1.1. Modbus on Sunrise SE-11

Table of contents:

2. Modbus

Modbus is a simple, open protocol for both PLC and sensors[1][2]. Details on Modbus can be found in the website www.modbus.org.

2.1. Modbus settings

Senseair Sunrise supports 9600 baud rate only.

Other Modbus settings are as follows:

Slave address:104 (default)
Baud rate: 9600 bps
Parity: NONE
DataBits: 8
StopBits:1

2.2. Modbus registers on sensor

The Modbus registers are mapped in memory, mapping is interpreted by the sensor at command reception.

Maps of registers (all registers are 16-bit words) are summarised in Table 1 and Table 2.
Associated number is Modbus register number, register address is calculated as (register number - 1)

Table 1 : Input Registers

IR#	Addr.	Name	Description (read only registers)		
IR1	0x00	ErrorStatus0			
			Bit	**Error description**	**Suggested action**
			0	**Fatal error**	Try to restart sensor by power on/off. Contact local distributor.
			1	**I2C communication error** 12C error.	Try to restart sensor by power on/off. Check wires, connectors and 12C protocol implementation. Contact local distributor.
			2	**Internal I2C operation error** Indicates instable communication on sensors local 12C bus.	Try to restart sensor by power on/off. Contact local distributor.
			3	**Calibration error** Indicate that calibration failed (ABC, zero, background or target calibration).	Try to repeat calibration, ensure environment is stable during calibration.
			4	**Self diagnostics error**	Try to restart sensor by power on/off. Contact local distributor.
			5	**Out of range** Indicate that measured concentration is not within sensors measurement range	Perform suitable CO2 calibration (zero, background or target calibration). Contact local distributor.
			6	**Memory error** Error during memory operations	Try to restart sensor by power on/off. Contact local distributor.
			7	**External I2C error** 12C error	Try to restart sensor by power on/off. Check wires, connectors and 12C protocol implementation. (see TDE5531) Contact local distributor.
			8	Reserved	
			9	Reserved	
			10	Reserved	
			11	Reserved	
			12	Reserved	
			13	Reserved	
			14	Reserved	
			15	Reserved	
IR2	0x01		Reserved		
IR3	0x02		Reserved		
IR4	0x03	Filtered CO2	CO2		
IR5	0x04	Unfiltered CO2	Unfiltered CO2 value		
IR6	0x05		Reserved		
IR7	0x06		Reserved		
IRS	0x07		Reserved		

Table 2: Holding Registers

HR#	Addr.	Name	Description (read/write registers)
HR1	0x00	ABC Status	0x0008: ABC calibration; 0x0004: Other types of calibration
HR2	0x01	SCR	Special Command Register SCR=0x00FF: System reset SCR=0x7C05: Target calibration SCR=0x7C06: Background calibration SCR=0x7C07: Zero calibration
HR3	0x02	Cale Target	Calibration target value
HR4	0x03	CO_2 Value Override	Override CO_2 value
HR5	0x04	ABC Time	Time passed in hours for ABC calibration
HR6	0x05	Cale Para0	Calibration parameter0
HR7	0x06	Cale Para1	Calibration Parameter1
HRS	0x07	Cale Para2	Calibration Parameter2
HR9	0x08	Cale Para3	Calibration Parameter3
HR10	0x09	Start Meas	Start to measure when 1 is written to the register in single measurement mode
HR11	0x0A	Meas Mode	Measurement mode (0: continuous mode, 1: single measurement mode) (System reset required after changing measurement mode)
HR12	0x0B	Meas Period[1]	Measurement period in seconds (ranged from 2 to 65535)
HR13	0x0C	Meas Num[2]	Number of measurements: recommended to be less than four times of measurement period (ranqed from 1 to 32767)
HR14	0x0D	ABC period	Time in hours to perform ABC calibration (System reset required after changinq configuration)
HR15	0x0E	Clear ErrorStatus 0	Write any number to this register to clear ErrorStatus 0
HR16	0x0F		Reserved
HR17	0x10		Reserved

Note: In single measurement mode, backup HR5 to HR9 to the host before shutting down the sensor and write HR5 to HR9 back to the sensor after turning it on. This is for proper background calibration. In addition, ABC Time must be supplied to the sensor correctly by the host in single measure mode.

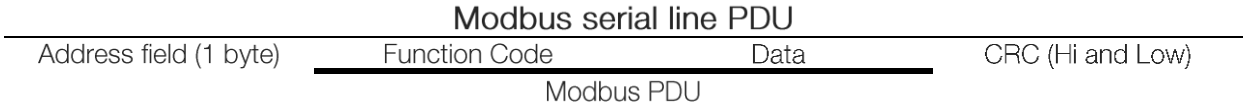
[1] When measurement period is set to be smaller than the time elapsed for the current measurement, current CO2 output will not be updated and the next measurement will start.

