Modbus on tSENSE

Table of contents:

1. Revision information

Table 1: Document revisions

Rev.	Date:	Author	Status:
1.00	Feb 20, 2014	LN/EN	Specification based on Modbus on CO2 Engine and eSense rev2_01.
1.01	May 05, 2015	LN/EN	HR60-61 added, HR62-63 changed. tSense rev1.03.
			HR57 added

2. General

Modbus is a simple, open protocol for both PLC and sensors. Details on Modbus can be found on www.modbus.org.

This specification describes Modbus functionality on tSENSE. Information about Modbus on tSENSE can be found in "Modbus on CO2 Engine and eSense rev2_01".

2.1. General overview of protocol and sensor implementation

Master – slave:
Only a Modbus master can initiate a transaction. The sensor is a slave and will never initiate communication.

Packet identification:
Every message (packet) starts and ends with a silent interval of 3.5 characters.
Silent interval between characters in message needs to be kept less than 1.5 characters.
An interval is from end of Stop-bit of previous byte to beginning of Start-bit of next byte.

Packet length:
Maximum 255 bytes including address and CRC, according to Modbus specification [1].

Modbus data model:
Four (4) primary data tables (addressable registers)

- Discrete Input (read only bit).
- Coil (read / write bit).
- Input register (read only 16 bit word, interpretation is up to application).
- Holding register (read / write 16 bit word).

Note: Sensor does not support bitwise access of registers.

Exception responses:
Sensor will send answer to master only in case of valid message structure. Sensor can send exception responses when detection of:

- Invalid function code.
- Invalid data address (requested register doesn't exist in given device).
- Invalid data.
- Error in execution of requested function.

3. Byte transmission

RTU transmission mode is only mode supported by sensor.

3.1. Byte format

The format for each byte in RTU mode differs between sensors default configuration and description on page 6 of MODBUS over serial line specification [2].

Table 2: Byte format differences

	MODBUS over serial line specification [2]	**Sensor default configuration**
Coding system	8-bit binary	8-bit binary
Bits per byte:	1 start bit	1 start bit
	8 data bits, least significant bit first	8 data bits, least significant bit first
	1 bit for even parity	NO parity
	1 stop bit	1 stop bit

The reason for this difference is historical, tSENSE can be configured to use standard byte format and baud rate (19200 bps and even parity).

3.2. Baud rate

Required default baud rate: 19200 bps, according to MODBUS over serial line specification [2], p. 13. Baud rate default: 9600 bps. Can be configured for baud rates 9600 – 115200 bps.

3.3. Physical layer

The sensor has an RS-485 driver and can be connected directly to a RS-485 network without additional components.

4. Modbus registers on sensor

Modbus registers are mapped in RAM and EEPROM memories of sensor. Mapping is interpreted by sensor firmware at command reception.

Following restrictive decisions are made:
1. Read only and read / write registers are not allowed to overlay.
2. Bit addressable items (i.e. Coils and Discrete inputs) are not implemented.
3. Only write single register functional codes are implemented.

All registers are 16 bit words (registers are summarized in appendix B). Register address is calculated as “register number -1”.

5. Serial line frame and addressing

5.1. Serial line frame

Modbus over serial line specification [2], distinguishes Modbus Protocol PDU and Modbus serial line PDU in following way (RTU mode only):

Modbus serial line PDU

Address field (1 byte)	Function Code	Data	CRC (High and Low)
	Modbus PDU		

5.2. Addressing rules

Address	Modbus over serial line V1.0	Sensor
0	Broadcast address	No broadcast commands currently implemented
1 – 247	Slave individual address	Slave individual address
248 – 253	Reserved	Slave individual
254	Reserved	“Any sensor” 1)
255	Reserved	Slave individual address

Notes:

1. “Any sensor” means that any sensor with any slave individual address will recognise serial line PDUs with address 254 as addressed to them.

6. Bus timing

Parameter	Min	Typ	Max	Units
Response time-out			200	ms

“Response time-out” is defined to prevent master (host system) from staying in “Waiting for reply” state indefinitely. Refer to page 5 of MODBUS over serial line specification [2].

For slave devices “Response time-out” represents maximum time allowed to take by “processing of required action”, “formatting normal reply” and “normal reply sent” alternatively by “formatting error reply” and “error reply sent”. Refer to slave state diagram on page 6 of document mentioned above.

