Manual Table of Contents

General

The Rheonics SME devices, offer the Modbus TCP service over the Ethernet interface. The Modbus message service provides a Client/Server communication between devices connected on an Ethernet TCP/IP network.

Wiring

The SME can be connected through the Ethernet RJ45 connector as part of a network, with intermediary network devices (router, switches) or directly to an Ethernet host in a direct connection.

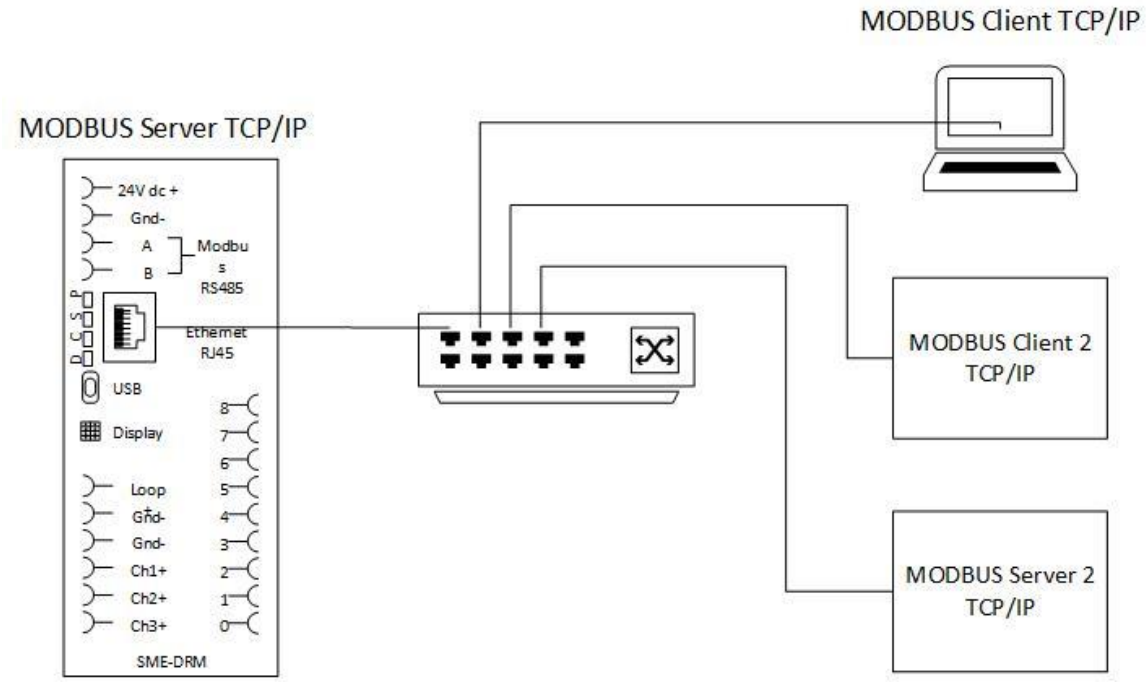

Figure 1 SME as part of an Ethernet network

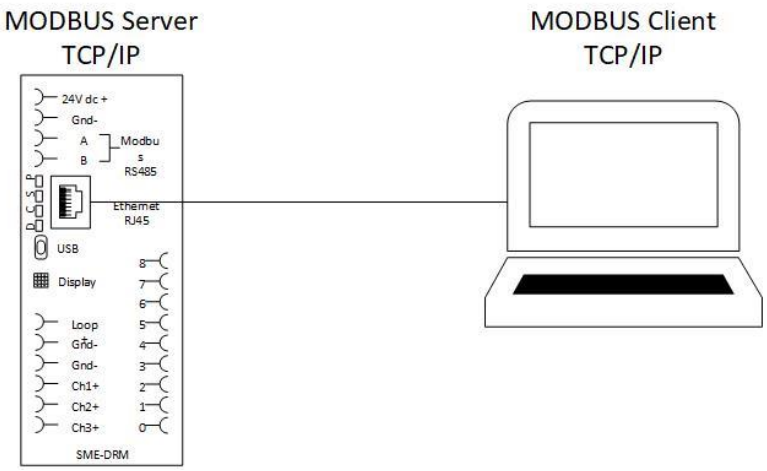

Figure 2 Direct connection between SME and ethernet Host

Configuration

The Rheonics SME device allows the configuration of the Ethernet IP address, mask, gateway and DNS using the RCP software. Please refer to the RCP software manual for instructions on setting these parameters. The factory default values are shown below.

