rheonics

inline process
density and viscosity
monitoring

MODBUS TCP Field Device Specification:
Rheonics, SME

Covers sensor Types: ***SR, SRV, SRD, DVP, DVM***

GLOBAL SUPPORT

For Support, visit: https://support.rheonics.com

EUROPE HEADQUARTERS
Rheonics GmbH
Klosterstrasse. 19
8406 Winterthur Switzerland
Tel: +41 52 511 32 00

US HEADQUARTERS
Rheonics, Inc.
3 Sugar Creek Center Blvd, Ste 100
Sugar Land, TX 77478
United States of America
Tel: +1 713 364 5427

Website: https://rheonics.com
Support Portal: https://support.rheonics.com
Sales E-Mail: info@rheonics.com
Support E-Mail: support@rheonics.com

Contents

1 Before you begin

1.1 About the manual

This manual provides information on Modbus TCP support on Rheonics devices. This document specifies all the device-specific features and Modbus TCP Protocol implementation details.

Important Instructions

This manual assumes that the following conditions apply:

- The sensor has been installed correctly and completely according to the installation.
- The installation complies with all applicable safety requirements.
- The user is trained in relevant safety standards.

1.2 Purpose

This specification is designed to complement the SME Installation Manual by providing a complete, clear description of this Field Device from a Modbus TCP Communication perspective.

1.3 Who should use this document?

The specification is designed to be a technical reference for Modbus TCP End Users. This document assumes the reader is familiar with Modbus Protocol requirements and terminology.

1.4 Warning

Before connecting the Modbus Communicator in an explosive atmosphere, make sure instruments are ordered and installed in accordance with intrinsically safe or EX classification-specific field wiring practices. Explosions can cause serious injury or death. Review Rheonics Ex Manual at https://rheonics.com/resources/manuals/.

1.5 Nomenclature

Abbreviation (short form)	Full-term	Meaning
SRV	Symmetric Resonator Viscometer	Viscosity sensor
SRD	Symmetric Resonator Densitometer	Density and Viscosity sensor
DVP	Density Viscosity Probe	HPHT inline probe
DVM	Density Viscosity Module	HPHT inline module
RCP	Rheonics Control Panel	Software for data acquisition and configuration
SME	Smart Module Electronics	Sensor electronics

Table 1. Defined Acronyms

1.6 Related Documentation

You can find all product documentation on the USB stick shipped with the SME and on our website at https://rheonics.com/resources

2 Product overview

The Rheonics SME provides clients Modbus TCP interface to get digital data on all devices ordered. This document provides guidance for field connection by an end user.

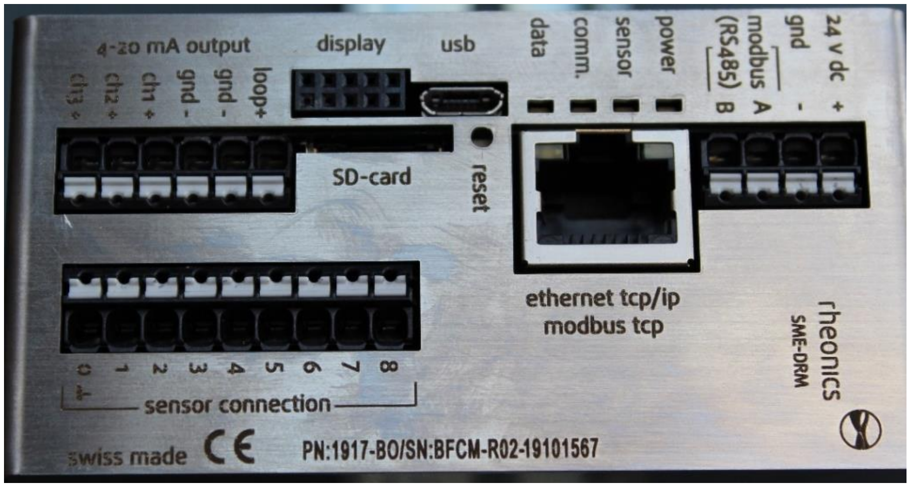


Figure 1. SME sensor electronics unit.

2.1 Process Interface

The SME is compatible with various Rheonics instruments. This includes Type: SR (SRV & SRD), Type: DV (DVP, DVM) and other instruments using the SME electronics from Rheonics.

Figure 2: Rheonics Sensor for Viscosity and density measurements.

2.2 Reference to other instruments.

Manuals and guides for digital instruments are modular. General instructions give information about the functioning and installation of instruments. Operational instructions explain the use of the digital instrument features and parameters. Fieldbus specific information explains the installation and use of the instrument on that Fieldbus network. Related manuals are listed next and can be found in https://rheonics.com/resources:

- SRV USER MANUAL
- SRD USER MANUAL
- DVP USER MANUAL
- DVM USER MANUAL
- SENSOR INSTALLATION MANUAL
- RHEONICS CONTROL PANEL USER MANUAL

3 Modbus TCP Installation

3.1 Instrument overview:

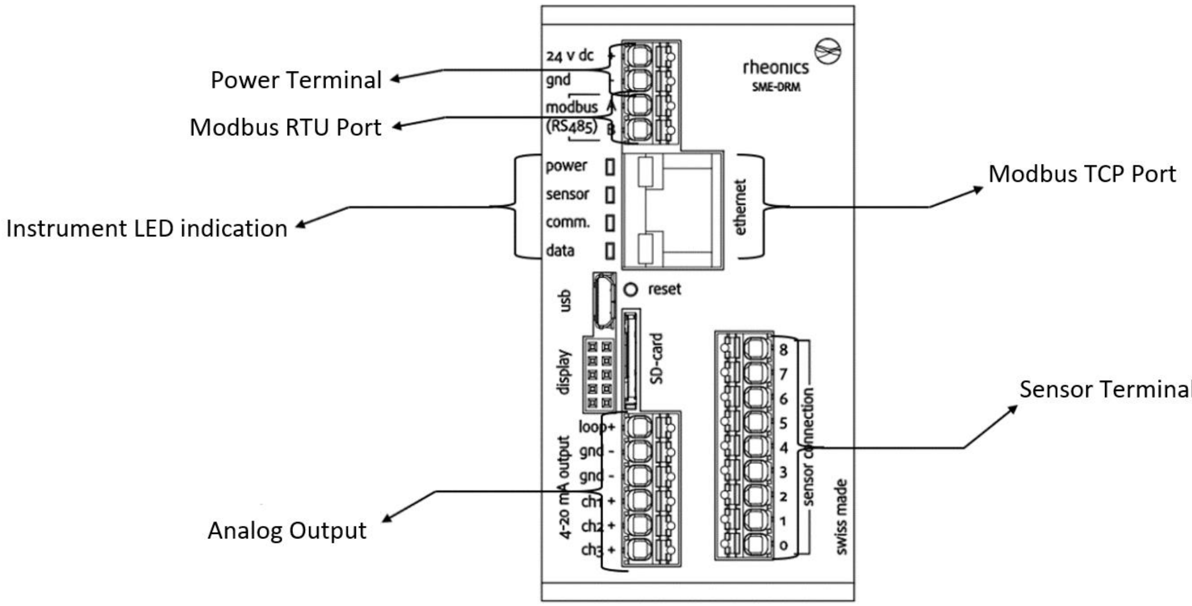


Figure 3. SME-DRM Diagram.

3.2 Ethernet PIN assignment Modbus TCP

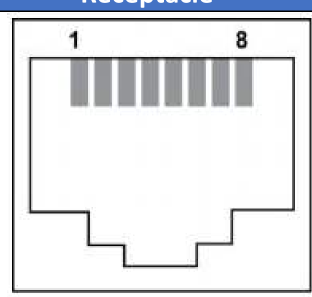
RJ45 Connector	Receptacle	Pin Number	Wire Color	Description
		1	Yellow	Transmit+
		2	Orange	Transmit-
		3	White	Receive+
		4		Not Used
		5		Not Used
		6	Blue	Receive-
		7		Not Used
		8		Not Used

Table 2: RJ45 Connector Pinout.

4 Getting started

4.1 Components Used

- Rheonics SRV, SRD, DVP or DVM w/ Firmware V03.30/0 or higher
- Modbus Communication software (e.g. Modbus Poll)
- Software sensor: Rheonics Control Panel (RCP)
- Windows 10 - 64 bit

4.2 System Connections

Connect the sensor to a PC (with RCP and Modbus communication software installed) with an Ethernet Cable connected in the sensor electronics RJ-45 port. This can be done through an Ethernet switch or hub, or a direct connection.

