Table of Contents

english

1 Description

These operating instructions describe the optional RS232/RS485 interfaces and their protocols.

2 Important notice

Read both these operating instructions relating to the optional interfaces and the operating instructions of the basic device before operating the device.

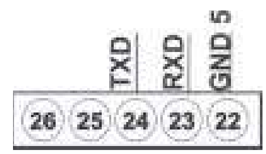
3 Interface

Signal lines are connected via 5 screw terminals with a spacing of 3.5 mm.

3.1 Optional RS232 board

The RS232 interface allows a point to point connection. No data flow control is supported.

Connections

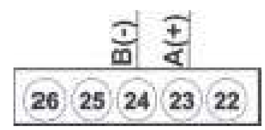


The maximum allowed cable length is 15 meters, or the cable length compatible with a capacity of 2500 pF.
The ground of the interface GND 5 is connected to the ground of the device.

3.2 Optional RS485 board

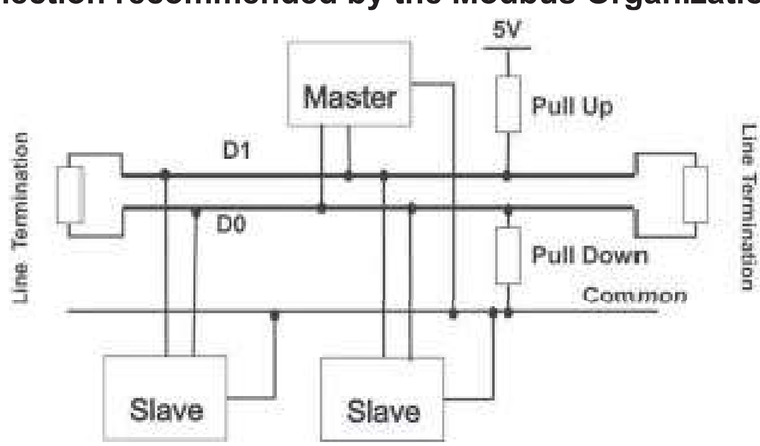
The RS485 interface allows a multipoint connection. Up to 32 devices may be connected to a bus segment.

Connections



The use of low-capacity and low-attenuation twisted pair cable allows achieving extremely reliable connections over long distances (maximum 1000 m).

3.3 Device connection recommended by the Modbus Organization [1]



Description	
MODBUS	**Device**
D0	B (-)
D1	A (+)
Common	

3.4 Interface configuration

MODBUS (RTU) Slave configuration

Interface	Baud rate	Data format	Address
RS232	9600	1 Start bit 8 Data bits Even Parity 1 Stop bit	1-247
RS485	9600	1 Start bit 8 Data bits Even Parity 1 Stop bit	1-247

CR/LF Master configuration

Interface	Baud rate	Data format	Address
RS232	9600	1 Start bit 8 Data bits No Parity 1Stop bit	1-99
RS485	9600	1 Start bit 8 Data bits No Parity 1Stop bit	1-99

3.5 Programming

INTERF.	Main interface programming menu.
PROTOC.	**Interface protocols**
MODBUS	MODBUS protocol
CR.LF	CR/LF protocol
ADRESS. 1	**Slave Address** CR/LF: 1 – 99 MODBUS: 1- 247
PR.TIME 0.5	**Cycle time (only for CR/LF)** 0,5 – 9999,9 sec (ON) 0 sec (OFF)
SOURCE	**Data sources (only for CR/LF)**
MAIN	Main counter
BATCH	Batch counter
TOTAL	Totalizer
MAI.BAT	Main and batch counter
MAI.TOT	Main counter and totalizer

The data sources can be set according to the output operations set for the device. If the concerned counter shows an overflow / underflow, it does not send the data value, but +oooooo / +uuuuuu.