DHCP = True
IP address=0.0.0.0
Mask= 0.0.0.0
Gateway = 0.0.0.0
DNS=0.0.0.0
TCP Port= 502
Slave ID = 255

NOTE: The default TCP Port, and slave ID can't be modified by the user.

Protocol

The structure of the request and response body in Modbus TCP connections, from the function code to the end of the data portion, has exactly the same layout and meaning as in the other MODBUS variants, such as Modbus RTU or Modbus ASCII. The only differences in these other cases are the form of any 'framing' sequence, error check pattern, and address interpretation.

The request and response are prefixed by eight bytes as follows:

byte 0:	transaction identifier – copied by server – usually 0
byte 1:	transaction identifier – copied by server – usually 0
byte 2:	protocol identifier = 0
byte 3:	protocol identifier = 0
byte 4:	length field (upper byte)
byte 5:	length field (lower byte)
byte 6:	unit identifier
byte 7:	MODBUS function code
byte 8 onwards:	Data

Table 1 Modbus TCP framing

The SME uses the Modbus function code x04, which allows to read the analog inputs of the device. All other function codes of the Modbus TCP protocol are unused and will not be mentioned in this document.

Function Code (Dec)	Name	Description
04	Read Input Registers	Read Analog Inputs (AI)

The measured parameters are mapped into the Modbus input registers; the variable associated with each parameter is listed in Table 2.

Parameter	Measurand	Description
Parameter 1	Viscosity median	Takes the median of the last 5 sensor measurements points
Parameter 2	Density median	
Parameter 3	Temperature median	
Parameter 4	Viscosity average	Calculates the average of the last 5 measurement points
Parameter 5	Density average	
Parameter 6	Viscosity raw 'process measurement'	Reads the direct measurement taken by the sensor.
Parameter 7	Density raw	
Parameter 8	Temperature raw	
Parameter 9	Viscosity Last Good	Keeps the last-good median measurement in memory when there is a measurement error
Parameter 10	Density Last Good	
Error	Error State	Error state of the sensor

Table 2 Measured parameters list

Input Registers

For each parameter there are four registers associated; the first two contain the value of the parameter in float data type, and the last two are the value scaled by 100 in Int16 format, and the status of the parameter in Uint16.

The table uses PDU addressing; addresses start at zero. Depending on the PLC used it might be required to add 1 to the addresses in the table below

Address (Dec)	Register (Dec)	Length (Reg.)	Type	Description
Parameter 1				
40	300041	2	Float	Parameter 1 value as float
42	300043	1	Int16	Parameter 1 value scaled by 100 as signed integer
43	300044	1	Uint16	Parameter 1 status (See data-types section)
Parameter 2				
48	300049	2	Float	Parameter 2 value as float
50	300051	1	Int16	Parameter 2 value scaled by 100 as signed integer
51	300052	1	Uint16	Parameter 2 status (See datatypes)
Parameter 3				
56	300057	2	Float	Parameter 3 value as float
58	300059	1	Int16	Parameter 3 value scaled by 100 as signed integer
59	300060	1	Uint16	Parameter 3 status (See datatypes)
Parameter 4				
64	300065	2	Float	Parameter 4 value as float
66	300067	1	Int16	Parameter 4 value scaled by 100 as signed integer
67	300068	1	Uint16	Parameter 4 status (See datatypes)
Parameter 5				
72	300073	2	Float	Parameter 5 value as float
74	300075	1	Int16	Parameter 5 value scaled by 100 as signed integer
75	300076	1	Uint16	Parameter 5 status (See datatypes)
Parameter 6				