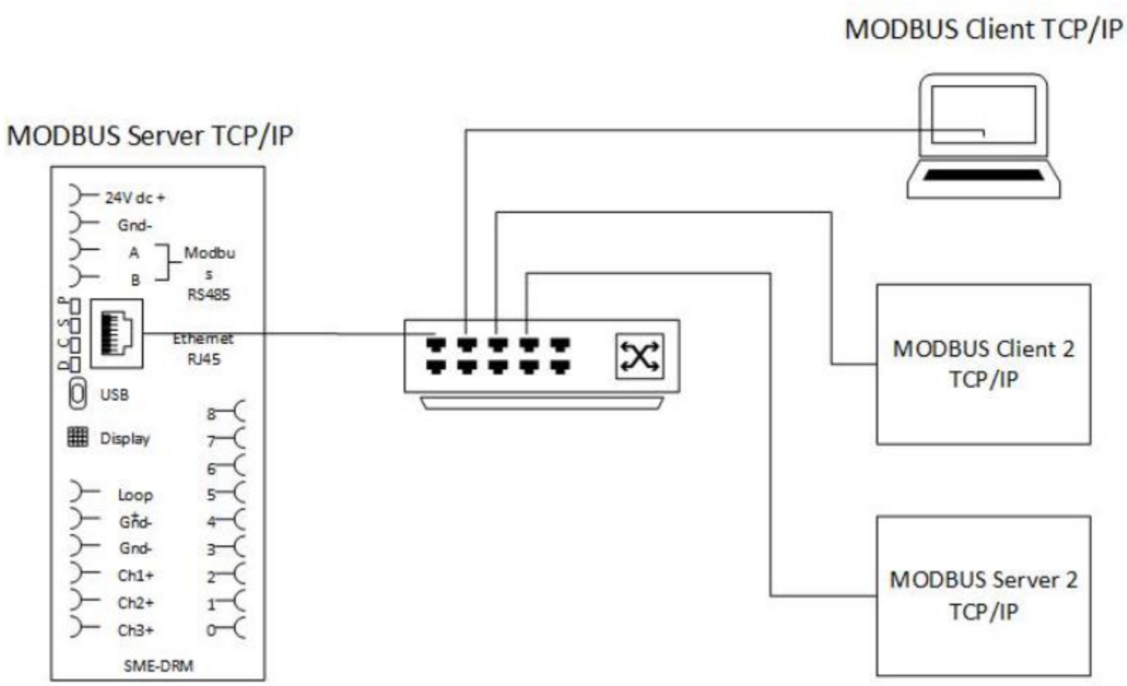


Figure 4. Communication diagram for Rheonics sensor-PLC and PC.

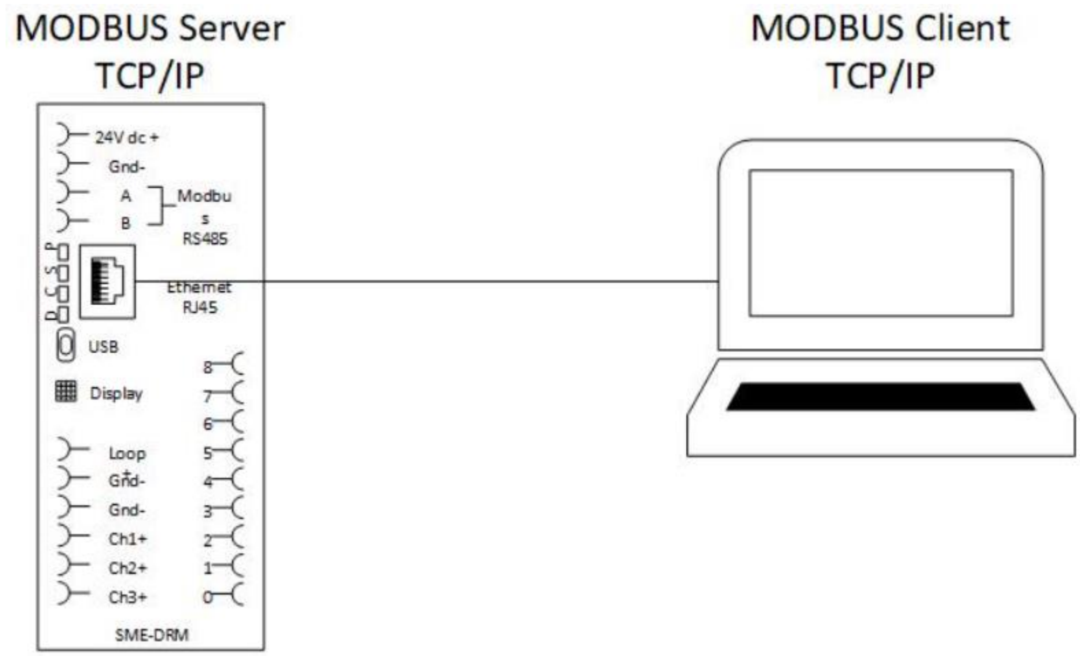


Figure 5: Direct connection between SME and PC

4.3 Configure Rheonics SME

In this section, the SME ethernet configuration is set. We will use the RCP software on the PC, connect to the SME using USB and configure the SME to disable DHCP. Make sure it has a valid IP address. Figure 7 shows a standard configuration that can be used for the correct performance of the system. See the next steps as guidance.

Open the RCP software. On Settings Communication Tab, select USB, identify the COM Port of the desired SME to connect and click Apply. Wait for the Status Led to go green.

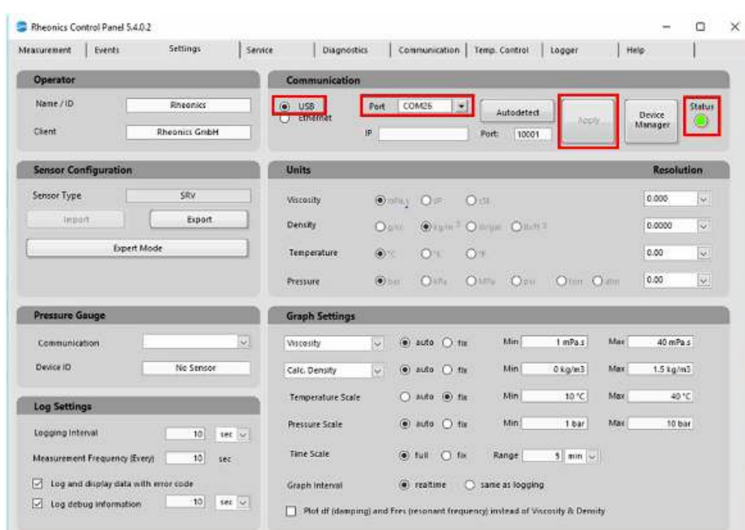


Figure 6: USB connection with SME.

For Static IP address, in the Ethernet section click the dropdown menu in DHCP and select false (this will disable the DHCP function) – input the static IP address, subnet and gateway to be used (Figure 7 is an example). Check the RCP manual for detailed instructions.

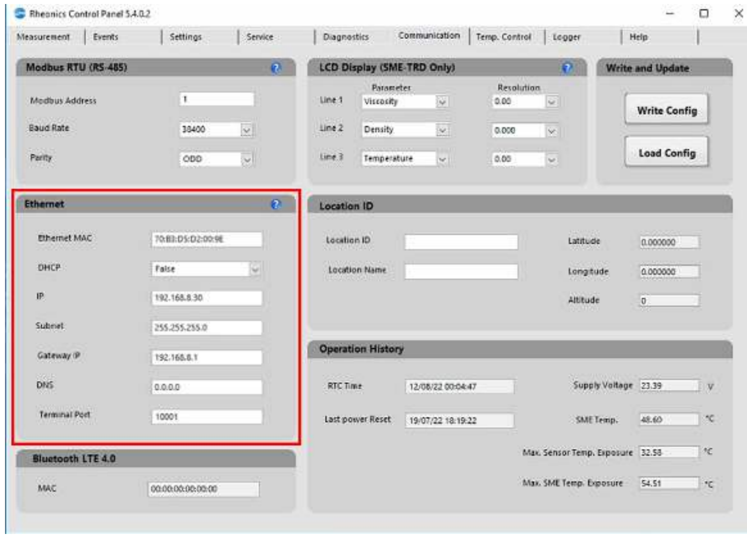


Figure 7: Configuration in RCP to use Ethernet with DHCP.

