

Aqua TROLL 400

Multiparameter Instrument

Modbus and SDI-12 Reference Guide



March 2014

rev. 003

Table of Contents

| | | |
|-------|--|----|
| 1. | Controller Requirements and Connections..... | 6 |
| 1.1 | Wiring Overview..... | 6 |
| 1.2 | Power Connections..... | 6 |
| 1.3 | SDI-12 Wiring Diagram..... | 7 |
| 1.4 | Modbus Master RS485 Wiring Diagram..... | 8 |
| 1.5 | Modbus Master RS232 Wiring Diagram (Converter Required)..... | 9 |
| 1.6 | RS485 Network Guidelines..... | 10 |
| 2. | SDI-12 Interface..... | 12 |
| 2.1 | Instrument Defaults..... | 12 |
| 2.2 | Device Identification..... | 13 |
| 2.3 | Basic Commands..... | 14 |
| 2.4 | Extended Commands..... | 17 |
| 2.5 | SDI-12 Configuration File..... | 25 |
| 2.5.1 | Connect the Communication Device and Comm Kit Software..... | 25 |
| 2.5.2 | SDI-12 Setup..... | 26 |
| 3. | Modbus Registers—Probe..... | 27 |
| 3.1 | Modbus Communication Setup..... | 27 |
| 3.2 | Automatic Salinity Correction..... | 27 |
| 3.3 | Automatic Barometric Correction..... | 28 |
| 3.4 | Automatic Density Correction..... | 28 |
| 4. | Modbus Registers—RDO Sensor..... | 28 |
| 4.1 | Header Registers—RDO..... | 28 |
| 4.2 | Sensor Serial Number..... | 29 |
| 4.3 | Sensor Status..... | 29 |
| 4.4 | Last Factory Calibration..... | 29 |
| 4.5 | Next Factory Calibration..... | 30 |
| 4.6 | Warm-Up Time..... | 30 |
| 4.7 | Fast Sample Rate..... | 30 |
| 4.8 | Parameter Registers—RDO..... | 30 |
| 4.9 | Calibration Registers—RDO..... | 32 |
| 4.10 | Live Salinity Value..... | 33 |
| 4.11 | Default Salinity Value..... | 33 |

| | | |
|--------|--|----|
| 4.12 | Live Barometric Pressure..... | 33 |
| 4.13 | Default Barometric Pressure | 33 |
| 4.14 | 100% Saturation Calibration Values..... | 33 |
| 4.15 | Zero% Saturation Calibration Values | 33 |
| 4.16 | Calibration Slope and Offset | 34 |
| 4.17 | Calibration Procedure—RDO | 34 |
| 4.17.1 | RDO Calibration Calculations | 35 |
| 5. | Modbus Registers—Conductivity Sensor | 36 |
| 5.1 | Header Registers—Conductivity | 36 |
| 5.2 | Parameter Registers—Conductivity | 38 |
| 5.3 | Calibration Registers—Conductivity..... | 42 |
| 5.3.1 | Cell Offset and Cell Constant | 42 |
| 5.4 | Calibration Procedure—Conductivity | 43 |
| 6. | Modbus Registers—Level Sensor..... | 43 |
| 6.1 | Header Registers..... | 43 |
| 6.1.1 | Warm-up Time | 44 |
| 6.1.2 | Fast Sample Rate..... | 45 |
| 6.2 | Parameter Registers—Level | 45 |
| 6.3 | Calibration Registers—Level..... | 46 |
| 6.3.1 | Pressure Calculation | 47 |
| 6.3.2 | Pressure Offset | 47 |
| 6.3.3 | Specific Gravity | 47 |
| 6.3.4 | Level Reference | 47 |
| 6.3.5 | Pressure Reference | 48 |
| 6.4 | Calibration Procedure—Pressure..... | 48 |
| 6.5 | Calibration Procedure—Level | 48 |
| 7. | Modbus Registers – pH/ORP Sensor..... | 49 |
| 7.1 | Header Registers..... | 49 |
| 7.2 | Parameter Registers..... | 51 |
| 7.3 | Calibration Registers—pH/ORP..... | 52 |
| 7.3.1 | pH Measured Value..... | 53 |
| 7.3.2 | ORP Measured Value | 54 |
| 7.4 | Calibration Procedure—pH | 54 |
| 7.5 | Calibration Procedure—ORP | 54 |
| 7.5.1 | Sensor Command Timeouts..... | 54 |

| | | |
|--------|---|----|
| 8. | Appendix—Modbus Additional Information..... | 55 |
| 8.1 | Modbus Tutorial | 55 |
| 8.2 | Modbus Modes | 55 |
| 8.3 | Protocol Overview..... | 56 |
| 8.4 | General Message Formats..... | 56 |
| 8.5 | RTU Message Format..... | 56 |
| 8.5.1 | Computer (Master) Message Format..... | 56 |
| 8.5.2 | Device (Slave) Response Format..... | 57 |
| 8.6 | ASCII Message Format..... | 57 |
| 8.6.1 | Computer (Master) Message Format..... | 57 |
| 8.6.2 | Device (Slave) Response Format..... | 57 |
| 8.7 | IP Message Format..... | 58 |
| 8.7.1 | Computer (Master) Message Format..... | 58 |
| 8.7.2 | Device (Slave) Message Format | 58 |
| 8.8 | Data Addresses (Registers) | 58 |
| 8.9 | Function Codes..... | 59 |
| 8.10 | Standard Message Formats..... | 59 |
| 8.10.1 | Read Holding Registers | 60 |
| 8.10.2 | Write Holding Register | 60 |
| 8.10.3 | Write Holding Registers | 60 |
| 8.10.4 | Mask Write Register..... | 61 |
| 8.10.5 | Report Slave ID | 62 |
| 8.11 | Instrument Manufacturer Data Types..... | 63 |
| 8.11.1 | Short..... | 63 |
| 8.11.2 | Unsigned Short | 63 |
| 8.11.3 | Long | 63 |
| 8.11.4 | Unsigned Long..... | 64 |
| 8.11.5 | Float | 64 |
| 8.11.6 | Double | 64 |
| 8.11.7 | Character..... | 64 |
| 8.11.8 | String | 64 |
| 8.11.9 | Time | 65 |
| 8.12 | Exception Codes..... | 66 |
| 8.13 | Extended Exception Codes..... | 67 |
| 8.14 | Probe Register Map Layout | 68 |

| | | |
|--------|---|----|
| 8.15 | Probe Common Registers..... | 69 |
| 8.16 | Communication Control Registers..... | 70 |
| 8.17 | Sensor Connection Registers | 70 |
| 8.18 | Current Loop Configuration Registers..... | 71 |
| 8.19 | Logged Record Registers | 72 |
| 8.20 | Register Map Template ID | 72 |
| 8.21 | Device ID | 73 |
| 8.22 | Device Serial Number | 73 |
| 8.23 | Manufacture Date | 73 |
| 8.24 | Firmware, Boot Code, Hardware Versions | 73 |
| 8.25 | Device Name | 73 |
| 8.26 | Site Name..... | 73 |
| 8.27 | Coordinates | 73 |
| 8.28 | Current Time..... | 73 |
| 8.29 | Device Status..... | 74 |
| 8.30 | Serial Communication Configuration..... | 75 |
| 8.31 | Baud Rates | 76 |
| 8.32 | RTU Settings | 76 |
| 8.33 | ASCII Settings | 76 |
| 8.34 | Max Message/Response Size..... | 76 |
| 8.35 | Message Counters..... | 76 |
| 8.36 | Sensor Connection Registers | 77 |
| 8.36.1 | Max Sensor Connections..... | 77 |
| 8.36.2 | Sensor Connection Status | 77 |
| 8.37 | Sensor Map Registers | 78 |
| 8.37.1 | Sensor ID Registers..... | 78 |
| 8.37.2 | Sensor Status Register | 78 |
| 8.37.3 | Sensor Command Register..... | 78 |
| 8.37.4 | Sensor Data Register Map Version Registers..... | 78 |
| 8.37.5 | Sensor Data Register Map Offsets..... | 79 |
| 8.37.6 | Sensor Data Cache Timeout..... | 79 |
| 8.38 | Current Loop Configuration..... | 79 |
| 8.39 | Last Logged Record Registers..... | 79 |

1. Controller Requirements and Connections

The Multi-PRO 400 may be connected to a controller or logger for communication via:

- SDI-12
- RS485 Modbus
- RS232 Modbus (with converter)

1.1 Wiring Overview

Refer to the diagrams on the following pages. Trim and insulate unused wires. The shielded wire should be wired to a chassis ground or earth ground.

Cable Stripped and Tinned

| Signal | Color |
|----------------|-------|
| Ground/Return | Black |
| External Power | Red |
| No Connection | Brown |
| RS485 (-) | Green |
| RS485 (+) | Blue |
| SDI-12 | White |

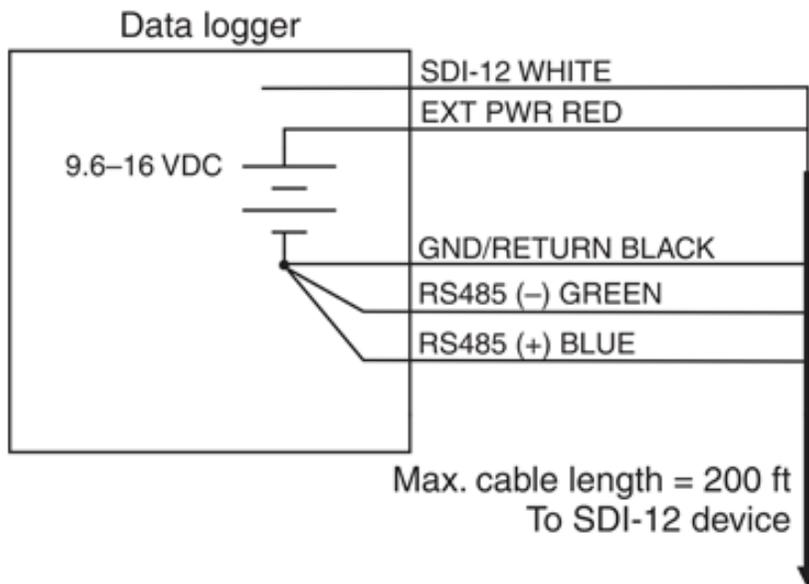
1.2 Power Connections

The Multi-PRO 400 requires an external 8 to 36 VDC power source. The red wire must be connected to the positive terminal of the power source. The black wire must be connected to the negative terminal of the power source, which is often referred to as the system ground or return.

1.3 SDI-12 Wiring Diagram

Cable length must not exceed 60.9 m (200 ft).

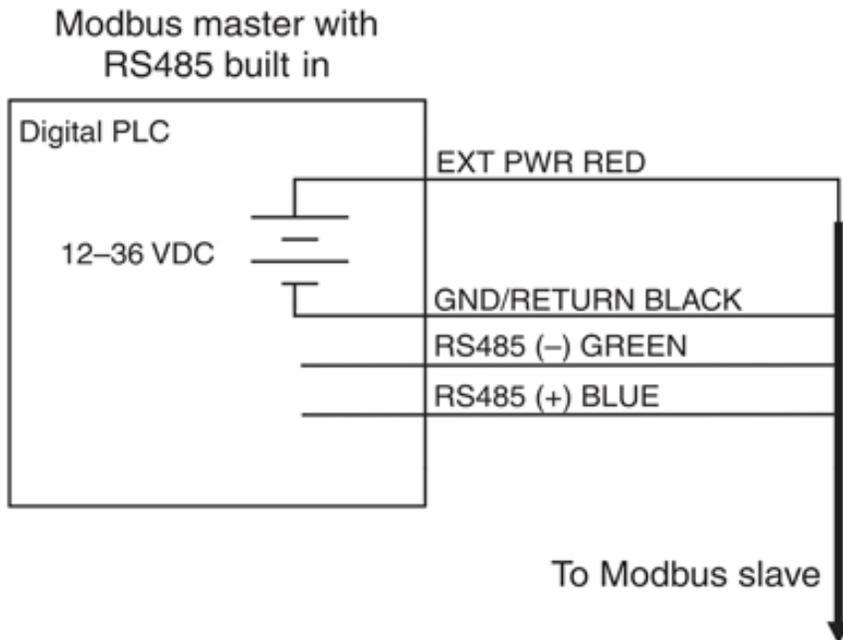
| Signal | Color |
|-----------------------------|-------|
| Ground/Return | Black |
| External Power (9.6-16 VDC) | Red |
| SDI-12 | White |



1.4 Modbus Master RS485 Wiring Diagram

Cable length must not exceed 1219.2 m (4000 ft).

