Modbus on K45 aSENSE

Table of contents

1. General

This document is valid for the following Senseair sensor models:

Model	Notes
aSENSE	Based on K45 platform

Modbus is a simple, open protocol for both PLC and sensors. Details on Modbus can be found on www.modbus.org.
There are some small differences between Modbus specification [1] and the default implementation in the sensor. The differences are listed in this document.

1.1. General overview of protocol and implementation in the sensor

Master – slave architecture: Only master can initiate transaction. The sensor is a slave and will never initiate communication. The host system initiates transactions to read/write values from / to the corresponding register. The host system shall also check status of the sensor periodically (e.g.2 sec) to determine if it is running without faults detected.

Packet identification: Any message (packet) starts with a silent interval of 3.5 characters. Another silent interval of 3.5 characters marks message end. Silence interval between characters in the message needs to be kept less than 1.5 characters.
Both intervals are from the end of Stop-bit of previous byte to the beginning of the Start-bit of the next byte.

Packet length: According to the Modbus specification, the packet length shall be maximum 255 bytes including address and CRC. We cannot support so large packets. Maximum length of packet (serial line PDU including address byte and 2 bytes CRC) supported by the sensor is 28 bytes. Packets of larger size are rejected without any answer from sensor even if the packet was addressed to the sensor. The number is selected in order to allow reading of Device ID strings of up to 15 bytes in length.

Modbus data model: There are 4 primary data tables (addressable registers), which may overlay:

- 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: The sensor does not support bitwise access of registers.

Exception responses: Slave will send answer to the master only in the case of valid message structure. Nevertheless, it can send exception response because of detection of:

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

2. Byte transmission.

RTU transmission mode is the only mode supported by the sensor.

2.1. Byte format

The format for each byte in RTU mode differs between the sensor default configuration and the description on page 12 of MODBUS over serial line specification [2].

Senseair

Table 1 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	2 stop bits

The reason for the difference is compatibility with test and production systems. Parity bit is configurable to odd or even.

2.2. Baud rate

The sensor has a baud rate of 9600 bps.

2.3. Physical layer

The sensor provides CMOS logical levels RxD and TxD lines for serial transmission. It's up to the system integrator to use them for direct communication with master processor or for connection to RS-232 (3-wires communication) or RS-485 (2-wires communication) drivers. R/T line is available.

The 3 communication lines are fed directly to micro controller with serial 56Ω protection resistors. Power supply to micro controller is 5.0V.

UART_RxD line is configured as digital input.
Input high level is 2.1V min
Input low level is 0.8V max

UART_TxD line is configured as digital output.
Output high level is 2.3V (assuming 5.0V supply) min.
Output low level is 0.75V max

UART_RxD input is pulled up to DVCC = 5.0V by 56kΩ
UART_TxD output is pulled up to DVCC = 5.0V by 56kΩ
UART_R/T is pulled down to DGND by 56kΩ

3. Serial line frame and addressing

3.1. Serial line frame

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

Modbus serial line PDU			
Address field (1 byte)	Function Code	Data	CRC (Hi and Low)
	Modbus PDU		

3.2. Addressing rules

Addressing rules are summarised in the following table:

Address	Modbus over serial line V1.0	K45 aSENSE Sensor
0	Broadcast address	No broadcast commands currently implemented
From 1 to 247	Slave individual address	Slave individual address
From 248 to 253	Reserved	Nothing[1]
254	Reserved	“Any sensor”
255	Reserved	Nothing[1]

Notes:
1. “Nothing” means that sensor doesn’t recognise Modbus serial line PDUs with this address as addressed to the sensor. Sensor does not respond.
2. “Any sensor” means that any sensor with any slave individual address will recognise serial line PDUs with address 254 as addressed to them. They will respond. However, this address is for production / test purposes only. It must not be used in the installed network.
   This is a violation against the Modbus specification [1].
3. Sensors default address is 0x68, for changing the address it’s possible by using Senseair program (UIP5).

4. Bus timing

Parameter	Min	Type	Max	Units
Response time-out			180	msec

Bus timing

“Response time-out” is defined to prevent master (host system) from staying in “Waiting for reply” state indefinitely. Refer to page 9 of MODBUS over serial line specification [2].

For slave device “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 the slave state diagram on page 10 of the document mentioned above.

