



## MODBUS Hardware & Software Starter Kit

### Temperature, Humidity, Dew point MODBUS Starter Kit

Master MODBUS in minutes with this starter kit. It includes: (1) Easy MODBUS Viewer software, (2) 2-in-1 MeteoTemp high precision temperature & humidity Modbus sensor with connector, (3) 5m (16ft) cable with connector & leads, (4) RS485-to USB converter and (5) helical MeteoShield-Professional, (6) outdoor mount. This kit offers a large discount over individual sensor & software pricing. MeteoWind 2 wind speed & direction starter kit is also available on the next page.



## Quick Start Guide - Connecting your Modbus software, hardware & USB converter to PC

### MODBUS hardware setup

#### RS485-to-USB converter setup

1. See wire connection diagram at right

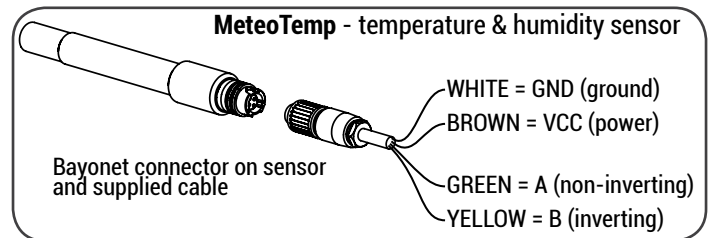
**!CONNECT SENSOR TO CONVERTER BEFORE CONNECTING TO PC!**

#### MeteoTemp sensor setup

2. Connect the communication wires Green & Yellow to the USB converter screw terminal block.
3. Connect first GND (ground), then VCC (power) wire to the USB converter screw terminal block.
4. Connect the bayonet connector on the other end of the supplied 5m cable to the sensor.

#### Plug the USB converter into a PC with the USB connector.

5. LED light colors on the USB converter indicate communication signals are being sent and received.
6. If lights are not blinking, either you are not sending commands to the sensors or there is an error in the setup.



### Links & Resources

- MODBUS SPECIFICATION: <http://modbus.org/specs.php>
- New MODBUS Protocol (after 2007): [http://modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf)
- Old MODBUS Over Serial Line LEGACY SPEC (before 2007): [http://modbus.org/docs/PI\\_MBUS\\_300.pdf](http://modbus.org/docs/PI_MBUS_300.pdf)
- MODBUS TCP/IP: <http://www.simplymodbus.ca/FC06.htm>
- MODBUS TCP/IP: [http://modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](http://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf)

### MODBUS Terminology

- **Modbus Slave** - a device that responds to the commands of a Master. Usually a sensor like MeteoTemp or MeteoWind.
- **Modbus Master** - a controller of the Modbus network that sends data requests to slave devices or other commands.
- **INPUT register** - 16bit (2 Byte) memory space in a Modbus Slave device from which data can be requested by a Master.
- **Slave ID** - address of a slave device on a Modbus network with multiple devices. 247 devices can be connected together.
- **Query type** - each query/request type has a number assigned to it which is sent as part of a request from a Master to a Slave and also from a Slave in the response message.
- **Hi & Lo Registers** - combine to form a 32bit integer = 4 Bytes
- **Register start Hi & Lo** - 16bit address of 1st register to read.
- **Register length Hi & Lo** - how many successive registers to read including the 1st register. Value 0 or 1 reads only 1st register.
- **CRC** - is a number which is calculated from the contents of a message by the sender (Master or Slave) and verified by the receiving device to make sure the received message is not corrupted and correct data is received.