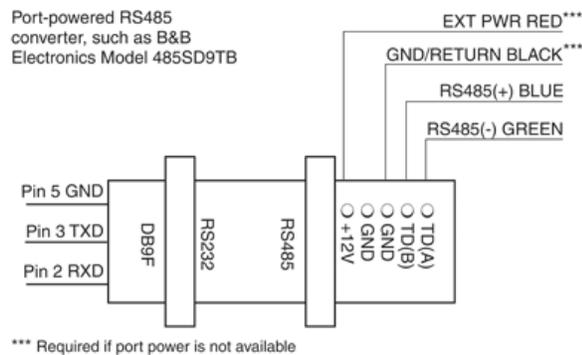
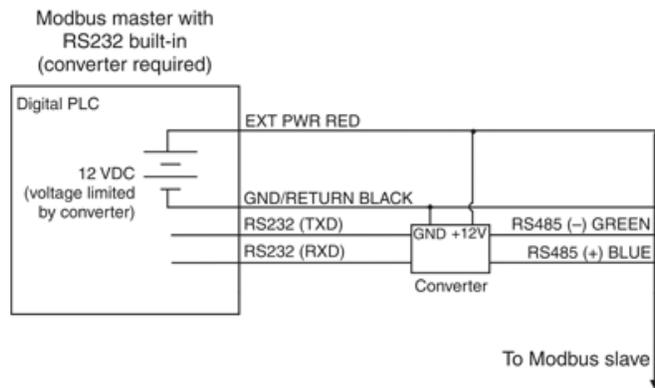
| Signal | Color |
|----------------------------|-------|
| Ground/Return | Black |
| External Power (12-36 VDC) | Red |
| RS485 (-) | Green |
| RS485 (+) | Blue |



1.5 Modbus Master RS232 Wiring Diagram (Converter Required)

Cable length between Master and Slave must not exceed 1219.2 m (4000 ft). Rugged Cable length between Master and Converter must not exceed 6 m (20 ft).

| Signal | Color |
|---|-------|
| Ground/Return | Black |
| External Power (12-36 VDC, voltage limited by converter) | Red |
| RS485 (-) | Green |
| RS485 (+) | Blue |

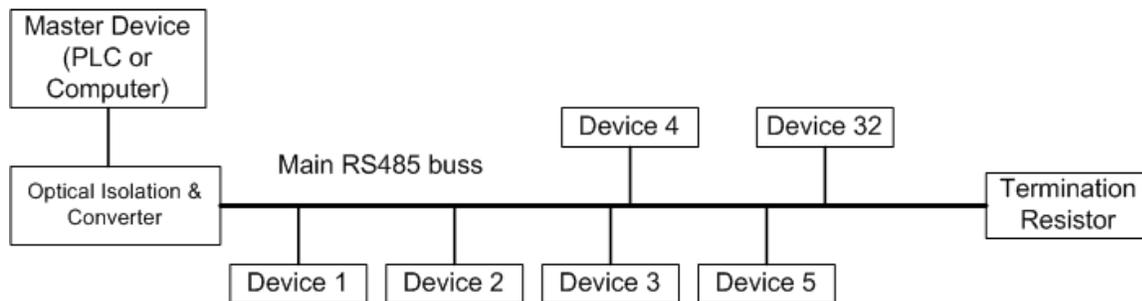


1.6 RS485 Network Guidelines

The manufacturer uses RS485 as its main digital communications link. RS485 is often used in an industrial setting as a small device network. There are some installation guidelines to follow when configuring an RS485 network with implications to use with the instrument.

RS485 Rule 1

RS485 is a bus network. It does not work when configured in a star network topology. This means that a user can have a network that looks like 1 long wire (up to 4000 ft) with short stubs hanging off the main branch with a device. Each stub must be less than 1 meter in length. See picture below:



RS485 Rule 2

The terminating resistor at the end of the network should be only 120 ohms. The bus is terminated on the long main bus wire at the opposite end from the Master.

RS485 Rule 3

This rule is not specific to RS485; rather it applies to any situation where you have long wires running across the ground or in the ground connected back to a computer. Always add an optical isolator to the link between the main bus wire and the Master device. This reduces the chance that a nearby lightning strike will damage the Master device.

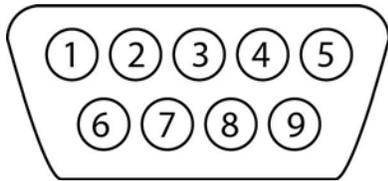
RS485 Rule 4

There can be only 32 devices per network, including the Master.

Implications to users of this instrument are as follows:

These devices are typically deployed on a cable of much greater length than the 1 meter stub supported by RS485. The above documented Rule 1 requires that only two devices are on an individual RS485 link, the PLC and the instrument. Many PLC's support multiple RS485 networks which can be used to connect multiple instruments to a single PLC.

DB-9 Pin Diagram



| Pin | Signal Name | |
|-----|----------------------|-----|
| 1 | Carrier Detector | DCD |
| 2 | Receive Data | RXD |
| 3 | Transmit Data | TXD |
| 4 | Data Terminal Ready | DTR |
| 5 | Signal Ground/Common | GND |
| 6 | Data Set Ready | DSR |
| 7 | Request to Send | RTS |
| 8 | Clear to Send | CTS |
| 9 | Ring Indicator | RI |

2. SDI-12 Interface

The Multi-PRO 400 Instrument adheres to the Serial Digital Interface Standard for Microprocessor-based sensors, version 1.3, and the extensions to the specifications identified in this document. This section identifies the device-specific implementation of the standard. For more information on SDI-12 see www.sdi-12.org.

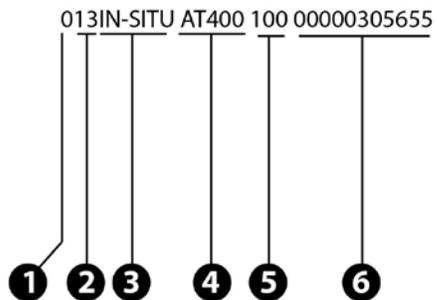
2.1 Instrument Defaults

The Multi-PRO 400 Instrument leaves the factory with the following settings:

| | |
|-----------------|---|
| Device Address | 0 |
| Parameters | 1: DO concentration in mg/L 2: DO percent saturation 3: Temperature in °C 4: Specific conductivity in µS/cm 5: Depth in feet 6: pH 7: ORP in millivolts |
| Output Sequence | The default sequence is oTsCLHr. o = Dissolved Oxygen Concentration T = Temperature s = Dissolved Oxygen Saturation C = Specific Conductivity L = Level H = pH r = ORP |

2.2 Device Identification

In response to the Send Identification command, the instruments respond as follows:



| | |
|----|------------------------------------|
| 1. | Device address |
| 2. | SDI-12 compatibility (Version 1.3) |
| 3. | Manufacturer |
| 4. | Instrument model |
| 5. | Firmware version (100 = 1.00) |
| 6. | Serial number |

2.3 Basic Commands

The following table lists each basic SDI-12 command, its format, and the format of each response.

| Name | Command | Response |
|---------------------|-------------|--|
| Address Query | ?! | a<CR><LF> The wildcard address '?' character is supported only for the Address Query command. It is ignored as an invalid address for all other commands. |
| Acknowledge Active | a! | a<CR><LF> Basic address characters in the range '0' to '9' and the extended address characters in the ranges 'A' to 'Z' and 'a' to 'z' are supported. All other characters are ignored as an invalid address. The default address is '0'. |
| Change Address | aAb! | b<CR><LF> Software changeable addresses and the Change Address command are supported. |
| Send Identification | al! | a13IN-SITU AT400 vv xxxxxxxxxxx<CR><LF> vv = device firmware version * 100 (120 = 1.20) xxx = 10-digit device serial number with leading zeroes |
| Start Verification | aV! | a0011<CR><LF> One result is available for reading by the Send Data command within 1 second. A service request (a<CR><LF>) will be sent when the result is ready. |
| Send Data | aD0! | a+n<CR><LF> n = lower 16 bits of the device status (0-65535) |
| Additional Data | aD1!...aD9! | a<CR><LF> No values are returned after an Additional Data command. |

| Name | Command | Response |
|---|---|---|
| <p>Start Measurement</p> <p>Start Measurement CRC</p> | <p>aM!</p> <p>aMC!</p> | <p>a004n<CR><LF></p> <p>n parameters will be available for reading by the Send Data command within 4 seconds. A service request (a<CR><LF>) will be sent when the parameters are ready. The number of parameters returned is determined by the SDI-12 configuration file. The default value for n is 7. The default parameters are:</p> <ol style="list-style-type: none"> 1: DO concentration in mg/L 2: DO percent saturation 3: Temperature in °C 4: Specific conductivity in uS/cm 5: Depth in feet 6: pH 7: ORP in millivolts <p>If the device is configured to output more than 9 parameters, only the first 9 parameters can be measured by this command.</p> |
| <p>Send Data</p> | <p>aD0!</p> | <p>a<values><CR><LF> or a<values><CRC><CR><LF></p> <p>The number and type of parameters returned is determined by the SDI-12 configuration file. At most 3 parameters are returned in a Send Data command. If more than 3 parameters are available, they are returned using the additional data command.</p> |
| <p>Additional Data</p> | <p>aD1!...aD9!</p> | <p>a<values><CR><LF> or a<values><CRC><CR><LF></p> <p>At most 3 parameters are returned per additional data request.</p> |
| <p>Additional Measurements</p> <p>Additional with CRC</p> <p>Send Data</p> <p>Additional Data</p> | <p>aM1! ... aM9!</p> <p>aMC1!... aMC9!</p> <p>aD0!</p> <p>aD1! ... aD9!</p> | <p>a0000<CR><LF></p> <p>No additional measurements shall be started by the device.</p> <p>a<CR><LF> or a<CRC><CR><LF></p> <p>No values are returned after an Additional Measurement command.</p> |

2.4 Extended Commands

The Multi-PRO 400 Instrument supports the following extended SDI-12 commands.