5. Modbus registers on sensor

The Modbus registers are mapped in memory, both RAM and EEPROM of the sensor.
Mapping is interpreted by sensor firmware at command reception.

5.1. Input registers on K45 aSENSE

IR#	#	Name																
IR1	0	Error status	DI 16	DI 15	DI 14	DI 13	DI 12	DI 11	DI 10	DI 9	DI 8	DI 7	DI 6	DI 5	DI 4	DI 3	DI 2	DI 1
			DI 1 - Fatal error DI 2 - Reserved [1] DI 3 - Algorithm Error DI 4 - Output Error DI 5 - Self diagnostics error DI 6 - Out Of Range error DI 7 - Memory error DI 8 - Warm up state DI 9 - Reserved [1] DI 10 - Reserved [1] DI 11 - Reserved [1] DI 12 - Reserved [1] DI 13 - Reserved [1] DI 14 - Reserved [1] DI 15 - Reserved [1] DI 16 - Reserved [1]															
IR2	1	Alarm status	DI 16	DI 15	DI 14	DI 13	DI 12	DI 11	DI 10	DI 9	DI 8	DI 7	DI 6	DI 5	DI 4	DI 3	DI 2	DI 1
			DI 17 - Reserved [1] DI 18 - Reserved [1] DI 19 - Reserved [1] DI 20 - Reserved [1] DI 21 - Reserved [1] DI 22 - Reserved [1] DI 23 - Reserved [1] DI 24 - Reserved [1] DI 25 - Reserved [1] DI 26 - Reserved [1] DI 27 - Reserved [1] DI 28 - Reserved [1] DI 29 - Reserved [1] DI 30 - Reserved [1] DI 31 - Reserved [1] DI 32 - Reserved [1]															

<table>
<tr><td>IR3</td><td>2</td><td>Output status</td><td>DI 16</td><td>DI 15</td><td>DI 14</td><td>DI 13</td><td>DI 12</td><td>DI 11</td><td>DI 10</td><td>DI 9</td><td>DI 8</td><td>DI 7</td><td>DI 6</td><td>DI 5</td><td>DI 4</td><td>DI 3</td><td>DI 2</td><td>DI 1</td></tr>
<tr><td></td><td></td><td></td><td colspan="16">DI 33 - Reserved [1]
DI 34 - Reserved [1]
DI 35 - Reserved [1]
DI 36 - Reserved [1]
DI 37 - Reserved [1]
DI 38 - Reserved [1]
DI 39 - Reserved [1]
DI 40 - Reserved [1]
DI 41 - Reserved [1]
DI 42 - Reserved [1]
DI 43 - Reserved [1]
DI 44 - Reserved [1]
DI 45 - Reserved [1]
DI 46 - Reserved [1]
DI 47 - Reserved [1]
DI 48 - Reserved [1]</td></tr>
<tr><td>IR4</td><td>3</td><td>CO_2 value</td><td colspan="16">The value is a ppm unit for standard sensors, on the other hand if the sensor is “%” then the value is divided by 10.</td></tr>
<tr><td>IR5</td><td>4</td><td>SpaceTemp [3]</td><td colspan="16">The unit is °C with two decimals, e.g. when the value = 2368 then it is 23.68°C</td></tr>
<tr><td>IR6</td><td>5</td><td>CH10</td><td colspan="16">Channel 10, word register</td></tr>
<tr><td>IR7</td><td>6</td><td>CH11</td><td colspan="16">Channel 11, word register</td></tr>
<tr><td>IR8</td><td>7</td><td>CH12</td><td colspan="16">Channel 12, word register</td></tr>
<tr><td>IR9</td><td>8</td><td>CH13</td><td colspan="16">Channel 13, word register</td></tr>
<tr><td>IR10</td><td>9</td><td>CH6</td><td colspan="16">Channel 6, word register</td></tr>
<tr><td>IR11</td><td>10</td><td>CH7</td><td colspan="16">Channel 7, word register</td></tr>
<tr><td>IR12</td><td>11</td><td>PS_CH4</td><td colspan="16">PS Channel 4, word register</td></tr>
<tr><td>IR13</td><td>12</td><td>Meas. status</td><td colspan="16">Status 0…3, word register</td></tr>
<tr><td>IR14</td><td>13</td><td>Meas. status</td><td colspan="16">Status 4…7, word register</td></tr>
<tr><td>IR15</td><td>14</td><td>Meas. status</td><td colspan="16">Status 8…11, word register</td></tr>
<tr><td>IR16</td><td>15</td><td>Meas. status</td><td colspan="16">Status 12…15, word register</td></tr>
<tr><td>IR17</td><td>16</td><td>Rg Ctrl</td><td colspan="16">Register control 0…3, word register</td></tr>
<tr><td>IR18</td><td>17</td><td>Rg Ctrl</td><td colspan="16">Register control 4…7, word register</td></tr>
<tr><td>IR19</td><td>18</td><td>Rg Ctrl</td><td colspan="16">Register control 8…11, word register</td></tr>
<tr><td>IR20</td><td>19</td><td>Rg Ctrl</td><td colspan="16">Register control 12…15, word register</td></tr>
<tr><td>IR21</td><td>20</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR22</td><td>21</td><td>Out1 value</td><td colspan="16">Value 0 to 16383 correlates to 0 - 10V</td></tr>
<tr><td>IR23</td><td>22</td><td>Out2 value</td><td colspan="16">Value 0 to 16383 correlates to 0 - 10V</td></tr>
<tr><td>IR24</td><td>23</td><td>Out3 value [2]</td><td colspan="16">Value = 0 is inactive state
Value = 16383 is active state
Values between 0 and 16383 means that output is within hysteresis region</td></tr>
</table>

