verzögerung Alarm aus	x	x
0022	Signed Int	Alarm 5:	Verzögerung Alarm aus	x	x
0023	Signed Int	Alarm 6:	Verzögerung Alarm aus	x	x
0024	Signed Int	Alarm 1:	Autoreset	x	x
0025	Signed Int	Alarm 2:	Autoreset	x	x
0026	Signed Int	Alarm 3:	Autoreset	x	x
0027	Signed Int	Alarm 4:	Autoreset	x	x
0028	Signed Int	Alarm 5:	Autoreset	x	x
0029	Signed Int	Alarm 6:	Autoreset	x	x
002A	Signed Int	Alarm 1:	Funktion	x	x
002B	Signed Int	Alarm 2:	Funktion	x	x
002C	Signed Int	Alarm 3:	Funktion	x	x
002D	Signed Int	Alarm 4:	Funktion	x	x
002E	Signed Int	Alarm 5:	Funktion	x	x
002F	Signed Int	Alarm 6:	Funktion	x	x
0030	Signed Int	Errorfunktion	1 = Relais ein bei Fehler, 2 = Relais aus bei Fehler	x	x
0031	Signed Int	Codesperre	0 = inaktiv, 1 = aktiv	x	x

0032	Signed Int	Sensor 1:	Max- Werte -> write 1 = Reset Wert	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	Sensorgruppe 1, 2, 3:		x	x
0039	Signed Int	Sensorgruppe 4, 5:		x	x
003A	Signed Int	Sensorgruppe 4, 5, 6:		x	x
003B	Signed Int	Sensorgruppe 1 – 6:		x	x
003C	Signed Int	Sensorgruppe 1, 2:		x	x
003D	Signed Int	Sensorgruppe 3, 4:		x	x
003E	Signed Int	Sensorgruppe 5, 6:		x	x
003F	Signed Int	Sensor 1:		Min- Werte -> write 1 = Reset Wert	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	Sensorgruppe 1, 2, 3:	x		x
0046	Signed Int	Sensorgruppe 4, 5:	x		x
0047	Signed Int	Sensorgruppe 4, 5, 6:	x		x
0048	Signed Int	Sensorgruppe 1 – 6:	x		x
0049	Signed Int	Sensorgruppe 1, 2:	x		x
004A	Signed Int	Sensorgruppe 3, 4:	x		x
004B	Signed Int	Sensorgruppe 5, 6:	x		x
004C	Signed Int	Sensor 1:	Ist- Messwert (Temperatur in °C) -> Werte bei Sensorfehlern: - 32767 = Sensor Kurzschluss - 32766 = Sensor Unterbrechung - 32748 = Sensor nc		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	Sensorgruppe 1, 2, 3:		Ist- Messwert (Temperatur in °C) -> Messwert des wärmsten Sensors einer Sensorgruppe	x
0053	Signed Int	Sensorgruppe 4, 5:	x		
0054	Signed Int	Sensorgruppe 4, 5, 6:	x		
0055	Signed Int	Sensorgruppe 1 – 6:	x		
0056	Signed Int	Sensorgruppe 1, 2:	x		
0057	Signed Int	Sensorgruppe 3, 4:	x		
0058	Signed Int	Sensorgruppe 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	Relais K 1: Status	0 = off 1 = on	x	
0061	Signed Int	Relais K 2: Status		x	
0062	Signed Int	Relais K 3: Status		x	
0063	Signed Int	Relais K 4: Status		x	
0064	Signed Int	Relais K 5: Status		x	
0065	Signed Int	Relais K 6: Status		x	
0066	Signed Int	Relais K 7: Status		x	
0067	Unsigned Int	Softwareversion 12360-14xx-yy -> HiByte = xx, LoByte = yy		x	

Parameter

Messwerte / Statusanzeigen

## 6 Fehlermeldungen

Das vom Master gesendete Telegramm wird vom Slave (TR600) geprüft. Im Fehlerfall wird eine Fehlermeldung generiert und an den Master zurückgesendet. Dabei wird das 7. Bit im Funktionsbyte auf „1“ gesetzt.

### Fehlertelegramm:

Byte Nr.	Bedeutung		1. Beispiel	2. Beispiel
1	Slave-Adresse		0x01	0x0A
2	Funktion		0x81	0x90
3	Fehlercode		0x02	0x03
4	Prüfsumme CRC-16	Lo-Byte	0xC1	0x7D
5		Hi-Byte	0x91	0xC3

### Folgende Fehlercodes sind möglich:

- 1 (01H) Ungültige Funktion
- 2 (02H) Ungültige Startadresse
- 3 (03H) Ungültiger Datenwert
- 4 (04H) Slave-Gerätefehler

### Fehler welche vom Slave nicht erkannt werden (Telegramm wird verworfen):

- Falsche Prüfsumme CRC-16
- Unbekannte Slave- Adresse

## 7 Prüfsumme CRC-16

Die Prüfsumme wird an jedes Modbus- Telegramm angehängt und dient der Erkennung von Übertragungsfehlern. Sie ist 2 Byte lang und wird aus allen Bytes eines Telegramms berechnet. Dabei werden als erstes das Lo-Byte und dann das Hi-Byte übertragen.

Details entnehmen Sie bitte der Modbus Originaldokumentation, zu finden unter <http://www.modbus.org>