80	300081	2	Float	Parameter 6 value as float
82	300083	1	Int16	Parameter 6 value scaled by 100 as signed integer
83	300084	1	Uint16	Parameter 6 status (See datatypes)
Parameter 7				
88	300089	2	Float	Parameter 7 value as float
90	300091	1	Int16	Parameter 7 value scaled by 100 as signed integer
91	300092	1	Uint16	Parameter 7 status (See datatypes)
Parameter 8				
96	300097	2	Float	Parameter 8 value as float
98	300099	1	Int16	Parameter 8 value scaled by 100 as signed integer
99	300100	1	Uint16	Parameter 8 status (See datatypes)
Parameter 9				
136	300136	2	Float	Parameter 9 value as float
138	300138	1	Int16	Parameter 9 value scaled by 100 as signed integer
139	300139	1	Uint16	Parameter 9 status (See datatypes)
Parameter 10				
144	300144	2	Float	Parameter 10 value as float
146	300146	1	Int16	Parameter 10 value scaled by 100 as signed integer
147	300147	1	Uint16	Parameter 10 status (See datatypes)
Error				
514	300514	1	Uint16	Error value

Table 3 Modbus input registers

Data-Types

Float: IEEE754 floating point. This data type spans two registers which should be read together. Depending on the endianness of the host it might be necessary to swap high/low byte and/or the respective registers.

Int16: Signed 16 bit integer (register)

Uint16: Unsigned 16 bit integer

Parameter status: The parameter status is a bitmask of individual status bits. As a general rule Bit 0 checking should be implemented where a set bit corresponds to an error/warning condition. In case of a critical error the parameter value is also set to NAN (not a number – See IEE754)

Bit 0: General error – Check other bits for more detailed configuration

Bit 1: Internal configuration error – Restart. If error persists call Rheonics for support

Bit 2: Hardware error, e.g. broken temperature sensor. Call Rheonics for support and arrange RMA

Bit 3: Dependency error: a parameter is calculated from other parameters and one of the source parameter has an error. Check all other parameters for errors.

Bit 4: Device is starting up and no results are ready.

Bit 5: Internal error – Restart. If error persists call Rheonics for support.

Bit 8: Parameter result is not stable. Check measurement conditions (flow, vibration, etc.)

Bit 15: Parameter exceeds sensor upper limit. No results available.

Bit 14: Parameter below sensor lower limit. No results available.

Bit 13: Parameter value above upper warning range. Measurements might not be accurate and sensor performance might be reduced.

Bit 12: Parameter value below lower warning range. Measurements might not be accurate and sensor performance might be reduced.

Error Status

The error register shows information about the sensor state and measurements. A 0 value means no error.

A combination of values from 1-13 means the sensor is either not attached to the SME or not locking to the right frequency to give an accurate measurement.

And error of 20 indicates that there is no power supply attached to the sensor (24 Volts).

Modbus TCP Command and Response Examples

Reading the parameter 3; Temperature median, in float32 data type.

Address =56 = 0x0038 (word) (Address in Hex of the first register)
Functional code= 04=0x0004 (word)-Read input registers
Number of registers = 02=0x0002 (word)
Unit ID = 255 = 0xFF

Command Structure:

Trans. ID (Hi Byte)	Trans. ID (Lo Byte)	Protocol ID (Hi Byte)	Protocol ID (Lo Byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Address 1st reg. (Hi byte)	Address 1st reg. (Lo Byte)	# of registers (Hi Byte)	# of registers (Lo Byte)
00	**00**	**00**	**00**	**00**	**06**	**FF**	**04**	**00**	**38**	**00**	**02**

Command Response

Functional Code	Number of Bytes	Value of the first register (Hi Byte)	Value of the first register (Lo Byte)	Value of the second register (Hi Byte)	Value of the second register (Lo Byte)
04	**04**	**41**	**B1**	**5C**	**29**

040441B15C29

Data = 0x41B15C29 = 22.17 (float32)

Reading the Parameter 1; Viscosity median, status.