IR25	24	Out4 value [2]	Value = 0 is inactive state Value = 16383 is active state Values between 0 and 16383 means that output is within hysteresis region
IR26	25		Reserved, returns "illegal data address" exception
IR27	26		Reserved, returns "illegal data address" exception
IR28	27		Reserved, returns "illegal data address" exception
IR29	28		Reserved, returns "illegal data address" exception
IR30	29		Reserved, returns "illegal data address" exception
IR31	30		Reserved, returns "illegal data address" exception
IR32	31		Reserved, returns "illegal data address" exception

Notes:

1. Reserved DIs return 0.
2. Out3, Out4 are not available in all models and consecutively these registers are not always present. 0x3FFF (16383Dec) represents 100% output, which refer in sensor's specification for voltage at 100% output.
3. Temperature measurement is not available in all models and consecutively these registers are not always present.

5.2. Holding registers on K45 aSENSE

<table>
<tr><th>HR#</th><th>#</th><th>Name</th><th colspan="16"></th></tr>
<tr><td>HR1</td><td>0</td><td>Acknowledgement register</td><td>DI 16</td><td>DI 15</td><td>DI 14</td><td>DI 13</td><td>DI 12</td><td>DI 11</td><td>DI 10</td><td>DI 9</td><td>DI 8</td><td>DI 7</td><td>DI 6</td><td>DI 5</td><td>DI 4</td><td>DI 3</td><td>DI 2</td><td>DI 1</td></tr>
<tr><td></td><td></td><td></td><td colspan="16">CI 1 - Reserved [1]
CI 2 - Reserved [1]
CI 3 - Reserved [1]
CI 4 - Reserved [1]
CI 5 - Reserved [1]
CI 6 - CO_2 background calibration has been performed
CI 7 - CO_2 nitrogen calibration has been performed
CI 8 - Reserved [1]
CI 9 - Reserved [1]
CI 10 - Reserved [1]
CI 11 - Reserved [1]
CI 12 - Reserved [1]
CI 13 - Reserved [1]
CI 14 - Reserved [1]
CI 15 - Reserved [1]
CI 16 - Reserved [1]</td></tr>
<tr><td>HR2</td><td>1</td><td>Special Command Register [2]</td><td colspan="5">Command</td><td colspan="11">Parameter</td></tr>
<tr><td></td><td></td><td></td><td colspan="5">0x7C</td><td colspan="11">0x6 - CO2 background calibration
0x7 - CO2 zero calibration</td></tr>
<tr><td>HR3</td><td>2</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>HR4</td><td>3</td><td>CO_2 value override</td><td colspan="16"></td></tr>
</table>