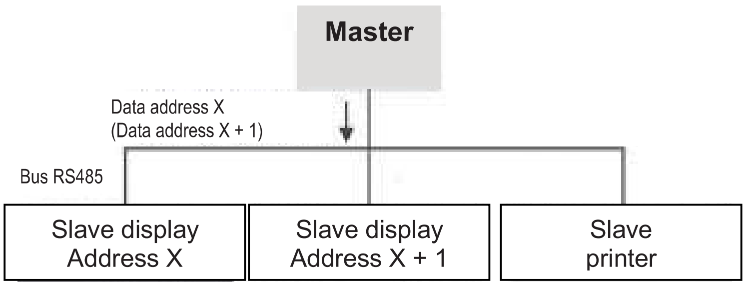
INPUT	Main menu for programming the signal and control inputs
MPI.INP.1	**User input 1**
MPI.INP.2	**User input 2**
PRINT	Triggering of the data transmission via the user input; possible cycle time ≥ 0, 5 sec (possible only for CR/LF and cycle time = 0)

Factory settings are highlighted in grey.

english

4 CR/LF protocol

In the CR/LF protocol, the device can transmit data values to a slave device according to the data source set.



4.1 Transmission

The data bytes are coded in ASCII.
Address 1: Slave address set in the device
Address 2: Slave address incremented by 1
Data value: 6 digits (+ decimal point)
Text: additional transmitted text for data source setting d) or e)

4.1.1 Transmission principle for data source main counter, batch counter or totalizer

Transmission: main or secondary counter

Address 1	Space	Sign	Data value	Carriage Return	Line Feed
2 bytes	1 byte	1 byte	6 or 7 bytes	1 byte	1 byte

Example for data source main counter:

01	**[SPACE]**	**-**	**123456**	**[CR]**	**[LF]**
3031_{hex}	20_{hex}	$2D_{hex}$	3132 3334 3536_{hex}	$0D_{hex}$	$0A_{hex}$

05	**[SPACE]**	**+**	**oooooo**	**[CR]**	**[LF]**
3035_{hex}	20_{hex}	$2B_{hex}$	6F6F 6F6F $6F6F_{hex}$	$0D_{hex}$	$0A_{hex}$

Example for data source totalizer:

01	**[SPACE]**	**+**	**000.456**	**[CR]**	**[LF]**
3031_{hex}	20_{hex}	$2B_{hex}$	30 3030 2E34 3536_{hex}	$0D_{hex}$	$0A_{hex}$

Bold: **ASCII**
Normal: HEX value

4.1.2 Transmission principle for data source main and batch counter / main counter and totalizer

Transmission: main and secondary counter

Address 1	Space	Text	Space	Sign	Data value	Carriage Return	Line Feed
2 bytes	1 byte	4 bytes	1 byte	1 byte	6 or 7 bytes	1 byte	1 byte

Address 2	Space	Text	Space	Sign	Data value	Carriage Return	Line Feed
2 bytes	1 byte	5 bytes	1 byte	1 byte	6 or 7 bytes	1 byte	1 byte

Example for data source main and batch counter:

15	**[SPACE]**	**MAIN**	**[SPACE]**	**+**	**000259**	**[CR]**	**[LF]**
3135_{hex}	20_{hex}	$4D41\ 494E_{hex}$	20_{hex}	$2B_{hex}$	$3030\ 3032\ 3539_{hex}$	$0D_{hex}$	$0A_{hex}$

16	**[SPACE]**	**BATCH**	**[SPACE]**	**+**	**999999**	**[CR]**	**[LF]**
3136_{hex}	20_{hex}	$42\ 4154\ 4348_{hex}$	20_{hex}	$2B_{hex}$	$3939\ 3939\ 3939_{hex}$	$0D_{hex}$	$0A_{hex}$

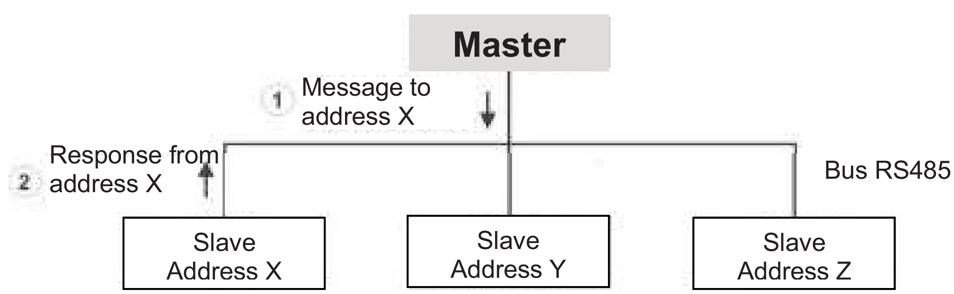
Bold: **ASCII**
Normal: HEX value

5 MODBUS protocol

The MODBUS data transmission in binary form takes place in the RTU (Remote Terminal Unit) operating mode, via the serial interface (RS485 or RS232). The protocol is implemented following the specifications (*MODBUS Application Protocol Specification V1.1b* and *MODBUS over Serial Line – Specification and Implementation Guide V1.02*.) of the MODBUS Organization.

