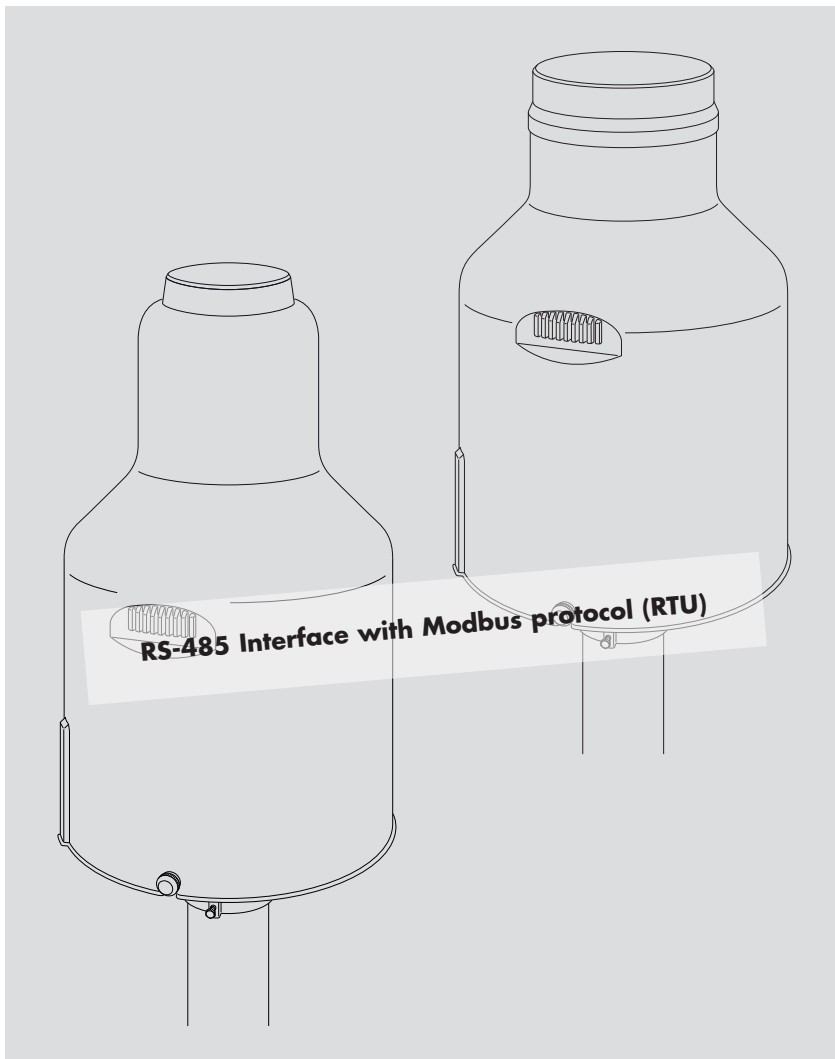


Protocol Description
Precipitation gauge
OTT Pluvio² L



We reserve the right to make technical changes and improvements without notice.

Table of contents

| | |
|---|----------|
| 1 Description RS-485 Interface with Modbus protocol (RTU) | 4 |
| 1.1 General note | 4 |
| 1.2 Preconditions | 4 |
| 1.3 Standard Data Formats | 4 |
| 1.4 Function "Read Holding Registers" (0x03) – Sensor description | 6 |
| 1.5 Function "Read Holding Registers" (0x03) – Sensor values | 9 |
| 1.6 Function "Read Holding Registers" (0x03) / "Write Single Register" (0x06) – Configuration | 10 |

1 Description RS-485 Interface with Modbus protocol (RTU)

1.1 General note

This "Protocol Description – Precipitation gauge OTT Pluvio² L" is an add on to the OTT Pluvio² L operating instructions if the Modbus transmission protocol is to be used.

Additionally, please also refer to

- ▶ "Operating Instructions – Precipitation gauge OTT Pluvio² L", particularly chapter 6.3 (document number: 70.040.000.B.E);
- ▶ "MODBUS Application Protocol Specification", available on "www.modbus.org";
- ▶ the operating instructions of the device to which the OTT Pluvio² L is connected to.