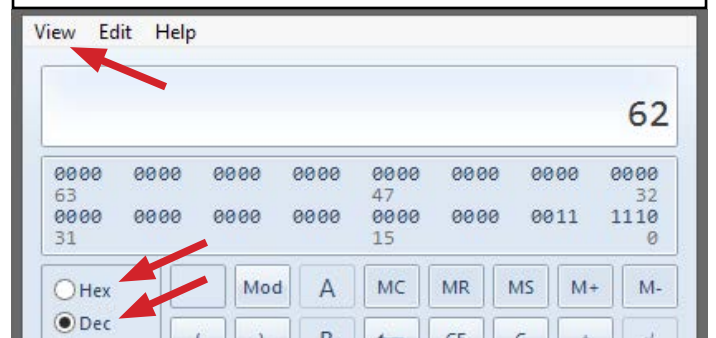
### MODBUS RTU message number format

**Hexidecimal** number format (base 16) is used in Modbus RTU because it is very compact. The number of digits used to represent a given number is usually less than in binary (base 2) and decimal (base 10) formats. **NOTE:** 0x in front of a number indicates that it is a HEX.

Counting up in Hex vs. Decimal system before two numbers have to be combined to form a larger number like 10 or 16, looks like:

- Hex: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Dec: 0,1,2,3,4,5,6,7,8,9

**Convert easily from Hex to Decimal to Binary** in Microsoft Windows with its calculator utility. Under view menu choose: Programmer mode as shown below:





## Wind Speed & Wind Direction MODBUS Starter Kit

### MODBUS Hardware & Software Starter Kit

Master MODBUS in minutes with this starter kit. It includes: (1) Easy MODBUS Viewer software, (2) 2-in-1 MeteoWind 2 Modbus anemometer with wind vane sensor, (3) 5m/16ft cable with connector & leads, (4) RS485-to-USB converter and (5) outdoor mount. This kit offers a large discount over individual sensor & software pricing.



## Quick Start Guide - Connecting your Modbus software, hardware, USB converter to PC

### MODBUS hardware setup

#### RS485-to-USB converter setup

1. See wire connection diagram at right

**!CONNECT SENSOR TO CONVERTER BEFORE CONNECTING TO PC!**

#### MeteoWind sensor setup

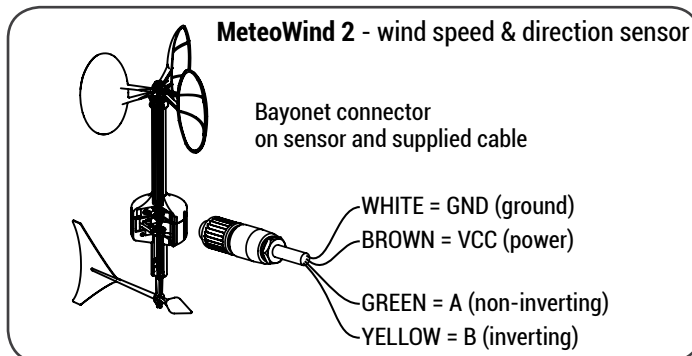
2. Connect the communication wires Green & Yellow to the USB converter screw terminal block.
3. Connect first GND (ground), then VCC (power) wire to the USB converter screw terminal block.
4. Connect the bayonet connector on the other end of the supplied 5m cable to the sensor.

#### Plug the USB converter into a PC with the USB connector.

5. LED light colors on the USB converter indicate communication signals are being sent and received.
6. If lights are not blinking, either you are not sending commands to the sensors or there is an error in the setup.

### MODBUS Terminology

- **Modbus Slave** - a device that responds to the commands of a Master. Usually a sensor like MeteoWind or MeteoWind.
- **Modbus Master** - a controller of the Modbus network that sends data requests to slave devices or other commands.
- **INPUT register** - 16bit (2 Byte) memory space in a Modbus Slave device from which data can be requested by a Master.
- **Slave ID** - address of a slave device on a Modbus network with multiple devices. 247 devices can be connected together.
- **Query type** - each query/request type has a number assigned to it which is sent as part of a request from a Master to a Slave and also from a Slave in the response message.
- **Hi & Lo Registers** - combine to form a 32bit integer = 4 Bytes
- **Register start Hi & Lo** - 16bit address of 1st register to read.
- **Register length Hi & Lo** - how many successive registers to read including the 1st register. Value 0 or 1 reads only 1st register.
- **CRC** - is a number which is calculated from the contents of a message by the sender (Master or Slave) and verified by the receiving device to make sure the received message is not corrupted and correct data is received.