| Name | Command | Response |
|-----------------------------|----------|--|
| Set Factory Defaults | aXFD! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. Restores all settings and calibration values to their factory defaults. |
| Send Data | aD0! | a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |
| Set Output Sequence | aXOnnn! | aOK<CR><LF> or aERROR<CR><LF> nnn = 1 to 15 parameter characters in the desired output order. The default sequence is oTsCLHr. o = Dissolved Oxygen Concentration T = Temperature s = Dissolved Oxygen Saturation p = Partial Pressure of Oxygen A = Actual Conductivity C = Specific Conductivity S = Salinity D = Total Dissolved Solids R = Resistivity d = Density of Water P = Pressure L = Level H = pH h = pH mV r = ORP |
| Read Output Sequence | aXO! | annn<CR><LF> nnn = 1 to 15 parameter characters as described for the XO command. |
| Set DO Concentration Units | aXoUnnn! | aOK<CR><LF> or aERROR<CR><LF> nnn = the DO concentration units ID Three digits are required. |
| Read DO Concentration Units | aXoU! | annn<CR><LF> nnn = DO concentration units ID |

| Name | Command | Response |
|--------------------------------|-------------------------|---|
| Set Temperature Units | aXTU \overline{nnn} ! | aOK<CR><LF> or aERROR<CR><LF> nnn = the temperature units ID Three digits are required. |
| Read Temperature Units | aXTU! | annn<CR><LF> nnn = temperature units ID |
| Restore DO Factory Calibration | aXoFC! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. Restores DO sensor slope and offset calibration values to their factory defaults. No change to other parameter settings. |
| Send Data | aD0! | a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |
| Set DO Slope | aXoSpd.d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = slope p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read DO Slope | aXoS! | apd.d<CR><LF> pd.d = slope |
| Set DO Offset | aXoOpd.d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = offset p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read DO Offset | aXoO! | apd.d<CR><LF> pd.d = offset |
| Set Conductivity Units | aXCU \overline{nnn} ! | aOK<CR><LF> or aERROR<CR><LF> nnn = the actual and specific conductivity units ID Three digits are required. |
| Read Conductivity Units | aXCU! | annn<CR><LF> nnn = actual and specific conductivity units ID |

| Name | Command | Response |
|--|-----------|--|
| Set Specific Conductivity Alpha | aXCp d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = alpha p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read Specific Conductivity Alpha | aXCA! | apd.d<CR><LF> pd.d = Alpha |
| Set Specific Conductivity Reference Temperature | aXCTp d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = reference temperature p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read Specific Conductivity Reference Temperature | aXCT! | apd.d<CR><LF> pd.d = reference temperature |
| Set TDS Units | aXDU nnn! | aOK<CR><LF> or aERROR<CR><LF> nnn = the TDS units ID Three digits are required. |
| Read TDS Units | aXDU! | annn<CR><LF> nnn = TDS units ID |
| Set TDS Factor | aXDFp d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = conversion factor p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read TDS Factor | aXDF! | apd.d<CR><LF> pd.d = conversion factor |
| Restore Conductivity Factory Calibration | aXCFC! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. Restores conductivity sensor cell constant to its factory default. No change to other parameter settings. |
| Send Data | aD0! | a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |

| Name | Command | Response |
|------------------------------------|-----------|--|
| Set Conductivity Cell Constant | aXCKpd.d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = cell constant p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read Conductivity Cell Constant | aXCK! | apd.d<CR><LF> pd.d = cell constant |
| Set Pressure Units | aXPUnnn! | aOK<CR><LF> or aERROR<CR><LF> nnn = the pressure units ID Three digits are required. |
| Read Pressure Units | aXPU! | annn<CR><LF> nnn = pressure units ID |
| Set Level Units | aXLUnnn! | aOK<CR><LF> or aERROR<CR><LF> nnn = the level units ID Three digits are required. |
| Read Level Units | aXLU! | annn<CR><LF> nnn = level units ID |
| Set Level Mode | aXLMnnn! | aOK<CR><LF> or aERROR<CR><LF> nnn = the level mode Three digits are required. |
| Read Level Mode | aXLM! | annn<CR><LF> nnn = level mode |
| Set Specific Gravity | aXLGpd.d! | aOK<CR><LF> or aERROR<CR><LF> pd.d = specific gravity p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) |
| Read Specific Gravity | aXLG! | apd.d<CR><LF> pd.d = specific gravity |

| Name | Command | Response |
|--|-----------------------|---|
| Restore Level Factory Calibration Send Data | aXLFC! aD0! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. Restores pressure and level calibrations to their factory defaults. No change to other parameter settings. a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |
| Zero Pressure Send Data | aXPZ! aD0! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. A service request (a<CR><LF>) will be sent when the result is ready. a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |
| Set Level Reference Send Data | aXLRpd.d! aD0! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. A service request (a<CR><LF>) will be sent when the result is ready. pd.d = level reference in current units p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional) a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |
| Restore pH/ORP Factory Calibration Send Data | aXHFC! aD0! | a0011<CR><LF> One result is available in 1 second for reading by the Send Data command. Restores pH and ORP calibrations to their factory defaults. a+s<CR><LF> s = command status, 1 = command successful, 0 = command failed. |

| Name | Command | Response |
|-----------------|-----------|--|
| Set ORP Offset | aXrOpd.d! | <p>a0011<CR><LF></p> <p>One result is available in 1 second for reading by the Send Data command. A service request (a<CR><LF>) will be sent when the result is ready.</p> <p>pd.d = ORP offset (+/- 1000 mV) p = polarity sign (+ or -) d = digits (1 to 7) . = decimal point (optional)</p> |
| Send Data | aD0! | <p>a+s<CR><LF></p> <p>s = command status, 1 = command successful, 0 = command failed (offset out of range, sensor not plugged in)</p> |
| Read ORP Offset | aXrO! | <p>a0011<CR><LF></p> <p>One result is available in 1 second for reading by the Send Data command.</p> |
| Send Data | aD0! | <p>a+pd.d<CR><LF></p> <p>pd.d = ORP offset (mV, returns -9999 if sensor is not plugged in)</p> |

| Name | Command | Response |
|---------------------------|-----------------|--|
| ISCO Compatibility | aXPR0! | <p>aIxIxIx<CR><LF></p> <p>Each Ix is a character pair identifying the parameter and units for each measurement. The number of Ix pairs equals the number of data values returned for the Start Measurement and Start Concurrent commands. The following pairs are supported:</p> <p>A0 = Temperature, °C A1 = Temperature, °F B0 = Actual Conductivity, mS/cm B1 = Actual Conductivity, uS/cm C0 = Specific Conductivity, mS/cm C1 = Specific Conductivity, uS/cm c2 = Resistivity, ohm-cm d0 = Pressure, psia d2 = Pressure, psig d3 = Pressure, mmHg F0 = Dissolved Oxygen, mg/L F1 = Dissolved Oxygen, percent saturation G0 = pH G1 = pH, mV H0 = ORP, mV I0 = Level, m I1 = Level, ft</p> <p>If not listed above, all other parameter and unit combinations will return “??”.</p> |
| ISCO Additional | aXPR1!...aXPR9! | <p>a<CR><LF></p> <p>No values are returned after an additional ISCO compatibility command.</p> |
| Communication Diagnostics | aXCD! | <p>a+A+C<CR><LF></p> <p>A = contents of Modbus device address register 9200 C = contents of Modbus serial communication configuration register 9201</p> |

2.5 SDI-12 Configuration File

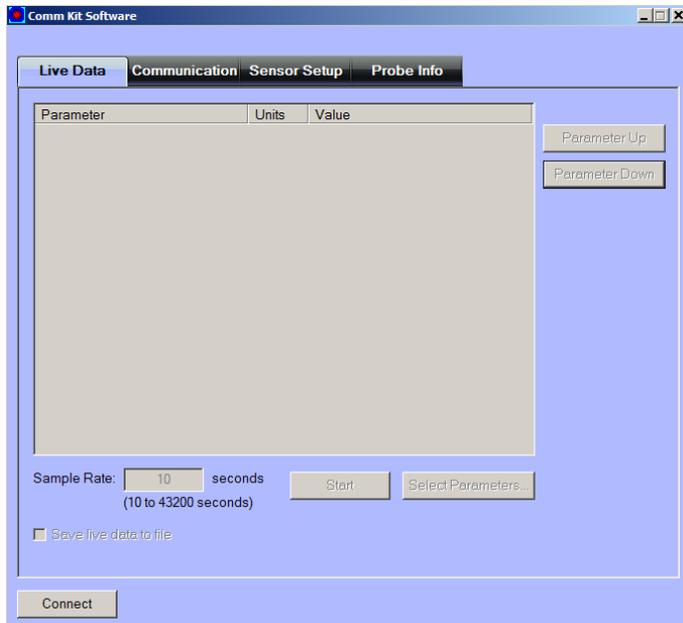
The SDI-12 configuration file can be edited using the Comm Kit Software and the Communication Device.

2.5.1 Connect the Communication Device and Comm Kit Software

The Communication Device connects a stripped-and-tinned cable and instrument to a computer via USB connection and enables a connection with the Comm Kit Software.

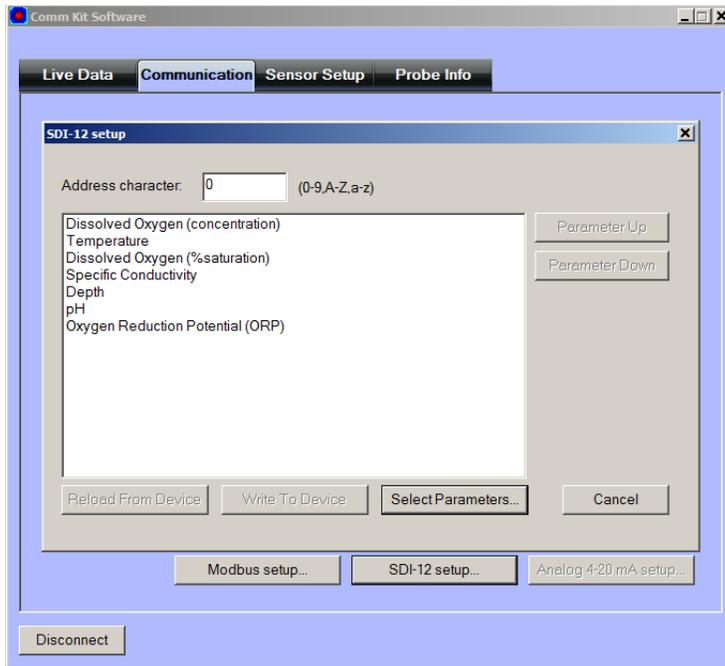


1. The communication device includes an electrical connection diagram label. To attach the sensor to the communication device, depress a lever and insert the appropriate wire in the location specified by the diagram on the label.
2. Attach the USB connection to a computer.
3. Open the Comm Kit Software and click the Connect button.



2.5.2 SDI-12 Setup

SDI-12 setup allows you to set the instrument address, select the parameters you intend to log, and select the order in which the parameters will appear in the log.

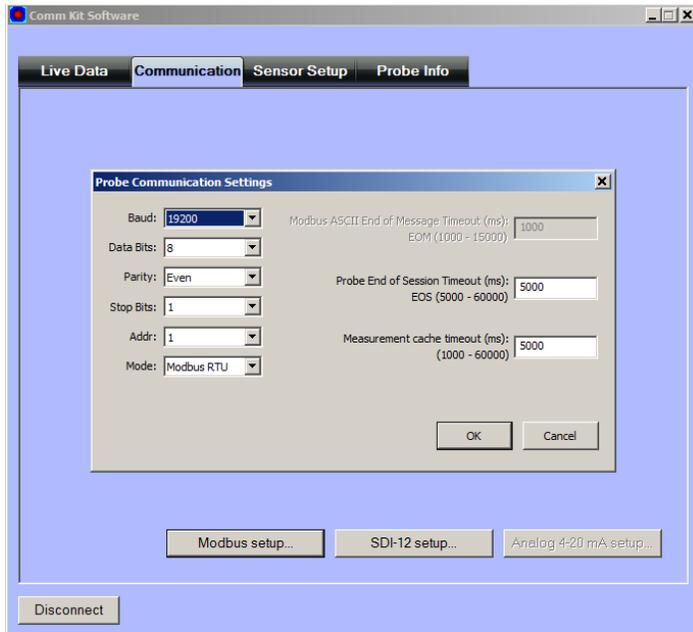


| Screen Element | Purpose |
|-----------------------------------|---|
| Address character (text field) | Allows you to assign a unique SDI-12 address to the instrument. Use 0-9, A-Z, or a-z. |
| Reload From Device (button) | Restores the settings that were last written to the instrument. |
| Write to Device (button) | Writes the parameters and the address to the instrument. |
| Select Parameters (button) | Allows you to enable and disable parameters. Click the Select Parameters button and click the checkboxes to select or clear parameters. |
| Parameter Up (button) | Allows you to move a parameter higher in the list. Click the parameter and then click the Parameter Up button until the desired location is reached. |
| Parameter Down (button) | Allows you to move a parameter lower in the list. Click the parameter and then click the Parameter Down button until the desired location is reached. |

3. Modbus Registers—Probe

3.1 Modbus Communication Setup

See Connect the Communication Device on page 25. Click the Modbus setup button and assign instrument settings according to the requirements of the controller.