7. Function codes descriptions

Description of exception responses.

PDU of received command has wrong format:

No response PDU, (sensor does not respond)

Function Code is not equal to any implemented function code:

Exception Response PDU,

Function code	1 byte	Function Code + 0x80
Exception code = ***Illegal Function***	1 byte	0x01

One or more of addressed registers is not assigned (register is reserved or quantity of registers is larger than maximum number of supported registers):

Exception Response PDU,

Function code	1 byte	Function Code + 0x80
Exception code = ***Illegal Data Address***	1 byte	0x02

7.1. 03 (0x03) Read Holding Registers (16 bits read / write registers)

Refer to Modbus specification [1].

Address of Modbus Holding Registers for 1- command reading is limited in range 0x0000..0x003F.

Request PDU

Function code	1 byte	**0x03**
Starting Address Hi	1 byte	Address Hi
Starting Address Lo	1 byte	Address Lo
Quantity of Registers Hi	1 byte	Quantity Hi
Quantity of Registers Lo	1 byte	Quantity Lo

Response PDU

Function code	1 byte	**0x03**
Byte Count	1 byte	2 x N*
Register Value	N* x 2 bytes	

* N = Quantity of Registers

If Address>0x003F or (Address + Quantity)>0x0040:
Exception Response PDU,

Function code	1 byte	**0x83**
Exception code = ***Illegal Data Address***	1 byte	0x02

If Quantity=0:
Exception Response PDU,

Function code	1 byte	**0x83**
Exception code = ***Illegal Data Value***	1 byte	0x03

7.2 04 (0x04) Read Input Registers (16 bits read only registers)

Refer to Modbus specification [1].

Address of Modbus Input Registers for 1-command reading is limited in range 0x0000..0x001F.

Request PDU

Function code	1 byte	**0x04**
Starting Address Hi	1 byte	Address Hi
Starting Address Lo	1 byte	Address Lo
Quantity of Registers Hi	1 byte	Quantity Hi
Quantity of Registers Lo	1 byte	Quantity Lo

Response PDU

Function code	1 byte	**0x04**
Byte Count	1 byte	2 x N*
Register Value	N* x 2 bytes	

* N = Quantity of Registers

If Address>0x001F or (Address + Quantity)>0x0020:
Exception Response PDU,

Function code	1 byte	**0x84**
Exception code = ***Illegal Data Address***	1 byte	0x02

If Quantity=0:
Exception Response PDU,

Function code	1 byte	**0x84**
Exception code = ***Illegal Data Value***	1 byte	0x03

7.3 06 (0x06) Write Single Register (16 bits read / write register)

Refer to Modbus specification [1].

Address of Modbus Holding Registers for 1-command reading/writing is limited in range 0x0000..0x003F.

Request PDU

Function code	1 byte	**0x06**
Starting Address Hi	1 byte	Address Hi
Starting Address Lo	1 byte	Address Lo
Register Value Hi	1 byte	Value Hi
Register Value Lo	1 byte	Value Lo

Response PDU (is an echo of the Request)

Function code	1 byte	**0x06**
Starting Address Hi	1 byte	Address Hi
Starting Address Lo	1 byte	Address Lo
Register Value Hi	1 byte	Value Hi
Register Value Lo	1 byte	Value Lo

If Address>0x003F:
Exception Response PDU,

Function code	1 byte	**0x86**
Exception code = ***Illegal Data Address***	1 byte	0x02

7.2. 43 (0x2B) Read Device Identification

Refer to Modbus specification [1].

The sensor supports only Read Device ID code 4, objects 0x00 to 0x02.

Object ID	**Object Name / Description**	**Type**	**Modbus status**	**Category**	**Implement. Status**
0x00	Vendor Name	ASCII string*	Mandatory	Basic	Implemented
0x01	Product Code	ASCII string*	Mandatory	Basic	Implemented
0x02	Major Minor Revision	ASCII string*	Mandatory	Basic	Implemented

Example:

Vendor Name	= “CO2Meter =
Product Code	“tSENSE”
Major Minor Revision	= “1.00”

Example: Read device identification.