5.1 Master - Slave principle

The protocol operates according to the Master-Slave principle. A Master can communicate with one or several Slaves. Only the slave explicitly addressed by the Master is allowed to send data back to the Master.



5.2 Transmission principles

The protocol has two different transmission principles.

Unicast mode

In Unicast mode, the Master addresses a determined Slave. The Slave processes the message and sends an answer back to the Master. Every Slave has a univocal address in the address range from 1 to 247. The transmission principle always consists in a request from the Master and the subsequent response from the Slave. If the Master does not receive a response within a defined period of time (Timeout), the Master can assume that errors occurred during the transmission.

Broadcast mode

In Broadcast mode, the Master sends a write instruction (request) to all Slave devices, which do not generate any response. Broadcast addressing uses address 0.

5.3 Timeout

If the Master device does not receive a response within 0.5 seconds, it can discard the last request.

5.4 Message cycle

A message cycle ≤ 0.5 sec. is supported.

5.5 Byte order

The Modbus protocol uses the Big Endian format.

5.6 Structure of a message

Slave Address	Function code	Data	CRC
1 byte	1 byte	N bytes	2 bytes

Pauses within the transmission of a message

Pauses with a length >= 1.5 characters between the single message bytes lead to an error in the data frame.

Start and end identification

A message pause with a length of 3.5 characters is specified as start and end identification of a message. This pause must be respected between the single messages.

5.7 Function codes

The following function codes are supported.

Function code	description
03_{hex}	Read several registers
10_{hex}	Write several registers
11_{hex}	Identification of the Slave

5.8 Data values

The 32 bit data values are transmitted via MODBUS either in the Integer or in the Float format. The register address defines the format to be used.

5.8.1 Float format (32 bits)

The Float values are coded in compliance with the IEEE 754 standard (single floating point) and have the following structure.

Address X + 1		Address X	
SEEEEEEE	**EMMMMMMM**	**MMMMMMMM**	**MMMMMMMM**
Byte 4	Byte 3	Byte 2	Byte 1

$$X = S \times M^E$$

S: Sign
E: Exponent
M: Mantissa
X: Total value

The data conversion in the floating point representation generates conversion errors (rounding error) due to the limited number of mantissa bits.

Machine precision

The relative machine precision indicates the maximum relative error that can be generated during rounding (conversion). For the Float values (single floating point), the machine precision is $\varepsilon_0 = 5{,}96 * 10^{-8}$.

5.8.2 Integer format (32 bits)

The Integer format is used in the representation as a two's complement. The device interprets the transmitted integer value. If the data value is represented in the device with the set number of decimal places, the decimal point is set for the integer in compliance with the set number of decimal places.

Example for a setting to 3 decimal places:

Transmitted integer value: 00000010_{hex}
Interpreted value: 000.016_{dec}

Address X + 1		Address X	
00_{hex}	**00_{hex}**	**00_{hex}**	**10_{hex}**
Byte 4	Byte 3	Byte 2	Byte 1

5.9 CRC (cyclic redundancy check)

A 16 bit CRC value is calculated according to a defined process for every data block and appended to the data block. For data check purposes, the same calculation process is applied to the data block, including the appended CRC value. If the result is zero, it can be assumed that the data block has not been corrupted.