Click “Write Config” to upload the configuration to the SME. Wait until the button turns green.

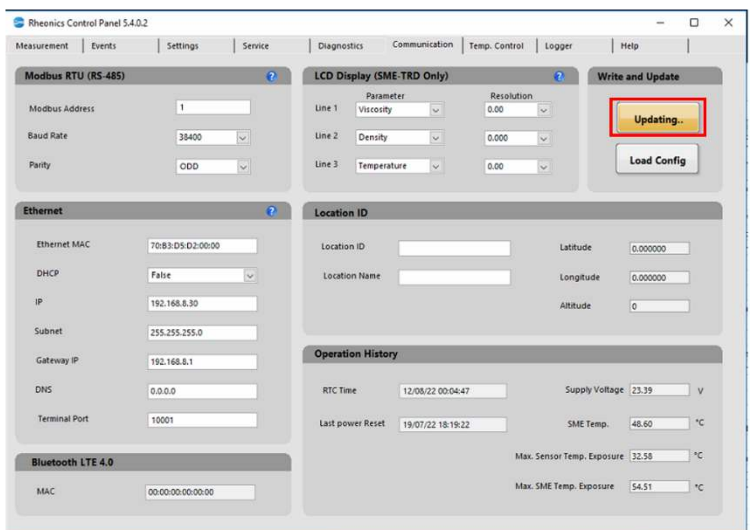


Figure 8. Upload configuration to the SME.

5 Modbus 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 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:	Address 1st Register (High Byte)
byte 9:	Address 1st Register (Low Byte)
byte 10:	Number of Registers (High Byte)
byte 11:	Number of Registers (Low Byte)

Table 3: Modbus TCP framing.

Unit Identifier (byte 6) for Rheonics SME is 255 (0xFF) and must be configured as such. For byte 7, the SME uses the Modbus function code x04, which allows to read the input registers of the device where all the parameters are mapped. All other function codes of the Modbus TCP protocol are unused and will not be mentioned in this document. Byte 8 onwards defines the Data with its first Register address and the number of registers to be read.

5.1 Measured Parameters List

The measured parameters are mapped into the Modbus input registers. There is a total of 23 Parameters available from the SME, as shown in Table 4. Their associated variable, the Sensor and Error status are also listed.

Parameter	Measurement	Description
Parameter 0	Viscosity median	Takes the median of the last 5 sensor measurements points
Parameter 1	Density median	
Parameter 2	Temperature median	
Parameter 3	Kinematic Viscosity	Measured value divided by density squared
Parameter 4	Density average	Calculates the average of the last 5 measurement points
Parameter 5	Viscosity raw 'process measurement'	Reads the direct measurement taken by the sensor.
Parameter 6	Density raw	
Parameter 7	Temperature raw	
Parameter 8	Resonant Frequency	Measured in Hertz
Parameter 9	Compensated Resonant Frequency	
Parameter 10	Damping	Damping from Measurement

Parameter 11	Coil Temperature	
Parameter 12	Viscosity Last Good	Keeps the last-good median measurement in memory when there is a measurement error
Parameter 13	Density Last Good	
Parameter 14	Displays of mapped value from Modbus register 512	
Parameter 15	Displays of mapped value from Modbus register 514	
Parameter 16	Displays of mapped value from Modbus register 516	
Parameter 17	Estimated Temperature	Calculated from internal temperature Algorithm
Parameter 18	Temperature from PT1000 sensor	Read from internal sensor
Parameter 19	Calculated parameter from viscosity model	Set from Advanced Calculations
Parameter 20	Calculated parameter from density model	
Parameter 21	Calculated parameter from concentration model	
Parameter 22	Sensor Cleanliness Ratio	Used for estimating sensor cleanliness
Sensor Status	Sensor Error Status	Status of the sensor
Error	Error State	Error state of the sensor

Table 4: Measured Parameters List.

5.2 Input Registers

For each parameter there are seven registers associated; the first two contain the value of the parameter in float data type, the third is the value scaled by 100 in Int16 format, fourth is the status of the parameter in UInt16. The fifth and sixth contain the uncalibrated value and the last one indicates the measurement units.
It's important to notice that Table 5 uses PDU addressing; addresses start at zero. Depending on the PLC used it might be required to add 1 to the addresses, to get the correct registers.

Address (Dec)	Register (Dec)	Length (Reg.)	Type	Description
Parameter 0	Viscosity median			
40	300041	2	Float32	Parameter 0 value as float
42	300043	1	SInt16	Parameter 0 value scaled by 100
43	300044	1	Uint16	Parameter 0 status (See section 8.2)
44	300045	2	Float32	Parameter 0 Uncalibrated value
46	300047	1	UInt16	Parameter 0 Unit (See section 9)
Parameter 1	Density median			
48	300049	2	Float32	Parameter 1 value as float
50	300051	1	SInt16	Parameter 1 value scaled by 100
51	300052	1	Uint16	Parameter 1 status

52	300053	2	Float32	Parameter 1 Uncalibrated value
54	300055	1	UInt16	Parameter 1 Unit
Parameter 2	Temperature median			
56	300057	2	Float32	Parameter 2 value as float
58	300059	1	SInt16	Parameter 2 value scaled by 100
59	300060	1	Uint16	Parameter 2 status
60	300061	2	Float32	Parameter 2 Uncalibrated value
62	300063	1	UInt16	Parameter 2 Unit
Parameter 3	Kinematic Viscosity			
64	300065	2	Float32	Parameter 3 value as float
66	300067	1	SInt16	Parameter 3 value scaled by 100
67	300068	1	Uint16	Parameter 3 status
68	300069	2	Float32	Parameter 3 Uncalibrated value
70	300071	1	UInt16	Parameter 3 Unit
Parameter 4	Density Average			
72	300073	2	Float32	Parameter 4 value as float
74	300075	1	SInt16	Parameter 4 value scaled by 100
75	300076	1	Uint16	Parameter 4 status
76	300077	2	Float32	Parameter 4 Uncalibrated value
78	300079	1	UInt16	Parameter 4 Unit
Parameter 5	Viscosity raw ‘process measurement’			
80	300081	2	Float32	Parameter 5 value as float
82	300083	1	SInt16	Parameter 5 value scaled by 100
83	300084	1	Uint16	Parameter 5 status
84	300085	2	Float32	Parameter 5 Uncalibrated value
86	300087	1	UInt16	Parameter 5 Unit
Parameter 6	Density raw			
88	300089	2	Float32	Parameter 6 value as float
90	300091	1	SInt16	Parameter 6 value scaled by 100
91	300092	1	Uint16	Parameter 6 status
92	300093	2	Float32	Parameter 6 Uncalibrated value
94	300095	1	UInt16	Parameter 6 Unit
Parameter 7	Temperature raw			
96	300097	2	Float32	Parameter 7 value as float
98	300099	1	SInt16	Parameter 7 value scaled by 100
99	300100	1	Uint16	Parameter 7 status
100	300101	2	Float32	Parameter 7 Uncalibrated value
102	300103	1	UInt16	Parameter 7 Unit
Parameter 8	Resonant Frequency			
104	300105	2	Float32	Parameter 8 value as float
106	300107	1	SInt16	Parameter 8 value scaled by 100
107	300108	1	Uint16	Parameter 8 status
108	300109	2	Float32	Parameter 8 Uncalibrated value
110	300111	1	UInt16	Parameter 8 Unit

