Modbus on Senseair K40 aSENSE

Table of contents:

1. General 2
2. Byte transmission. 3
3. Serial line frame and addressing. 4
4. Bus timing. 4
5. Supported Modbus commands 5
6. Modbus registers on sensor. 7
7. Appendix A: Application examples 13
8. References 16

1. General

This document is valid for the following Senseair sensor models:

Model	Notes
aSENSE	Based on K40 platform

Modbus is a simple, open protocol for both PLC and sensors. Details on Modbus can be found at 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. The sensor do not support so large packets, maximum packet length (serial line PDU including address byte and 2 bytes CRC) supported 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].

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

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 (2wires communication) drivers. R/T line is available.

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

UART_RxD line is configured as digital input.
Input high level is 4.0V min
Input low level is 0.8V max

UART_TxD line is configured as digital output.
Max output current level is 4mA (DVCC = 5.0V).

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 summarized in the following table:

Address	Modbus over serial line V1.0	K40-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”[2]
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 individual address can be set/changed using UIP5.

4. Bus timing.

Parameter	Min	Typ	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. 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

5.1. 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

5.2. 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**

Senseair

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

5.3.06 (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

6. 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.

6.1. Input registers on K40-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 32	DI 31	DI 30	DI 29	DI 28	DI 27	DI 26	DI 25	DI 24	DI 23	DI 22	DI 21	DI 20	DI 19	DI 18	DI 17
			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 48	DI 47	DI 46	DI 45	DI 44	DI 43	DI 42	DI 41	DI 40	DI 39	DI 38	DI 37	DI 36	DI 35	DI 34	DI 33

			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]
IR4	3	CO_2 value	The unit is ppm, for high concentration sensors (%) the value is divided by 10, e.g. 1000 read from sensor means 10000ppm.
IR5	4	Space Temp	The unit is °C with two decimals, e.g. register value 2368 means 23.68°C
IR6	5	Ch6	Channel 6, depends on product: aSENSE-VAV SetPtTemp aSENSE-DUOAH aSENSE-VAV-RH RH aSENSE-mIII SetPtCO (unit ppm x 10)
IR7	6	Ch7	Channel 7, depends on product: aSENSE-VAV SetPtCO2 aSENSE-DUODewPoint aSENSE-VAV-RH SetPtTemp aSENSE-mIII SetPtCO2 aSENSE-CC-FC T SptLocal
IR8	7	Ch3	Channel 3, depends on product: aSENSE-DUORH aSENSE-CC-FC FlowRaw
IR9	8	Ch9	Channel 9, depends on product: aSENSE-DUOTemp (F) aSENSE-VAV-RH SetPtRH aSENSE-mIII CO (unit ppm x 10) aSENSE-CC-FC Raw Temp
IR10	9	Ch8	Channel 8, depends on product: aSENSE-DUODewPoint (F) aSENSE-VAV-RH SetPtCO2 aSENSE-CC-FC Flow regulation SetPt
IR11	10	Ch11	Channel 11, depends on product: aSENSE-DUOEnthalpy