The following Modbus registers are specific to the instrument. More information about Modbus, including protocol specifications can be downloaded from www.modbus.org.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|--|
| 7000 | 1 | R1 | ushort | Probe Register Map Template Version (1) |
| 7001 | 1 | R1/W3 | ushort | Automatic Salinity Correction |
| 7002 | 1 | R1/W3 | ushort | Automatic Barometric Pressure Correction |
| 7003 | 1 | R1/W3 | ushort | Automatic Density Correction |
| 7004 | 1 | R1 | ushort | External Power Voltage (millivolts) |

3.2 Automatic Salinity Correction

Set this register to 0 to disable automatic salinity correction. Set to 1 to enable automatic salinity correction. The default is disabled. If disabled, the RDO sensor shall apply its live salinity register value to correct its oxygen concentration for salinity. If enabled, the RDO sensor shall use the salinity value from the conductivity sensor to correct its oxygen concentration for salinity.

3.3 Automatic Barometric Correction

Set this register to 0 to disable automatic barometric correction. Set to 1 to enable automatic barometric correction. The default is disabled. If enabled, the level sensor shall subtract the live barometric pressure register from its pressure sensor readings. This register shall not affect how the RDO sensor utilizes the live barometric pressure register to calculate its parameters.

3.4 Automatic Density Correction

Set this register to 0 to disable automatic density correction. Set to 1 to enable automatic density correction. The default is disabled. If disabled, the level sensor shall apply its specific gravity value to convert pressures to level. If enabled, the level sensor shall use the density value from the conductivity sensor to convert pressures to level.

4. Modbus Registers—RDO Sensor

4.1 Header Registers—RDO

The sensor map data-register offset points to the first register in the sensor data header block. The current value for this sensor data register map version is 1.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|---------------------|------------------------------|--------------|--|
| 0001 | 1 | R1 | ushort | Sensor ID = 42 |
| 0002 | 2 | R1 | ulong | Sensor serial number (0 = no cap) |
| 0004 | 1 | R1 | 16 bits | Sensor status |
| 0005 | 3 | R1 | time | Last factory calibration (cap start time, 0 = no cap) |
| 0008 | 3 | R1 | time | Replace cap 0 = no cap 8 = Service period exceeded or moderate abrasion detected on cap |
| 0011 | 3 | R1 | time | Last user calibration |
| 0014 | 3 | R1/W2 | time | Next user calibration (0 = none required) |
| 0017 | 1 | R1 | ushort | Warm-up time in milliseconds = 2000 |
| 0018 | 1 | R1 | ushort | Fast sample rate = 1200 milliseconds |
| 0019 | 1 | R1 | ushort | Number of sensor parameters (N) = 4 |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|---------------------|------------------------------|--------------|--|
| 0020 | 1 | R1/W3 | ushort | Alarm/warning parameter number (1 – N, default = 1) |
| 0021 | 1 | R1/W3 | 16 bits | Alarm and warning enable bits (default = 0) Bit 0 = High alarm enabled Bit 1 = High warning enabled Bit 2 = Low warning enabled Bit 3 = Low alarm enabled Bit 4 = Sensor calibration warning |
| 0022 | 2 | R1/W3 | float | High alarm set value (default = 0.0) |
| 0024 | 2 | R1/W3 | float | High alarm clear value (default = 0.0) |
| 0026 | 2 | R1/W3 | float | High warning set value (default = 0.0) |
| 0028 | 2 | R1/W3 | float | High warning clear value (default = 0.0) |
| 0030 | 2 | R1/W3 | float | Low warning clear value (default = 0.0) |
| 0032 | 2 | R1/W3 | float | Low warning set value (default = 0.0) |
| 0034 | 2 | R1/W3 | float | Low alarm clear value (default = 0.0) |
| 0036 | 2 | R1/W3 | float | Low alarm set value (default = 0.0) |

4.2 Sensor Serial Number

This field returns the serial number of the installed sensor cap. The probe returns zero if the sensor cap is not installed or is expired.

4.3 Sensor Status

Bit 9 of the sensor status register is set to indicate when the sensor cap is not installed.

4.4 Last Factory Calibration

This field returns the start time of the installed sensor cap. If the cap has not been used, the cap manufactured time is returned. The probe returns zero if the sensor cap is not installed.

4.5 Next Factory Calibration

This field returns the expiration time of the installed sensor cap. This value is the lesser of the cap manufactured time plus 24 months, or the cap start time plus 12 months. The probe returns zero if the sensor cap is not installed.

4.6 Warm-Up Time

If automatic salinity correction is disabled, this register returns the RDO sensor warm-up time. If automatic salinity correction is enabled, this register returns the sum of the RDO sensor and conductivity sensor warm-up times.

4.7 Fast Sample Rate

If automatic salinity correction is disabled, this register returns the RDO sensor fast sample rate. If automatic salinity correction is enabled, this register returns the sum of the RDO sensor and conductivity sensor fast sample rates.

4.8 Parameter Registers—RDO

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|---|------------------|---------------------------|-----------|--|
| Dissolved Oxygen Concentration | | | | |
| Dissolved oxygen concentration is calculated from the oxygen partial pressure and temperature using settings contained in the sensor calibration registers. | | | | |
| If the parameter value is read when the sensor cap is not installed, the register returns the sentinel value and a Data Quality ID of 7 (sensor communication error). | | | | |
| 0038 | 2 | R1 | float | Measured value, C ₀ |
| 0040 | 1 | R1 | ushort | Parameter ID = 20 |
| 0041 | 1 | R1/W2 | ushort | Units ID 117 = mg/L (default) 118 = µg/L |
| 0042 | 1 | R1 | ushort | Data Quality ID |
| 0043 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0045 | 1 | R1 | 16 bits | Available Units = 0x0030 (48) |
| Temperature | | | | |
| 0046 | 2 | R1 | float | Measured value |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|---|---------------------|------------------------------|--------------|--|
| 0048 | 1 | R1 | ushort | Parameter ID = 1 |
| 0049 | 1 | R1/W2 | ushort | Units ID 1 = °C (default) 2 = °F |
| 0050 | 1 | R1 | ushort | Data Quality ID |
| 0051 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0053 | 1 | R1 | 16 bits | Available Units = 0x0003 (3) |
| Dissolved Oxygen %Saturation | | | | |
| The saturation parameter is calculated from dissolved oxygen concentration and settings contained in the sensor calibration registers. | | | | |
| If the parameter value is read when the sensor cap is not installed, the register returns the sentinel value and a Data Quality ID of 7 (sensor communication error). | | | | |
| 0054 | 2 | R1 | float | Measured value |
| 0056 | 1 | R1/W2 | ushort | Parameter ID = 21 |
| 0057 | 1 | R1/W2 | ushort | Units ID 177 = percent saturation (default) |
| 0058 | 1 | R1 | ushort | Data Quality ID |
| 0059 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0061 | 1 | R1 | 16 bits | Available Units = 0x0001 (1) |
| Oxygen Partial Pressure | | | | |
| Oxygen partial pressure is calculated using calibration information stored in the sensor cap. | | | | |
| If the parameter value is read when the sensor cap is not installed, the register returns the sentinel value and a Data Quality ID of 7 (sensor communication error). | | | | |
| 0062 | 2 | R1 | float | Measured value |
| 0064 | 1 | R1 | ushort | Parameter ID = 30 (oxygen partial pressure) |
| 0065 | 1 | R1/W2 | ushort | Units ID 26 = torr (default) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|---|
| 0066 | 1 | R1 | ushort | Data Quality ID |
| 0067 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0069 | 1 | R1 | 16 bits | Available Units = 0x0200 (512) |

4.9 Calibration Registers—RDO

Values in the calibration registers determine how the sensor parameters are calculated.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|---|
| 0118 | 2 | R1/W3 | float | Live salinity value (PSU) |
| 0120 | 2 | R1/W3 | float | Default salinity value (PSU, default = 0.0) |
| 0122 | 2 | R1/W3 | float | Live barometric pressure (P_B , mbar) |
| 0124 | 2 | R1/W3 | float | Default barometric pressure (mbar, default = 1013.25) |
| 0126 | 2 | R1/W3 | float | 100% saturation calibration reading (mg/L) |
| 0128 | 2 | R1/W3 | float | 100% saturation temperature reading (°C) |
| 0130 | 2 | R1/W3 | float | 100% saturation salinity value (PSU) |
| 0132 | 2 | R1/W3 | float | 100% saturation barometric pressure (mbar) |
| 0134 | 2 | R1/W3 | float | 0% saturation calibration reading (mg/L) |
| 0136 | 2 | R1/W3 | float | 0% saturation temperature reading (°C) |
| 0138 | 2 | R1/W3 | float | Calibration slope (default = 1.0) |
| 0140 | 2 | R1/W3 | float | Calibration offset (default = 0.0) |

4.10 Live Salinity Value

The live salinity value is used to correct the oxygen concentration value for salinity when automatic salinity correction is disabled. Values must be written in Practical Salinity Units (PSU) in the range 0 to 42 PSU. This is not a measured parameter. If automatic salinity correction is enabled, the salinity value is obtained from the conductivity sensor and the live salinity will have no effect.

4.11 Default Salinity Value

The default salinity value is loaded into the live salinity value register when power is applied to the probe or when the default value is changed. It is used in calculations until a live salinity value is written. This is not a measured parameter. If automatic salinity correction is enabled, the salinity value is obtained from the conductivity sensor and the default salinity will have no effect.

4.12 Live Barometric Pressure

The live barometric pressure is used in the calculation of percent saturation and to determine the theoretical saturation point during calibration. Values must be written in millibars in the range 506.625 to 1114.675 mbar. This is not a measured parameter.

4.13 Default Barometric Pressure

The default barometric pressure is loaded into the live barometric pressure register when power is applied to the probe or when the default value is changed. It is used in calculations until a live barometric pressure is written. This is not a measured parameter.

4.14 100% Saturation Calibration Values

These values represent the sensor conditions while the probe is in a 100% saturation calibration environment. These are not measured values, they are written by the controller during the calibration process.

Writes to these registers are only accepted if the probe is in the calibration mode. The probe will return exception 0x92 (invalid sensor mode) if an attempt is made to write these registers when the calibration mode is off.

4.15 Zero% Saturation Calibration Values

These values represent the sensor conditions while the probe is in a 0 % saturation calibration environment. These are not measured values, they are written by the controller during the calibration process.

Writes to these registers are only accepted if the probe is in the calibration mode. The probe will return exception 0x92 (invalid sensor) if an attempt is made to write these registers when the calibration mode is off.

4.16 Calibration Slope and Offset

These values represent the slope and offset that will be applied to the raw concentration reading from the sensor to generate the final values reported by the sensor parameters. Writes to these registers are only accepted if the probe is in the calibration mode. The probe will return exception 0x92 (invalid sensor) if an attempt is made to write these registers when the calibration mode is off. These registers may be written independently of the normal internal calibration procedure.

4.17 Calibration Procedure—RDO

The RDO sensor is calibrated using the following procedure.

1. Write the Calibration Mode On command (0xE000) to the Sensor Command register.
2. Update the live salinity and barometric pressure registers if necessary.
3. Prompt the user to place the probe in a 100% saturation environment.
4. Read the oxygen concentration and temperature parameters. When these values have reached equilibrium, record them in their respective 100% saturation calibration registers. Write the current live salinity and barometric pressure readings to their respective calibration registers.
5. Prompt the user to place the probe in a 0% saturation environment. When these registers have reached equilibrium, record them in their respective 0% saturation calibration registers. If a zero calibration is not to be performed, these registers can be set to zero or left at their previous values.
6. Write the Calibration Update command (0xE001) to the Sensor Command register. The sensor will calculate a new slope and offset, will write the current time to the Last User Calibration Time register, and set the Next User Calibration Time register to zero (disabled). If the concentrations at 100% and 0% saturation are equal, the probe will return an exception response with code 0x97 (invalid calibration) and not attempt to compute a new slope and offset due to possible division by zero. If the slope does not calculate between 0.85 and 1.20 inclusive, or the offset does not calculate between -0.2 and +0.2 inclusive, the probe will return an exception response with code 0x97 (invalid calibration). The slope and offset will be available for read but will not be committed to flash memory.
7. Optionally, read the Last User Calibration Time register, add the next calibration interval, and write the result to the Next User Calibration Time register.
8. Write the Calibration Mode Off command (0xE002) to the sensor command register to place the sensor in normal operation. If the calibration mode is turned off without a calibration update command, or the calibration command returned an exception, the previous calibration shall be restored.