Request PDU, Object ID 0x00 to 0x02

Function code	1 byte	**0x2B**
MEI Type	1 byte	0x0E
Read Device ID code	1 byte	0x04 (individual access only)
Object ID	1 byte	0x00..0x02

Response PDU, Object ID 0x00 to 0x02

Function code	1 byte	**0x2B**
MEI Type	1 byte	0x0E
Read Device ID code	1 byte	0x04, same as in request
Conformity level	1 byte	0x81, basic identification for individual or stream access
More Follows	1 byte	0x00
Next Object ID	1 byte	0x00
Number of objects	1 byte	0x01
Object ID	1 byte	0x00..0x02
Object length	1 byte	0x0B or 0x0E or 0x05 (see definition of ASCII strings)
Object value	n byte	Object Data

8. References

- [1] MODBUS Application Protocol Specification V1.1b3
- [2] MODBUS over serial line specification and implementation guide V1.02

Appendix A: Application examples

Prerequisites for the application examples:

1. A single slave (sensor) is assumed (address “any sensor” is used).
2. Values in <..> are hexadecimal.

CO_2 read sequence:

Sensor is addressed as “Any address” (0xFE).
CO_2 value from IR4 is read by using “Read input registers” (function code 04). Hence, starting address is 0x0003 (register number-1) and Quantity of registers is 0x0001. CRC calculated to 0xC5D5 is sent with low byte first.

Assumption in this example:
Measured CO_2 value by sensor is 400ppm*.
Sensor replies with CO_2 reading 400ppm (400 ppm = 0x190 hexadecimal).

Master request:
<FE> <04> <00> <03> <00> <01> <D5> <C5>

Slave response:
<FE> <04> <02> <01> <90> <AC> <D8>

Sensor status read sequence:

Sensor is addressed as “Any address” (0xFE).
Status from IR1 is read by using “Read input registers” (function code 04). Hence, Starting address is 0x0000 (register number-1) and Quantity of registers 0x0001. CRC calculated to 0xC525 is sent with low byte first.

Sensor replies with status 0.

Master request:
<FE> <04> <00> <00> <00> <01> <25> <C5>

Slave response:
<FE> <04> <02> <00> <00> <AD> <24>

Background calibration sequence:

Sensor is addressed as “Any address” (0xFE).

1. Clear acknowledgement register by writing 0 to HR1. Starting address is 0x0000 and Register value 0x0000. CRC calculated as 0xC59D is sent with low byte first.

   Master request:
   <FE> <06> <00> <00> <00> <00> <9D> <C5>

   Slave response:
   <FE> <06> <00> <00> <00> <00> <9D> <C5>

2. Write command to start background calibration. Parameter for background calibration is 6 and for nitrogen calibration is 7. We write command 0x7C with parameter 0x06 to HR2. Starting address is 0x0001 and Register value 0x7C06. CRC calculated as 0xC76C is sent with low byte first
   Master request:
   <FE> <06> <00> <01> <7C> <06> <6C> <C7>

   Slave response:
   <FE> <06> <00> <01> <7C> <06> <6C> <C7>

3. Wait at least 15 seconds (15s measurement period).

4. Read acknowledgement register. Function 3 “Read Holding register” is used to read HR1. Starting address is 0x0000 and Quantity of registers is 0x0001. CRC calculated as 0x0590 is sent with low byte first.

   Master request:
   <FE> <03> <00> <00> <00> <01> <90> <05>

   Slave response:
   <FE> <03> <02> <00> <20> <AD> <88>

   Check that bit 5 (CI6) is 1, it is an acknowledgement of that the sensor has performed the calibration operation. Sensor can leave out calibration if signal is unstable due to changing CO_2 concentration at the moment of calibration request.

Read Device ID, Vendor Name:

Sensor is addressed as “Any address” (0xFE).
Read Device ID is used to read Vendor Name (object 0, basic access). This object is an ASCII string containing “SenseAir AB”.

Function code is 0x2B, MEI Type 0x0E. Read Device ID code must be 0x04 (since the sensor only supports individual access.) Object ID is 0x00. CRC calculated to 0x3367 is sent with low byte first.

Sensor replies with a packet containing the 11-byte string.