1.2 Preconditions

- ▶ Interface EIA-485 (RS-485)
 - ▶ Transmission parameter
 - Data bits 8
 - Stop bits 1 · 2 (only with parity „None“)
 - Parity Even* · Odd · None
 - ▶ Transmission speed 9 600* · 19 200 · 57 600 · 115 200 bit/s
 - ▶ Bus address 1* ... 247
- * factory setting

1.3 Standard Data Formats

Implementations are restricted to data points in the following standard data format:

- int: signed integer value
- uint: unsigned integer value
- bitfield: a collection of bits, multi-valued alarms or state
- string: a null terminated or fixed length value

▶ 16-bit Integer values

Values are stored in big-endian order per the Modbus specification and consist of a single register. All integer values are documented as signed or unsigned. All signed values are represented using two's complement format.

int range: -32 767 ... +32 767 not implemented: 0x8000
 uint range: 0 ... 65 534 not implemented: 0xFFFF
 bitfield16 range: 0 ... 0xFFFF not implemented: 0xFFFF

| Modbus Register | 1 | | | | | | | | | | | | | | | |
|-----------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Byte | 0 | | | | | | | | 1 | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

▶ 32-bit Integer values

32-bit integers are stored using two registers in big-endian order.

int range: -214 483 647 ... +214 483 647 not implemented: 0x80 000 000
 uint range: 0 ... +4 294 967 294 not implemented: 0xFF FFF FFF

| Modbus Register | 1 | | | | 2 | | | |
|-----------------|-----------|--|-----------|--|----------|--|-----------|--|
| Byte | 0 | | 1 | | 2 | | 3 | |
| Bit | 31 ... 24 | | 23 ... 16 | | 15... 08 | | 07 ... 00 | |

► **Floating point values**

Floating point values are 32 bits and encoded according to the IEEE 754 floating point standard.

float32 range: see IEEE 754

not implemented: 0x7F C00 000 (NaN)

| | | | | | | | | | | | | | | | | |
|------------------------|------|----------|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|
| Modbus Register | 1 | | | | | | | | | | | | | | | |
| Byte | 0 | | | | | | | | 1 | | | | | | | |
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IEEE 754 | sign | Exponent | | | | | | | Fraction | | | | | | | |

| | | | | | | | | | | | | | | | | |
|------------------------|----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Modbus Register | 2 | | | | | | | | | | | | | | | |
| Byte | 2 | | | | | | | | 3 | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IEEE 754 | Fraction least | | | | | | | | | | | | | | | |

► **String values**

Store variable length string values in a fixed size register range using a NULL (0 value) to terminate or pad the string. For example, up to 16 characters can be stored in 8 contiguous registers as follows:

not implemented values: all registers filled with NULL or 0x0000

| | | | | | | | | | | | | | | | | |
|------------------------|---|---|---|---|---|---|---|-----|---|---|----|----|----|----|----|------|
| Modbus Register | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | | | | | |
| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Bit | E | X | A | M | P | L | E | spc | S | T | R | I | N | G | ! | NULL |

- **Note:** The OTT Pluvio² L has only one "Holding Register" Block.

1.4 Function "Read Holding Registers" (0x03) – Sensor description

| Register name | Register-number ¹⁾ | Data type | Length | Access mode | min. / max. | Description |
|--|-------------------------------|-----------|--------|-------------|-------------|--|
| ▶ <i>Protocol ID</i> | 1 (0) | uint32 | 2 | R | | OTT HydroMet device assignment starting with register number 41001 and beginning with the 32-bit "OTTP" identifier (0x4F545420). This allows for discovery of OTT HydroMet Modbus protocol compatible devices. |
| ▶ <i>Protocol description ID</i> | 3 (2) | uint16 | 1 | R | | 0x0002 "Common Block Pluvio 2S/L" |
| ▶ <i>Protocol description length</i> | 4 (3) | uint16 | 1 | R | | Total number of 16-bit registers (registers names are marked in italics) |
| ▶ <i>Product ID</i> | 5 (4) | uint32 | 2 | R | | Product ID 70030 (0x0001118E) |
| ▶ <i>Device ID</i> | 7 (6) | uint32 | 2 | R | | Device ID 001 (0x00000001) |
| ▶ <i>Firmware version</i> | 9 (8) | uint32 | 2 | R | | V1.23.4 = 123400 (0x0001E208) |
| ▶ <i>Bootloader version</i> | 11 (10) | uint32 | 2 | R | | V1.23.4 = 123400 (0x0001E208) |
| ▶ <i>Physical element reference system</i> | 13 (12) | uint16 | 1 | R | | 0x002 = OTT (0x001 = SHEF) |
| ▶ <i>Unit reference system</i> | 14 (13) | uint16 | 1 | R | | 0x002 = OTT (0x001 = SHEF) |
| ▶ <i>Number of channels</i> | 15 (14) | uint16 | 1 | R | 1 ... 40 | Number of sensor channels |
| ▶ Channel 1 – physical element definition | 16 (15) | uint16 | 1 | R | | Intensity RT 'PA' |
| ▶ Channel 1 – unit | 17 (16) | uint16 | 1 | R | | 0x0020: mm/h 0x0021: mm/min 0x0022: inch/h 0x0023: inch/min |
| ▶ Channel 1 – unit string | 18 (17) | uint16 | 5 | R | | e.g. mm/h (register is 5 x 16 bit long and contains a char[10] array) |
| ▶ Channel 2 – physical element definition | 23 (22) | uint16 | 1 | R | | Accu RT-NRT 'PB' |
| ▶ Channel 2 – unit | 24 (23) | uint16 | 1 | R | | 0x0007: inch 0x0009: mm |
| ▶ Channel 2 – unit string | 25 (24) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 3 – physical element definition | 30 (29) | uint16 | 1 | R | | Accu NRT 'PC' |
| ▶ Channel 3 – unit | 31 (30) | uint16 | 1 | R | | 0x0007: inch 0x0009: mm |
| ▶ Channel 3 – unit string | 32 (31) | uint16 | 5 | R | | Refer to channel 1 unit string example |