HR5	4	SpaceTemp value override	
HR6	5	Out1 set-point	Named RDB (regulator dead band) in some documents
HR7	6		Reserved, returns "illegal data address" exception
HR8	7	Out2 set-point	
HR9	8		Reserved, returns "illegal data address" exception
HR10	9	Out3 set-point [3]	
HR11	10		Reserved, returns "illegal data address" exception
HR12	11	Out4 set-point [3]	
HR13	12		Reserved, returns "illegal data address" exception
HR14	13	Out1 min limit	
HR15	14	Out1 max limit	
HR16	15	Out2 min limit	
HR17	16	Out2 max limit	
HR18	17	Out3 min limit [3]	
HR19	18	Out3 max limit*	
HR20	19	Out4 min limit [3]	
HR21	20	Out4 max limit [3]	
HR22	21	Out1 override	
HR23	22	Out2 override	
HR24	23	Out3 override [3]	
HR25	24	Out4 override [3]	
HR26	25	Out1 PRC	PRC = proportional regulator constant, parameter is named “pband” in some documents
HR27	26		Reserved, returns "illegal data address" exception
HR28	27	Out2 PRC	
HR29	28	Out3 hysteresis [3]	
HR30	29	Out4 hysteresis [3]	
HR31	30	Temperature correction [3]	
HR32	31	ABC Period [4]	

Notes:

1. Reserved DIs return 0.
2. Special Command Register is write-only.
3. Out3, Out4 and temperature measurement are not available in all models and consecutively these registers are not always present.

4. Writing to ABC_Period zero value suspends ABC function. ABC samples and ABC time counting will not be reset. To resume ABC function with prior ABC samples and ABC time write to ABC_Period non-zero value.

6. Supported Modbus commands

Sensor supports following subset of Modbus commands:

1. 03 (0x03) Read Holding Registers
2. 04 (0x04) Read Input Registers
3. 06 (0x06) Write Single Register

6.1. 03 (0x03) Read Holding Registers (16 bits read / write registers)

Refer to Modbus specification [1]

Quantity of Registers is limited to 8.

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

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>0x001F or (Address + Quantity)>0x0020:

Exception Response PDU,

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

If Quantity=0 or Quantity>8

Exception Response PDU,

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

6.2. 04 (0x04) Read Input Registers (16 bits read only registers)

Refer to Modbus specification [1].

Quantity of Registers is limited to 8.

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 or Quantity>8:
Exception Response PDU,

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

6.3.06 (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..0x001F.

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>0x001F:
Exception Response PDU,

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

7. Appendix A: Application examples

Prerequisites for the application examples:

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

7.1. CO_2 read sequence

The sensor is addressed as “Any address” (0xFE).
We read CO_2 value from IR4 using “Read input registers” (function code 04). Hence, starting address will be 0x0003 (register number-1) and Quantity of registers 0x0001. CRC calculated to 0xC5D5 is sent with low byte first.
We assume in this example that by sensor measured CO_2 value is 400ppm*.

Sensor replies with CO_2 reading 400ppm (400 ppm = 0x190 hexadecimal).

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

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

* Note that some models have a different scale factor on the ppm reading. The reading on these models is divided by 10 (i.e. when ambient CO_2 level is 400ppm the sensor will transmit the number 40). In this example the reply from these models would be 40 (= 0x28 hexadecimal).

7.2. Sensor Register control read sequence

The sensor is addressed as “Any address” (0xFE).
We read Register control from IR1 using “Read input registers” (function code 04). Hence, starting address will be 0x0000 (register number-1) and Quantity of registers 0x0001. CRC calculated to 0xC525 is sent with low byte first.

Sensor replies with Register control 0.

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

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

Sensor Register control and CO_2 read sequence

The sensor is addressed as “Any address” (0xFE).
Here we read both Register control and CO_2 in one command by reading IR 1 to 4 using “Read input registers” (function code 04). Hence, starting address will be 0x0000 (register number-1) and Quantity of registers 0x0004. CRC calculated to 0xC6E5 is sent with low byte first.
We assume in this example that by sensor measured CO_2 value is 400ppm*.

Sensor replies with Register control=0 and CO_2 value 400ppm (0x190 hexadecimal).

Master Transmit:
<FE> <04> <00> <00> <00> <04> <E5> <C6>

Slave Reply:
<FE> <04> <08> <00> <00> <00> <00> <00> <00> <01> <90> <16> <E6>
| Reg control | | CO_2 value |

* Note that some models have a different scale factor on the ppm reading. The reading on these models is divided by 10 (i.e. when ambient CO_2 level is 400ppm the sensor will transmit the number 40). In this example the reply from these models would be 40 (= 0x28 hexadecimal).