Parameter 9	Compensated Resonant Frequency			
112	300113	2	Float32	Parameter 9 value as float
114	300115	1	SInt16	Parameter 9 value scaled by 100
115	300116	1	Uint16	Parameter 9 status
116	300117	2	Float32	Parameter 9 Uncalibrated value
118	300119	1	UInt16	Parameter 9 Unit
Parameter 10	Damping Frequency			
120	300121	2	Float32	Parameter 10 value as float
122	300123	1	SInt16	Parameter 10 value scaled by 100
123	300124	1	Uint16	Parameter 10 status
124	300125	2	Float32	Parameter 10 Uncalibrated value
126	300127	1	UInt16	Parameter 10 Unit
Parameter 11	Coil Temperature			
128	300129	2	Float32	Parameter 11 value as float
130	300131	1	SInt16	Parameter 11 value scaled by 100
131	300132	1	Uint16	Parameter 11 status
132	300133	2	Float32	Parameter 11 Uncalibrated value
134	300135	1	UInt16	Parameter 11 Unit
Parameter 12	Viscosity Last Good			
136	300137	2	Float32	Parameter 12 value as float
138	300139	1	SInt16	Parameter 12 value scaled by 100
139	300140	1	Uint16	Parameter 12 status
140	300041	2	Float32	Parameter 12 Uncalibrated value
142	300043	1	SInt16	Parameter 12 Unit
Parameter 13	Density Last Good			
144	300145	2	Float32	Parameter 13 value as float
146	300147	1	SInt16	Parameter 13 value scaled by 100
147	300148	1	Uint16	Parameter 13 status
148	300149	2	Float32	Parameter 13 Uncalibrated value
150	300151	1	SInt16	Parameter 13 Unit
Parameter 14	Displays of mapped value from Modbus register 512			
152	300153	2	Float32	Parameter 14 value as float
154	300155	1	SInt16	Parameter 14 value scaled by 100
155	300156	1	Uint16	Parameter 14 status
156	300157	2	Float32	Parameter 14 Uncalibrated value
158	300159	1	SInt16	Parameter 14 Unit
Parameter 15	Displays of mapped value from Modbus register 514			
160	300161	2	Float32	Parameter 15 value as float
162	300163	1	SInt16	Parameter 15 value scaled by 100
163	300164	1	UInt16	Parameter 15 status
164	300165	2	Float32	Parameter 15 Uncalibrated value
166	300167	1	SInt16	Parameter 15 Unit
Parameter 16	Displays of mapped value from Modbus register 516			
168	300169	2	Float32	Parameter 16 value as float

170	300171	1	SInt16	Parameter 16 value scaled by 100
171	300172	1	UInt16	Parameter 16 status
172	300173	2	Float32	Parameter 16 Uncalibrated value
174	300175	1	SInt16	Parameter 16 Unit
Parameter 17	Estimated Temperature			
176	300177	2	Float32	Parameter 17 value as float
178	300179	1	SInt16	Parameter 17 value scaled by 100
179	300180	1	UInt16	Parameter 17 status
180	300181	2	Float32	Parameter 17 Uncalibrated value
182	300183	1	SInt16	Parameter 17 Unit
Parameter 18	Temperature from PT1000 sensor			
184	300185	2	Float32	Parameter 18 value as float
186	300187	1	SInt16	Parameter 18 value scaled by 100
187	300188	1	UInt16	Parameter 18 status
188	300189	2	Float32	Parameter 18 Uncalibrated value
190	300191	1	SInt16	Parameter 18 Unit
Parameter 19	Calculated parameter from viscosity models			
192	300193	2	Float32	Parameter 19 value as float
194	300195	1	SInt16	Parameter 19 value scaled by 100
195	300196	1	UInt16	Parameter 19 status
196	300197	2	Float32	Parameter 19 Uncalibrated value
198	300199	1	SInt16	Parameter 19 Unit
Parameter 20	Calculated parameter from density models			
200	300201	2	Float32	Parameter 20 value as float
202	300203	1	SInt16	Parameter 20 value scaled by 100
203	300204	1	UInt16	Parameter 20 status
204	300205	2	Float32	Parameter 20 Uncalibrated value
206	300207	1	SInt16	Parameter 20 Unit
Parameter 21	Calculated parameter from concentration models			
208	300209	2	Float32	Parameter 21 value as float
210	300211	1	SInt16	Parameter 21 value scaled by 100
211	300212	1	UInt16	Parameter 21 status
212	300213	2	Float32	Parameter 21 Uncalibrated value
214	300215	1	SInt16	Parameter 21 Unit
Parameter 22	Sensor Cleanliness Ratio			
216	300217	2	Float32	Parameter 22 value as float
218	300219	1	SInt16	Parameter 22 value scaled by 100
219	300220	1	UInt16	Parameter 22 status
220	300221	2	Float32	Parameter 22 Uncalibrated value
222	300223	1	SInt16	Parameter 22 Unit
Sensor Status	Sensor Error Status			
39	300040	1		
Error	Error State			
514	300515	1	Uint16	Error value

Table 5: Input Registers.

5.3 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. To transform the value to decimal form, follow the IEEE 754 Standard for Floating Point Binary Arithmetic.
Int16: Signed 16 bit integer (register).
Uint16: Unsigned 16 bit integer.

6 Linking tags with Modbus Poll

The Rheonics Modbus TCP instruments offer many different modules and parameters. To select these modules/parameters, the client's configuration tooling software is used, in this case, Modbus Poll.

6.1 Open the software and follow the steps:

Create a new Connection.

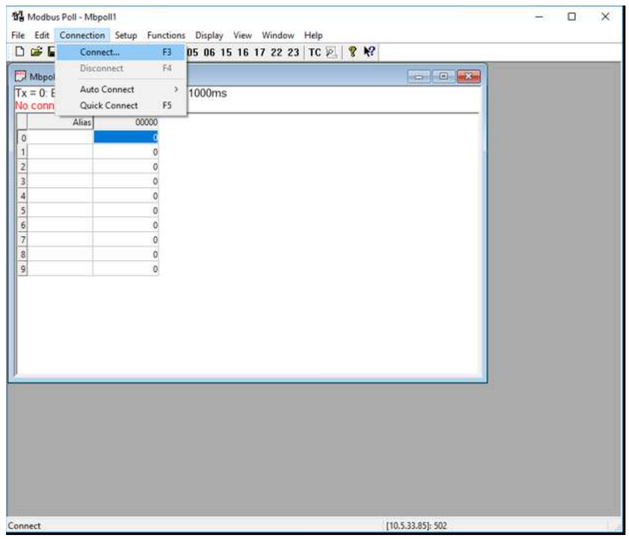


Figure 9: Modbus Poll new connection.

With the new window, set a Modbus TCP/IP connection type and include the SME IP address that was set before.

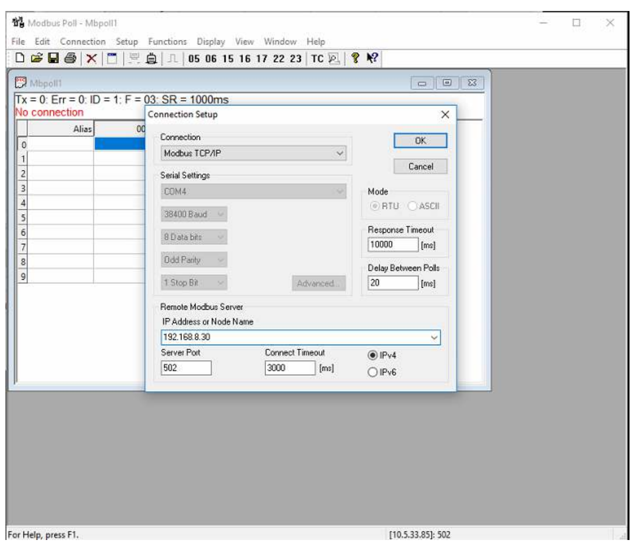


Figure 10: Connection Setup.

Fill in every other parameter as shown and click Ok.

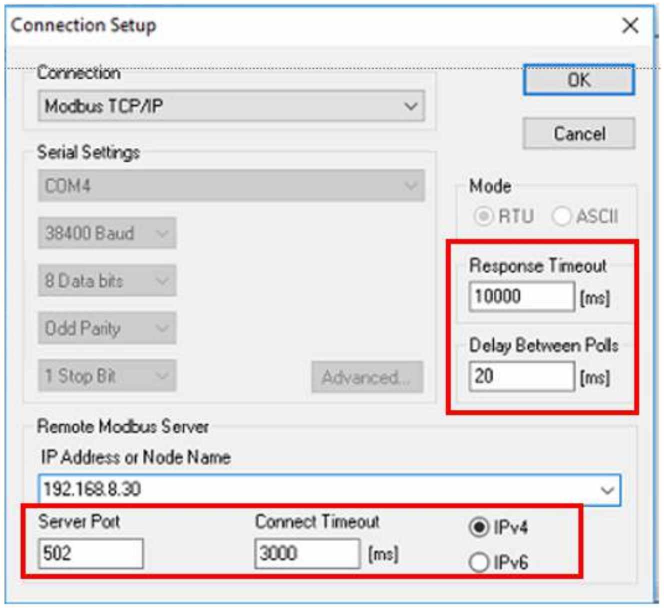


Figure 11: Connection Setup details.

