thermokon
Sensortechnik GmbH

Protocol Description
WRF04 RS485 ModBus CO2

Version 1.0, 04.02.2011

Index of Changes

Version	Date	Description
1.0	09.02.2009	First Draft

1 WRF04-RS485-Modbus-CO2

The present document describes the serial interface of the room operating unit WRF04-RS485-MODBUS-CO2. The MODBUS-Protocol developed by the comany Modicon is a disclosed protocol for communication of several intelligent Master-Slave based devices.

For further information and definitions on the topic MODBUS, please see www.modbus.org

1.1 Hardware Installation

The room operating unit can be connected by means of a twisted-pair cable (line resistance 120 Ohm). For detailed information on installation and mounting, please see the product data sheet WRF04-RS485-Modbus-CO2 and the data sheet wiring_rs485_network.pdf.

1.2 RS485 Transceiver

The maximum number of bus participants without use of a repeater is preset by the RS485-transceiver. The transceiver used enables 32 devices per bus segment at maximum.

1.3 Protocol

The room operating unit WRF04-RS485-Modbus is a slave-bus participant only allowed to send to the bus on demand of the master. The protocol corresponds to the defaults of:
- MODBUS Application Protocol Specification V1.1
- MODBUS via Serial Line Specification & Implementation guide V1.0

1.4 Configuration Options

By means of the DIP switch the device can be adapted to the corresponding bus topology.
5pole DIP switch:

- Bus address of device (1 - 31) via5 pole DIP switch; DIP switch: 1-5 = 6pole DIP switch:
- Transmitting mode
  - DIP 1 off: RTU
  - DIP 1 on: ASCII
- Baud rate
  - DIP 2 off + DIP 3 off: 9600
  - DIP 2 on + DIP 3 off: 19200
  - DIP 2 off + DIP 3 on: 38400
  - DIP 2 on + DIP 3 on: 57600
- Parity
  - DIP 4 off + DIP 5 off: non
  - DIP 4 on + DIP 5 off: even
  - DIP 4 off + DIP 5 on: odd
- Bus terminating resistor 120 Ohm
  - DIP 6 off
  - DIP 6 on
- The number of data bits is fixed and preset to: RTU 8 data bits and ASCII 7 data bits

As the data sheet contains a detailed description of position and meaning of the jumpers, please refer to the file „Produktblatt_wrf04_rs485.pdf“.

Important notice for operation in the Master/Slave system:

!! The bus address must be adjusted differently for each device

!! Transmission mode, baud rate and parity must be identical

2 WRF04-RS485-Modbus-CO2 Protocol

2.1 Control Commands Supported

The following MODBUS – control commands are supported:

Description	Function code	
Read bits	01 (hex)	1 (dez)
	02 (hex)	2 (dez)
Read register	03 (hex)	3 (dez)
	04 (hex)	4 (dez)
Write individual bit	05 (hex)	5 (dez)
Write individual register	06 (hex)	6 (dez)
Write several bits	0F (hex)	15 (dez)
Write several registers	10 (hex)	16 (dez)

Table 1

2.2 Data Administration

All data in a MODBUS-Slave are assigned to addresses. Data access (read or write) is made by the corresponding control command and the indication of the corresponding data address.

Procedure	RTU	ASCII
Read register	20	10
Write register	20	10
Read coils	16	8
Write coils	8	8

2.3 EEprom – non volatile memory

Configuration parameters are not allowed to write permanently. Device has maximum write cycles of nonvolatile memory. (dimension: <10000).

2.4 Register Definition

2.4.1 Configuration Register (Holding Register R/W)

Register	Data Address	Value Range	Description	
1 R	**0x0000**	**0x0003**	**Device coding, not changeable**	
2 R	**0x0001**	**0x0012**	**Firmware version, not changeable**	
2 – 11	**0x0002 – 0x000A**	**Configuration of the operating unit, EEPROM- data – !! Don't update permanently EEprom !!**		
3 R/W	0x0002	0x0001-0x0001	Device Type	CO2
4 R/W	0x0003	0x0000-0xFFFF	Device location identification	(default = 0x0000)
5 R/W	0x0004	0x0000-0x07D0	Switch yellow LED	(default = 800)
6 R/W	0x0005	0x0000-0x07D0	Switch red LED	(default = 1200)
7 R/W	0x0006	0x0000-0xFFFF	Updating interval of display in seconds	(default = 0x0A)
8 R/W	0x0007	0x0000-0x0C80	Min-Response-Delay-Time	signed int, (max 3100 ms) (default = 0x0A = 10 ms)
9 R/W	0x0008	0x0000-0x00FF	Temperature-Offset for calibration of temperature sensor signed int, e.g. 10_{dec} = +1.0 K, -5_{dez} = -0.5 K	(default = 0x00)
10 R/W	0x0009	0x0000-0xFFFF	Offset for calibration of humidity sensor signed int, e.g. 50_{dec} = +5.0%, -30_{dez} = -3.0%	
11 R/W	0x000A	0x0000-0xFFFF	Offset for calibration of CO2 sensor signed int, e.g. 50_{dec} = +50ppm, -50_{dez} = -50ppm	