4.17.1 RDO Calibration Calculations

Calibrated Oxygen reading:

$$O_{2RC} = c_0 + c_1 \times O_{2RU}$$

where:

$$c_1 = \frac{O_2 100\% Sat}{O_{2RUS} - O_{2RUZ}}$$

$$c_0 = -c_1 \times O_{2RUZ}$$

where:

$O_2 100\% Sat$ is the theoretical 100% saturation point.

O_{2RUS} is the un-calibrated reading at 100% saturation.

O_{2RUZ} is the un-calibrated reading at 0% saturation.

References:

Standard Methods:

4500-0 C. Azide Modification

5. Modbus Registers—Conductivity Sensor

5.1 Header Registers—Conductivity

The sensor map data register offset points to the first register in the sensor data header block. The current value for this sensor data register map version is 1.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|---------------------|------------------------------|--------------|--|
| 0501 | 1 | R1 | ushort | Sensor ID = 35 |
| 0502 | 2 | R1 | ulong | Sensor serial number (same as probe) |
| 0504 | 1 | R1 | 16 bits | Sensor status |
| 0505 | 3 | R1/W4 | time | Last factory calibration |
| 0508 | 3 | R1/W4 | time | Next factory calibration (0 = none required) |
| 0511 | 3 | R1 | time | Last user calibration |
| 0514 | 3 | R1/W2 | time | Next user calibration (0 = none required) |
| 0517 | 1 | R1 | ushort | Warm-up time in milliseconds = 1200 |
| 0518 | 1 | R1 | ushort | Fast sample rate = 1200 milliseconds |
| 0519 | 1 | R1 | ushort | Number of sensor parameters (N) = 7 |
| 0520 | 1 | R1/W3 | ushort | Alarm/warning parameter number (1 – N, default = 1) |
| 0521 | 1 | R1/W3 | 16 bits | Alarm and warning enable bits (default = 0) Bit 0 = High alarm enabled Bit 1 = High warning enabled Bit 2 = Low warning enabled Bit 3 = Low alarm enabled Bit 4 = Sensor calibration warning |
| 0522 | 2 | R1/W3 | float | High alarm set value (default = 0.0) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|-----------------|-----------------------------|--|----------------------|---|
| 0524 | 2 | R1/W3 | float | High alarm clear value (default = 0.0) |
| 0526 | 2 | R1/W3 | float | High warning set value (default = 0.0) |
| 0528 | 2 | R1/W3 | float | High warning clear value (default = 0.0) |
| 0530 | 2 | R1/W3 | float | Low warning clear value (default = 0.0) |
| 0532 | 2 | R1/W3 | float | Low warning set value (default = 0.0) |
| 0534 | 2 | R1/W3 | float | Low alarm clear value (default = 0.0) |
| 0536 | 2 | R1/W3 | float | Low alarm set value (default = 0.0) |

5.2 Parameter Registers—Conductivity

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------------------------|---------------------|------------------------------|--------------|--|
| Actual Conductivity | | | | |
| 0538 | 2 | R1 | float | Measured value, AC |
| 0540 | 1 | R1 | ushort | Parameter ID = 9 (actual conductivity) |
| 0541 | 1 | R1/W2 | ushort | Units ID 65 = microsiemens per centimeter (default) 66 = millisiemens per centimeter |
| 0542 | 1 | R1 | ushort | Data Quality ID |
| 0543 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0545 | 1 | R1 | 16 bits | Available Units = 0x0003 (3) |
| Temperature | | | | |
| 0546 | 2 | R1 | float | Measured value |
| 0548 | 1 | R1 | ushort | Parameter ID = 1 (temperature) |
| 0549 | 1 | R1/W2 | ushort | Units ID 1 = °C (default) 2 = °F |
| 0550 | 1 | R1 | ushort | Data Quality ID |
| 0551 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0553 | 1 | R1 | 16 bits | Available Units = 0x0003 (3) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|--|---------------------|------------------------------|--------------|--|
| <p>Specific Conductivity</p> <p>The default units for specific conductivity are uS/cm. Conversion to other units is as follows. $mS/cm = uS/cm / 1000$</p> <p>Specific conductivity is calculated from actual conductivity and temperature using the following equation.</p> $SC = AC * (\beta_0 + \beta_1 T + \beta_2 T^2 + \dots + \beta_7 T^7) / (1 + \alpha (T - T_{ref}))$ <p>Where T_{ref}, α, and β_{0-7} are specified in the sensor calibration registers. The factory default coefficients calculate specific conductivity per Standard Methods 2510B.</p> | | | | |
| 0554 | 2 | R1 | float | Measured value, SC |
| 0556 | 1 | R1 | ushort | Parameter ID = 10 (specific conductivity) |
| 0557 | 1 | R1/W2 | ushort | Units ID 65 = microsiemens per centimeter (default) 66 = millisiemens per centimeter |
| 0558 | 1 | R1 | ushort | Data Quality ID |
| 0559 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0561 | 1 | R1 | 16 bits | Available Units = 0x0003 (3) |
| <p>Salinity</p> <p>Salinity is calculated using actual conductivity and temperature.</p> | | | | |
| 0562 | 2 | R1 | float | Measured value, S |
| 0564 | 1 | R1 | ushort | Parameter ID = 12 (salinity) |
| 0565 | 1 | R1/W2 | ushort | Units ID 97 = Practical Salinity Units PSU (default) |
| 0566 | 1 | R1 | ushort | Data Quality ID |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|--|------------------|---------------------------|-----------|---|
| 0567 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0569 | 1 | R1 | 16 bits | Available Units = 0x0001 (1) |
| <p>Total Dissolved Solids</p> <p>The default units for TDS are ppt. Conversion to other units is as follows.</p> $\text{ppm} = \text{ppt} * 1000$ <p>TDS is calculated from specific conductivity using the following equation.</p> $\text{TDS} = \text{CF}_{\text{TDS}} * \text{SC}$ <p>Where CF_{TDS} is the TDS conversion factor in ppm units as specified in the sensor calibration registers. The default conversion factor is 0.65.</p> | | | | |
| 0570 | 2 | R1 | float | Measured value, TDS |
| 0572 | 1 | R1 | ushort | Parameter ID = 13 (TDS) |
| 0573 | 1 | R1/W2 | ushort | Units ID 113 = parts per million 114 = parts per thousand (default) |
| 0574 | 1 | R1 | ushort | Data Quality ID |
| 0575 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0577 | 1 | R1 | 16 bits | Available Units = 0x0003 (3) |
| <p>Resistivity</p> <p>Resistivity is calculated from actual conductivity using the following equation.</p> $R (\text{ohm-cm}) = 10^6 / \text{AC} (\text{uS/cm})$ <p>If the actual conductivity is zero, R is reported as 10,000,000 ohm-cm.</p> | | | | |
| 0578 | 2 | R1 | float | Measured value, R |
| 0580 | 1 | R1 | ushort | Parameter ID = 11 (resistivity) |
| 0581 | 1 | R1/W2 | ushort | Units ID 81 = ohm-cm (default) |
| 0582 | 1 | R1 | ushort | Data Quality ID |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|--|------------------|---------------------------|-----------|---|
| 0583 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0585 | 1 | R1 | 16 bits | Available Units = 0x0001 (1) |
| Density of Water Density of water is calculated from salinity and temperature. | | | | |
| 0586 | 2 | R1 | float | Measured value, p |
| 0588 | 1 | R1 | ushort | Parameter ID = 14 (density of water) |
| 0589 | 1 | R1/W2 | ushort | Units ID 129 = g/cm ³ (default) |
| 0590 | 1 | R1 | ushort | Data Quality ID |
| 0591 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 0593 | 1 | R1 | 16 bits | Available Units = 0x0001 (1) |

5.3 Calibration Registers—Conductivity

Values in the calibration registers determine how sensor parameters are calculated.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|---------------------|------------------------------|--------------|--|
| 0628 | 2 | R1/W3 | float | Cell Offset, K_0 (default = 0.0) |
| 0630 | 2 | R1/W3 | float | Cell Constant, K (default = 1.0) |
| 0632 | 2 | R1/W3 | float | Reference Temperature, T_{ref} in °C (default = 25) |
| 0634 | 2 | R1/W3 | float | Alpha Coefficient α (default = 0.0191) |
| 0636 | 2 | R1/W3 | float | Beta Coefficient β_0 (default = 1.0) |
| 0638 | 2 | R1/W3 | float | Beta Coefficient β_1 (default = 0.0) |
| 0640 | 2 | R1/W3 | float | Beta Coefficient β_2 (default = 0.0) |
| 0642 | 2 | R1/W3 | float | Beta Coefficient β_3 (default = 0.0) |
| 0644 | 2 | R1/W3 | float | Beta Coefficient β_4 (default = 0.0) |
| 0646 | 2 | R1/W3 | float | Beta Coefficient β_5 (default = 0.0) |
| 0648 | 2 | R1/W3 | float | Beta Coefficient β_6 (default = 0.0) |
| 0650 | 2 | R1/W3 | float | Beta Coefficient β_7 (default = 0.0) |
| 0652 | 2 | R1/W3 | float | TDS Conversion Factor CF_{TDS} in ppm (software limited range 0.001 to 10.000, default = 0.65, resolution 0.001) |

5.3.1 Cell Offset and Cell Constant

These values are used to calibrate conductivity to user standards. These registers shall only be able to be written when the sensor is in the calibration mode. The probe will return exception 0x92 (invalid sensor mode) if an attempt is made to write these registers when the calibration mode is off.

Actual conductivity (AC) is calculated as follows.

$$AC = K_0 + K * AC_f$$

Where AC_f is the actual conductivity value computed using the factory calibrated cell constant. For a single point calibration, K_0 is set to zero.

5.4 Calibration Procedure—Conductivity

1. Write the Calibration Mode On command (0xE000) to the Sensor Command register. Reading the actual conductivity parameter in calibration mode shall present actual conductivity as AC_f (the current cell offset and cell constant shall not be applied).
2. Instruct the user to place the conductivity sensor into one or more conductivity standards.
3. Read the actual conductivity and temperature parameters at each conductivity standard.
4. Calculate new values for the cell offset K_0 and cell constant K and write these values to their corresponding registers.
5. Write the Calibration Update command (0xE001) to the Sensor Command register. The sensor sets the Last User Calibration Date to the current date and sets the Next User Calibration Date to zero (none required).
6. Optionally, read the Last User Calibration Time register, add the next calibration interval, and write the result to the Next User Calibration Time register.
7. Write the Calibration Off command (0xE002) to the sensor command register to place the sensor in normal operation.

6. Modbus Registers—Level Sensor

6.1 Header Registers

The sensor map data register offset points to the first register in the sensor data header block. The current value for this sensor data register map version is 1.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|---|
| 1001 | 1 | R1/W4 | ushort | Sensor ID 34 = 30 foot full-scale level, absolute pressure 32 = 100 foot full-scale level, absolute pressure 33 = 250 foot full-scale level, absolute pressure |
| 1002 | 2 | R1/W4 | ulong | Sensor serial number |
| 1004 | 1 | R1 | 16 bits | Sensor status |
| 1005 | 3 | R1/W4 | time | Last factory calibration |
| 1008 | 3 | R1/W4 | time | Next factory calibration (0 = none required) |
| 1011 | 3 | R1 | time | Last user calibration |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|--|
| 1014 | 3 | R1/W2 | time | Next user calibration (0 = none required) |
| 1017 | 1 | R1 | ushort | Warm-up time in milliseconds = 800 |
| 1018 | 1 | R1 | ushort | Fast sample rate = 800 milliseconds |
| 1019 | 1 | R1 | ushort | Number of sensor parameters (N) = 3 |
| 1020 | 1 | R1/W3 | ushort | Alarm/warning parameter number (1 – N, default = 1) |
| 1021 | 1 | R1/W3 | 16 bits | Alarm and warning enable bits (default = 0) Bit 0 = High alarm enabled Bit 1 = High warning enabled Bit 2 = Low warning enabled Bit 3 = Low alarm enabled Bit 4 = Sensor calibration warning |
| 1022 | 2 | R1/W3 | float | High alarm set value (default = 0.0) |
| 1024 | 2 | R1/W3 | float | High alarm clear value (default = 0.0) |
| 1026 | 2 | R1/W3 | float | High warning set value (default = 0.0) |
| 1028 | 2 | R1/W3 | float | High warning clear value (default = 0.0) |
| 1030 | 2 | R1/W3 | float | Low warning clear value (default = 0.0) |
| 1032 | 2 | R1/W3 | float | Low warning set value (default = 0.0) |
| 1034 | 2 | R1/W3 | float | Low alarm clear value (default = 0.0) |
| 1036 | 2 | R1/W3 | float | Low alarm set value (default = 0.0) |

6.1.1 Warm-up Time

If automatic density correction is disabled, this register returns the level sensor warm-up time as shown. If automatic density correction is enabled, this register returns the sum of the level sensor and conductivity sensor warm-up times.