Go to "Set up" tab and select "Read/Write Definition".

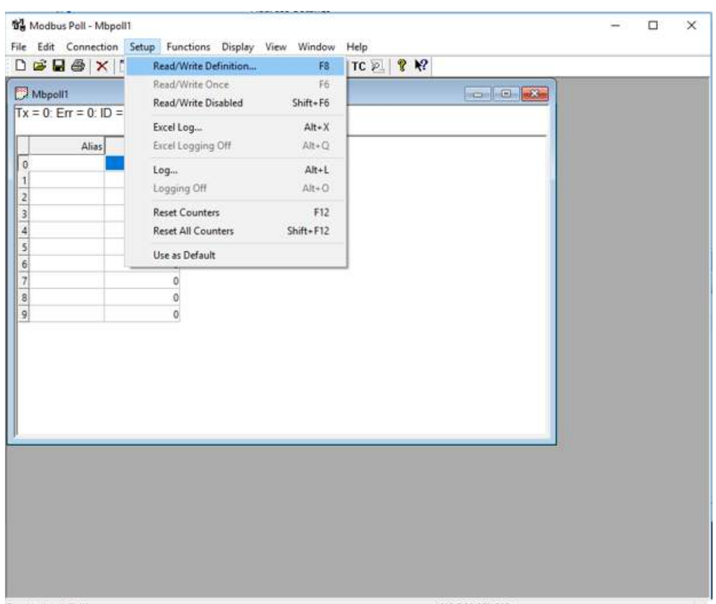


Figure 12: Reading Definition.

Once the new window is opened, set all the parameters as shown below (only quantity can be varied).

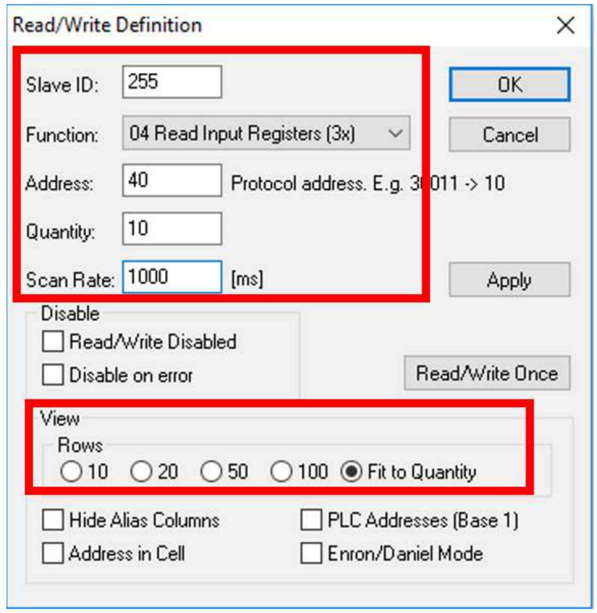


Figure 13: Configure Read/Write Definition.

Read Parameters, at first, may look like Figure 14, which shows the first 10 registers starting from address 40.

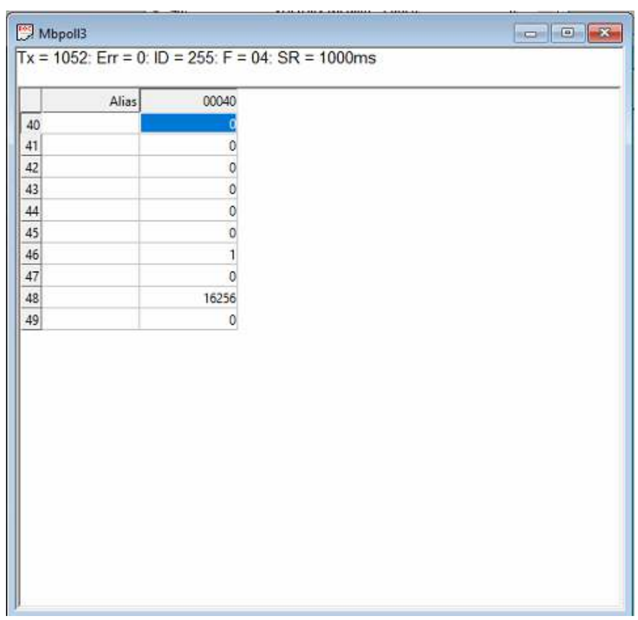


Figure 14: Example of Registers read.

By right-clicking any data, it can be seen that by default the data type is "signed". This needs to be changed for every data according to Table 5.

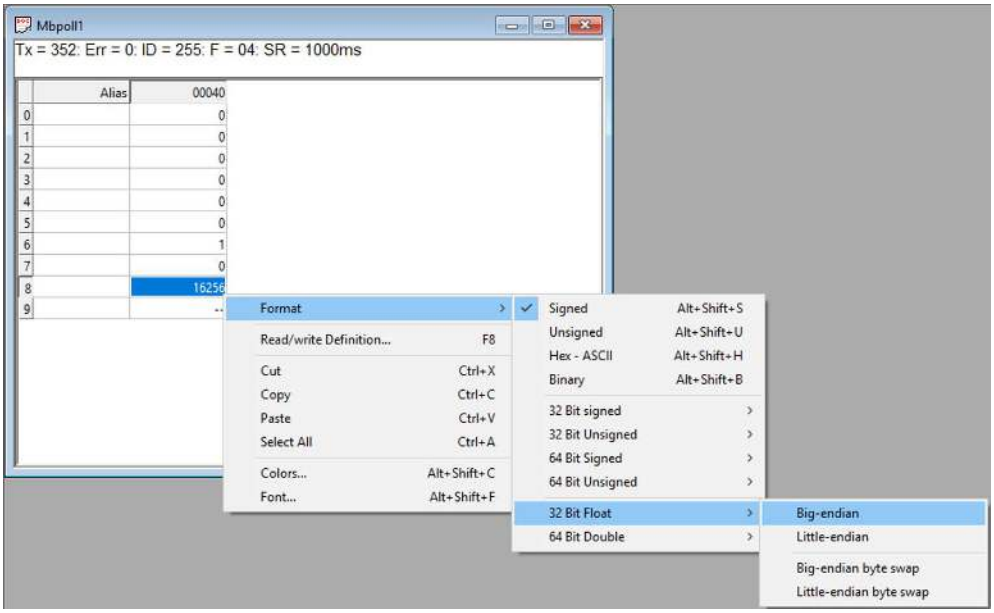


Figure 15: Configure Data format.

In this case, when Address 48 is configured as Float32 Big-endian, the value changes to 1, the current Density median. The data is better shown by configuring the correct data type and adding Alias to each address. The result would look like in the following Figure.

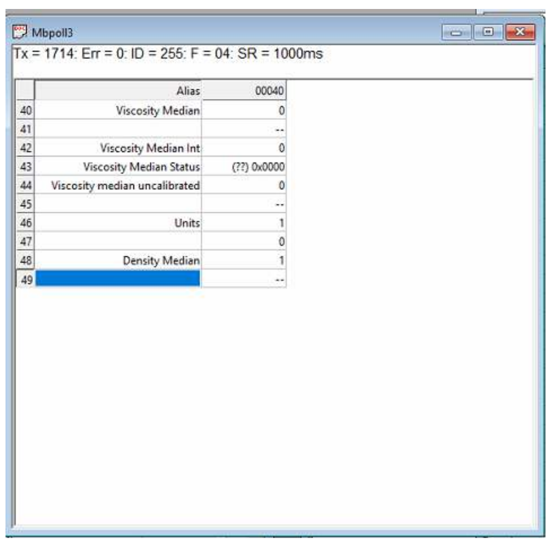


Figure 16: Data with correct format and alias.

6.2 Import Modbuspoll pre-configured files for Rheonics Modbus Devices

To read values through Modbus Poll easily, pre-configured files can be used. These already have the correct data types and alias. You can find them on Rheonics Support web page here. To import them, follow the next steps.

Click Open on the File Tab.