### Links & Resources

- MODBUS SPECIFICATION: <http://modbus.org/specs.php>
- New MODBUS Protocol (after 2007): [http://modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf)
- Old MODBUS Over Serial Line LEGACY SPEC (before 2007): [http://modbus.org/docs/PI\\_MBUS\\_300.pdf](http://modbus.org/docs/PI_MBUS_300.pdf)
- MODBUS TCP/IP: [http://modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](http://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf)

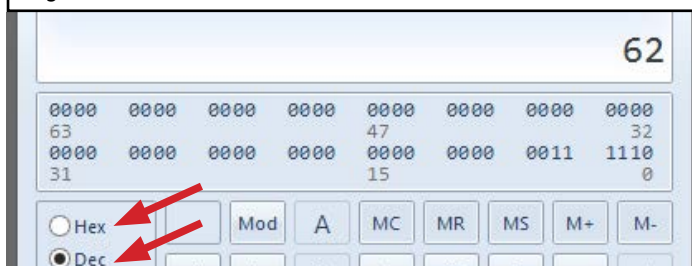
### MODBUS RTU message number format

**Hexidecimal** number format (base 16) is used in Modbus RTU because it is very compact. The number of digits used to represent a given number is usually less than in binary (base 2) and decimal (base 10) formats. **NOTE:** 0x in front of a number indicates that it is a HEX.

Counting up in Hex vs. Decimal system before two numbers have to be combined to form a larger number like 10 or 16, looks like:

- Hex: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Dec: 0,1,2,3,4,5,6,7,8,9

**Convert easily from Hex to Decimal to Binary** in Microsoft Windows with its calculator utility. Under view menu choose: Programmer mode as shown below:





## MeteoTemp Quick Start Guide - Getting Modbus data from your MeteoTemp sensor

### Default Communication settings

MODBUS RTU: 19200/8/N/1 (Sensor Slave ID address = 1)  
 (Speed=19200 Baud / 8 data bits / parity=None / stop bit = 1)  
 \* Baud speed/parity/stop-bits/Slave address are user selectable.

Holding registers contain settings and Input registers contain measured data from sensors.

Modbus request/query to read communication settings values from holding registers

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Low	CRC
01	03	00	04	00	02	D4 0A

Modbus response with communication settings values 2 Bytes:

Slave ID	Query type	DATA Length	DATA (2 Bytes) Hi Byte, Low Byte	CRC
01	03	02	01 06	39 D6

High Byte value **01** = 00000001 in binary format is divided into two sets of bits as follows: 0 & 0000001 (see table below)

- 0 = RTU (1 = ASCII)
- 0000001 converted to decimal = Slave ID 1

High Byte of Holding register 5	
1 bit	7 bits
0 = RTU or 1 = ASCII	Slave ID as a 7 bit binary number

Low Byte value **06** = 00000110 in binary format is divided into 3 sets of bits as follows: 0 & 00 & 00110 (see table below)

- 0 = 1 stop bit
- 00 = parity None (01 = Odd, 10 = Even parity)
- 00110 (binary) = 19200 Baud rate

Low Byte of Holding register 5		
1 bit	2 bits	5 bits
Stop bit	Parity	Baud rate (bps)
0 or 1	0 = none or 1 = odd or 2 = even	see table below

Baud Rate (bps)	1200	2400	4800	9600	14400	19200
Binary value	01	10	11	100	101	110

Write command for holding register 5 (offset 4) with COM settings

Slave ID	Query type	Register start Hi	Register start Lo	Write this value into register	CRC
01	06	00	04	01 06	49 99

Query type 06 writes a single register and type 16 writes multiple registers. Writing to Holding register 5 will change communication parameters immediately until device is powered off. If subsequently **0x0008 (HEX) is written to Holding register 11 (offset 0x000B) the new value will be saved** and Holding reg. 11 will change to 0x0000 as acknowledgment of the save command.