CRC generation algorithm:

1. Loading of the value $FFFF_{hex}$ in the 16 bit CRC register
2. Exclusive OR operation on the first byte sent or received with the low byte of the CRC register; store the result in the CRC register
3. Shift the CRC register 1 bit to the right and fill the msb with 0
4. If a 0 has been shifted out in step 3, this step is repeated. If a 1 has been shifted out in step 3, an exclusive OR operation is carried out on the CRC register with the value $A001_{hex}$.
5. Steps 3 and 4 are repeated until 8 bit shifts have been performed.
6. Steps 2 to 5 are repeated for the other bytes sent or received.
7. The result of the CRC generation is now in the CRC register. They are appended to the sent message after a byte exchange.

msb: most significant bit

english

5.10 Reading several registers

This instruction allows reading 16-bit registers.

XX_{hex} valid value in Hex-Format
LSB: least significant byte
MSB: most significant byte

Instruction (Master):

Field name	Value
Slave address	XX_{hex}
Function Read	03_{hex}
Start at register address (MSB)	XX_{hex}
Start at register address (LSB)	XX_{hex}
Number of registers (MSB)	XX_{hex}
Number of registers (LSB)	XX_{hex}
CRC value (LSB)	XX_{hex}
CRC value (MSB)	XX_{hex}

Response (Slave):

Field name	Value
Slave address	XX_{hex}
Function Read	03_{hex}
Number of data bytes	XX_{hex}
Register value 1 (MSB)	XX_{hex}
Register value 1 (LSB)	XX_{hex}
…	…
Register value N (MSB)	XX_{hex}
Register value N (LSB)	XX_{hex}
CRC value (LSB)	XX_{hex}
CRC value (MSB)	XX_{hex}

5.11 Writing several registers

This instruction allows writing 16-bit registers.

Instruction (Master):

Field name	Value
Slave address	XX_{hex}
Function Write	10_{hex}
Start at register address (MSB)	XX_{hex}
Start at register address (LSB)	XX_{hex}
Number of registers (MSB)	XX_{hex}
Number of registers (LSB)	XX_{hex}
Number of data bytes (2 * number of registers)	XX_{hex}
Register value 1 (MSB)	XX_{hex}
Register value 1 (LSB)	XX_{hex}
…	…
Register value N (MSB)	XX_{hex}
Register value N (LSB)	XX_{hex}
CRC value (LSB)	XX_{hex}
CRC value (MSB)	XX_{hex}

Response (Slave):

Field name	Value
Slave address	XX_{hex}
Function Write	10_{hex}
Start at register address (MSB)	XX_{hex}
Start at register address (LSB)	XX_{hex}
Number of registers (MSB)	XX_{hex}
Number of registers (LSB)	XX_{hex}
CRC value (LSB)	XX_{hex}
CRC value (MSB)	XX_{hex}

5.12 Identification

This instruction allows reading the identification of the Slave.

Both the Slave ID and the software version are transmitted in ASCII format.

Instruction (Master):

Field name	Value
Slave address	XX_{hex}
Function Identification	11_{hex}
CRC value (LSB)	XX_{hex}
CRC value (MSB)	XX_{hex}

Response (Slave):

Field name	Value
Slave address	XX_{hex}
Function Identification	11_{hex}
Number of data bytes (MSB)	00_{hex}
Number of data bytes (LSB)	11_{hex}
Slave ID byte 1	XX_{hex}
Slave ID byte 2	XX_{hex}
Slave ID byte 3	XX_{hex}
Slave ID byte 4	XX_{hex}
Slave ID byte 5	XX_{hex}
Slave ID byte 6	XX_{hex}
Slave ID byte 7	XX_{hex}
Slave ID byte 8	XX_{hex}
Status	FF_{hex}
Software Version byte 1	XX_{hex}
Software Version byte 2	XX_{hex}
Software Version byte 3	XX_{hex}
Software Version byte 4	XX_{hex}
Software Version byte 5	XX_{hex}
Software Version byte 6	XX_{hex}
Software Version byte 7	XX_{hex}
Software Version byte 8	XX_{hex}
CRC (LSB)	XX_{hex}
CRC (MSB)	XX_{hex}

5.13 Error protocol

The Slave sends no response if it receives a message sent by the Master including transmission errors.
Detected transmission errors:

- Parity errors
- No stop bit detected
- Error in the data frame
- Overrun error (data buffer overflow)
- CRC error

If the Slave device cannot carry out the uncorrupted message it received, it sends back to the Master an error response including the error code.