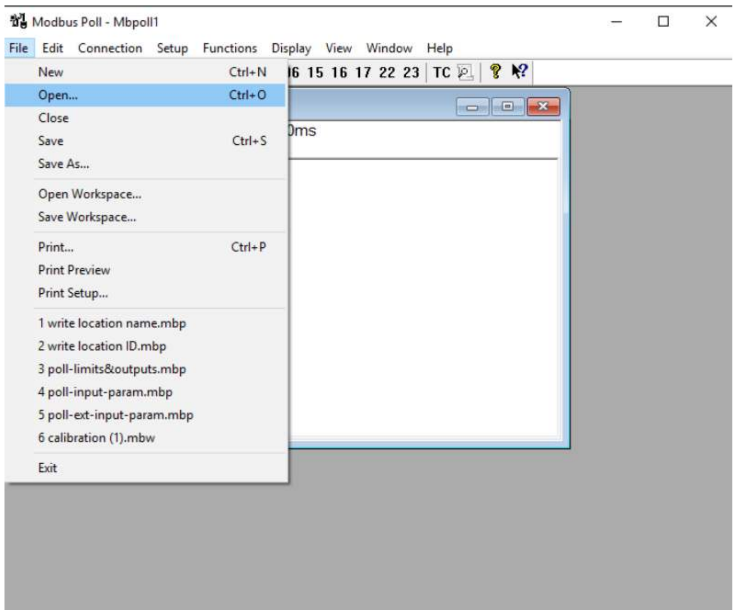


Figure 17: Import pre-configured file.

Choose the desired file between the files available on Rheonics web page.

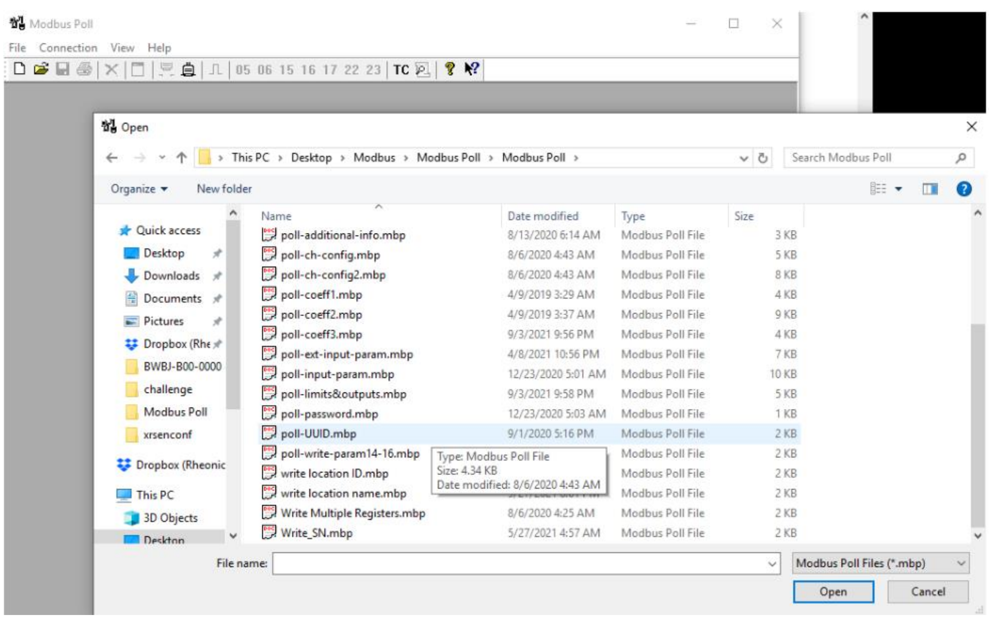


Figure 18: Select a pre-configured file.

The chosen File's Registers will appear with the correct data type and name.

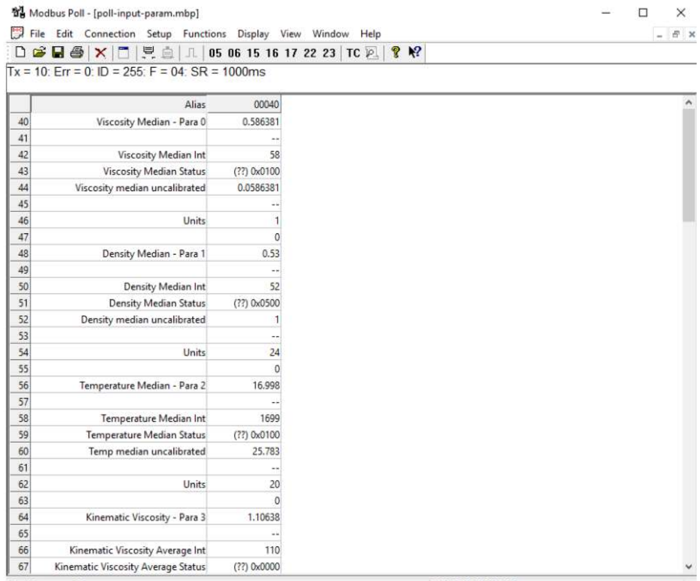


Figure 19: Loaded pre-configured file.

6.3 Density Input through Modbuspoll

In the case of the SRV sensor, the viscosity output (Parameter 0) is the product of dynamic viscosity and density. The default measurement output on the viscosity channel assumes a constant density of 1.0 g/cc (1000 kg/m3). To get the actual dynamic or kinematic viscosity from the SRV, the sensor supports input of a constant or temperature-dependent density. For more information see the Support Article on Rheonics web page.
With density input, dynamic and kinematic viscosity can be directly output by the sensor over digital and analog channels without the need of any PC.

The steps below show how the user can set the density on the sensor using Modbus TCP to get dynamic or kinematic viscosity directly from the sensor electronics as well as on the RCP software. This configuration can also be done via the RCP software.

Introduction to density input for SRV

The Density SRV output can be described as a polynomial of the form:

$$D_{out} = D_{o5}.T^5 + D_{o4}.T^4 + D_{o3}.T^3 + D_{o2}.T^2 + D_{o1}.T + D_{o0}$$

With this model, we can describe either a constant density or a temperature-dependent density.

a. Constant density input
To describe a constant density, the coefficient Do0 should be set equal to the constant density value and all the other coefficients should be set to zero.

b. Temperature-dependent density input

To describe a temperature dependent density, assign values to the coefficients, that reflect the density behavior of your fluid with temperature. If you have a fluid reference table (for example Cannon fluids, Nippon grease), you can fit the density to a polynomial model to get the coefficients.

The equation above expects temperature in °C. If you have data in °F or °K, then convert temperature to the corresponding °C before estimating the coefficients. Once you have the coefficients (Do0, Do1, ...), they can be set into the sensor electronics (SME).

Setting Density Input through Modbus TCP

Over Modbus (either RTU or TCP) there are a set of Registers associated with the density input for the SRV sensors. There are two holding registers associated with each of the D_o coefficients of the polynomial. Each pair of Holdings should form a float 32 data type in Big Endian format.
The registers are password protected. This means, you should input the user password in the holding register 0 to be able to write the coefficients. The Modbus Function to write to multiple registers is 16 and 03 to read the holding registers.

Register	Holding Register Address	Data Type
Password	Address = 0; User Password = 0x5312	Uint16
D_{o0}	1212	Float32
D_{o1}	1214	Float32
D_{o2}	1216	Float32
D_{o3}	1218	Float32
D_{o4}	1220	Float32
D_{o5}	1222	Float32

Table 6: Density Coefficients Registers.

Follow the next steps to configure the Density Input, in this case, using ModbusPoll.

6.3.1.1 Create a new Definition with the following data.

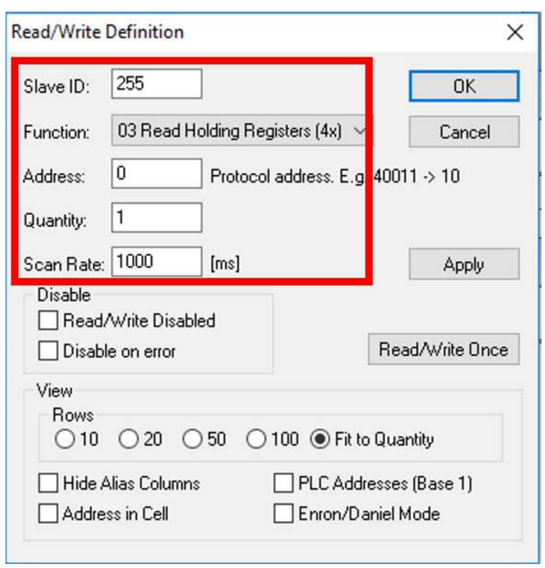


Figure 20: Reading Address 0 from Holding Register.

6.3.1.2 The register 0 from the Holding Register will appear, change the format to Hex and write the password as shown in Figure.