7.3. Background calibration sequence

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

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 Transmit:
<FE> <06> <00> <00> <00> <00> <9D> <C5>

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

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 Transmit:
<FE> <06> <00> <01> <7C> <06> <6C> <C7>

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

Wait at least 2 seconds for standard sensor with 2 sec lamp cycle.

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

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

Slave Reply:
<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. The sensor may skip calibration; an example of a reason for this could be unstable signal due to changing CO_2 concentration at the moment of the calibration request.

7.4. Read ABC parameter, ABC_PERIOD

One of the ABC parameters, 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.

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

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

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

In the slave reply we can see:

Address = 0xFE	
Function code = 0x03	
Byte count = 0x02	- We 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 low byte first

7.5. Disable ABC function

We can disable the ABC function by setting ABC_PERIOD to 0.

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

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

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

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

We can see the reply which is an echo of the transmitted sequence.

7.6. Enable ABC function

We can enable the ABC function by setting ABC_PERIOD to some value other than 0. In this example, we set it to 7,5 days.

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

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

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

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

We can see the reply which is an echo of the transmitted sequence.

8. Configuring K45 aSENSE by Modbus

This section describes how to modify the sensors settings and override values by writing to Modbus holding registers.

8.1. Override

Holding registers 4,5 and 22-25 can be used to override current CO2, temperature and output values. To inactivate an override write 32767 (0x7FFF) to the register, the overrides are set to inactive in sensor initialisation at power on.

CO2 override example
Override CO2 to 400 ppm
Write 400 to HR4

Master transmit:
<FE> <06> <00> <03> <01> <90> <6C> <39>

Slave reply:
<FE> <06> <00> <03> <01> <90> <6C> <39>

We can see the reply which is an echo of the transmitted sequence.

Inactivate override
Write 32767 to HR4

Master transmit:
<FE> <06> <00 <03> <7F> <FF> <0D> <B5>

Slave reply:
<FE> <06> <00> <03> <7F> <FF> <0D> <B5>

We can see the reply which is an echo of the transmitted sequence.

Temperature override example
Override temperature to 22°C
Write 2200 to HR5 (unit = °C x 100)

Master transmit:
<FE> <06> <00> <04> <08> <98> <DA> <6E>

Slave reply:
<FE> <06> <00> <04> <08> <98> <DA> <6E>

We can see the reply which is an echo of the transmitted sequence.

Inactivate override
Write 32767 to HR5

Master transmit:
<FE> <06> <00> <04> <7F> <FF> <BC> <74>

Slave reply:
<FE> <06> <00> <04> <7F> <FF> <BC> <74>

We can see the reply which is an echo of the transmitted sequence.

Senseair

Output value override example

Override out1 to 4V (0-10V configuration)

Write 6553 to HR22, the value 16383 gives max output voltage on the output, 0.4 x 16383 = 6553 gives 4V on the output

Master transmit:
<FE> <06> <00> <15> <19> <99> <47> <FB>

Slave reply:
<FE> <06> <00> <15> <19> <99> <47> <FB>

We can see the reply which is an echo of the transmitted sequence.

Inactivate out1 override

Write 32767 to HR22

Master transmit:
<FE> <06> <00 <15> <7F> <FF> <EC> <71>

Slave reply:
<FE> <06> <00> <15> <7F> <FF> <EC> <71>

We can see the reply which is an echo of the transmitted sequence.

Override out3 to active state

Write 16383 to HR24
Master transmit:
<FE> <06> <00 <17> <3F> <FF> <7C> <71>

Slave reply:
<FE> <06> <00> <17> <3F> <FF> <7C> <71>

Override out3 to inactive state

Write 0 to HR24
Master transmit:
<FE> <06> <00 <17> <00> <00> <2D> <C1>

Slave reply:
<FE> <06> <00> <17> <00> <00> <2D> <C1>

Inactivate out3 override
Write 32767 to HR24
Master transmit:
<FE> <06> <00 <17> <7F> <FF> <4D> <B1>

Slave reply:
<FE> <06> <00> <17> <7F> <FF> <4D> <B1>

Note. For digital outputs (out3 and out4) only the override values 0, 16383 and 32767 should be used, for other values the output will keep its current state (output/relay will act as within the hysteresis region).

