Modbus on Senseair S88

1. General

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

This specification is based on the specification of Modbus implementation on aSense, eSense and Sensor Core and aims to support backwards compatibility with them. The differences between the Modbus specification [1] and the default implementation in the sensor are listed in this document.

Table of contents:

1. General 1
2. Byte transmission. 3
3. Modbus registers on sensor. 4
4. Serial line frame and addressing. 9
5. Bus timing. 9
6. Function codes descriptions (PUBLIC). 10
7. References 14
8. Appendix A: Application examples 15
9. Appendix B: Understanding Output configuration 22
10. Appendix C: Compatibility with Senseair S8 and S88 Modbus definitions. 24
11. Revision history 26

General overview of protocol and implementation in the sensor

Master – slave:
Only master can initiate a transaction. The sensor is a slave and will never initiate communication. The host system initiates transactions to read CO_2 value from the corresponding register. The host system shall also check status of the sensor periodically (e.g. every two (2) seconds) in order 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 [1], the packet length shall be maximum 252 bytes including address and CRC. Packets of larger size are rejected without any answer from sensor

Modbus data model:
There are four (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].

	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
Data bits	8 data bits, least significant bit first		8 data bits, least significant bit first
	1 bit for even parity	No parity bit	NO parity bit
	1 stop bit	2 stop bits	1 stop bit for receiving 2 stop bits at transmission

Table 1: Byte format differences

Implementation of 1 stop bit on receive and 2 stop bits at transmit provides compatibility with masters using both 1 and 2 stop bits.

2.2. Baud rate (data signalling rate)

9600 bps and 19200 bps are required baud rates and required default baud rate according to MODBUS over serial line specification [2], page 20, is 19200 bps.

Senseair S88 supports 9600 baud rate only.

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 or RS-485 drivers. Sensor has support for R/T control line.

Communication lines are fed directly to the micro-controller of the sensor. Please refer to technical description for electrical specifications of particular model.

3. 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.

Presently, the 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) will not be implemented.
3. The total number of registers should be limited. Present decision is to limit number of input registers to 32 and number of holding registers to 33.
4. Larger amount of data should be transferred as file. It is not implemented at the current stage of development.

Maps of registers (All registers are 16 bit word) are summarised in Table 3 and Table 4. Associated number is Modbus register number: Register address is calculated as (register number -1

IR#	#	Name																
IR1	0	MeterStatus	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 DI 7 - Memory error DI 8 - Warm Up 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	AlarmStatus	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]															
IR3	2	Output 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 33 - Alarm Output status (inverted due to Open Collector) DI 34 - PWM Output status (1 means full output) 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]
IR4	3	Space CO_2	Space CO_2
IR5	4	Space Temp	Sensor temperature, usually above environment temperature due to sensor's self-heating
IR6	5	Synchro	Increments every measurement period
IR7	6	Vbb	VBB voltage during active lamp ramp time, LSB - 1mV
IR8	7		Reserved, returns "illegal data address" exception
IR9	8		Reserved, returns "illegal data address" exception
IR10	9		Reserved, returns "illegal data address" exception
IR11	10		Reserved, returns "illegal data address" exception
IR12	11		Reserved, returns "illegal data address" exception
IR13	12		Reserved, returns "illegal data address" exception
IR14	13		Reserved, returns "illegal data address" exception
IR15	14		Reserved, returns "illegal data address" exception
IR16	15		Reserved, returns "illegal data address" exception
IR17	16		Reserved, returns "illegal data address" exception
IR18	17		Reserved, returns "illegal data address" exception
IR19	18		Reserved, returns "illegal data address" exception
IR20	19		Reserved, returns "illegal data address" exception
IR21	20		Reserved, returns "illegal data address" exception
IR22	21	PWM Output[2]	PWM Output[2]
IR23	22		Reserved, returns "illegal data address" exception
IR24	23	ETC High[6]	Elapsed Time counter, increments every hour.
IR25	24	ETC Low[6]	
IR26	25		Sensor Type ID High[3]