[2] When the Number of Measurements (NOM) is set to be smaller than the current NOM, the current CO2 output will be calculated immediately if the new NOM has been reached, and the actual NOM for the average is undefined, which is between the old NOM and the new NOM.

2.3. Serial line frame and addressing

Serial line frame

Modbus over serial line specification [2] distinguishes Modbus Protocol PDU and Modbus serial line PDU in the following way:



Addressing rules

Addressing rules are summarised in the table:

Address	Modbus over serial line V1.0	Senseair Sunrise 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
254	Reserved	"Any sensor"
255	Reserved	Nothing

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].

2.4. Bus timing

Parameter	Min	Typ	Max	Units
Response time-out			180	msec

"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.

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

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

Not implemented.

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

Not implemented.

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

Refer to Modbus specification [1].

Reauest POU

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

Response POU

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

* N = Quantity of Registers

If Address is out of range:

Exception Response POU.

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

If Quantity=0 or Quantity>Number of Registers:

Exception Response POU.

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

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

Refer to Modbus specification [1].

Quantity of Registers is limited to 32.

Reguest POU

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

ReSQOnse POU

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

* **N** = Quantity of Registers

If Address is out of range:

ExceQtion ResQonse POU,

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

If Quantity=0 or Quantity>Number of registers:

ExceQtion ResQonse POU,

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

05 (0x05) Write Single Coil (one bit read / write register)

Not implemented.

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

Not implemented.

15 (0x0F) Write Multiple Coils (one bit read/ write registers)

Not implemented.

16 (0x10) Write Multiple Registers (16 bits read/ write register)

Refer to Modbus specification [1].

Address of Modbus Holding Registers for , -command reading/writing is limited in range 0xOOOO..0xOOI F.

Reguest POU

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

* N = Quantity of Registers

Response POU (is an echo of the Reguest)

Function code	1 byte	**0x10**
Startinq Address Hi	1 bvte	Address Hi
Startinq Address Lo	1 byte	Address Lo
Number of Reqister written Hi	1 byte	Value Hi
Number of Reqister written Lo	1 byte	Value Lo

If Address is out of range:
Exception Response POU.

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

20 (0x14) Read File record

Not implemented.

21 (0x15) Write File record

Not implemented.

22 (0x16) Mask Write Register (16 bits read/ write register)

Not implemented.

23 (0x17) Read/ Write Multiple Registers (16 bits read/ write register)

Not implemented.

3. References

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

4. Examples

Set measurement mode to continuous mode:

Write 0 to holding register 10. Note that after measurement mode has been written to the sensor it has to be restarted before it will change to the new measurement mode.

Request:
0x68 0x10 0x00 0x0A 0x00 0x01 0x02 0x00 0x00 0x64 0xA8

Response:
0x68 0x10 0x00 0x0A 0x00 0x01 0x28 0xF2

Set measurement mode to single measurement mode:

Write 1 to holding register 10. Note that after measurement mode has been written to the sensor it has to be restarted before it will change to the new measurement mode.

Request:
0x68 0x10 0x00 0x0A 0x00 0x01 0x02 0x00 0x01 0xA5 0x68

Response:
0x68 0x10 0x00 0x0A 0x00 0x01 0x28 0xF2

Single measurement sequence

1. Drive EN pin high

2. Wait for 35ms (or longer) for sensor start-up and stabilisation

3. Write stored sensor state data from previous measurement cycle to HR5-HR9
   Request:
   0x68 0x10 0x00 0x04 0x00 0x05 0x0A 0x00 0x00 0x7B 0x2C 0xFE 0x7E 0x0D 0x14 0x00 0x06 0xAE 0x5B

   Response:
   0x68 0x10 0x00 0x04 0x00 0x05 0x48 0xF2

   State data written to the sensor in request above is just an example, it is very important that correct state data is written to the sensor.