Error codes:

Code	Name	Description
01_{hex}	Function not allowed	The function code is not defined in the device.
02_{hex}	Address not allowed	The address is not available.
03_{hex}	Data value not allowed	The received data value cannot be written. Data structure / data length is not correct.
04_{hex}	Device error	Data value / data format is not correct. The device cannot or can only partially process the instruction.
10_{hex}	Err1	Set value smaller than 0 is not allowed.
11_{hex}	Err2	Set value larger than Preset2 is not allowed.

Error response:

Description	Value
Slave address	XX_{hex}
Function	80_{hex} + Function code
Error code	XX_{hex}
CRC value (LSB)	XX_{hex}
CRC value (MSB)	XX_{hex}

english

5.14 Registers

All data values extend over 2 MODBUS registers and can only be processed as a whole.

Float register (4 bytes)

Address	Value	Access	Description	Note
0000_{hex}	Main counter	r/w	Writing any value resets the main counter.	
0002_{hex}	Secondary counter	r/w	Writing a value resets the main counter and the secondary counter.	
0004_{hex}	Preset 1	r/w	Preset 1	
0006_{hex}	Preset 2	r/w	Preset 2	
0008_{hex}	Multiplication factor	-/w	Multiplication factor	
$000A_{hex}$	Division factor	-/w	Division factor	
$000C_{hex}$	Store set value	-/w	Storage of the set value (no setting is performed)	
$000E_{hex}$	Perform set function	-/w	Setting (writing any value performs the setting)	
0010_{hex}	Preset 1 sign setting	-/w	Byte 1: Sign value Byte 2: 0 Byte 3: 0 Byte 4: 0	Sign values: 1: + values 2: -- values 3: +/- values
0012_{hex}	Decimal point setting	r/w	Byte 1:decimal places Byte 2: 0 Byte 3: 0 Byte 4: 0	Decimal places: 0: 0 decimal places 1: 1 decimal place 2: 2 decimal places 3: 3 decimal places 4: 4 decimal places 5: 5 decimal places
0014_{hex}	Status	r/o	Byte 1: bit1: output 1 bit2: output 2 Byte 2: bit1 - bit4: main counter bit5 - bit8: secondary counter Byte 3: 0 Byte 4: 0	Output: 1: ON 0: OFF Counter status: 0: counter status: regular 1 :counter status: overflow 2: counter status: underflow

r/o: Read-only access
r/w: Read and write access
-/w: Write access

Integer register (4 bytes)

Address	Value	Access	Description	Note
8000_{hex}	Main counter	r/w	Writing any value resets the main counter.	
8002_{hex}	Secondary counter	r/w	Writing a value resets the main counter and the secondary counter.	
8004_{hex}	Preset 1	r/w	Preset 1	
8006_{hex}	Preset 2	r/w	Preset 2	
8008_{hex}	Multiplication factor	-/w	Multiplication factor	
$800A_{hex}$	Division factor	-/w	Division factor	
$800C_{hex}$	Store set value	-/w	Storage of the set value (no setting is performed)	
$800E_{hex}$	Perform set function	-/w	Setting (writing any value performs the setting)	
8010_{hex}	Preset 1 sign setting	-/w	Byte 1: Sign value Byte 2: 0 Byte 3: 0 Byte 4: 0	Sign values: 1: + values 2: -- values 3: +/- values
8012_{hex}	Decimal point setting	r/w	Byte 1:decimal places Byte 2: 0 Byte 3: 0 Byte 4: 0	Decimal places: 0: 0 decimal places 1: 1 decimal place 2: 2 decimal places 3: 3 decimal places 4: 4 decimal places 5: 5 decimal places
8014_{hex}	Status	r/o	Byte 1: bit1: output 1 bit2: output 2 Byte 2: bit1 - bit4: main counter bit5 - bit8: secondary counter Byte 3: 0 Byte 4: 0	Output: 1: ON 0: OFF Counter status: 0: counter status: regular 1 :counter status: overflow 2: counter status: underflow

r/o: Read-only access
r/w: Read and write access
-/w: Write access

General guidelines:

- Read/Write: registers must be requested in compliance with the register layout described above. So registers can only be read and written as a whole and according to the authorized access.
- Write: writing instructions that cannot be performed because of format or over-range errors generate an error response with the error code *Device error.*
- Read/Write: if the device is in the programming menu or in the parameter setting menu, it will not perform any instruction or send back any error response.
- Read: the status request informs about overflows or underflows in the main and/or secondary counter and about the outputs switched on.
  Status: 00001103_{hex} (output1 and output2 switched on; main counter: overflow; secondary counter: overflow)
- The Loc input has no effect on the functionality of the MODBUS.
- The sign alarm value 1 is relevant only in Trail Mode
- The main and/or secondary counters will only be reset when the reset mode of the counter is programmed to MAN.EL or MAN.RES.