IR27	26		Sensor Type ID Low[3]
IR28	27		Memory Map version
IR29	28		FW version Main.Sub[4]
IR30	29		Sensor ID High[5]
IR31	30		Sensor ID Low[5]
IR32	31		Reserved, returns "illegal data address" exception

Table 2 : Input Registers

[1] – Reserved DIs return 0.
[2] – 0x3FFF represents 100% output. Refer to sensor model's specification for voltage at 100% output.
[3] – IR26 low byte + IR27 contains Sensor Type ID 3-bytes value.
[4] – IR29 high byte is FW Main revision, low byte – FW Sub revision.
[5] – IR30 + IR31 – 4-bytes Sensor's Serial Number.
[6] – IR24 + IR25 – 4-bytes ETC.

<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 - Reset calibration has been performed
CI 2 - Reserved [6]
CI 3 - Force ABC calibration has been performed
CI 4 - Reserved [6]
CI 5 - CO_2 target calibration has been performed
CI 6 - CO_2 background calibration has been performed
CI 7 - CO_2 zero calibration has been performed
CI 8 - Reserved [6]
CI 9 - Reserved [6]
CI 10 - Reserved [6]
CI 11 - Reserved [6]
CI 12 - Reserved [6]
CI 13 - Reserved [6]
CI 14 - Reserved [6]
CI 15 - Reserved [6]
CI 16 - Reserved [6]</td></tr>
<tr><td>HR2</td><td>1</td><td>Special Command Register[7]</td><td colspan="6">Command</td><td colspan="10">Parameter</td></tr>
<tr><td></td><td></td><td></td><td colspan="6">0x7C</td><td colspan="10">0x1 - Reset CO_2 calibration to factory defaults
0x3 - Force ABC calibration
0x5 - CO_2 target calibration
0x6 - CO_2 background calibration
0x7 - CO_2 zero calibration</td></tr>
<tr><td>HR3</td><td>2</td><td>Calibration Target Register</td><td colspan="16">Target concentration for “CO_2 target calibration”</td></tr>
<tr><td>HR4</td><td>3</td><td>Pressure</td><td colspan="16">Use 0 or negative value to disable pressure compensation. LSB - 0.1hPa</td></tr>
<tr><td>HR5</td><td>4</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>HR6</td><td>5</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>HR7</td><td>6</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>HR8</td><td>7</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>HR9</td><td>8</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
<tr><td>HR10</td><td>9</td><td>OUT0 RDB</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR11</td><td>10</td><td>OUT0 PRC</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR12</td><td>11</td><td>OUT1 RDB</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR13</td><td>12</td><td>OUT1 PRC</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR14</td><td>13</td><td>OUT1 MinLimit</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR15</td><td>14</td><td>OUT1 MaxLimit</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR16</td><td>15</td><td>OUT1 Offset</td><td colspan="16">See B: Understanding Output configuration. EEPROM mapped</td></tr>
<tr><td>HR17</td><td>16</td><td></td><td colspan="16">Reserved, returns "illegal data address" exception</td></tr>
</table>

HR18	17		Reserved, returns "illegal data address" exception
HR19	18		Reserved, returns "illegal data address" exception
HR20	19	Communication address	Communication address. Valid values in 1...254 range. EEPROM mapped
HR21	20		Reserved, returns "illegal data address" exception
HR22	21	OUT1 Override	Use to override "PWM Output" variable to test PWM output
HR23	22		Reserved, returns "illegal data address" exception
HR24	23		Reserved, returns "illegal data address" exception
HR25	24		Reserved, returns "illegal data address" exception
HR26	25		Reserved, returns "illegal data address" exception
HR27	26	Default pressure	EEPROM mapped, optional. Use 0 or negative value to disable pressure compensation. Load on start-up into HR4_Pressure. LSB - 0.1hPa Pressure compensation disabled by default
HR28	27	Static IIR config	IIR filter configuration: 0 - filtering disabled, 1 to 16 - static filter parameter, 255 - factory default settings. **Note:** If value of the variable is not equal to 255, ABC functionality will omit environment gas stability checks.
HR29	28		Reserved, returns "illegal data address" exception
HR30	29		Reserved, returns "illegal data address" exception
HR31	30		Reserved, returns "illegal data address" exception
HR32	31	ABC Period	ABC Period in hours[8] EEPROM mapped
HR33	32	ABC Target	ABC target in concentration bins. EEPROM mapped

Table 3: Holding Registers

[6] – Reserved CIs return 0.
[7] – Special Command Register is write-only.
[8] – 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.