Master request:
<FE> <2B> <0E> <04> <00> <67> <33>

Slave response:
<FE> <2B> <0E> <04> <81> <00> <00> <01> <00> <0B> <53> <65> <6E> <73> <65> <41> <69> <72> <20> <41> <42> <BE> <18>

Slave response contents:
Address = 0xFE
Function code = 0x2B
MEI Type = 0x0E
Read Device ID code = 0x04
Conformity level = 0x81
More Follows = 0x00
Next Object ID = 0x00
Number of objects = 0x01
Object ID = 0x00
Object Length = 0x0B (11 bytes)
Object Value = 0x53 … 0x42 (11 bytes with ASCII codes for “SenseAir AB”)
CRC = 0x18BE sent with low byte first

Read ABC parameter, ABC_PERIOD:

ABC_PERIOD, is available for modification as it is mapped as a holding register.
This example shows how to read ABC_PERIOD by accessing HR32.

Sensor is addressed as “Any address” (0xFE).
Read current setting of ABC_PERIOD by reading HR32. Function code 03 “Read Holding registers” is used. Starting address is 0x001f and Quantity of Registers 0x0001. CRC calculated as 0xC3A1 is sent with the low byte first.

Master request:
<FE> <03> <00> <1F> <00> <01> <A1> <C3>

Slave response:
<FE> <03> <02> <00> <B4> <AC> <27>

Slave response contents:
Address = 0xFE
Function code = 0x03
Byte count = 0x02 - read 2 bytes (1 register of 16 bits)
Register value = 0x00B4 - 0xB4 hexadecimal = 180 decimal;
180 hours / 24 equals 7,5 days.

CRC = 0x27AC - CRC sent with the low byte first

Disable ABC function

ABC function disables by setting ABC_PERIOD to 0.

Sensor is addressed as “Any address” (0xFE).

Function code 06 “Write Single Register” is used to write to HR32. Register address is 0x001f, register value 0x0000. CRC calculated as 0x03AC is sent with the low byte first.

Master request:
<FE> <06> <00> <1F> <00> <00> <AC> <03>

Slave response:
<FE> <06> <00> <1F> <00> <00> <AC> <03>

The response is an echo of the transmitted sequence.

Enable ABC function

Enable ABC function by setting ABC_PERIOD to any value except 0.
In this example it is set to 7,5 days.

Sensor is addressed as “Any address” (0xFE).

Function code 06 “Write Single Register” is used to write to HR32. Register address is 0x001f, register value is 0x00B4 (7,5 days * 24 hours = 180; 180 in hexadecimal format is 0xB4). CRC calculated as 0x74AC is sent with low byte first.

Master request:
<FE> <06> <00 <1F> <00> <B4> <AC> <74>

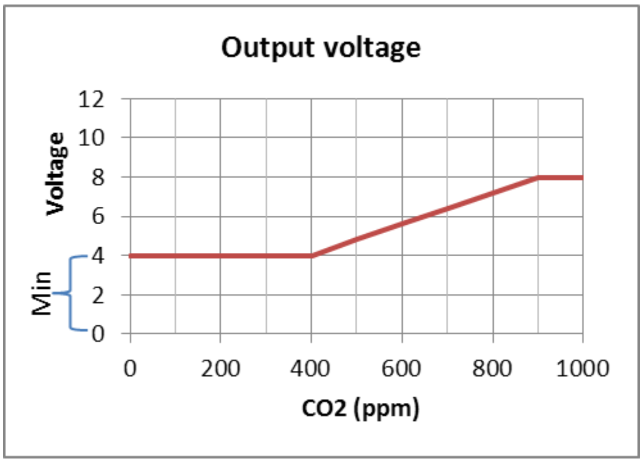
Slave response:
<FE> <06> <00> <1F> <00> <B4> <AC> <74>