¹⁾ the corresponding register start addresses are given in brackets (register number - 1 = register start address)

| Register name | Register-number ¹⁾ | Data type | Length | Access mode | min. / max. | Description |
|---|--------------------------------------|------------------|---------------|--------------------|--------------------|--|
| ▶ Channel 4 – physical element definition | 37 (36) | uint16 | 1 | R | | Accu total NRT 'PD' |
| ▶ Channel 4 – unit | 38 (37) | uint16 | 1 | R | | 0x0007: inch 0x0009: mm |
| ▶ Channel 4 – unit string | 39 (38) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 5 – physical element definition | 44 (43) | uint16 | 1 | R | | Bucket RT 'PE' |
| ▶ Channel 5 – unit | 45 (44) | uint16 | 1 | R | | 0x0007: inch 0x0009: mm |
| ▶ Channel 5 – unit string | 46 (45) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 6 – physical element definition | 51 (50) | uint16 | 1 | R | | Bucket NRT 'PE' |
| ▶ Channel 6 – unit | 52 (51) | uint16 | 1 | R | | 0x0007: inch 0x0009: mm |
| ▶ Channel 6 – unit string | 53 (52) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 7 – physical element definition | 58 (57) | uint16 | 1 | R | | Temperature of load cell 'TL' |
| ▶ Channel 7 – unit | 59 (58) | uint16 | 1 | R | | 0x0010: DEGREE C 0x0011: DEGREE F |
| ▶ Channel 7 – unit string | 60 (59) | uint 16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 8 – physical element definition | 65 (64) | uint16 | 1 | R | | Status of the heating 'SH' |
| ▶ Channel 8 – unit | 66 (65) | uint16 | 1 | R | | 0x0001: none |
| ▶ Channel 8 – unit string | 67 (66) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 9 – physical element definition | 72 (71) | uint16 | 1 | R | | Status of the device 'SD' |
| ▶ Channel 9 – unit | 73 (72) | uint16 | 1 | R | | 0x0001: none |
| ▶ Channel 9 – unit string | 74 (73) | uint16 | 5 | R | | Refer to channel 1 unit string example |

¹⁾ the corresponding register start addresses are given in brackets (register number - 1 = register start address)

| Register name | Register-number ¹⁾ | Data type | Length | Access mode | min. / max. | Description |
|--|-------------------------------|-----------|--------|-------------|-------------|--|
| ▶ Channel 10 – physical element definition | 79 (78) | uint16 | 1 | R | | Temperature of electronics unit 'TE' |
| ▶ Channel 10 – unit | 80 (79) | uint16 | 1 | R | | 0x0010: DEGREE C 0x0011: DEGREE F |
| ▶ Channel 10 – unit string | 81 (80) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 11 – physical element definition | 86 (85) | uint16 | 1 | R | | Supply voltage 'VB' |
| ▶ Channel 11 – unit | 87 (86) | uint16 | 1 | R | | 0x0030: V |
| ▶ Channel 11 – unit string | 88 (87) | uint16 | 5 | R | | Refer to channel 1 unit string example |
| ▶ Channel 12 – physical element definition | 93 (92) | uint16 | 1 | R | | Temperature of orifice rim 'TR' |
| ▶ Channel 12 – unit | 94 (93) | uint16 | 1 | R | | 0x0010: DEGREE C 0x0011: DEGREE F |
| ▶ Channel 12 – unit string | 95 (94) | uint16 | 5 | R | | Refer to channel 1 unit string example |