Note: The new settings in the registers (excluding HR1-HR3, HR22) applies only after sensor restart

Note: Holding registers are EEPROM mapped parameters (excluding HR1-HR4, HR22), this means that too frequent writes to these registers will lead to a EEPROM corrupt. Total number of EEPROM write cycles should be less than 10000. When writing multiple registers in one sequence then this write cycle will be counted as just one write cycle out of the 10000 that are allowed writes to the EEPROM. If sensor is powered down/reset when EEPROM write operations are ongoing it may result in corrupt parameters. Please wait at least 180ms before powering down/reset sensor after holding registers change (or wait sensor response on corresponding Modbus request).

4. Serial line frame and addressing.

4.1. 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		

4.2. Addressing rules

Addressing rules are summarised in the table:

Address	Modbus over serial line V1.0	Senseair S88 Sensor
0	Broadcast address	Broadcast address
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)]

Table 4: Summarised addressing rules

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. So that 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].

4.3. Broadcast address

Modbus specification [1] requires execution of all write commands in the broadcast address mode.

Current status for the sensor: Broadcast commands are implemented.

5. Bus timing.

Parameter	Min	Type	Max	Units
Response time-out			180	msec

Table 5: 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.
Note: Due to hardware limitations, probability for end-customers to get the response timeout on a Modbus request in normal conditions could be 0.000007. For reliable sensor reading, the host shall use retries in communication.

6. Function codes descriptions (PUBLIC).

Description of exception responses

If the PDU of the received command has wrong format:

No Response PDU, (sensor doesn't respond)

If Function Code isn't equal to any implemented function code:

Exception Response PDU,

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

If 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

6.1. 01 (0x01) Read Coils (one bit read / write registers).

Not implemented.

6.2. 02 (0x02) Read Discrete Inputs (one bit read only registers).

Not implemented.

6.3. 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..0x0020.

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

Senseair

If Address>0x0020 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>0x007D:
Exception Response PDU,

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

6.4. 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 or Quantity>0x007D:
Exception Response PDU,

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

6.5. 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..0x0020.

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>0x0020:
Exception Response PDU,

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

6.6. 16 (0x10) Write Multiple Registers (16 bits read / write 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..0x0020.

Request PDU

Function code	**1 byte**	**0x10**
Starting Address Hi	1 byte	Address Hi
Starting Address Lo	1 byte	Address Lo
Number of Register Hi	1 byte	Value Hi
Number of Register Lo	1 byte	Value Lo
The Number of Data Bytes	1 byte	2 x N*
Register Value to Write	2 x N* bytes	Value to write

* N = Quantity of Registers

Response PDU (is an echo of the Request)

Function code	**1 byte**	**0x10**
Starting Address Hi	1 byte	Address Hi
Starting Address Lo	1 byte	Address Lo
Number of Register written Hi	1 byte	Value Hi
Number of Register written Lo	1 byte	Value Lo

If Address is out of range:
Exception Response PDU

Function code	**1 byte**	**0x90**
Exception code = *Illegal Data Address*	1 byte	0x02

43 / 14 (0x2B / 0x0E) Read Device Identification.