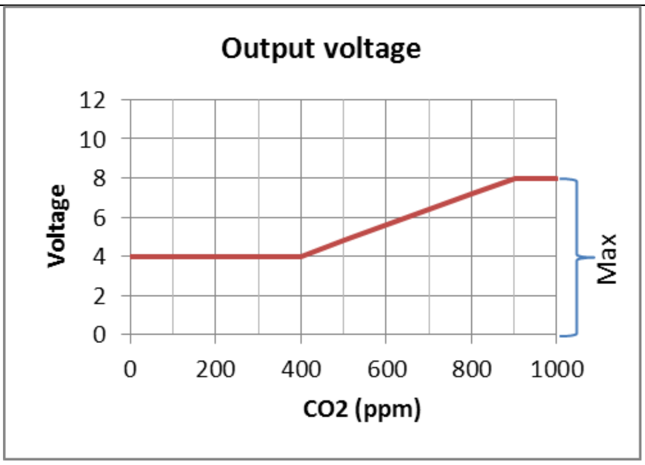
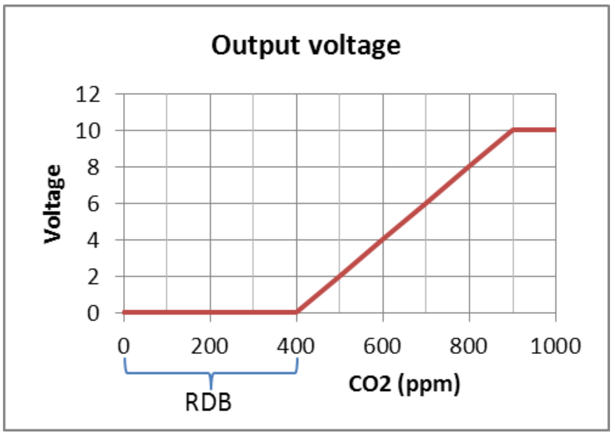
Appendix B: Modbus registers

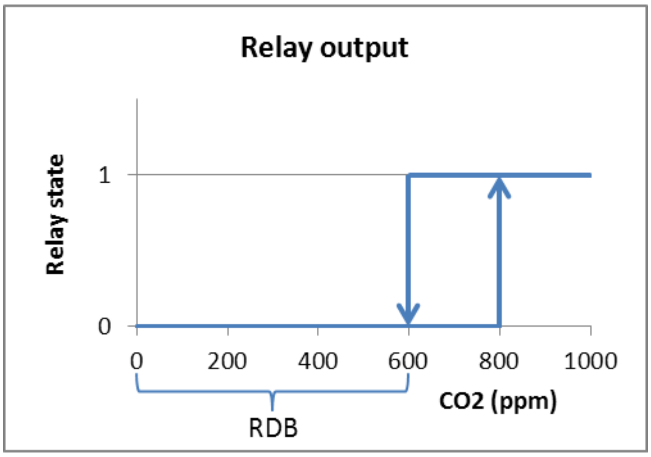
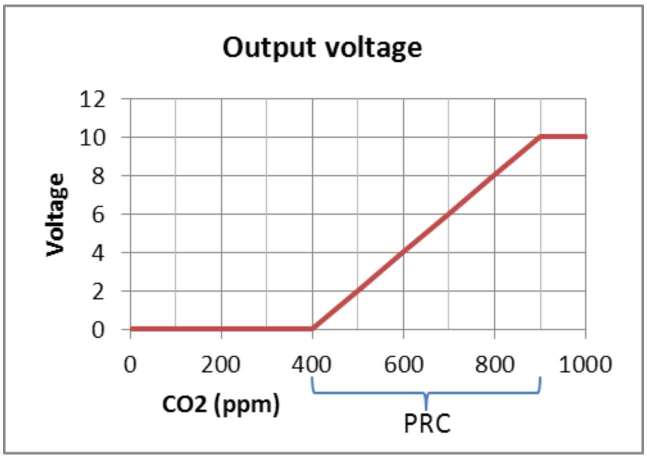
Input registers