Address =43 = 0x002B (word) (Address in Hex of the first register)
Functional code= 04=0x0004 (word)-Read input registers

Number of registers = 01=0x0001 (word)
Unit ID = 255 = 0xFF

Command Structure:

Trans. ID (Hi Byte)	Trans. ID (Lo Byte)	Protocol ID (Hi Byte)	Protocol ID (Lo Byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Address 1st reg. (Hi byte)	Address 1st reg. (Lo Byte)	# of registers (Hi Byte)	# of registers (Lo Byte)
00	**00**	**00**	**00**	**00**	**06**	**FF**	**04**	**00**	**2B**	**00**	**01**

Command Response

Functional Code	Number of Bytes	Value of the first register (Hi Byte)	Value of the first register (Lo Byte)
04	**02**	**00**	**01**

04020001

Data = 0x0001 = 1 (Uint16)

Referring to the parameter status, this value means a general error occurred.

Reading the Parameter 8; Temperature raw, Int16 value.

Address =98 = 0x0062 (word) (Address in Hex of the first register)
Functional code= 04=0x0004 (word)-Read input registers
Number of registers = 01=0x0001 (word)
Unit ID = 255 = 0xFF

Command Structure:

Trans. ID (Hi Byte)	Trans. ID (Lo Byte)	Protocol ID (Hi Byte)	Protocol ID (Lo Byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Address 1st reg. (Hi byte)	Address 1st reg. (Lo Byte)	# of registers (Hi Byte)	# of registers (Lo Byte)
00	**00**	**00**	**00**	**00**	**06**	**FF**	**04**	**00**	**62**	**00**	**01**

Command Response

Functional Code	Number of Bytes	Value of the first register (Hi Byte)	Value of the first register (Lo Byte)

04	02	08	9E

0402089E

Data = 0x089E = 2206 (Int16)

Since the register value is the temperature scaled by 100. The actual temperature value is 22.06 °C.

This register is useful for devices in which single point conversion requires too much computation.

Reading Multiple parameters; Viscosity Average value in float, Uint and status.

Address =64 = 0x0040 (word) (Address in Hex of the first register)
Functional code= 04=0x0004 (word)-Read input registers
Number of registers = 04=0x0004 (word)
Unit ID = 255 = 0xFF

Command Structure:

Trans. ID (Hi Byte)	Trans. ID (Lo Byte)	Protocol ID (Hi Byte)	Protocol ID (Lo Byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Address 1st reg. (Hi byte)	Address 1st reg. (Lo Byte)	# of registers (Hi Byte)	# of registers (Lo Byte)
00	**00**	**00**	**00**	**00**	**06**	**FF**	**04**	**00**	**40**	**00**	**04**

Command Response

Functional Code	Number of Bytes	1th reg. (Hi Byte)	1th reg. (Lo Byte)	2th reg. (Hi Byte)	2th reg. (Lo Byte)	3th reg. (Hi Byte)	3th reg. (Lo Byte)	4th reg. (Hi Byte)	4th reg. (Lo Byte)
04	**08**	**41**	**F4**	**51**	**EC**	**0B**	**EE**	**00**	**00**

0408**41F451EC0BEE0000**

Viscosity Average (float32) = 30.54 cP.

Viscosity Average (UInt16) = 3054 cP

Viscosity Average Status (UInt16) = 0

Changelog

Version	Description	Date	Name
1.0	Version Creation	27.03.2017	C. Walter
2.0	Update for map, function code, and examples	22.01.2019	M. Hernandez
3.0	Add last good and error registers	13.02.2020	M. Hernandez