Refer to Modbus specification …
The sensor only supports Read Device ID code 4, individual access.
Objects 0x00 ..0x02 (basic identification) are available (see table)

Object ID	Object Name / Description	Type	Modbus status	Category	Implementation status
0x00	Vendor Name	ASCII string*	Mandatory	Basic	Implemented
0x01	ProductCode	ASCII string*	Mandatory	Basic	Implemented
0x02	MajorMinorRevision	ASCII string*	Mandatory	Basic	Implemented
0x03	VendorUrl	ASCII string*	Optional	Regular	Not Implemented
0x04	ProductName	ASCII string*	Optional	Regular	Not Implemented
0x05	ModelName	ASCII string*	Optional	Regular	Not Implemented
0x06	UserApplicationName	ASCII string*	Optional	Regular	Not Implemented
0x07.. 0x7F	Reserved				
0x80	Memory map version	1 byte unsigned	Optional	Extended	Not Implemented
0x81	Firmware revision, consists of: Firmware type, Revision Main, Revision Sub	3 bytes unsigned	Optional	Extended	Not Implemented
0x82	Sensor serial number (sensor ID)	4 bytes unsigned	Optional	Extended	Not Implemented
0x83	Sensor type	3 bytes unsigned	Optional	Extended	Not Implemented

*The ASCII strings are different for different models and firmware revision. Product Code is the sensors article number. As an example:
Vendor Name = “Senseair” (length 8 bytes)
MajorMinorRevision = “1.00” (length 4 bytes)
Product Code = “004-1-0100” (length 10 bytes)
Firmware revision 1.00 and later example:

Example: Read objects of category “Basic”
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 0x07 or 0x04 (see definition of ASCII strings)
Object value	n byte	Object Data

If wrong MEI Type:
Exception Response PDU

Function code	**1 byte**	**0xAB**
Exception code = *Illegal Function Code*	1 byte	0x01

If Object ID is not in range 0x00..0x03:
Exception Response PDU

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

If wrong Device ID:
Exception Response PDU

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

Note: The exception response for function code 43 is implemented according to the RFC "RFC Non extended Exception code format of 43 Encapsulated Transport.doc" which is in status "Recommended for approval" at time of writing. This is in contrast with the Modbus specification [1] where the exception responses for function code 43 also have a MEI type field.

7. References

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

8. 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.

8.1. 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 future models in the Senseair S8 family of sensors may 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 one of these models would be 40 (= 0x28 hexadecimal).

8.2. Sensor status read sequence:

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

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

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

8.3. Sensor status and CO_2 read sequence:

The sensor is addressed as “Any address” (0xFE).
Here we read both status 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 status=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>
| Status | | CO2 |

* Note that some future models in the Senseair S88 family of sensors may 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 one of these models would be 40 (= $0x28_{16}$).

8.4. Background calibration sequence:

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

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

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

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

4. 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.

8.5. Target calibration sequence:

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

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

2. Write target concentration into HR3 register, for example 600 ppm (0x0258 in hexadecimal):

Master Transmit:
<FE> <06> <00> <02> <02> <58> <3C> <9F>

Slave Reply:
<FE> <06> <00> <02> <02> <58> <3C> <9F>

3. Write command to start target calibration. Parameter for target calibration is 5. We write command 0x7C with parameter 0x05 to HR2. Starting address is 0x0001 and Register value 0x7C05. CRC calculated as 0xC62C is sent with low byte first.

Master Transmit:
<FE> <06> <00> <01> <7C> <05> <2C> <C6>

Slave Reply:
<FE> <06> <00> <01> <7C> <05> <2C> <C6>

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

5. 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> <10> <AD> <9C>

Check that bit 4 (CI5) 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.

8.6. 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

8.7. Disable ABC function:

Disable the ABC function by setting ABC PERIOD to 0.
The 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 low byte first.

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

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

The reply can be seen as an echo of the transmitted sequence.

8.8. Enable ABC function:

Enable the ABC function by setting ABC PERIOD to any value except 0. Set to 7.5 days in this example.

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

Function code 06 “Write Single Registers” is used to write to HR32. Register address is 0x001f, register value 0x00B4 (7.5 days x 24 hours = 180_{10}; 180_{10} = $B4_{16}$).
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>
The reply can be seen as an echo of the transmitted sequence.