2.4.2 Output Register (Input Registers R)

Register	Data Address	Value Range	Description	
257 – 259R	**0x0100 – 0x0102**	**Measuring value (data output)**		
257 R	0x0100	0x0000-0xFFFF	Temperature	signed int, e.g. 184_{dec} = 18.4 °C
258 R	0x0101	0x0000-0x000F	Humidity	unsigned int, z.B. 500_{dez} = 50.0%
259 R	0x0102	0x0000-0xFFFF	CO2	unsigned int, z.B. 630_{dez} = 630 ppm

Input register (Holding Registers R/W)

Register	Data Address	Value Range	Description	
513 - 515	**0x0200 – 0x0202**	**Control (ext. data default)**		
513 R/W	0x0200	0x0000-0xFFFF	Ext. temperature default	signed int, e.g. 170_{dez} = 17.0°C
514 R/W	0x0201	0x0000-0x03E8	Ext. humidity default	unsigned int, e.g. 1000_{dez} = 100.0%
515 R/W	0x0202	0x0000-0xFFFF	Ext. CO2 default	unsigned int, e.g. 652_{dez} = 652ppm

Data- Address	Description
0xFF00 – 0xFFFF	**Range defined by the manufacturer, not allowed to be changed!**

2.5 Bit Allocation / Coil Definition

2.5.1 Configuration bits (Coils R/W)

Bit	Data Address	Description		
0x0000 – 0x0007	Configuration of Operating unit Bit-Register, EEPROM- Data Configuration of Display – !! Don't update permanently EEprom !!			
1 R/W	0x0000	Room temperature	1 = display 0 = do not display	(default)
2 R/W	0x0001	Humidity	1 = display 0 = do not display	(default)
3 R/W	0x0002	CO2 temperature	1 = display 0 = do not display	(default)
4 R/W	0x0003	°C/°F	1 = °C 0 = °F	(default)
5 R/W	0x0004	Traffic LED	1 = display 0 = do not display	(default)
6 R/W	0x0005	Comma room temperature	1 = display 0 = do not display	(default)
7 R/W	0x0006	Comma Humidity	1 = display 0 = do not display	(default)
8 R/W	0x0007	Humidity display	1 = with RH 0 = without RH	(default)

2.5.2 Input bits (Coils R/W)

Bit	Data Address	Description	
0x0100 – 0x0103	**Input Value of Operating unit Bit-Register Override of Controller**		
257 R/W	0x0100	Trigger LED[1]	0 – OFF (default) 1 – ON
258 R/W	0x0101	Green LED	0 – OFF (default) 1 – ON
259 R/W	0x0102	Yello LED	0 – OFF (default) 1 – ON
260 R/W	0x0103	Red LED	0 – OFF (default) 1 – ON

[1] The *LED* can only be controlled, if the configuration register for the LED control is parameterized, accordingly!

3 Data Transmission

3.1 Master/Slave Protocol

One master and one or more slaves are connected to the serial bus. The communication between master and slave is exclusively controlled by the master. The slaves are only allowed to send if they have been addressed by the master before. Slaves are only sending back to the master, never to another slave.

3.2 Data Frame

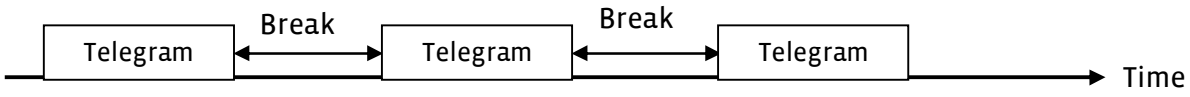
The data are sent to the bus in accordance to severely defined defaults:

Address	Control command	Data	Checksum

In general, a MODBUS telegram starts with the address of the slave, followed by a control command (e.g. read register) and the data. By means of the checksum at the telegram end, the bus participants can recognize transmission errors.

3.3 Transmission Mode RTU

In the transmission mode RTU telegrams are separated by means of transmission breaks:



The period of the transmission breaks for separating telegrams is depending on the adjusted baud rate and amounts to 3,5 * word transmission time (11 bit). With 9600 baud at least 4 ms must pass by and with 57600 at least 1 ms must pass by between two telegrams.