Remarks about the Float registers:

- If the Timer function is programmed with the time format HH:MM:SS, the integer part of the float value will be used in the decimal form for setting the time value.
  Examples:
  - Float data value $48DBD8E0_{hex}$ (450247_{dec}) will be interpreted as 45:02:47 (HH:MM:SS)
  - Float data value $48DBDB60_{hex}$ (450267_{dec}) cannot be represented. The time display 45:02:67 (HH:MM:SS) is not allowed. A device error is generated. Writing is not performed.
  - In the time format HH:MM:SS, writing the register for programming the decimal places is not possible. Reading the register is still possible.
- In case of Counter setting with input mode A / B and A % B, the multiplication and division factors cannot be set.
- In case of Tachometer setting, the tachometer value of the device cannot be reset. Setting and using a set value is not possible. The sign of Preset 1 cannot be described.

Remarks about the Integer registers:

- If the Timer function is programmed with the time format HH:MM:SS, the transmitted integer value will be interpreted as a time value.
  Example:
  - Integer value 00010078_{hex} (65656_{dec}) will be interpreted as 6:56:56 (HH:MM:SS).
  - In the time format HH:MM:SS, writing the register for programming the decimal places is not possible. Reading the register is still possible.
- In case of Counter setting with input mode A / B and A % B, the multiplication and division factors cannot be set.
- In case of Tachometer setting, the tachometer value of the device cannot be reset. Setting and using a set value is not possible. The sign of Preset 1 cannot be described.

5.15 Message transmission examples

5.15.1 Example Read

Register address (Float): 0000_{hex} (main counter)
Set Slave address: 01_{hex}
Read data value: 3F80 0000_{hex} (1_{dec})

Master message:

Slave address	Function	Register address	Number of registers	CRC
01_{hex}	03_{hex}	0000_{hex}	0002_{hex}	$C40B_{hex}$

Slave response:

Slave address	Function	Number of bytes	Data value	CRC
01_{hex}	03_{hex}	04_{hex}	3F80 0000_{hex}	$F7CF_{hex}$

5.15.2 Example Write

Register address (Integer): 8014_{hex} (status)
Set Slave address: 01_{hex}
Error code: 04_{hex} (device error)

Master message:

Slave address	Function	Register address	Number of registers	Number of bytes	CRC
01_{hex}	10_{hex}	8014_{hex}	0002_{hex}	04_{hex}	$0DDD_{hex}$

Slave response:

Slave address	Function	Error code	CRC
01_{hex}	90_{hex}	04_{hex}	$4DC3_{hex}$

5.15.3 Example Identification

Set Slave address: 01_{hex}
Slave ID: 3536 302E 302E 3035_{hex} (560.0.A5)
Software version: 5645 2E30 322E 3031_{hex} (VE.02.01)

Master message:

Slave address	Function	Register address	Number of registers	Number of bytes	CRC
01_{hex}	10_{hex}	8014_{hex}	0002_{hex}	04_{hex}	$0DDD_{hex}$

Slave response:

Slave address	Function	Number of bytes	Slave ID	Status	Software version	CRC
01_{hex}	90_{hex}	04_{hex}	3536 302E 302E 3035_{hex}	FF_{hex}	5645 2E30 322E 3031_{hex}	$4DC3_{hex}$

6 Parameter sets

The following settings apply to parameter sets 1 to 3.