6.1.2 Fast Sample Rate

If automatic density correction is disabled, this register returns the level sensor fast sample rate as shown. If automatic density correction is enabled, this register shall return the sum of the level sensor and conductivity sensor fast sample rates.

6.2 Parameter Registers—Level

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|--------------------|------------------|---------------------------|-----------|--|
| Pressure | | | | |
| 1038 | 2 | R1 | float | Measured value, P_m |
| 1040 | 1 | R1 | ushort | Parameter ID = 2 (pressure) |
| 1041 | 1 | R1/W2 | ushort | Units ID 17 = PSI (default) 19 = KPa 20 = bar 21 = mbar 22 = mmHg |
| 1042 | 1 | R1 | ushort | Data Quality ID |
| 1043 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 1045 | 1 | R1 | 16 bits | Available Units = 0x003D (61) |
| Temperature | | | | |
| 1046 | 2 | R1 | float | Measured value |
| 1048 | 1 | R1 | ushort | Parameter ID = 1 (temperature) |
| 1049 | 1 | R1/W2 | ushort | Units ID 1 = °C (default) 2 = °F |
| 1050 | 1 | R1 | ushort | Data Quality ID |
| 1051 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 1053 | 1 | R1 | 16 bits | Available Units = 0x0003 (3) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|--------------|------------------|---------------------------|-----------|---|
| Level | | | | |
| 1054 | 2 | R1 | float | Measured value, L_m |
| 1056 | 1 | R1/W2 | ushort | Parameter ID 3 = level, depth (default) 4 = level, top of casing 5 = level, elevation |
| 1057 | 1 | R1/W2 | ushort | Units ID 33 = millimeters 34 = centimeters 35 = meters 37 = inches 38 = feet (default) |
| 1058 | 1 | R1 | ushort | Data Quality ID |
| 1059 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 1061 | 1 | R1 | 16 bits | Available Units = 0x0037 (55) |

6.3 Calibration Registers—Level

Values in the calibration registers determine how sensor parameters are calculated.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|---|
| 1100 | 2 | R1/W3 | float | Specific Gravity (default = 1.0) |
| 1102 | 2 | R1/W3 | float | Pressure Offset, P_O (default = 0.0) |
| 1104 | 2 | R1/W3 | float | Level Reference, L_R (default = 0.0) |
| 1106 | 2 | R1/W3 | float | Pressure Reference, P_R (default = 0.0) |

6.3.1 Pressure Calculation

If automatic barometric pressure correction is disabled, the raw pressure reading (P) is simply the pressure reading of the sensor (P_S).

$$P = P_S$$

If automatic barometric pressure correction is enabled, the live barometric pressure value (P_B , see RDO sensor) is subtracted from the pressure sensor reading (P_S) to generate the raw pressure reading.

$$P = P_S - P_B$$

6.3.2 Pressure Offset

The pressure offset (P_O) is subtracted from the raw pressure reading (P) to correct for offset errors in the pressure sensor. This register shall only be able to be written when the sensor is in the calibration mode. The probe will return exception 0x92 (invalid sensor mode) if an attempt is made to write these registers when the calibration mode is off. Master software must ensure that the pressure offset is written in the currently selected units. This value shall be converted to the appropriate units when the Pressure Units ID register is written. The measured pressure is presented as:

$$P_M = P - P_O$$

6.3.3 Specific Gravity

If automatic density correction is disabled, pressures (PSI) are converted to depth (meters) accounting for the specific gravity (SG) of the fluid according to the following equation. Values shall be in the range 0.1 to 10.0 inclusive.

$$D(P) = (P * 0.70307) / SG$$

If automatic density correction is enabled, pressures (PSI) are converted to depth (meters) accounting for the measured fluid density (ρ) according to the following equation. The value contained in the specific gravity register is ignored.

$$D(P) = (P * 0.70307) / \rho$$

6.3.4 Level Reference

This value is used to reference a level reading to an independently established value. This register shall only be able to be written when the sensor is in the calibration mode. The probe will return exception 0x92 (invalid sensor mode) if an attempt is made to write these registers when the calibration mode is off. Master software must ensure that the level value is written in the currently selected units. When this register is written, the device shall measure and record the current pressure reading P_M as reference pressure P_R in the currently selected pressure units. It shall also calculate and record the current depth $D_R = D(P_R)$ in the currently selected level units. L_R and D_R shall be converted to the appropriate units when the Level Units ID register is written. The following equations shall be used to calculate level based on the level parameter ID selection.

| Parameter | Description | Equation |
|-----------|----------------------|------------------------------|
| 3 | Level, depth | $L_M = D(P_M)$ |
| 4 | Level, top of casing | $L_M = L_R - (D(P_M) - D_R)$ |
| 5 | Level, elevation | $L_M = L_R + (D(P_M) - D_R)$ |

6.3.5 Pressure Reference

Master software may optionally overwrite the pressure reference value recorded by the device when the Level Reference register is written. This register can be written only when the sensor is in the calibration mode. The probe will return exception 0x92 (invalid sensor mode) if an attempt is made to write these registers when the calibration mode is off. Master software must ensure that the pressure value is written in the currently selected units. This value shall be converted to the appropriate units when the Pressure Units ID register is written.

6.4 Calibration Procedure—Pressure

1. Write the Calibration On command (0xE000) to the Sensor Command register. Reading the pressure parameter in the calibration mode shall present the raw pressure (P).
2. Automatic barometric pressure compensation must be set to the desired mode.
3. Instruct the user to place the pressure sensor in open air.
4. Read the pressure parameter P.
5. Write P as the new value of P_O to the pressure offset register.
6. Write the Calibration Update command (0xE001) to the Sensor Command register. The sensor sets the last user calibration date to the current date and sets the next user calibration date to zero (none required).
7. Optionally, read the last user calibration time, add the next calibration interval, and write the result to the next user calibration time register.
8. Write the Calibration Off command (0xE002) to the Sensor Command register to place the sensor in normal operation.

6.5 Calibration Procedure—Level

1. Write the Calibration On command (0xE000) to the sensor command register.
2. Instruct the user to deploy the probe and to independently determine the reference level (L_R).
3. Reading the level parameter in the calibration mode shall present depth using the current level configuration settings to assist the user in properly deploying the device.
4. Write the new value of L_R to the Level Reference register.
5. The sensor measures the current pressure P_M (with offset correction) and stores it as P_R .
6. Optionally, the master device can overwrite the pressure reference register with a calculated value.
7. Write the Calibration Update command (0xE001) to the Sensor Command register. The sensor sets the Last User Calibration Date to the current date and sets the Next User Calibration Date to zero (none required).

8. Optionally, read the Last User Calibration Time, add the next calibration interval, and write the result to the Next User Calibration Time register.
9. Write the Calibration Off command (0xE002) to the Sensor Command register to place the sensor in normal operation.

7. Modbus Registers – pH/ORP Sensor

The Sensor Map Data Register Offset points to the first register in the sensor data header block. The current value for this sensor data register map version is 1.

The pH/ORP sensor registers remain valid even if the sensor is not installed.

7.1 Header Registers

The sensor map data register offset points to the first register in the sensor data header block.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|--|
| 1501 | 1 | R1/W4 | ushort | Sensor ID 0 = not installed 27 = pH/ORP |
| 1502 | 2 | R1/W4 | ulong | Sensor serial number (0 – 65535) |
| 1504 | 1 | R1 | 16 bits | Sensor status |
| 1505 | 3 | R1/W4 | time | Last factory calibration |
| 1508 | 3 | R1 | time | Next factory calibration (0 = none required) |
| 1511 | 3 | R1 | time | Last user calibration |
| 1514 | 3 | R1/W2 | time | Next user calibration (0 = none required) |
| 1517 | 1 | R1 | ushort | Warm-up time = 1300 milliseconds |
| 1518 | 1 | R1 | ushort | Fast sample rate = 1300 milliseconds |
| 1519 | 1 | R1 | ushort | Number of sensor parameters (N = 3) |
| 1520 | 1 | R1/W3 | ushort | Alarm/warning parameter number (1 – N, default = 1) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|---|
| 1521 | 1 | R1/W3 | 16 bits | Alarm and warning enable bits (default = 0) Bit 0 = High alarm enabled Bit 1 = High warning enabled Bit 2 = Low warning enabled Bit 3 = Low alarm enabled Bit 4 = Sensor calibration warning |
| 1522 | 2 | R1/W3 | float | High alarm set value (default = 0.0) |
| 1524 | 2 | R1/W3 | float | High alarm clear value (default = 0.0) |
| 1526 | 2 | R1/W3 | float | High warning set value (default = 0.0) |
| 1528 | 2 | R1/W3 | float | High warning clear value (default = 0.0) |
| 1530 | 2 | R1/W3 | float | Low warning clear value (default = 0.0) |
| 1532 | 2 | R1/W3 | float | Low warning set value (default = 0.0) |
| 1534 | 2 | R1/W3 | float | Low alarm clear value (default = 0.0) |
| 1536 | 2 | R1/W3 | float | Low alarm set value (default = 0.0) |

7.2 Parameter Registers

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|---|------------------|---------------------------|-----------|---|
| pH | | | | |
| If the parameter value is read when the sensor is not plugged in, it returns the sentinel value and a Data Quality ID of 7 (sensor communication error). | | | | |
| If the parameter value calculates to a value outside of the range 0 to 14 pH, it shall return the sentinel value and a Data Quality ID of 3 (error). | | | | |
| 1538 | 2 | R1 | float | Measured value |
| 1540 | 1 | R1 | ushort | Parameter ID = 17 (pH) |
| 1541 | 1 | R1/W2 | ushort | Units ID 145 = pH (default) |
| 1542 | 1 | R1 | ushort | Data Quality ID |
| 1543 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 1545 | 1 | R1 | 16 bits | Available Units = 0x0001 (1) |
| pH/mV | | | | |
| If the parameter value is read when the sensor is not plugged in, it shall return the sentinel value and a Data Quality ID of 7 (sensor communication error). | | | | |
| 1546 | 2 | R1 | float | Measured value |
| 1548 | 1 | R1 | ushort | Parameter ID = 18 (pH mV) |
| 1549 | 1 | R1/W2 | ushort | Units ID 162 = mV (default) |
| 1550 | 1 | R1 | ushort | Data Quality ID |
| 1551 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 1553 | 1 | R1 | 16 bits | Available Units = 0x0002 (2) |
| ORP/mV | | | | |
| If the parameter value is read when the sensor is not plugged in, it shall return the sentinel value and a Data Quality ID of 7 (sensor communication error). | | | | |
| 1554 | 2 | R1 | float | Measured value |
| 4556 | 1 | R1 | ushort | Parameter ID = 19 (ORP) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|---|
| 4557 | 1 | R1/W2 | ushort | Units ID 162 = mV (default) |
| 4558 | 1 | R1 | ushort | Data Quality ID |
| 4559 | 2 | R1/W3 | float | Off line sentinel value (default = 0.0) |
| 4561 | 1 | R1 | 16 bits | Available Units = 0x0002 (2) |

7.3 Calibration Registers—pH/ORP

These registers can be written only when the sensor is in the calibration mode. The probe will return exception 0x92 (invalid sensor mode) if an attempt is made to write these registers when the calibration mode is off. pH solution data must be written into the registers in order of increasing pH.