8.2. Output configuration

The outputs behaviour can be modified by writing to the sensors holding registers, examples below shows how to change configuration for analogue and digital (relay) outputs.

8.2.1. Analogue out1 example

Change configuration for an K45 aSENSE standard sensor to 500-1500ppm and 4-10V as described in table below.

	K45 aSENSE standard configuration	New configuration	Comment
Setpoint/RDB	0	500	
PRC	2000	1000	
Min limit	2 or 0 V (depend on jumper position)	4 or 0V	Output voltage 0 - 10V correlates to min/max limit 0 – 16383 (0V ⇔ limit = 0, 10V ⇔ limit = 16383)
Max limit	10 V	8	

Step 1, write 500 to HR6 to change the setpoint to 500ppm

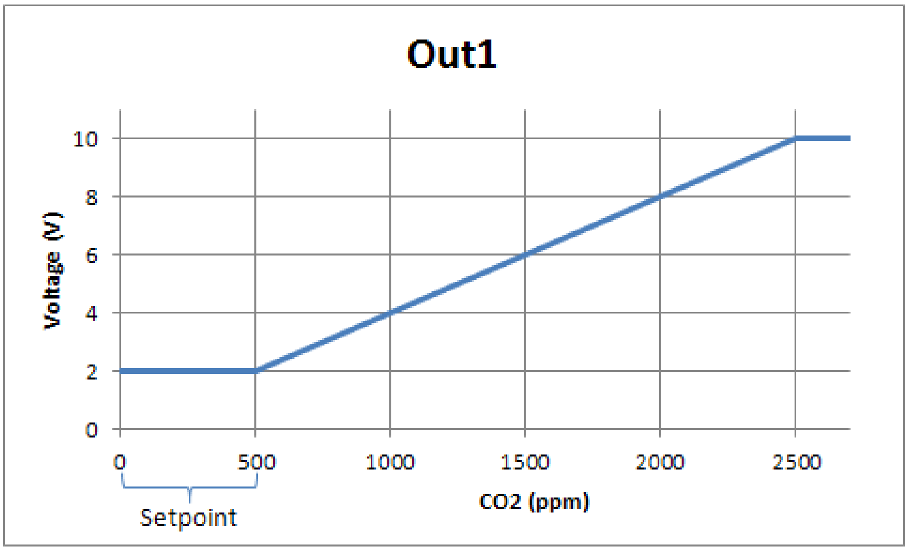

Figure 1 Out1 configuration after increasing setpoint to 500ppm

Step 2, write 1000 to HR26 to set PRC to 1000ppm

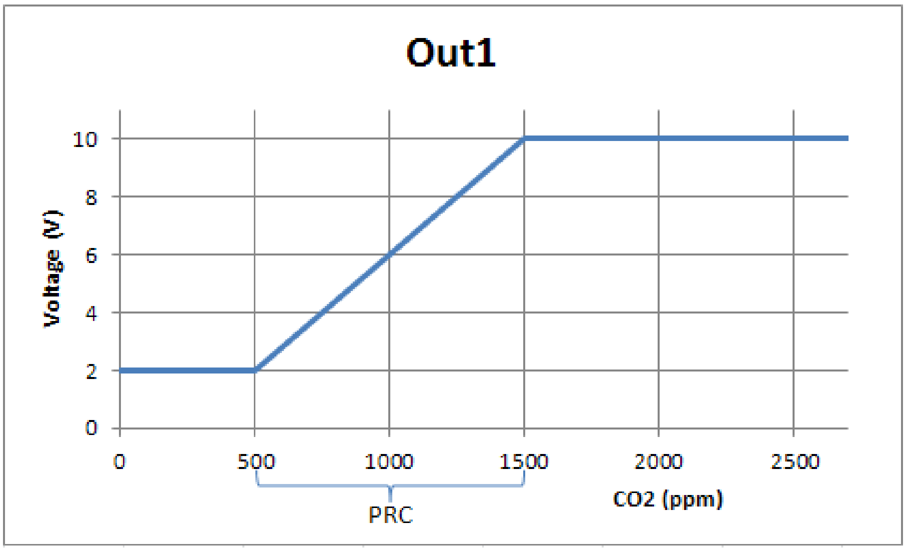

Figure 2 Out1 configuration after increasing PRC to 1000 ppm

Step 3, write 6553 to HR14 to set min limit to 4V (0.4 x 16383 = 6553)