¹⁾ the corresponding register start addresses are given in brackets (register number - 1 = register start address)

Example

Read Holding Register, register number 9 "Firmware version":

000088-Tx: 01 03 00 08 00 02 45 C9

000089-Rx: 01 03 04 00 00 04 24 F8 E8

▶ Transmit (Tx)

– Modbus address 01₁₆
– function "Read Holding Register" 03₁₆
– start address 00 08₁₆ → 8₁₀
– number of registers to read 00 02₁₆ → 2₁₀
– checksum 45 C9₁₆

▶ Receive (Rx)

– Modbus address 01₁₆
– function "Read Holding Register" 03₁₆
– number of data bytes 04₁₆ → 4₁₀
– response 00 00 04 24₁₆ → 1060₁₀ → firmware version V1.06.0 (unit32)
– checksum F8 E8₁₆

(000088/000089: sequential line number of the communication software; example only, not relevant)

1.5 Function "Read Holding Registers" (0x03) – Sensor values

| Register name | Register-number ¹⁾ | Data type | Length | Access mode | Description |
|--------------------|-------------------------------|-----------|--------|-------------|--|
| ▶ Timestamp – date | 101 (100) | unit32 | 2 | R | Date; format: 0x00ddMMyy dd = day; MM = month; yy = year |
| ▶ Timestamp – time | 102 (101) | unit32 | 2 | R | Time; format: 0x00HHmmss HH = hour; mm = minutes; ss = seconds |
| ▶ Channel 1 | 105 (104) | float32 | 2 | R | Intensity RT |
| ▶ Channel 2 | 107 (106) | float32 | 2 | R | Accu RT-NRT |
| ▶ Channel 3 | 109 (108) | float32 | 2 | R | Accu NRT |
| ▶ Channel 4 | 111 (110) | float32 | 2 | R | Accu total NRT |
| ▶ Channel 5 | 113 (112) | float32 | 2 | R | Bucket RT |
| ▶ Channel 6 | 115 (114) | float32 | 2 | R | Bucket NRT |
| ▶ Channel 7 | 117 (116) | float32 | 2 | R | Temperature of load cell |
| ▶ Channel 8 | 119 (118) | float32 | 2 | R | Status of the heating |
| ▶ Channel 9 | 121 (120) | float32 | 2 | R | Status of the device |
| ▶ Channel 10 | 123 (122) | float32 | 2 | R | Temperature of electronics unit |
| ▶ Channel 11 | 125 (124) | float32 | 2 | R | Supply voltage |
| ▶ Channel 12 | 127 (126) | float32 | 2 | R | Temperature of orifice rim |

¹⁾ the corresponding register start addresses are given in brackets (register number - 1 = register start address)

1.6 Function "Read Holding Registers" (0x03) / "Write Single Register" (0x06) – Configuration

Register values/factory settings are described in Chapter 6 "SDI-12 commands and responses" of OTT Pluvio² L Operating Instructions.

Changes to the Modbus communication settings will cause a Modbus timeout because the internal communication is restarted and the stack cannot respond. Successful changes are answered with a regular Modbus response, invalid data with "illegal data value" and unsupported register addresses with "illegal data address".

Please note: Changes to the SD-112 address will reset the entire system and cause a Modbus timeout.