PR.TIME	1.0
SOURCE	MAIN
PROTOC.	MODBUS
ADRESS.	1

7 Technical data

7.1 Pulse counter

Response time of the outputs at the maximum counting frequency (see 8.1):

Add;Sub;Trail	< 13 ms
for automatic repetition	< 13 ms
A / B; (A-B) / A	< 70 ms

7.2 Frequency meter

Response time of the outputs:

1-channel operation:	< 100 ms @ 40 kHz
	< 160 ms @ 55 kHz
2 -channel operation:	< 150 ms @ 40 kHz
	< 250 ms @ 55 kHz

7.3 Timer

Response time of the outputs:

Setting Seconds:	< 15 ms
Setting Minutes:	< 75 ms
Measuring error:	< 100 ppm

8 Frequencies (typical)

NOTE: switching level of the inputs

Switching level with AC power supply:

HTL level	Low:	0..4 VDC
	High:	12..30 VDC
5V level	Low:	0..2 VDC
	High:	3.5..30 VDC

Switching level with DC power supply:

HTL level	Low:	0..0.2 × UB
	High:	0.6 × UB..30 VDC
5V level	Low:	0..2 VDC
	High:	3.5..30 VDC

8.1 Pulse counter

HTL level, rectangular signal shape 1:1

AC power supply		typ.Low	2.5V
		typ.High	22V
DC power supply	12V	typ.Low	2V
		typ.High	10V
DC power supply	24V	typ.Low	2.5V
		typ.High	22V

	Add Sub Trail	AddAr SubAr AddBat SubBat TrailAr	AddTot SubTot
Cnt.Dir	45 kHz	2.4 kHz	2.2 kHz
Up.Dn; Up.Up	25 kHz	2.2 kHz	2.1 kHz
Quad; Quad2	22 kHz	1.0 kHz	1.0 kHz
Quad4	17 kHz	0.8 kHz	0.8 kHz
A / B; (A – B) / A	27 kHz		

5V level, rectangular signal shape 1:1

typ. Low	1,0V
typ. High	4,0V

	Add Sub Trail	AddAr SubAr AddBat SubBat TrailAr	AddTot SubTot
Cnt.Dir	9 kHz	2,1 kHz	2,0 kHz
Up.Dn; Up.Up	9 kHz	2,0 kHz	2,0 kHz
Quad; Quad2	9 kHz	1,0 kHz	1,0 kHz
Quad4	9 kHz	0,8 kHz	0,8 kHz
A / B; (A – B) / A	9 kHz		

8.2 Frequency meter

HTL level, rectangular signal shape 1:1

AC power supply		typ.Low	2.5V
		typ.High	22V
DC power supply	12V	typ.Low	2V
		typ.High	10V
DC power supply	24V	typ.Low	2.5V
		typ.High	22V

5V level, rectangular signal shape 1:1

typ.Low	1.0V
typ.High	4.0V

	HTL	5V
A	55 kHz	9 kHz
A - B; A + B; A / B; (A – B) / A	55 kHz	9 kHz
Quad	27 kHz	9 kHz

9 Help texts

INTERF.		MAIN MENU INTERFACES
PROTOC.	MODBUS	MODBUS PROTOCOL
PROTOC.	CR.LF	CRLF PROTOCOL
ADRESS.	1-247 / 1-99	INTERFACE ADDRESS MODBUS / INTERFACE ADDRESS CRLF
PR.TIME	0,5 – 9999,9	SENDING CYCLE CRLF PROTOCOL
SOURCE	MAIN	CRLF DATA MAIN COUNTER
SOURCE	BATCH	CRLF DATA BATCH COUNTER
SOURCE	TOTAL	CRLF DATA TOTALIZER
SOURCE.	MAI.BAT.	CRLF DATA MAIN AND BATCH COUNTER
SOURCE.	MAI.TOT.	CRLF DATA MAIN COUNTER AND TOTALIZER
MP.INP.1	PRINT	MP INPUT1 FUNCTION PRINT
MP.INP.2	PRINT	MP INPUT2 FUNCTION PRINT

10 References

Information about the MODBUS protocol:

[1] **Modbus specification**
MODBUS Application Protocol Specification V1.1b
MODBUS over Serial Line – Specification and Implementation Guide V1.02.
www.modbus.org

Information about RS232:

[2] **ANSI/EIA/TIA-232-F-1997**

Information about RS485:

[3] **ANSI/TIA/EIA-485-A-98**