8.9. Change Modbus address:

To change sensor's Modbus address, write correct Modbus address into HR20 register and restart sensor.

The sensor is addressed as "Any address" (0xFE), address to set is 0x21:

Master transmit:
<FE> <06> <00> <13> <00> <21> <AC> <18>

Slave reply:
<FE> <06> <00> <13> <00> <21> <AC> <18>

Apply new setting by restarting sensor using power-off/power-on sequence.

8.10. Set current pressure:

If host system has information about environment pressure it can be used to enable pressure compensation in sensor's concentration reading. Write current pressure value into HR4 register.

The sensor is addressed as "Any address" (0xFE), pressure value set is 1013.1 hPa (10311 in 0.1 hPa, which is 0x2793 in hexadecimal):

Master transmit:
<FE> <06> <00> <03> <27> <93> <36> <58>

Slave reply:
<FE> <06> <00> <03> <27> <93> <36> <58>

Note: For consistent concentration filtering, pressure value shall be written before or just after first measurement of the sensor after power-up/reset. If this rule could not be applied, the host system shall take in consideration possible delay in change of concentration reading due to pressure compensation due to filtering.

8.11. Set default pressure:

If the sensor is used at high altitude and host system has no possibility to write current pressure into the sensor (or host system is absent), it is possible to enable static pressure compensation which sensor will load at every power-on/reset conditions. Write static pressure value into HR27 register.

The sensor is addressed as "Any address" (0xFE), static pressure value set is 1013.1 hPa (10311 in 0.1 hPa, which is 0x2793 in hexadecimal):

Master transmit:
<FE> <06> <00> <1A> <27> <93> <E7> <9F>

Slave reply:
<FE> <06> <00> <1A> <27> <93> <E7> <9F>

The static pressure value will be loaded into HR4 register at next sensor power-up/reset cycle.

8.12. Change filter configuration:

To disable filter for concentration, write 0 into HR28:

Master transmit:
<FE> <06> <00> <1B> <00> <00> <ED> <C2>

Slave reply:
<FE> <06> <00> <1B> <00> <00> <ED> <C2>

To restore default filtering parameters write 0xFF into HR28:

Master transmit:
<FE> <06> <00> <1B> <00> <FF> <AD> <82>

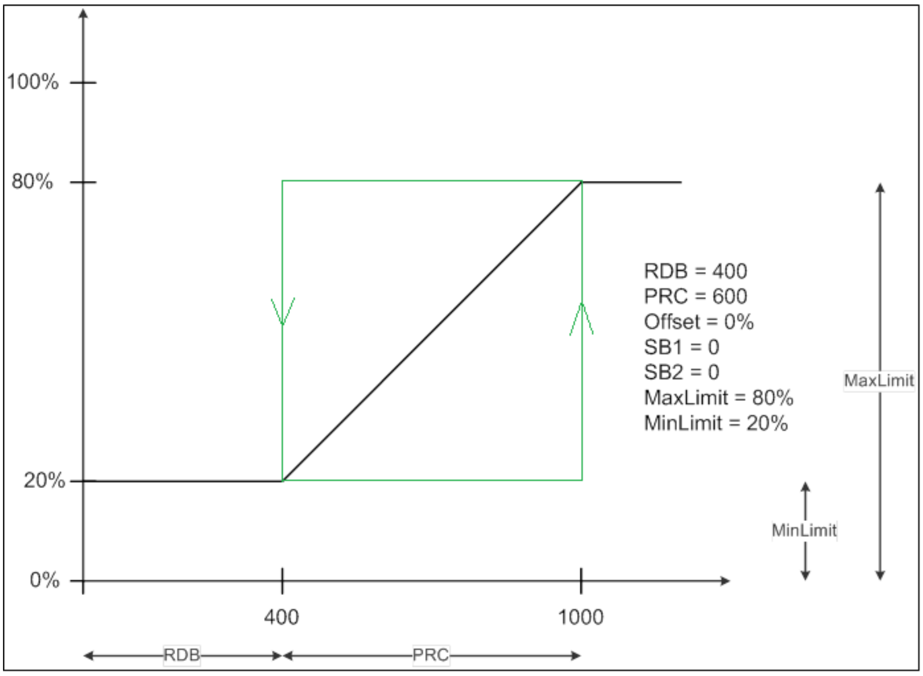
Slave reply:
<FE> <06> <00> <1B> <00> <FF> <AD> <82>