4. Write 1 to HR10 to start measurement
   Request:
   0x68 0x10 0x00 0x09 0x00 0x01 0x02 0x00 0x01 0xA5 0x5B

   Response:
   0x68 0x10 0x00 0x09 0x00 0x01 0xD8 0xF2

5. Wait until ready pin goes low or 6s (for default configuration)

6. Read CO2 value, in example CO2 value is 0x0D53 = 3411d ppm
   Request:
   0x68 0x04 0x00 0x03 0x01 0xC8 0xF3
   Response:
   0x68 0x04 0x02 0x0D 0x53 0xA1 0x94

7. Read sensor state data from HR5-HR9 and save it for next measurement
   Request:
   0x68 0x03 0x00 0x04 0x00 0x05 0xCD 0x31

   Response:
   0x68 0x03 0x0A 0x00 0x00 0x7B 0x2C 0xFE 0x7E 0x0D 0x14 0x00 0x06 0xEC 0xF4

8. Drive EN pin low

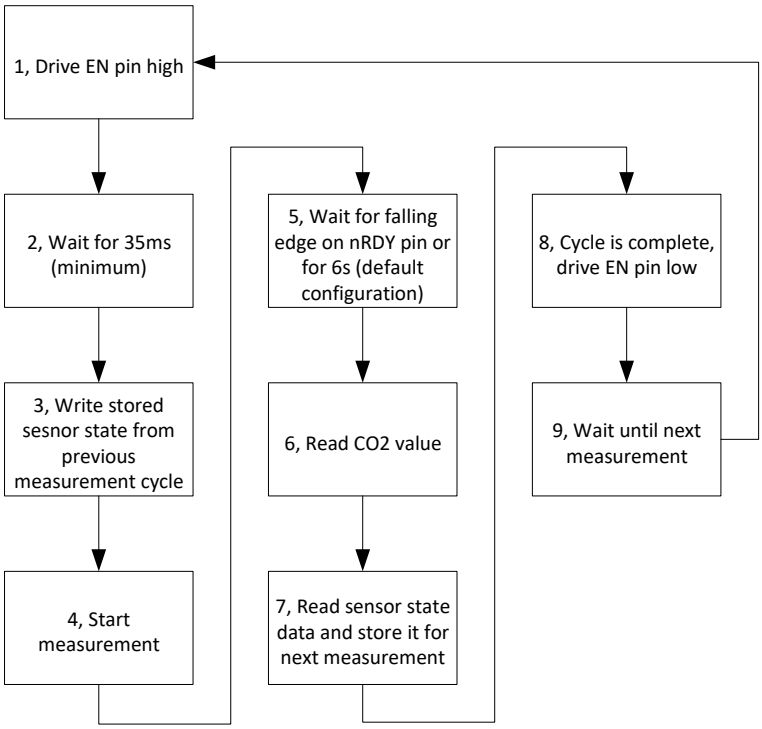
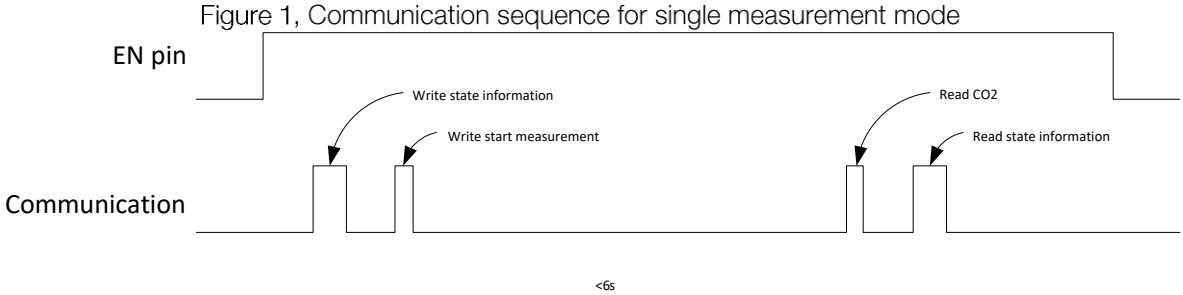


Figure 1, Communication sequence for single measurement mode



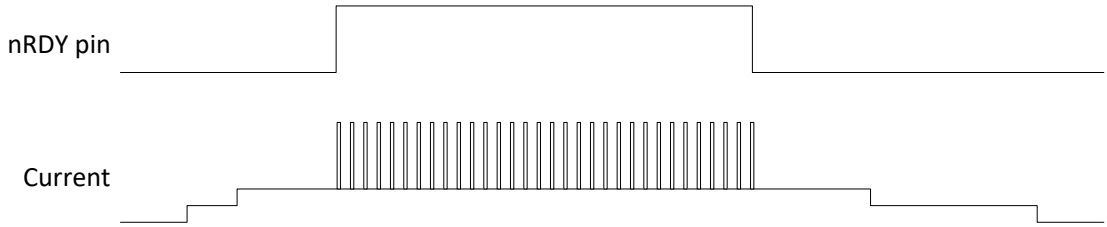


Figure 2, Timing diagram for single measurement mode

GasLab Inc

131 Business Center Drive, Ste A3 Ormond Beach, FL 32174 USA

Phone: (386) 872-7668

Web page: Gaslab.com

See GasLab, Inc. Terms & Conditions: https://gaslab.com/pages/terms-conditions