<table>
<tr><td>IR12</td><td>11</td><td>Ch12</td><td>Channel 12, depends on product:</td></tr>
<tr><td>IR13</td><td>12</td><td>Meas. status</td><td><table><tr><td>Status CH7</td><td>Status CH6</td><td>Status Temp</td><td>Status CO2</td></tr></table>
Bit 0 – Fault warning
Bit 1 – Fault
Bit 2 – Overridden
Bit 3 - OutOfService</td></tr>
<tr><td>IR14</td><td>13</td><td>Meas. status</td><td><table><tr><td>Status CH11</td><td>Status CH8</td><td>Status CH9</td><td>Status CH3</td></tr></table>
Bit 0 – Fault warning
Bit 1 – Fault
Bit 2 – Overridden
Bit 3 - OutOfService</td></tr>
<tr><td>IR15</td><td>14</td><td>Meas. status</td><td><table><tr><td>Read zero</td><td>Read zero</td><td>Read zero</td><td>Status CH12</td></tr></table>
Bit 0 – Fault warning
Bit 1 – Fault
Bit 2 – Overridden
Bit 3 - OutOfService</td></tr>
<tr><td>IR16</td><td>15</td><td></td><td>Reserved</td></tr>
<tr><td>IR17</td><td>16</td><td></td><td>Reserved</td></tr>
<tr><td>IR18</td><td>17</td><td></td><td>Reserved</td></tr>
<tr><td>IR19</td><td>18</td><td></td><td>Reserved</td></tr>
<tr><td>IR20</td><td>19</td><td></td><td>Reserved</td></tr>
<tr><td>IR21</td><td>20</td><td></td><td>Reserved</td></tr>
<tr><td>IR22</td><td>21</td><td>Out1 value</td><td>Value 16383 represents 100% output</td></tr>
<tr><td>IR23</td><td>22</td><td>Out2 value</td><td>Value 16383 represents 100% output</td></tr>
<tr><td>IR24</td><td>23</td><td>Out3 value</td><td>Value 16383 represents 100% output</td></tr>
<tr><td>IR25</td><td>24</td><td>Out4 value</td><td>Value 16383 represents 100% output</td></tr>
<tr><td>IR26</td><td>25</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR27</td><td>26</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR28</td><td>27</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR29</td><td>28</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR30</td><td>29</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR31</td><td>30</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
<tr><td>IR32</td><td>31</td><td></td><td>Reserved, returns "illegal data address" exception</td></tr>
</table>

Notes:

1. Reserved DIs return 0.

6.2. Holding registers on K40 aSENSE

HR#	#	Name		
HR1	0	Acknowledgement register	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	
			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]	
HR2	1	Special Command Register	Command	Parameter
			0x7C	0x06 - CO2 background calibration 0x07 - CO2 zero calibration
HR3	2		Reserved, returns "illegal data address" exception	
HR4	3		Reserved, returns "illegal data address" exception	
HR5	4		Reserved, returns "illegal data address" exception	
HR6	5		Reserved, returns "illegal data address" exception	
HR7	6		Reserved, returns "illegal data address" exception	
HR8	7		Reserved, returns "illegal data address" exception	
HR9	8		Reserved, returns "illegal data address" exception	
HR10	9	Set point adjustment (value to be added/subtracted from setpoint)	SPC Local | bPar0 Depends on product: aSENSE-VAV Temp setpoint correction aSENSE-mIII CO setpoint correction Example VAV: SPC Local = 2 means temp setpoint = 2300 + (2 x 20) = 2340 = 23.40°C (default base setpoint + (SPC Local x SPC Local resolution)) Example mIII: SPC Local = 10 means CO setpoint = 35 + 10 = 45ppm (default base setpoint + (SPC Local x SPC Local resolution))	

HR11	10	Set point adjustment	bPar2 | bPar1
HR12	11	Set point adjustment	bPar4 | bPar3
HR13	12	Set point	CSP CO2 setting | CSP CO2 resolution Depends on product: aSENSE-VAV CO2 setpoint aSENSE-mIII CO2 setpoint Example: CSP CO2 setting = 12, CSP CO2 resolution = 50 (default) means CO2 setpoint is 12 x 50 = 600ppm
HR14	13	Out1 min limit	Range 0 – 16383 (0 – 100%)
HR15	14	Out1 max limit	Range 0 – 16383 (0 – 100%)
HR16	15	Out2 min limit	Range 0 – 16383 (0 – 100%)
HR17	16	Out2 max limit	Range 0 – 16383 (0 – 100%)
HR18	17	Out3 min limit	Range 0 – 16383 (0 – 100%)
HR19	18	Out3 max limit	Range 0 – 16383 (0 – 100%)
HR20	19	Out4 min limit	Range 0 – 16383 (0 – 100%)
HR21	20	Out4 max limit	Range 0 – 16383 (0 – 100%)
HR22	21		Reserved, returns "illegal data address" exception
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		Reserved, returns "illegal data address" exception
HR28	27		Reserved, returns "illegal data address" exception
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 [2]	

Notes:

1. Reserved CIs return 0.
2. 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.

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>

Senseair

7.3. 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).

7.4. 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.

7.5. 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

7.6. 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.

7.7. 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. References

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