9. Appendix B: Understanding Output configuration

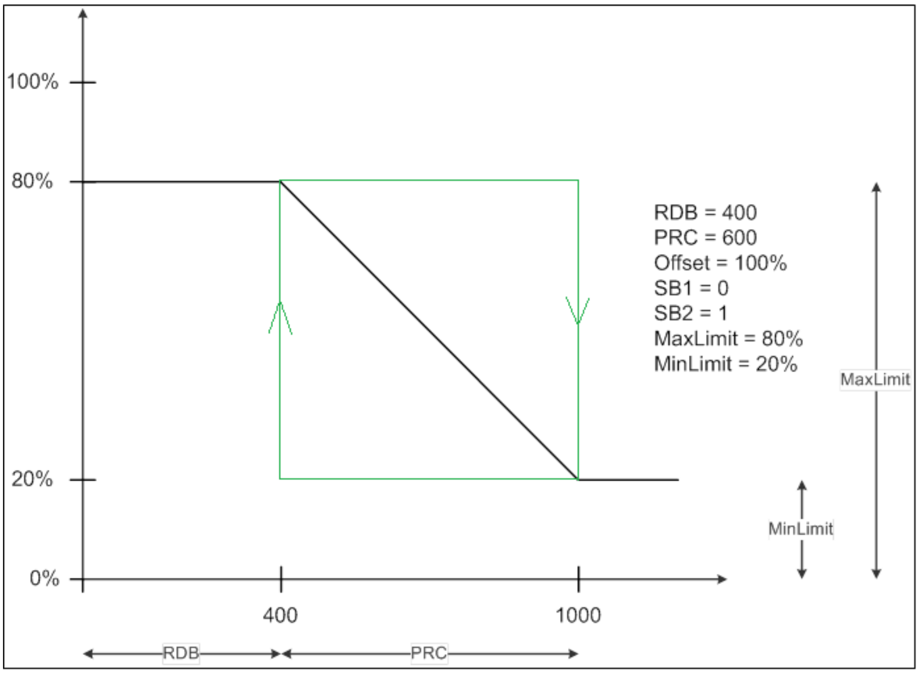
Senseair S88 has several different output possibilities and can be configured with both software and hardware. In terms of software, the output can be either PWM or digital high/low. Through hardware configuration, the PWM signal can be translated to a voltage or a current output. The following parameters can be configured in software:

MaxLimit – Max output duty cycle (100% = 1023= 0x03FF)
MinLimit – Min output duty cycle (0% = 0 = 0x00)
RDB – Regulator dead band
PRC – Proportional regulator constant
SB1 – ShapeBit1, RDB high bit – mirror in y-axis
SB2 – ShapeBit2, PRC high bit – mirror in x-axis
Offset – X-axis offset (100% = 1023= 0x03FF)

For PWM output (OUT0), the duty cycle will increase from MinLimit to HighLimit linearly when the measured concentration exceeds RDB and is below RDB + PRC.
For Alarm output (OUT1), the output pin will go high when measured concentration is larger than RDB + PRC and return to low when measured concentration is lower than RDB. This mode disregards MinLimit and MaxLimit.
Offset and ShapeBits can be used to mirror behavior.
Configuration examples can be seen below:
Green lines shows behavior of the digital output, black lines show PWM duty cycle output.