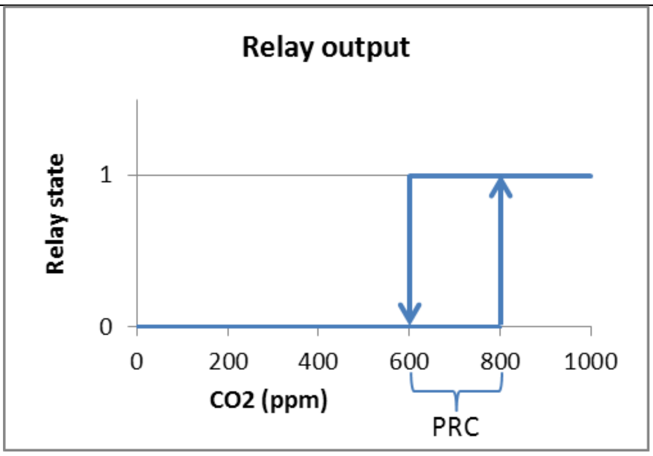
<table>
<tr><th>Register</th><th>Name</th><th>Description/comment</th></tr>
<tr><td>IR1</td><td>Error status</td><td><table>
<tr><td>Bit in register</td><td>Error</td></tr>
<tr><td>0</td><td>CO2 module communication error</td></tr>
<tr><td>1</td><td>CO2 measurement error</td></tr>
<tr><td>2</td><td>NTC measurement error</td></tr>
<tr><td>3</td><td>RH/temperature module communication error</td></tr>
<tr><td>4</td><td>RH measurement error</td></tr>
<tr><td>5</td><td>Temperature measurement error</td></tr>
<tr><td></td><td></td></tr>
<tr><td>8</td><td>Output configuration error</td></tr>
</table></td></tr>
<tr><td>IR2</td><td>Reserved</td><td></td></tr>
<tr><td>IR3</td><td>Reserved</td><td></td></tr>
<tr><td>IR4</td><td>CO2 value</td><td>S16, unit ppm</td></tr>
<tr><td>IR5</td><td>Temperature</td><td>S16, unit °C/°F*100 (register value = 2223 equals 22.23°C)</td></tr>
<tr><td>IR6</td><td>RH</td><td>S16, unit %RH*100 (register value = 4587 means 45.87 % RH)</td></tr>
<tr><td>IR7</td><td>NTC</td><td>S16, unit °C/°F*100 (register value = 2252 equals 22.52°C)</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>IR12</td><td>CO2 status</td><td>Bit 1 in register = CO2 status</td></tr>
<tr><td>IR13</td><td>Temperature status</td><td>Bit 1 in register = temperature status</td></tr>
<tr><td>IR14</td><td>RH status</td><td>Bit 1 in register = RH status</td></tr>
<tr><td>IR15</td><td>NTC status</td><td>Bit 1 in register = NTC status</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>IR22</td><td>Out1 value</td><td>Register value range 0 – 16383, corresponds to 0 – 10V</td></tr>
<tr><td>IR23</td><td>Out2 value</td><td>Register value range 0 – 16383, corresponds to 0 – 10V</td></tr>
<tr><td>IR24</td><td>Out3 value</td><td>Register value range 0 – 16383, corresponds to 0 – 10V</td></tr>
<tr><td>IR25</td><td>Out4 (relay) value</td><td>Register value range 0 – 16383, 0 means inactive relay, 16383 active relay, between 0 and 16383 the relay is in its hysteresis region</td></tr>
<tr><td>IR26</td><td>Out1 status</td><td>Bit 1 in register = Out1 status</td></tr>
<tr><td>IR27</td><td>Out2 status</td><td>Bit 1 in register = Out2 status</td></tr>
<tr><td>IR28</td><td>Out3 status</td><td>Bit 1 in register = Out3 status</td></tr>
<tr><td>IR29</td><td>Out4 (relay) status</td><td>Bit 1 in register = Out4/relay status</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

Holding registers

Register	Name	Description/comment
HR1	CO2 calibration status	After executed calibration (for example trigged by writing to HR2 or pushing button) status is written to HR1. Bit 1 Background calibration performed Bit 2 Zero calibration performed
HR2	CO2 calibration	Background and zero calibration can be trigged by writing 31750 respective 31751 to HR2. Calibration status is written by sensor to HR1 after performed calibration. It is recommended that HR1 is cleared before a calibration is trigged, by writing to HR2, otherwise status from previous calibrations can still be written to HR1. After writing to HR2 it can take up to 15s before calibration is executed. Calibration Register value Background 31750 (0x7C06) Zero 31751 (0x7C07)
HR3	Reserved	
HR4	CO2 value override	S16, default value = 32767 (no override)
HR5	Temperature override	S16, default value = 32767 (no override)
HR6	RH override	S16, default value = 32767 (no override)
HR14	Out1 min value	Defines outputs min voltage, range 0 – 16383 (0 – 10V). Min value must be less than max value  Min value = 0.4* 16383 = 6553 (4V) in figure above
HR15	Out2 min value	Defines outputs min voltage, range 0 – 16383 (0 – 10V). Min value must be lower than max value
HR16	Out3 min value	Defines outputs min voltage, range 0 – 16383 (0 – 10V). Min value must be lower than max value
HR17	Out4 min value	Defines outputs min voltage, for relay output always set min value to 0
HR18	Out1 max value	Outputs max voltage, range 0 – 16383 (0 – 10V). Max value must be higher than min value