3.3.1 Telegram Layout

Address 1 Byte	Control command 1 Byte	Data 0 - 100 byte	Checksum	
			CRC Low	CRC High

3.3.2 Calculation of CRC-Checksum

The CRC checksum (Cyclical Redundancy Check) is calculated by the sender out of all bytes transmitted and is attached to the message.

The receiver re-calculates the CRC checksum and compares it with the checksum received. If the values do not correspond, a transmission error is assumed and the data received are rejected.

The least significant byte of the 16 bit checksum is set to the penultimate location and the most significant byte is set at last location.

Calculation of checksum (Programming example in C):

```
crc = 0xFFFF;                                   // CRC-Check, Initialisation
for(i = 0; i < Telegram length-2; i++)
        crc = crc_calc(crc, Telegram data[i]);

crc_low  = crc & 0x00FF;                        // Low-Byte
crc_high = (crc & 0xFF00) >> 8;                 // High-Byte

// Function definition CRC calculation
unsigned int     crc_calc(unsigned int   crc_temp, unsigned int data)
{
        unsigned int    Index_CC=0;             // Loop counter
        unsigned int    LSB=0;                  // Help variable

        // Exclusive-Oder des 8Bit-Char with the lower 8Bit of CRC
        crc_temp = ( ( crc_temp ^ data) | 0xFF00) & (crc_temp | 0x00FF) ;

        for(Index_CC = 0; Index_CC<8; Index_CC++)
        {
                LSB = (crc_temp & 0x0001);
                crc_temp        >>= 1;
                if(LSB)
                crc_temp = crc_temp ^ 0xA001;           // calculation polynominal für CRC16
        }

        return(crc_temp);
}
```

3.4 Transmission Mode ASCII

The ASCII transmission mode does not make that high demands on the computer speed of the bus participants. The telegrams are not separated by break times, but by ASCII control characters.

3.4.1 Telegram Layout

The ASCII control character „: “always identifies the beginning of a telegram. The ASCII control characters „CR“and „LF“identify the end of a telegram. The telegram data are output hexa-decimal in the ASCII format:

e.g.: 197dez (1Byte) = C5hex (1 Byte) = C (1 Byte) 5 (1 Byte) ASCII

As one data byte is displayed by 2 ASCII characters, the number of data bytes to be transmitted is doubled compared with the RTU mode.

Start 1 char	Address 2 char	Control command 2 char	Data 0 - 2 x 100 char	Checksum LRC 2 char	End 2 char
:					**CR LF**

3.4.2 Calculation of LRC-Checksum

The LRC checksum (Longitudinal Redundancy Check) is calculated by the sender out of all bytes transmitted (without „:“, „CR“, „LF“) and pasted in the message of „CR„ and „LF“. The receiver re-calculates the LRC checksum and compares it with the checksum received. If the values do not correspond, a transmission error is assumed and the data received are rejected.

The most significant ASCII character of the 8 bit checksum is sent in the telegram before the least significant ASCII character.

Calculation of checksum (programming example in C):

```
lrc = 0;
for(i = 1; i < Telegram length -4; i++)
        lrc = lrc + Telegram data [i];

lrc = 0xFF - lrc;
lrc = lrc + 1;
```

4 Examples: Telegrams

4.1 Register

The operating unit has different registers for the configuration, for the display of values and for the input values.

4.1.1 Configuration of parameter

The operating unit can be parameterized by the configuration registers 3-11 and the control commands „Write Register“(10hex or 06hex).

Example: Digital input 1 as breaker dew point and digital input 2 as maker energy hold off.

Master - Telegram in Transmission Mode RTU:

Device	command	Start address		Number of registers		Number of Bytes	Data Register 1F		Data Register 20		Check Sum	
		H Byte	L Byte	H Byte	L Byte		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	10	00	1F	00	02	04	00	01	00	12	CRC	

Slave – Response Telegram in Transmission Mode RTU:

Device	command	Start Address		Number of Register		Check Sum	
		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	10	00	06	00	02	CRC	

4.1.2 Read-Out of Output Register

Values are stored in the output registers.

Master - Telegram in Mode RTU		**Slave – Response Telegram in Mode RTU**	
Description	Value (Hex)	Description	Value (Hex)
Slave Address	02	Slave Adresse	02
Command	03	Command	03
Start address High	01	Number of Bytes	12
Start address Low	00	Register value High (0100) Temperature	00
Number of Registers High	00	Register value Low (0100) Temperature	DC
Number of Registers Low	03	Register value High (0101) Humidity	01
Checksum Low	CRC	Register value Low (0101) Humidity	3B
Checksum High		Register value High (0102) CO2	02
		Register value Low (0102) CO2	1C
		Checksum Low	CRC
		Checksum High	

4.1.3 Setting of Input Registers

By means of the input registers different values can be overwritten in the operating unit.