Example 1


100%
80%
20%
0%
400
1000
RDB
PRC
RDB = 400
PRC = 600
Offset = 100%
SB1 = 0
SB2 = 1
MaxLimit = 80%
MinLimit = 20%
MaxLimit
MinLimit
Example 2

10. Appendix C: Compatibility with Senseair S8 and S88 Modbus definitions.

	Senseair S8	**Senseair S88**
IR1	MeterStatus	MeterStatus
IR2	Alarm Status	Alarm Status
IR3	Output Status	Output Status
IR4	Space CO2	Space CO2
IR5	*Reserved*	Space Temp
IR6	*Reserved*	***Synchro***
IR7	*Reserved*	***VBB***
IR8	*Reserved*	*Reserved*
IR9	*Reserved*	*Reserved*
IR10	*Reserved*	*Reserved*
IR11	*Reserved*	*Reserved*
IR12	*Reserved*	*Reserved*
IR13	*Reserved*	*Reserved*
IR14	*Reserved*	*Reserved*
IR15	*Reserved*	*Reserved*
IR16	*Reserved*	*Reserved*
IR17	*Reserved*	*Reserved*
IR18	*Reserved*	*Reserved*
IR19	*Reserved*	*Reserved*
IR20	*Reserved*	*Reserved*
IR21	*Reserved*	*Reserved*
IR22	Output PWM	Output PWM
IR23	*Reserved*	*Reserved*
IR24	*Reserved*	***ETC High***
IR25	*Reserved*	***ETC Low***
IR26	Type ID High	Type ID High
IR27	Type ID Low	Type ID Low
IR28	Map version	Map version
IR29	FW version	FW version
IR30	Sensor ID High	Sensor ID High
IR31	Sensor ID Low	Sensor ID Low
IR32	*Reserved*	*Reserved*

Table 6: Input Registers compatibility
Bold & italic cells are introduced for Senseair S88 registers

	Senseair S8	**Senseair S88**
HR1	Acknowledgement register	Acknowledgement register
HR2	Command Register	Command Register
HR3	*Reserved*	***Calibration Target***
HR4	*Reserved*	***Pressure***
HR5	*Reserved*	*Reserved*
HR6	*Reserved*	*Reserved*
HR7	*Reserved*	*Reserved*
HR8	*Reserved*	*Reserved*
HR9	*Reserved*	*Reserved*
HR10	*Reserved*	***OUT0_RDB***
HR11	*Reserved*	***OUT0_PRC***
HR12	*Reserved*	***OUT1_RDB***
HR13	*Reserved*	***OUT1_PRC***
HR14	*Reserved*	***OUT1 MinLimit***
HR15	*Reserved*	***OUT1 MaxLimit***
HR16	*Reserved*	***OUT1_Offset***
HR17	*Reserved*	*Reserved*
HR18	*Reserved*	*Reserved*
HR19	*Reserved*	*Reserved*
HR20	*Reserved*	***Communication address***
HR21	*Reserved*	*Reserved*
HR22	*Reserved*	***Override OUT1***
HR23	*Reserved*	*Reserved*
HR24	*Reserved*	*Reserved*
HR25	*Reserved*	*Reserved*
HR26	*Reserved*	*Reserved*
HR27	*Reserved*	***Default pressure***
HR28	*Reserved*	***Static IIR configuration***
HR29	*Reserved*	*Reserved*
HR30	*Reserved*	*Reserved*
HR31	*Reserved*	*Reserved*
HR32	ABC period	ABC period
HR33	*Not implemented*	***ABC Target***

Table 7: Holding Registers compatibility
Bold & italic cells are introduced for Senseair S88 registers

11. Revision history

Date	Revision	Page (s)	Description
2024-02-15	1		Preliminary
2024-02-19	2		Remove empty page 16.
2024-02-22	3	1 9 All 5 15-20	Footer from ©2023 to ©2024 Table 4 Changed Font and header placement Changed footer hight Fixed ETC registers naming. Add WarmUp bit in MeterStatus register Examples added.
2024-03-04	4	22 & 24	Changed footer hight Page 1 Page 9, Table 4: changed Font and header placement
2024-09-04	5	7	Fixed HR1 register description regarding “Force ABC calibration” bit

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

www.senseair.com

Senseair