

REDUNDANTER FEUCHTE-, TEMPERATUR- UND DRUCKSENSOR

RS485 · MODBUS RTU · QUICK START

01 ELEKTRISCHER ANSCHLUSS / Electrical connection

Kürzel	Aderfarbe	Funktion	Anschluss
bn	Braun	+5 V DC	Versorgung +
ge	Gelb	RS485 A	A / D+
ws	Weiß	RS485 B	B / D-
gn	Grün	GND	0 V / Masse

WARNUNG Versorgung und RS485-Leitungen vor dem Einschalten prüfen. Eine falsche Verdrahtung kann das Gerät oder den RS485-Adapter beschädigen.

02 MONTAGE UND HANDHABUNG / Installation

- Sensorkopf frei anströmen lassen und Temperaturgradienten entlang des Gehäuses vermeiden.
- Filterkappe im Betrieb montiert lassen; Sensorelemente nicht berühren oder mechanisch belasten.
- RS485 als Bus verdrahten. Bei langen Leitungen Abschluss nur an den beiden Busenden vorsehen.

03 KOMMUNIKATION UND ERSTER TEST / Modbus setup

Baudrate	Datenbits	Parität	Stopbits	Slave-ID
9600	8	None	1	234

Empfohlener Funktionstest

Funktion	Start	Anzahl	Datentyp	Auswertung
FC03	4000	13	INT16	siehe Seite 2

HINWEIS Vollständige Kommunikationseinstellungen, Änderung der Slave-ID sowie alle Register sind in der „Modbus Programmierhilfe BRH 100RA“ beschrieben.

04 KOMPAKTE REGISTERÜBERSICHT / Holding registers · FC03/FC04

Messwert	INT16	Skalierung	FLOAT32	Einheit
Temperatur 1	4000	÷100	1000/1001	°C
Feuchte 1	4001	÷100	1002/1003	%RH
Taupunkt 1	4002	÷100	1004/1005	°C
Abs. Feuchte 1	4003	÷10	1006/1007	g/m³
Temperatur 2	4004	÷100	1008/1009	°C
Feuchte 2	4005	÷100	1010/1011	%RH
Druck	4008	÷10	1016/1017	hPa
TempCalc / RHCalc	4010 / 4011	÷100	1018...1021	°C / %RH
QualityIndex	4012	÷10	1022/1023	0...100

FLOAT32: zwei Register pro Wert, Wortreihenfolge Low-Word → High-Word.

05 KURZE CODEBEISPIELE / Python · pymodbus

INT16 lesen

```
from pymodbus.client import ModbusSerialClient
c = ModbusSerialClient(port='COM5', baudrate=9600,
    parity='N', stopbits=1, bytesize=8, timeout=.5)
if c.connect():
    r = c.read_holding_registers(4000, 13, slave=234)
    t1 = r.registers[0] / 100.0
    rh1 = r.registers[1] / 100.0
    p = r.registers[8] / 10.0
    print(t1, rh1, p)
c.close()
```

FLOAT32 lesen

```
import struct

def regs_to_float(lo, hi):
    raw = struct.pack('<HH', lo & 0xFFFF,
        hi & 0xFFFF)
    return struct.unpack('<F', raw)[0]

r = c.read_holding_registers(1000, 4, slave=234)
t1 = regs_to_float(r.registers[0], r.registers[1])
rh1 = regs_to_float(r.registers[2], r.registers[3])
```

Installation: pip install pymodbus

06 SCHNELLE FEHLERPRÜFUNG / Troubleshooting

Beobachtung	Prüfung
Keine Antwort	COM-Port, +5 V/GND, Slave-ID, 9600 8N1 und A/B prüfen; A/B testweise tauschen.
Werte zu groß	Skalierung ÷100 bzw. ÷10 beachten.
FLOAT unplausibel	Low-Word vor High-Word verwenden.