### MeteoTemp - Data registers (input registers)

Input Register	Measure	Offset	Register start Hi	Register start Lo
1	Temperature	0	00	00
2	Dew Point / Frost Point	1	00	01
3	Pressure (Low register)	2	00	02
4	Pressure High (Hi register)	3	00	03
5	Relative Humidity	4	00	04

### MODBUS RTU query & response format

Request message composition (length is 8 Bytes) (1Byte = 8bits)

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Low	CRC
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes

Response message composition (length is variable)

Slave ID	Query type	DATA Length	DATA (n * 8bits)	CRC
1 Byte	1 Byte	1 Byte	2 to 250 Bytes	2 Bytes

### Read single registers from MeteoTemp

Modbus request for Temperature

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Low	CRC
01	04	00	00	00	01	31 CA

Response with Temperature: 2221 = 22.21 °C **Negative numbers are represented by values ≥ 0x8000 (32767) see Dew point below**

Slave ID	Query type	DATA Length	DATA (2 * 8bits)	CRC
01	04	02	08 AD	7F 4D

Request for Dew point

01	04	00	01	00	01	60 0A
----	----	----	----	----	----	-------

Response with Dew point: FF6B = 111111101101011. Since the leftmost bit is a 1, value is **negative** and calculated using the two's complement: FF6B -(2<sup>16</sup>) = 65387 - 65536 = -149 = -1.49 °C

01	04	02	FF 6B	B9 2F
----	----	----	-------	-------

Request for Relative Humidity

01	04	00	04	00	01	70 0B
----	----	----	----	----	----	-------

Request for Pressure calls values from two registers (2 & 3) because pressure is a large number requiring 4 Bytes (32 bits).

01	04	00	02	00	02	D0 0B
----	----	----	----	----	----	-------

Response for Pressure contains 4 Bytes from registers 2 & 3.

Pressure Low (reg 2) = 8A 88, Pressure Hi (reg 3) = 00 01,

Combine Pressure Hi & Lo values together in this order. Result is a 32bit unsigned integer value from two 16bit registers. Converting 00018A88 from Hex format to Decimal: **Pressure = 101000 Pa**

01	04	04	8A 88 00 01	90 76
----	----	----	-------------	-------

### Read multiple Modbus registers together

Request for Temperature, Dew point, Pressure, Humidity encompasses 5 registers together

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Lo	CRC
01	04	00	00	00	05	30 09

Response: Temperature, Dew point, Pressure, Humidity together

Slave ID	Query type	DATA Length	DATA (10 * 8bits)	CRC
01	04	0A	08 AD 02 42 8A 88 00 01 0D 74	B9 DF

**NOTE:** 0x0008 is a hexadecimal number. The 0x indicates that it should be evaluated in base 16 instead of 10. Without it, you will not know whether 0008 is decimal, hexadecimal or octal...







## MeteoWind Quick Start Guide - Getting Modbus data from your MeteoWind sensor

### Default Communication settings

MODBUS RTU: 19200/8/N/1 (Sensor Slave ID address = 1)  
 (Speed=19200 Baud / 8 data bits / parity=None / stop bit = 1)  
 \* Baud speed/parity/stop-bits/Slave address are user selectable.

**Holding registers** contain settings and **Input registers** contain measured data from sensors.

Modbus request/query to read communication settings values from holding registers

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Low	CRC
01	03	00	04	00	02	D4 0A

Modbus response with communication settings values 2 Bytes:

Slave ID	Query type	DATA Length	DATA (2 Bytes) Hi Byte, Low Byte	CRC
01	03	02	01 06	39 D6

High Byte value **01** = 00000001 in binary format is divided into two sets of bits as follows: 0 & 0000001 (see table below)

- 0 = RTU (1 = ASCII)
- 0000001 converted to decimal = Slave ID 1

High Byte of Holding register 5	
1 bit	7 bits
0 = RTU or 1 = ASCII	Slave ID as a 7 bit binary number