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|------------------|---------------------------|-----------|--|
| 1618 | 1 | R1/W3 | ushort | Number of pH calibration points 1, 2, or 3 (default = 1) |
| 1619 | 2 | R1/W3 | float | pH value of solution 1 (default = 0) |
| 1621 | 2 | R1/W3 | float | mV response in solution 1 (default = 0) |
| 1623 | 2 | R1/W3 | float | Temperature of solution 1, T_1 in °C (default = 25) |
| 1625 | 2 | R1/W3 | float | pH value of solution 2 (default = 0) |
| 1627 | 2 | R1/W3 | float | mV response in solution 2, mV_2 (default = 0) |
| 1629 | 2 | R1/W3 | float | Temperature of solution 2, T_2 in °C (default = 25) |
| 1631 | 2 | R1/W3 | float | pH value of solution 3 (default = 0) |
| 1633 | 2 | R1/W3 | float | mV response in solution 3 (default = 0) |
| 1635 | 2 | R1/W3 | float | Temperature of solution 3, T_3 in °C (default = 25) |
| 1637 | 2 | R1/W3 | float | Slope, m_1 in pH/mV (default = -0.016903313) |
| 1639 | 2 | R1/W3 | float | Offset, b_1 in pH (default = 7.0) |

| Register | Size (registers) | Mode & Access Level (R/W) | Data Type | Description |
|----------|---------------------|------------------------------|--------------|---|
| 1641 | 2 | R1/W3 | float | Slope, m_2 in pH/mV (default = -0.016903313) |
| 1643 | 2 | R1/W3 | float | Offset, b_2 in pH (default = 7.0) |
| 1645 | 1 | R1/W3 | ushort | ORP solution type (default = 0) |
| 1646 | 2 | R1/W3 | float | mV value of solution (default = 0) |
| 1648 | 2 | R1/W3 | float | mV response in solution (default = 0) |
| 1650 | 2 | R1/W3 | float | Temperature of solution, T_{orp} in °C (default = 25) |
| 1652 | 2 | R1/W3 | float | Slope, m_{orp} in mV/mV (default = 1.0) |
| 1654 | 2 | R1/W3 | float | Offset, b_{orp} in mV (default = 0.0, limit = +/-1000) |

7.3.1 pH Measured Value

The measured value is derived from either one line segment (created from a one-point or two-point calibration) or two line segments (created from a three-point calibration).

For one line segment:

$$pH = (T_{1K} / T_K) \times m_1 \times mV + b_1$$

For two line segments:

$$mV > mV_2: pH = (T_{1K} / T_K) \times m_1 \times mV + b_1$$

$$mV \leq mV_2: pH = (T_{2K} / T_K) \times m_2 \times mV + b_2$$

Where:

mV is the measured sensor millivolts

T_K is the measured temperature in °K

T_{1K} is the calibration temperature converted to °K. ($T_{1K} = T_1 + 273.15$)

T_{2K} is the calibration temperature converted to °K. ($T_{2K} = T_2 + 273.15$)

m_1 and b_1 are slope and offset for segment one respectively

m_2 and b_2 are slope and offset for segment two respectively

pH is the calculated result

7.3.2 ORP Measured Value

The measured value is derived as follows:

$$\text{ORP} = m_{\text{orp}} \times \text{mV} + b_{\text{orp}}$$

Where:

mV is the measured sensor millivolts

m_{orp} and b_{orp} are slope and offset respectively

ORP is the calculated value

7.4 Calibration Procedure—pH

1. Write the Calibration Mode On command (0xE000) to the Sensor Command register.
2. Place the pH sensor into one or more pH standards.
3. Read the pH mV and temperature parameters at each pH standard.
4. Calculate new pH calibration values and write these values to their corresponding registers.
5. Write the Calibration Update command (0xE001) to the Sensor Command register. The sensor sets the Last User Calibration Date to the current date and sets the Next User Calibration Date to zero (none required).
6. Optionally, read the Last User Calibration Time register, add the next calibration interval, and write the result to the Next User Calibration Time register.
7. Write the Calibration Off command (0xE002) to the Sensor Command register to place the sensor in normal operation.

7.5 Calibration Procedure—ORP

1. Write the Calibration Mode On command (0xE000) to the Sensor Command register.
2. Place the ORP sensor into a standard.
3. Read the ORP mV and temperature parameters of the standard. Reading the ORP parameter in the calibration will return raw ORP millivolts, without the current calibration applied.
4. Calculate new ORP calibration values and write these values to their corresponding registers.
5. Write the Calibration Update command (0xE001) to the Sensor Command register. The sensor sets the last user calibration date to the current date and sets the Next User Calibration Date to zero (none required).
6. Optionally, read the Last User Calibration Time register, add the next calibration interval, and write the result to the Next User Calibration Time register.
7. Write the Calibration Off command (0xE002) to the Sensor Command register to place the sensor in normal operation.

7.5.1 Sensor Command Timeouts

Both the Sensor Cal Defaults command (0xE003) and the Restore Sensor Defaults command (0xE004) require a response timeout of 1000 milliseconds.

8. Appendix—Modbus Additional Information

The Multi-PRO 400 Instrument supports Modbus as its primary communication protocol. This appendix contains a brief Modbus tutorial intended to accelerate learning for a person who is not familiar with the protocol. This document is not an official protocol document. More information about Modbus, including protocol specifications, can be downloaded from www.modbus.org.

The remaining sections of this document provide the information required to program a PLC/DCS and/or SCADA system to poll data. The user is expected to have a working knowledge of Modbus.

8.1 Modbus Tutorial

Modbus is a lightweight communication protocol developed in the late 1970's by Modicon as a digital communication protocol for its PLC's. The protocol requires very little code space and processing power to implement and has become a worldwide favorite for embedded devices. Modbus is royalty free and the specifications and standards can be downloaded from the web.

Modbus is a transport protocol. What this means is that Modbus does not have any protocol features that indicate what types of data are being transported in a message packet. This is similar to TCP/IP, the internet protocol standard. If we use TCP/IP as an example, consider that when a person connects to a web site, the primary data content being moved via TCP/IP is HTML. When a person downloads an instrument manual from an FTP site, typically the data content being moved is an Adobe PDF file. In both cases, TCP/IP is used as the protocol. The TCP/IP protocol simply ensures all the bytes (whatever they represent) are transferred from point A to point B correctly. Modbus is similar in concept. It provides a transport mechanism to move data safely over a communication link from a device to a computer.

Modbus can be used on a wide variety of communication links. In most applications, the protocol is used over an RS485 or RS232 link. This is because these types of communication links are inexpensive and efficient, perfectly suited for communication with embedded devices. Modbus can also be used over wireless radios, satellites, TCP/IP (Ethernet, token ring, etc.) and any other protocol-independent communication link.

Modbus is designed to be a Master/Slave protocol. This means that the protocol assumes that a single Master computer will initiate control and commands to the slave devices. The slave devices do not send any data on the communication link unless specifically asked for that data by the Master.

8.2 Modbus Modes

Modbus message packets can be formatted in three ways, RTU, ASCII and IP.

- RTU is the format of choice for hard connected serial connections such as RS485 or RS232 because it is the most compact and therefore the most efficient.

- ASCII format is required for any kind of wireless serial communications because it eliminates the message timing requirements needed for RTU mode. Message timing can be erratic over a wireless link.
- IP formatted messages can be used when the messages are transported using a secondary transport protocol such as TCP/IP. In this case the secondary transport protocol ensures that all of the packet bytes are transported correctly. Additionally, this format provides for packet sequence numbering because the secondary transport layer eliminates the synchronous Master/Slave nature of the serial RTU/ASCII formats.

8.3 Protocol Overview

Modbus provides message structures to read and write data to/from a device. The protocol also provides for the extension of the protocol to permit customization of the message structures. The instrument manufacturer adheres to the standard read/write message structures in its implementations. Modbus does not provide as part of the standard, a suitable file transfer message structure. For this function, the manufacturer has used a protocol extension to satisfy the downloading of data files from the device.

8.4 General Message Formats

The general message format consists of a device address, a function code, a data payload and a message validity checksum. The message format for each of the three modes changes as described below.

8.5 RTU Message Format

The RTU message format allows the transmission of bytes of data encompassing the full range of values per byte 0-255. There are no characters indicating the start or end of the packet. The end of packet is signaled by a time delay equivalent to a 3 byte transfer time on the communication link without any data transmitted.

8.5.1 Computer (Master) Message Format

| | | | |
|----------------|---------------|--------------|-----|
| Device Address | Function Code | Data Payload | CRC |
|----------------|---------------|--------------|-----|

- Device Address: 1 byte field with a value ranging from 1 to 247. Broadcast address is 0.
- Function Code: 1 byte field with a value range 1-127 representing the standard or extended function code. The function code tells the device what operation is to take place (i.e. Read/Write).
- Data Payload: 0-N bytes with information required to complete the requested function code operation.
- CRC: 2 bytes with a value computed mathematically from the message bytes. This value can be computed on both sides of the link and verified to ensure that the bits of the message were transmitted and received correctly.

8.5.2 Device (Slave) Response Format

| | | | |
|----------------|---------------|--------------|-----|
| Device Address | Function Code | Data Payload | CRC |
|----------------|---------------|--------------|-----|

- Device Address: Echo of device address sent in the message to the device. A broadcast message will not generate a response.
- Function Code: Echo of the function code sent to the device in the message packet. If an error occurs, the top bit of the byte is set and the data payload is the 1 byte error code from the device.
- Data Payload: 0-N bytes with response data from the device. For an error response, the payload will be a 1 byte value 1-255.
- CRC: 2 bytes with a value computed mathematically from the message bytes. This value can be computed on both sides of the link and verified to ensure that the bits of the message were transmitted and received correctly.

8.6 ASCII Message Format

The ASCII formatted message is almost identical in content to the RTU formatted message with the addition of the Start Of Packet (SOP) and End Of Packet (EOP) characters. The SOP character is a ':' and the EOP is the combination carriage return <CR> (0x0D) followed by a linefeed <LF> (0x0A).

The contents of the packet are converted to 2 byte hex characters 0-9 and A-Z. For example, the 1 byte device address 25 would be two bytes 0x3235 where 0x32 is the ASCII character '2' and 0x35 is the ASCII character '5'. This ensures that the message contents never contain SOP or EOP characters.

The check value uses a different mathematical algorithm and is called an LRC.

8.6.1 Computer (Master) Message Format

| | | | | | |
|--------------|----------------|---------------|--------------|-----|------------|
| Start Packet | Device Address | Function Code | Data Payload | LRC | End Packet |
|--------------|----------------|---------------|--------------|-----|------------|

- Start Packet: the ':' character signals the start of an ASCII packet.
- Device Address: 2 byte field containing the device address 1-247 in hex characters.
- Function Code: 2 byte field with a value range 1-127 representing the standard or extended function code in hex characters.
- Data Payload: 0-N bytes with information required to complete the requested function code operation. Each data byte is represented in its two byte hex character format.
- LRC: 2 bytes represented in hex characters with a value computed mathematically from the message bytes. This value can be computed on both sides of the link and verified to ensure that the bits of the message were transmitted and received correctly.
- End Packet the <CR><LF>characters.

8.6.2 Device (Slave) Response Format

| | | | | | |
|--------------|----------------|---------------|--------------|-----|------------|
| Start Packet | Device Address | Function Code | Data Payload | LRC | End Packet |
|--------------|----------------|---------------|--------------|-----|------------|

- Start Packet: the ':' character signals the start of an ASCII packet.
- Device Address: 2 byte field containing the device address 1-247 in hex characters.