		 Max value = 0.8 * 16383 = 13106 (8V) in figure above
HR19	Out2 max value	Outputs max voltage, range 0 – 16383 (0 – 10V). Max value must be higher than min value
HR20	Out3 max value	Outputs max voltage, range 0 – 16383 (0 – 10V). Max value must be higher than min value
HR21	Out4 max value	Outputs max voltage, range 0 – 16383, for the relay output always set max value to 16383
HR22	Out1 override	Default value = 32767 (no override), override range 0 – 16383 (0 – 10V)
HR23	Out2 override	Default value = 32767 (no override), override range 0 – 16383 (0 – 10V)
HR24	Out3 override	Default value = 32767 (no override), override range 0 – 16383 (0 – 10V)
HR25	Out4 (relay) override	Default value = 32767 (no override), only 0 and 16383 (full range) will give a predictable state on the relay
HR29	Reserved	
HR30	RH offset	S16, two decimals, for example “200” gives an offset of 2.00%RH
HR31	Temperature offset	S16, two decimals, for example: “110” gives a offset of - 1.10°C/1.10°F
HR32	ABC period	ABC period in hours, ABC period = 0 disables ABC
HR33	Out1 RDB	Regulators dead-band  RDB = 400 in figure above

HR34	Out2 RDB	Regulator dead-band
HR35	Out3 RDB	Regulator dead-band
HR36	Out4 RDB	Regulator dead-band, defines the inactive region for the relay output  RDB = 600 in figure above
HR37	Out1 sec RDB	Second regulator dead-band for a configurations with more than one input, for example max of CO2 and temp
HR38	Out2 sec RDB	Second regulator dead band
HR39	Out3 sec RDB	Second regulator dead band
HR40	Out4 sec RDB	Second regulator dead band
HR41	Out1 PRC	Proportional regulator constant  PRC = 500 in figure above
HR42	Out2 PRC	Proportional regulator constant
HR43	Out3 PRC	Proportional regulator constant
HR44	Out4 PRC	Defines hysteresis region of relay output

		 PRC = 200 in figure above
HR45	Out1 sec PRC	Second proportional regulator constant for configurations with more than one input, for example max of CO2 and temp
HR46	Out2 sec PRC	Second proportional regulator constant
HR47	Out3 sec PRC	Second proportional regulator constant
HR48	Out4 sec PRC	Second proportional regulator constant
HR49	Out1 OOS override	Permanent override of output. Output will be set to this value even after, for example, power cycling. Default value = 32767 (no OOS override), override range 0 – 16383 (0 – 10V)
HR50	Out2 OOS override	Permanent override, range 0 – 16383 (0 – 10V)
HR51	Out3 OOS override	Permanent override, range 0 – 16383 (0 – 10V)
HR52	Out4 OOS override	Permanent override, range 0 – 16383 (0 – 10V)
HR53	Out1 Error demand	Regulator value used if there is an error on the outputs input value (default CO2 for out1) and the error indication is enabled. Range 0 – 16383 (0 – 10V)
HR54	Out2 Error demand	Error demand for out2, range 0 – 16383
HR55	Out3 Error demand	Error demand for out3, range 0 – 16383
HR56	Out4 Error demand	For relay output only 0 (inactive) and 16383 (active) error demands will give predictable outputs states
HR57	Temperature unit	Temperature unit Register value = 0 mean °C Register value = 1 mean °F
HR60	Stop bits	Modbus stop bits Register value = 1 means one stop bit Register value = 2 means two stop bits All other register values results in one stop bit A sensor reset is needed in order to activate the new setting
HR61	Parity	Modbus parity Register value = 0 gives no parity Register value = 1 gives odd parity Register value = 2 gives even parity Register value higher than 2 results in no parity A sensor reset is needed in order to activate the new setting

HR62	Baud rate (high bits)	Modbus baud rate U32, high bits in HR62 (bit16-31), low bits in HR63 (bit0-15). Baudrate can be calculates as HR62 x 65536 + HR63, for example: Baudrate HR62 HR63 9600 0 9600 19200 0 19200 38400 0 38400 57600 0 57600 76800 1 11264 (1 x 65536 + 11264 = 76800) 115200 1 49664 (1 x 65536 + 49664 = 115200) A sensor reset is needed in order to activate the new setting/baudrate
HR63	Baud rate (low bits)	
HR64	MAC address	Modbus address, valid range 1 - 253

The product and product specification are subject to change without notice. Contact CO2Meter to confirm that the information in this product description is up to date.