Master - Telegram in Transmission mode RTU:

Device	Command	Start address		Data Register 513		Checksum	
		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	06	02	00	01	02	CRC	

Slave – Response telegram in Transmission mode RTU:

Device	Command	Start address		Number of Registers		Checksum	
		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	10	02	00	00	02	CRC	

4.2 Coil / Bit Allocation

The operating unit has different configuration bits for adjusting the display value of the display. By means of the input bits different LEDs can be controlled.

Writing Configuration Bits
By means of the control command „Write Bit(s)“ (0Fhex or 05hex) a configuration bit (or more) can be written with the value „1“ or „0“.

Example: Dispay Set point temperature

Master - telegram in Transmission mode RTU:

Slave Address	commandl	Start address		Number of bits		Number of bytes	Data	Checksum	
		H Byte	L Byte	H Byte	L Byte		H Byte	L CRC	H CRC
02	0F	00	04	00	01	01	01	CRC	

Slave – response telegram Transmission mode RTU:

Slave Address	command	Start address		Number of bits		Checksum	
		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	0F	00	04	00	01	CRC	

4.2.1 Read Bits

By means of the control command „Read bits“(01hex or 02hex) one or more bits can be read out.

Master - Telegram in Mode RTU		Slave – Response Telegram in Mode RTU	
Description	Value (Hex)	Description	Value (Hex)
Device	02	Device	02
Command	01	Command	01
Start address High	00	Number of bytes	01
Start address Low	00	Bit value 0,0,0,0,0,0,bit1,bit0	02
Number of bits High	00	Checksum Low	CRC
Number of bits Low	02	Checksum High	
Checksum Low	CRC		
Checksum High			

5 Configuration Software

By means of a RS485-interface (e.g. RS232-RS485-level converter e.g. ADAM-4520) it is possible to access to the Modbus by the configuration software. The configuration software is not obligatory necessary for the installation of the WRF04-RS485 Modbus CO2. It is possible to use any programme producing Modbus telegrams which is suitable to set registers.

6 Software Installation

For the installation of the configuration software, the setup files „WRF04_Modbus_ Config_Setup.exe “must be started. Please note that you must have administrator rights for the installation. During the installation, please follow the screen instructions.

After a successful operation, the configuration software can be started via the “Starting Menu/Programs/Thermokon“

Operating systems supported: Windows9x; WindowsNT; WindowsMe; Windows2000; WindowsXP; WindowsServer

7 Configuration of WRF04-RS485-Modbus-CO2

7.1 Software Configuration

By means of the configuration software the configuration registers can be clearly adjusted. Output registers of the WRF04 can be read out and input registers can be set. The load of the individual registers is described in chapter 2.4!

Via the menu points "File" and "Saving of Parameter" respectively "Loading of Parameter", the configuration registers can be stored in a text file and can be reloaded into the WRF04-RS485-Modbus-CO2.

7.2 Parameter-Frame

The Modbus can be accessed via the configuration software by means of a COM-Port. In the "Parameter"-Frame hardware settings can be made. They must be in conformity with the Modbus receiver, in order to produce a connection.

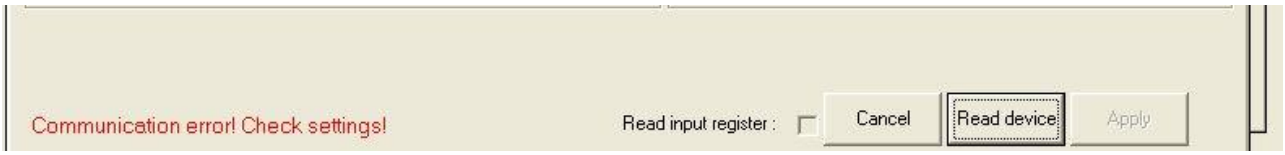
The following options can be selected:

- COM-Port
- Baud rate 9600 , 19200, 57600
- Parity none, even, odd
- Modus for setting of transmission ASCII or RTU
- Modbus address (1-31)

In the field "Modbus address" the address of the WRF06-RS485 Modbus that shall be configured is entered (value between 1 and 31).

Via the selection menu behind "COM-Port" the port can be opened "open" and closed "close".
If the COM-Port is used alreasdy, an error message is shown.

After having opened the COM-Port, the current register values of the device can be read out via the button “read out device“. If no connection to the device can be made, the same is shown by an error message.



Picture 7-1: Communication Problems

7.3 Register

The configuration registers can be set in the different tabs. Furthermore, the output registers can be read and the input registers can be set.

Changes are sent to the WRF04-RS485 Modbus CO2 after having pressed the button "take over". By actuating the button "Cancel" the registers of the WRF04-RS485-Modbus-CO2 are read out again.

By activating the hook "read output register" all output registers are read out cyclically.

Picture 7-2: Data