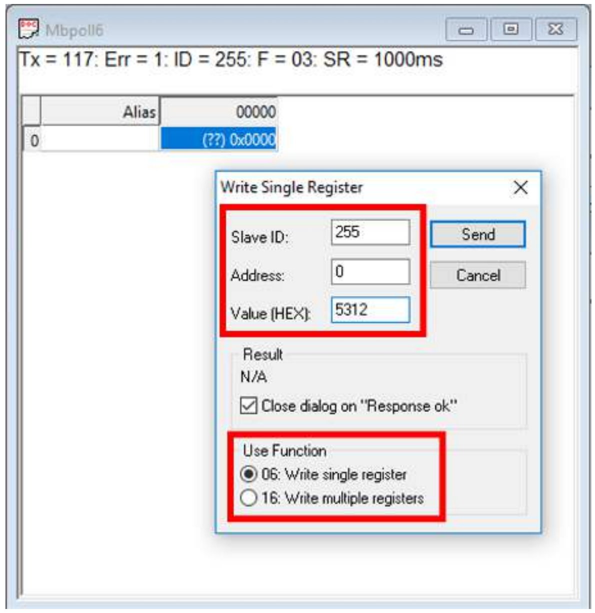


Figure 21: Writing Address 0 from Holding Register.

6.3.1.3 Create another Definition to read and write the D_o coefficients.

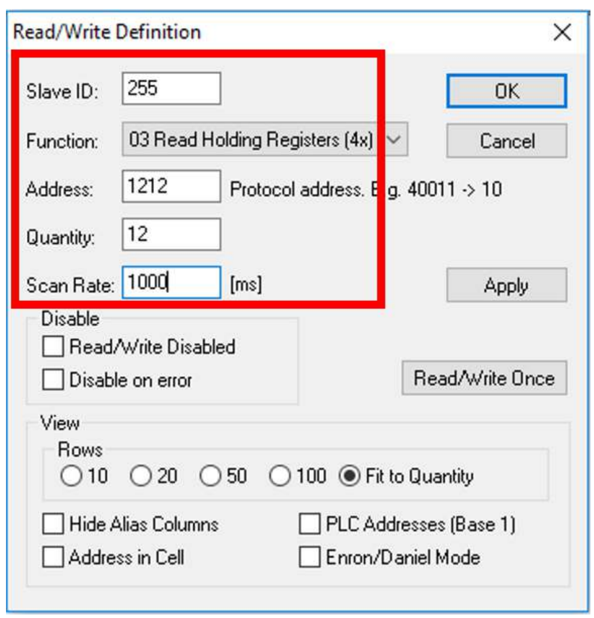


Figure 22: Reading Density Coefficients Registers.

6.3.1.4 Configure these registers with a float 32 format. For a constant density set D_{o0} to the desired value and all others to zero.

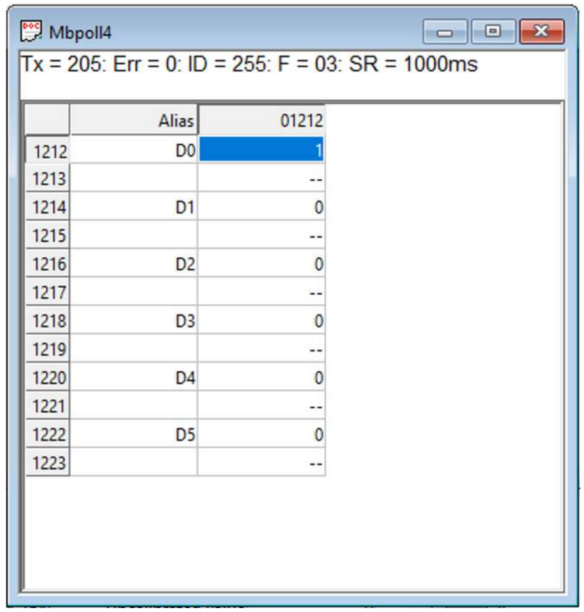


Figure 23: Example for constant density coefficients.

If a temperature dependent density is required, configure the other $\mathrm{D_o}$ parameters as needed.

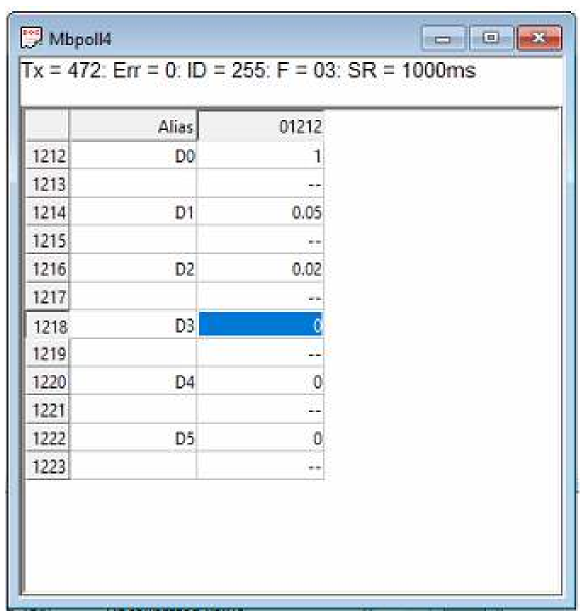


Figure 24: Example for temperature dependent density coefficients.

7 Modbus TCP Command and Response examples

This section helps with the syntax for Modbus TCP Commands and Responses for the mostly used Rheonics parameters. Modbus TCP syntax was previously discussed in Section 5.

7.1 Reading the parameter 3, Temperature median

This is a float data type and the syntax would be as follows:

- 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 (Hy Byte)	Trans ID (Lo Byte)	Protoc. of ID (Hi byte)	Protoc. of ID (Lo byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Add. 1st Reg. (Hi Byte)	Add. 1st Reg. (Lo Byte)	# of Reg. (Hi Byte)	# of Reg. (Lo Byte)
00	00	00	00	00	06	FF	04	00	38	00	02

Response Structure:

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

Data = **0x41B15C29** = 22.17 (Float)
Temperature median = 22.17 °C (default unit)

7.2 Reading the parameter 1, Viscosity median Status

Status value is a UInt16 data type and the syntax would be as follows:

- 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 (Hy Byte)	Trans ID (Lo Byte)	Protoc. of ID (Hi byte)	Protoc. of ID (Lo byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Add. 1st Reg. (Hi Byte)	Add. 1st Reg. (Lo Byte)	# of Reg. (Hi Byte)	# of Reg. (Lo Byte)
00	00	00	00	00	06	FF	04	00	2B	00	01

Response Structure:

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

Data = **0x0001** = 1 (Uint16) Referring to the parameter status, this value means a general error occurred.

7.3 Reading Viscosity Average Float and Int Value, and its status

This is a Int data type and the syntax would be as follows:

- 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 (Hy Byte)	Trans ID (Lo Byte)	Protoc. of ID (Hi byte)	Protoc. of ID (Lo byte)	Length (Hi byte)	Length (Lo byte)	Unit ID	Function Code	Add. 1st Reg. (Hi Byte)	Add. 1st Reg. (Lo Byte)	# of Reg. (Hi Byte)	# of Reg. (Lo Byte)
00	00	00	00	00	06	FF	04	00	40	00	04

Response Structure:

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

Data = **0408 41F4 51EC 0BEE 0000**
Viscosity Average (float32) = 30.54 cP.
Viscosity Average (UInt16) = 3054 cP
Viscosity Average Status (UInt16) = 0

8 Sensor status and parameter status

All Rheonics sensors (SRV, SRD, DVP, DVM) for inline viscosity and density monitoring have inbuilt status. The status can be read through Modbus with Address 39 (Refer to Table 5: Input Registers.) and can be used over digital communication channels to understand when the sensor is operating correctly and when there is an issue.

8.1 Sensor Error Status

The sensor status can take any of the following values OR a combination of them:

Bit	Hex	**Name**	**Comment**
Bit 0	0x0001	PLL frequency mismatch	The PLL frequency does not match the sensor frequency. Derived from the ASB string (E10)
Bit 1	0x0002	PLL not locked	The PLL is not locked. Derived from the ASB string (E01)
Bit 2	0x0004	PLL lock incorrect	The PLL has locked on a wrong frequency. Derived from the ASB string (E02)
Bit 3	0x0008	ASB communication error	Issues with sensor electronics
Bit 4	0x0010	Temperature sensor failed	The temperature sensor has failed. Derived from the ASB string if temperature is -273.0
Bit 5	0x0020	Sensor too hot	If temperature is above the hardcoded physical temperature limit.
Bit 6	0x0040	ASB communication error	Communication issue between two electronics boards in the SME
Bit 7	0x0080	Serial Changed	
Bit 8	0x0100	Status not clean	Sensor is not clean (only SRV)
Bit 9	0x0200	Status in Air	Determines if sensor is in air
Bit 10 -15	Unused		

Table 7: Sensor error status bit code and description.