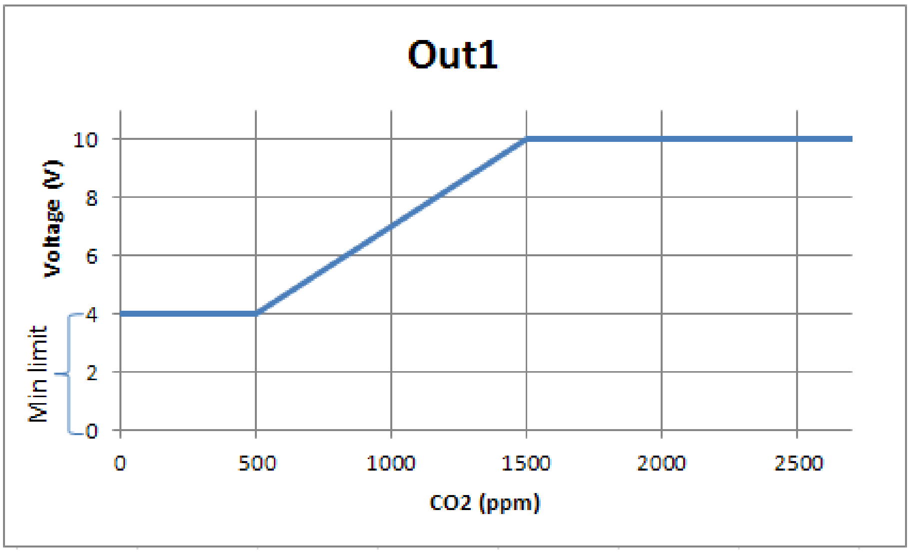

Figure 3 Out1 configuration after changing min voltage to 4V

Step 4, write 13106 to HR15 to set min limit to 8V (0.8 x 16383 = 13106)

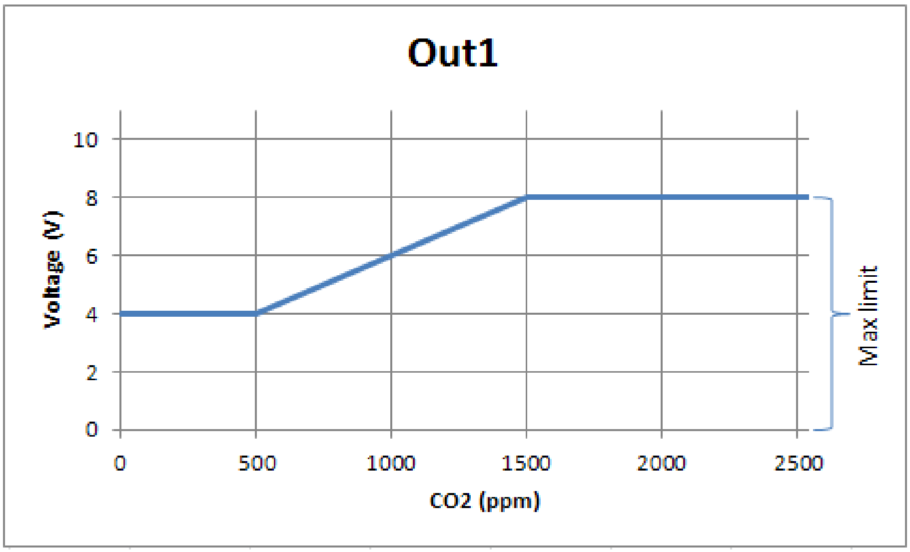

Figure 4 Out1 configuration after changing max voltage to 8V

Out1 configuration after changing setpoint to 500ppm, PRC to 1000ppm, min limit to 4V and max limit to 4V

8.2.2. Digital out3 example

Change configuration for an K45 aSENSE RL sensor as described in table below.