Low Byte value **06** = 00000110 in binary format is divided into 3 sets of bits as follows: 0 & 00 & 00110 (see table below)

- 0 = 1 stop bit
- 00 = parity None (01 = Odd, 10 = Even parity)
- 00110 (binary) = 19200 Baud rate

Low Byte of Holding register 5		
1 bit	2 bits	5 bits
Stop bit	Parity	Baud rate (bps)
0 or 1	0 = none or 1 = odd or 2 = even	see table below

Baud Rate (bps)	1200	2400	4800	9600	14400	19200
Binary value	01	10	11	100	101	110

Write command for holding register 5 (offset 4) with COM settings

Slave ID	Query type	Register start Hi	Register start Lo	Write this value into register	CRC
01	06	00	04	01 06	49 99

Query type 06 writes a single register and type 16 writes multiple registers. Writing to Holding register 5 will change communication parameters immediately until device is powered off, unless subsequently 0x0008 (HEX) is written to Holding register 11 (offset 0x000B). Immediately, its value will change to 0x0000 as acknowledgment of the save command.

### Calibrating Wind direction (setting offset)

Angle OFFSET (bits)=1024 - ([measured °angle] / 360° \* 1024)  
 It is the angular offset from direction sensor zero based on a circle being divided into 1024 bit increments.

Request for Wind direction OFFSET from Holding register 74

01	03	00	49	00	01	55 DC
----	----	----	----	----	----	-------

Response with Wind direction OFFSET: 0x1770 = 6000 = 21.09°

01	03	02	17 70	B6 50
----	----	----	-------	-------

WRITE new Wind direction OFFSET: 211°/360\*1024 = 0xEA72

01	06	00	49	EA 72	97 59
----	----	----	----	-------	-------

SAVE Wind direction OFFSET by writing 0x0008 to Holding reg 11

01	06	00	0B	00 08	F9 CE
----	----	----	----	-------	-------

Immediately, its value will change to 0x0000 as acknowledgment.

01	06	00	49	00 00	58 1C
----	----	----	----	-------	-------

### Setting sampling/averaging time period

MeteoWind samples wind speed and direction 10x per second (10Hz) and the sampling period can be adjusted to in 1 second increments by writing with query type 06 to Holding register 10.

### MeteoWind - Data registers (input registers)

Input Register	Measure	Offset	Register start Hi	Register start Lo
6	Current Wind Speed	5	00 05	
7	Current Wind Direction	6	00 06	
8	Average speed in period	7	00 07	
9	Max. Wind speed in period	8	00 08	
10	Average direction in period	9	00 09	

### MODBUS RTU query & response format

Request message composition (length is 8 Bytes) (1Byte = 8bits)

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Low	CRC
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes

Response message composition (length is variable)

Slave ID	Query type	DATA Length	DATA (n * 8bits)	CRC
1 Byte	1 Byte	1 Byte	2 to 250 Bytes	2 Bytes

### Read single registers from MeteoWind

Modbus request for Wind speed

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Low	CRC
01	04	00	05	00	01	21 CB

Modbus response with Wind speed: 10m/s = 1000 = 0x03E8

Slave ID	Query type	DATA Length	DATA (2 * 8bits)	CRC
01	04	02	03 E8	B9 8E

Request for Wind direction

01	04	00	06	00	01	D1 CB
----	----	----	----	----	----	-------

Response with Wind direction: 359.00° = 35900 = 0x8C3C

01	04	02	8C 3C	DD E1
----	----	----	-------	-------

### Read multiple Modbus registers together

Request for Wind speed, Wind direction, Average wind speed encompasses 3 registers together

Slave ID	Query type	Register start Hi	Register start Lo	Length Hi	Length Lo	CRC
01	04	00	05	00	03	A0 0A

Response: Wind speed, Wind direction, Average speed together

Slave ID	Query type	DATA Length	DATA (6 * 8bits)	CRC
01	04	0A	01 AD 71 D8 6C 44	B6 AE