8.2 Parameter Status

The Parameter Status can take any of the following values, OR a combination of these states.
For example: If there is a config error, the status value will be 0x0003
If there is a config error and an internal error, the status will take a value of 0x0023

Bit	Hex	Name	Comment
Bit 0	0x0001	General error	This bit is always set in case there is an issue with the parameter. It can be used by the general user or application programmer to alert an issue with that parameter output. For details check the other bits
Bit 1	0x0002	Config error	The parameter is not configured or there exists a configuration error.
Bit 2	0x0004	Hardware error	The parameter cannot be calculated as the hardware failed. Example: Temperature sensor has failed.

Bit 3	0x0008	Dependent error	A parameter source for a dependent parameter is not available. Example: In case of a free formula a referenced parameter is NAN.
Bit 4	0x0010	Not ready	No result is yet available. Example: No measurement has been taken yet. The algorithm requires a run-in time
Bit 5	0x0020	Internal error	Internal error - Report to Rheonics
Bit 6	0x0040	Calibration Error	Diagnostics
Bit 7	0x0080	Further use	
Bit 8	0x0100	Parameter Calibrated	Triggered when parameter has a calibration/scale factor/coefficient applied to it.
Bit 9	0x0200	Model Loaded	Active when a model has been loaded in script parameters. Only valid for parameters 19,20,21
Bit 10	0x0400	Filtering Active	Active when there is a filter loaded for that parameter
Bit 11	0x0800	Not stable	Parameter result not yet stable Example: Set for example on viscosity if sensor status is not okay.
Bit 12	0x1000	Warning lower	Below lower warning limit (if configured for parameter)
Bit 13	0x2000	Warning upper	Above upper warning limit (if configured for parameter)
Bit 14	0x4000	Alarm lower	Below alarm limit (Hardcoded depending on parameter type)
Bit 15	0x8000	Alarm upper	Above alarm limit (Hardcoded depending on parameter type)

Table 8: Parameter Status bit code and description.

8.3 How to read sensor status?

Sensor status is a WORD data type, these status bits can be used over digital communication channels to understand when the sensor is operating correctly and when there is an issue.

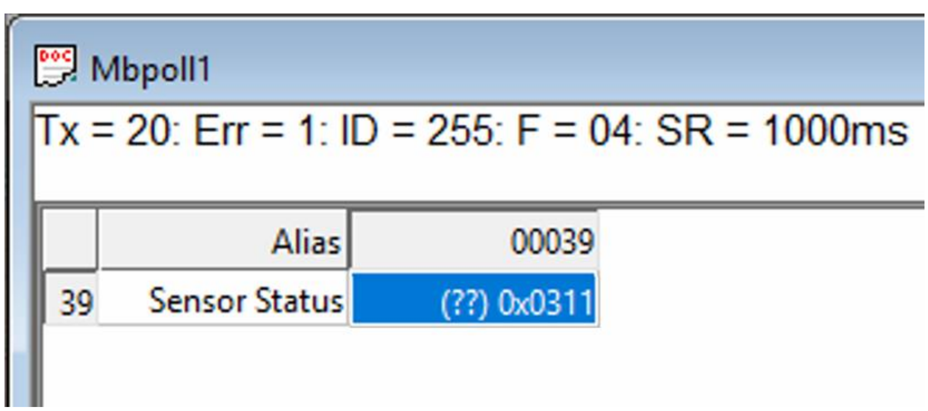


Figure 25: Sensor status example.

In the scenario described in Figure 25, **Sensor Status has a value 0x311.** Sensor error status is the combination of any bit and in this case is the combination (OR) of the following error bits:

Bit	H	Comments
0	0x001	The PLL frequency does not match the sensor frequency. Derived from the ASB string (E10)
4	0x010	The temperature sensor has failed. Derived from the ASB string if temperature is -273.0
8	0x100	Sensor is not clean (only SRV)
9	0x200	Status in Air
Result	0x311	

Table 4. Sensor status bits for status 0117.

8.4 Which parameters should I read?

Each of the 23 parameters from Rheonics sensor are shown in Section 5.2.

For up-to-date parameters information, please check the page:

https://support.rheonics.com/support/solutions/articles/81000393235-parameter-list-access-for-field-devices

https://support.rheonics.com/support/solutions/articles/81000393237-units-translation-table-for-field-devices

Customers normally choose Parameters 12 (Viscosity Median and last good), 13 (Density Median and last good), 2 (Temperature Median) and sensor status for their process as they provide good information for the fluid.

For custom parameters like concentration, compensated viscosity, and density; parameters 19,20, and 21 can be used. **Contact Rheonics support for more information about mathematical models that the sensors support natively.**

9 Units table

Unit Index	Unit Display	Unit Index	Unit Display
0		**35**	°Baumé
1	mPa.s	**36**	°Brix
2	cP	**37**	%wt/v
3	Pa.s	**38**	%v/v
4	Poise	**39**	%vol
5	Reyn	**40**	Bar
6	mm^2/s	**41**	psi
7	cSt	**42**	m^3/s
8	St	**43**	sccm
9	m^2/s	**44**	gpm
10	in^2/s	**45**	pH
11	SUS	**46**	m^3
12	VI	**47**	gal
13	AV	**48**	STP
14	PV	**49**	Tref
15	YP	**50**	n_D
16	sec	**51**	%wt
17	µ	**52**	%Vol
18	η	**53**	mol/m^3
19	ν	**54**	alcohol
20	°C	**55**	ethanol
21	°F	**56**	Hz
22	°K	**57**	rhe
23	$ref_{XX°Y}$	**58**	°P
24	g/cc		
25	Kg/m^3		
26	lb/ft^3		
27	lbm/gal		
28	lbs/gal		
29	ppg		
30	pptf		
31	$slug/ft^3$		
32	SG		
33	ρ		
34	°API		

Table 7. Units translation table.

10 Troubleshooting

Electronics Issues	
No LED lights, display, or output signals	1. Check power supply and cabling
Viscosity is not stable	1. SRV viscosity output is compared against only dynamic viscosity of the calibration fluids. 2. Sensor is not fully in fluid.
Probe Issues	
NaN values are displayed on the RCP software or PLC	1. Check wiring from the probe to SME is correct 2. Measure the internal resistance from the probe
Communication Issues	
No data is visible	1. Try to reset the SME and/or restart your master 2. Verify default port 502 is used 3. Check all settings are correct with RCP and controller side 4. Verify there is no duplicated address in the network 5. Try a different cable from the network 6. Try to use the ping command to get a response from either the PLC or SME
Data issues	
Measured value is different from my reference standards	1. SRV viscosity output is compared against only dynamic viscosity of the calibration fluids
Wrong Unit from the data stream	1. Unit is received as HEX representation, convert to decimal
Wrong readings for each parameter	1. Verify each parameter is correctly mapped, some bytes are swapped
Sensor Status does not match any bit from the sensor status table	1. Sensor status byte is swapped
CRC Error	1. Check that Pre-Delay is at least 50 mS

https://support.rheonics.com/support/solutions/articles/81000397718-modbus-tcp-ip-troubleshooting-guide	

- Contact RHEONICS support desk
    - support@rheonics.com
    - https://support.rheonics.com

https://support.rheonics.com/support/home	

11 Reviews and approvals

Version	Nature of changes	Approval	FW version	Date
1	Original	C. Arroyo	3.30/15	16.02.2024

12 Notes/Errata

Contact Rheonics support for customization of system settings.

Notes

EUROPE HEADQUARTERS
Rheonics GmbH
Klosterstrasse. 19
8406 Winterthur Switzerland
Tel: +41 52 511 32 00

US HEADQUARTERS
Rheonics, Inc.
3 Sugar Creek Center Blvd, Ste 100
Sugar Land, TX 77478
United States of America
Tel: +1 713 364 5427

Website: https://rheonics.com
Support Portal: https://support.rheonics.com
Support E-Mail: support@rheonics.com
Sales E-Mail: info@rheonics.com

Modbus

rheonics
inline process
density and viscosity
monitoring