	K45 aSENSE RL configuration	New configuration
RDB	900	1800
PRC	100	200
Min limit	Min (0)	Min (0)
Max limit	Max (16383)	Max (16383)

Step 1, write 1800 to HR10 to change RDB to 1800ppm

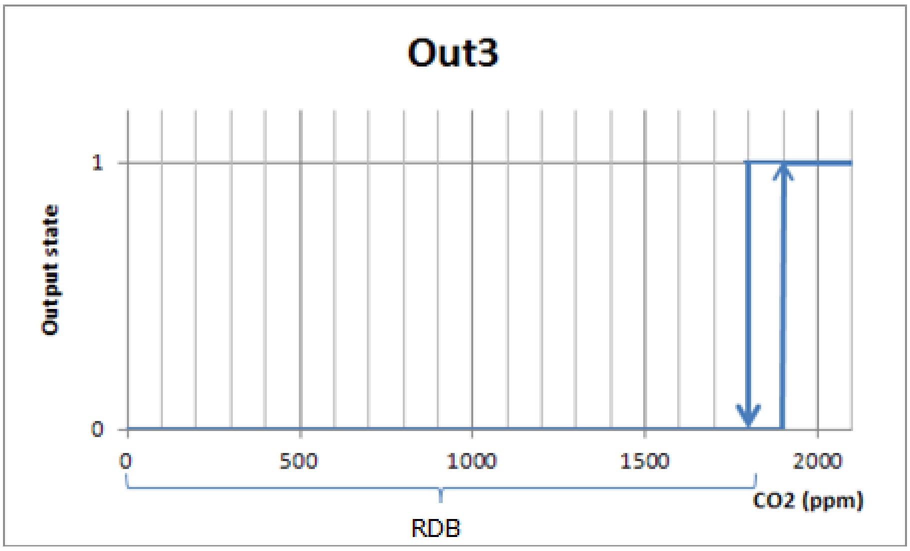

Figure 5 Out3 configuration after increasing RDB to 1800 ppm

Step2, write 200 to HR29 to change hysteresis to 200 ppm

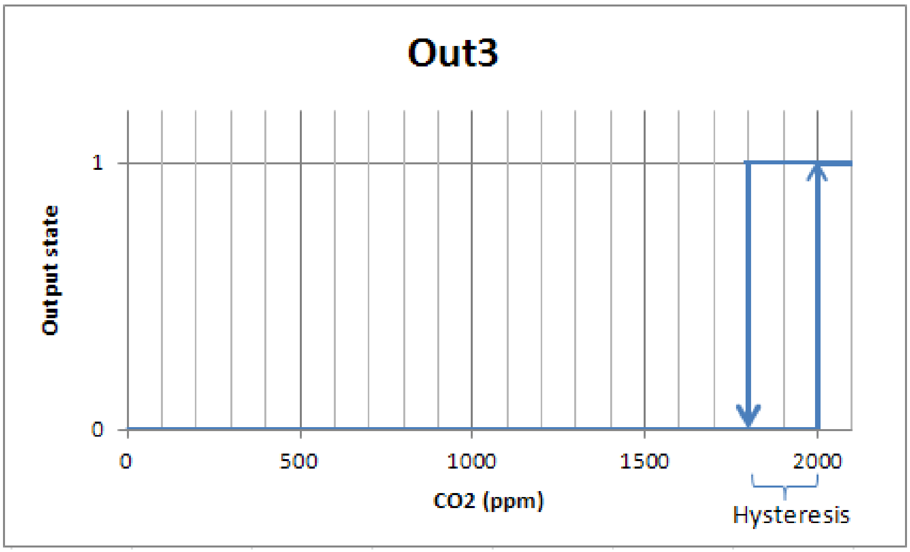

Figure 6 Out3 configuration after changing hysteresis to 200 ppm

There is no need to change max/min limits for the digital outputs, they should be set to 0 respective 16383 as in the standard product.

8.3. Temperature adjustment

Holding register 31 gives the possibility to add an offset to the temperature value, by default this value is set to zero. The unit for the temperature correction is °C x 100 as for the temperature, the sign of the offset is negative, this means that when writing a positive value to HR31 the temperature value shown by the sensor will decrease.

Temperature adjustment example:
Decrease current temperature value by 0.7°C
Write 70 to HR31

Master transmit:
<FE> <06> <00 <1E> <00> <46> <7C> <31>

Slave reply:
<FE> <06> <00> <1E> <00> <46> <7C> <31>

9. References

[1] MODBUS Application Protocol Specification V1.1a
[2] MODBUS over serial line specification and implementation guide V1.01