| Register name | Register-number ¹⁾ | Data type | Length | Access mode | Description |
|--|-------------------------------|-----------|--------|-------------|--|
| ▶ Unit intensity | 201 (200) | uint16 | 1 | R/W | Setting the intensity unit 0x0020: mm/h 0x0021: mm/min 0x0022: inch/h 0x0023: inch/min |
| ▶ Unit temperature | 202 (201) | uint16 | 1 | R/W | Setting the temperature unit 0x0010: degree Celsius 0x0011: degree Fahrenheit |
| ▶ Pulse output rate | 203 (202) | uint16 | 1 | R/W | Setting the pulse output rate 0x0002: 2 Hz 0x0005: 5 Hz |
| ▶ Pulse output factor | 204 (203) | uint16 | 1 | R/W | Setting the pulse output rate 0x0000: pulse rate 0.5 0x0001: pulse rate 1 0x0002: pulse rate 2 0x0003: pulse rate 5 0x0004: pulse rate 10 |
| ▶ Orifice rim heater mode | 205 (204) | uint16 | 1 | R/W | Setting the orifice rim heater mode 0x0000: deactivated 0x0001: heater mode 1 0x0002: heater mode 2 0x0003: heater mode 3 0x0004: heater mode 4 |
| ▶ Orifice rim heater target temperature | 206 (205) | int16 | 1 | R/W | Setting the orifice rim target temperature 0x0002 ... 0x0009: +2 ... +9 °C |
| ▶ Orifice rim heater lower temperature limit | 207 (206) | int16 | 1 | R/W | Setting the orifice rim lower temperature limit 0xFFD8 ... 0x0009: -40 ... +9 °C Refer to data format "16-bit Integer values"! |
| ▶ Orifice rim heater on-time/after-run time | 208 (207) | uint16 | 1 | R/W | Setting the orifice rim heater on-time/after-run time 0x0001 ... 0x05A0: 1 ... 1440 minutes |
| ▶ Orifice rim heater starting time (hour) | 209 (208) | uint16 | 1 | R/W | Setting the orifice rim heater starting time; 0x0000 ... 0x017: 0 ... 23 (o'clock) |
| ▶ Orifice rim heater starting time (minutes) | 210 (209) | uint16 | 1 | R/W | Setting the orifice rim heater starting time; 0x0000 ... 0x003B: 0 ... 59 minutes |
| ▶ Orifice rim heater starting time (seconds) | 211 (210) | uint16 | 1 | R/W | Setting the orifice rim heater starting time; 0x0000 ... 0x003B: 0 ... 59 seconds |

¹⁾ the corresponding register start addresses are given in brackets (register number - 1 = register start address)

| Register name | Register-number ¹⁾ | Data type | Length | Access mode | Description |
|---|-------------------------------|-----------|--------|-------------|---|
| ▶ Orifice rim heater self-test interval | 212 (211) | uint16 | 1 | R/W | Setting the orifice rim heater self-test interval 0x0000 ... 0xFDE8: 0 ... 65000 min |
| ▶ Serial interface | 213 (212) | uint16 | 2 | R/W | Setting the serial interface 0x0000: SDI-12 0x0001: RS-485 2-wire 0x0002: RS-485 4-wire |
| ▶ RS-485 interface mode | 215 (214) | uint16 | 1 | R/W | Setting the serial interface 0x0000: SDI-12 protocol 0x0001: ASCII (RS-485 command line mode) 0x0002: – (special application) 0x0003: Modbus (RTU) protocol |
| ▶ SDI-12 address | 216 (215) | uint16 | 1 | R/W | Setting the SDI-12 address ASCII value!; e.g. SDI-12 address "0" = 0x0030 0x0030 ... 0x0039: 0 ... 9 0x0041 ... 0x005A: A ... Z 0x0061 ... 0x007A: a ... z |
| ▶ Modbus bus address | 217 (216) | uint16 | 1 | R/W | Setting the Modbus bus address 0x0001 ... 0x00F7: 1 ... 247 |
| ▶ RS-485 baud rate | 218 (217) | uint16 | 1 | R/W | Setting the Modbus baud rate 0x0000: 9 600 bit/s 0x0001: 19 200 bit/s 0x0002: 57 600 bit/s 0x0003: 115 200 bit/s |
| ▶ Modbus parity framing | 219 (218) | uint16 | 1 | R/W | Setting the Modbus parity framing 0x0000: None (1 stop bit) 0x0001: None (2 stop bits) 0x0002: Odd (1 stop bit) 0x0003: Even (1 stop bit) |
| ▶ Reset Accu total NRT | 220 (219) | uint16 | 1 | R/W | Resetting the "Accu total NRT" value; write "0x0001" into register to reset the value |

¹⁾ the corresponding register start addresses are given in brackets (register number - 1 = register start address)

Example

Write Single Register, register number 201, setting the intensity unit to mm/min:

000242-Tx: 01 06 00 C8 00 21 C8 2C

000243-Rx: 01 06 00 C8 00 21 C8 2C

- ▶ Transmit (Tx) / Receive (Rx)
 - Modbus address 01₁₆
 - function "Write Single Register" 06₁₆
 - start address 00 C8₁₆ → 200₁₀
 - write 00 21₁₆ → mm/min
 - checksum C8 2C₁₆

(000242/000243: sequential line number of the communication software; example only, not relevant)

Document number
70.040.001.B.E 01-0623



OTT HydroMet GmbH

Ludwigstrasse 16
87437 Kempten · Germany
Phone +49 831 5617-0
Fax +49 831 5617-209
euinfo@otthydromet.com
www.otthydromet.com