- Function Code: 2 byte field with a value range 1-127 representing the standard or extended function code in hex characters.
- Data Payload: 0-N bytes with response data from the device in hex characters. For an error response, the payload will be a 1 byte value 1-255.
- LRC: 2 bytes represented in hex characters with a value computed mathematically from the message bytes. This value can be computed on both sides of the link and verified to ensure that the bits of the message were transmitted and received correctly.
- End Packet: the <CR><LF> characters.

8.7 IP Message Format

The IP message format is based on the RTU format but eliminates the CRC because TCP/IP will ensure that the message bytes are transmitted correctly. The instrument does not support IP format. However, the protocol can be used through a connected telemetry device.

TCP/IP is an asynchronous protocol. The significance is that the device must send its response to the address of the sender (computer). This differs from the serial RTU and ASCII formats where the protocol assumes only a single master device (computer) which does not have an address. For this reason, the IP format has a different header than the RTU serial formatted message.

8.7.1 Computer (Master) Message Format

| Xac ID | Protocol ID | Msg Length | Device Address | Function Code | Data Payload |
|--------|-------------|------------|----------------|---------------|--------------|
|--------|-------------|------------|----------------|---------------|--------------|

- Xac ID: 2 byte transaction ID to provide for asynchronous messages.
- Protocol ID: 2 byte field – always 0.
- Length: 2 byte field represents number of bytes following.
- Device Address: 1 byte field with a value ranging from 1 to 247. Broadcast address is 0.
- Data Payload: 0-N bytes with information required to complete the requested function code operation.

8.7.2 Device (Slave) Message Format

| Xac ID | Protocol ID | Msg Length | Device Address | Function Code | Data Payload |
|--------|-------------|------------|----------------|---------------|--------------|
|--------|-------------|------------|----------------|---------------|--------------|

- Xac ID: transaction ID echoed from message.
- Protocol ID: 2 byte field – always 0.
- Length: 2 byte field represents number of bytes following.
- Device Address: Echoed from message. No response for broadcast.
- Data Payload: 0-N bytes with information required to complete the requested function code operation.

8.8 Data Addresses (Registers)

As shown in the section pertaining to message formats, the function code from the Master's message tells the Slave device what operation to perform. There are a number of standard function codes defined by Modbus. These functions typically operate on atomic chunks of data historically and generically called registers. A register is quite simply a data address. A Modbus

device will always have a published “Register Map” that defines the numerical addresses of data values that can be accessed in the device.

Note: Register Maps typically use a 1-based numbering system whereas the protocol requires the data address/register number passed to a device to be 0-based. In this document, the register maps are 1-based.

Data addresses are associated with two atomic sizes of memory, 1 bit and 2 bytes. These are divided into Read-Only Bits, Read/Write Bits, Read-Only Words and Read/Write Words each with their own associated name as follows:

- Discretes: Read Only Bits
- Coils: Read/Write Bits
- Input Register: Read Only Word
- Holding Register: Read/Write Word

All Manufacturer Registers are Holding Registers.

8.9 Function Codes

Function codes in a message packet tell the device what operation to perform. The function code is always in the byte following the device address. Some of the standard function codes are as follows:

- 01: Read Coil
- 02: Read Discrete
- 03: Read Holding Registers
- 04: Read Input Registers
- 05: Write Coil
- 06: Write single Holding Register
- 15: Write multiple Coils
- 16: Write multiple Holding Registers
- 17: Read slave device ID
- 22: Mask write Holding Register

The slave device ID is an implementation defined response that will vary with manufacturer and/or device.

Function codes from 65 to 72 and 100 to 110 can be used as custom function codes.

8.10 Standard Message Formats

The manufacturer has implemented a sub-set of the standard function codes in every Modbus-enabled instrument. These message formats are defined in this section.

8.10.1 Read Holding Registers

This command reads one or more registers from a device.

| Message (8 bytes): | | | Response (5 + N bytes): | | |
|--------------------|---------|-------------|-------------------------|---------|-----------|
| Address | 1 Byte | 1-247 | Address | 1 Byte | 1-247 |
| Function Code | 1 Byte | 0x03 | Function Code | 1 Byte | 0x03 |
| Data Address | 2 Bytes | 0 to 0xFFFF | Byte Count | 1 Byte | 0 to 0xFA |
| Register Count | 2 Bytes | 0 to 0x7D | Data Payload | N Bytes | |
| CRC | 2 Bytes | | CRC | 2 Bytes | |

Where Byte Count is the #bytes in the Data Payload (does not include CRC bytes).

Byte Count = 2 * Register Count.

8.10.2 Write Holding Register

This command sets a **single** register in a device.

| Message (8 bytes): | | | Response (8 bytes – message echo): | | |
|--------------------|---------|-------------|------------------------------------|---------|-------------|
| Address | 1 Byte | 1-247 | Address | 1 Byte | 1-247 |
| Function Code | 1 Byte | 0x06 | Function Code | 1 Byte | 0x06 |
| Data Address | 2 Bytes | 0 to 0xFFFF | Data Address | 2 Bytes | 0 to 0xFFFF |
| Data Payload | 2 Bytes | 0 to 0xFFFF | Data Payload | 2 Bytes | 0 to 0xFFFF |
| CRC | 2 Bytes | | CRC | 2 Bytes | |

8.10.3 Write Holding Registers

This command sets **one or more** registers in a device.

| Message (9 + N bytes): | | | Response (8 bytes): | | |
|------------------------|---------|-------------|---------------------|---------|-------------|
| Address | 1 Byte | 1-247 | Address | 1 Byte | 1-247 |
| Function Code | 1 Byte | 0x10 | Function Code | 1 Byte | 0x10 |
| Data Address | 2 Bytes | 0 to 0xFFFF | Data Address | 2 Bytes | 0 to 0xFFFF |
| Register Count | 2 Bytes | 1 to 0x7B | Register Count | 2 Bytes | 0 to 0x78 |
| Byte Count | 1 Byte | 2 to 0xF0 | CRC | 2 Bytes | |
| Data Payload | N Bytes | | | | |
| CRC | 2 Bytes | | | | |

Where Byte Count is the #bytes in the Data Payload (does not include CRC bytes).

Byte Count = 2*Register Count.

The register count is limited to a single data format field. If an attempt is made to write a data field with an incorrect register count, the device will return a Modbus exception response with error code 0x80.

8.10.4 Mask Write Register

This command will set and/or clear one or more bits in a single register.

| Message : | | | Response: | | |
|---------------|---------|-------------|---------------|---------|-------------|
| Address | 1 Byte | 1-247 | Address | 1 Byte | 1-247 |
| Function Code | 1 Byte | 0x16 | Function Code | 1 Byte | 0x16 |
| Data Address | 2 Bytes | 0 to 0xFFFF | Data Address | 2 Bytes | 0 to 0xFFFF |
| And_Mask | 2 Bytes | 0 to 0xFFFF | And_Mask | 2 Bytes | 0 to 0xFFFF |
| Or_Mask | 2 Bytes | 0 to 0xFFFF | Or_Mask | 2 Bytes | 0 to 0xFFFF |
| CRC | 2 Bytes | | CRC | 2 Bytes | |

Register = (Register Value AND And_Mask) OR (Or_Mask AND (NOT And_Mask))

And_Mask: 0 = bits to change, 1 = bits to leave unchanged.

Or_Mask: 0 = bits to clear, 1 = bits to set.

Example: - set bit 1 (LSB), clear bit 2, leave remaining bits unchanged

| | | |
|------------|--------|---------------------|
| Old Value: | 0x007E | 0000 0000 0111 1110 |
| And_Mask: | 0xFFFC | 1111 1111 1111 1100 |
| Or_Mask: | 0x0001 | 0000 0000 0000 0001 |
| <hr/> | | |
| New Value: | 0x007D | 0000 0000 0111 1101 |

This command is useful in a bit mapped register where the Master wants to set some bits of a register that are mapped to a feature without disturbing the other bits of the register that might be mapped to a different feature.

8.10.5 Report Slave ID

This command query's a device for ID information.

| Message (4 bytes): | | | Response (N bytes): | | |
|--------------------|---------|-------|---------------------|---------|-----------|
| Address | 1 Byte | 1-247 | Address | 1 Byte | 1-247 |
| Function Code | 1 Byte | 0x11 | Function Code | 1 Byte | 0x11 |
| CRC | 2 Bytes | | Byte Count | 1 Byte | 0 to 0xFF |
| | | | Slave ID | 1 Byte | 0 to 0xFF |
| | | | Run Status | 1 Byte | 0 to 0xFF |
| | | | Data Payload | N Bytes | |
| | | | CRC | 2 Bytes | |

- The Data Payload layout is defined in the Slave ID Format section.
- The Byte Count field is the number of bytes from the Slave ID field to the end of the Data Payload (excludes the 2 byte CRC).
- Run Status – must be 0x00 or 0xFF.

Slave ID Layout Example

| Byte Offset | Field Description | Type | Value |
|----------------------------------|------------------------------|--------|-----------------------|
| 0 | Device Address | byte | 1-247 |
| 1 | Function Code | byte | 0x11 |
| 2 | Byte Count | byte | 23 |
| 3 | Slave ID | byte | 0x49 ('I') |
| 4 | Run Status Indicator | byte | 0x00 = Off, 0xFF = On |
| Device Specific Info | | | |
| 5 | Slave ID Format Version ID | byte | 0x01 |
| Format Version 1 ID Block | | | |
| 6-7 | Manufacturer ID | ushort | 0x5349 ('SI') |
| 8-9 | Device ID | ushort | |
| 10-11 | Application Firmware Version | ushort | Version * 100 |
| 12-13 | Boot Code Firmware Version | ushort | Version * 100 |
| 14-15 | Hardware Version | ushort | |
| 16-17 | Register Map Template ID | ushort | |

| Byte Offset | Field Description | Type | Value |
|-------------|--------------------------------------|--------|-------|
| 18-21 | Device Serial Number | ulong | |
| 22-23 | Max Message/Response Size (bytes) | ushort | |
| 24-25 | Max Baud Rate ID | ushort | |
| 26-27 | CRC | | |

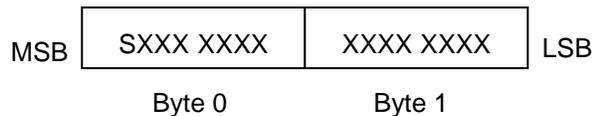
The specific field values such as device ID, baud ID, etc. are documented in the sections that follow.

8.11 Instrument Manufacturer Data Types

Modbus defines all I/O in terms of 2 byte blocks called registers. Modbus does not formally define blocks for floating point values or strings. In the manufacturer's implementation, these fundamental types and others are handled by combining two or more registers. The manufacturer data type implementations are defined in the following sections.

8.11.1 Short

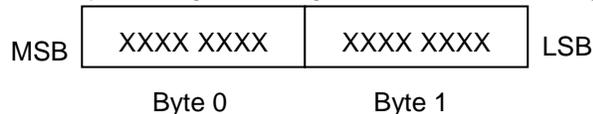
A 2-byte signed integer contained in a single register data address. IEEE standard.



Where S = sign bit

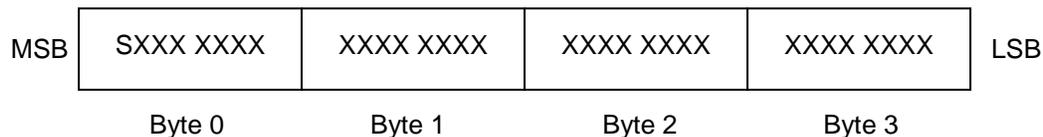
8.11.2 Unsigned Short

A 2-byte unsigned integer contained in a single register data address. IEEE standard.



8.11.3 Long

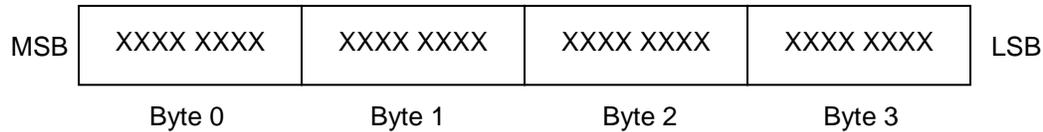
A 4-byte signed integer contained in two register data address's. IEEE standard.



Where S = sign bit

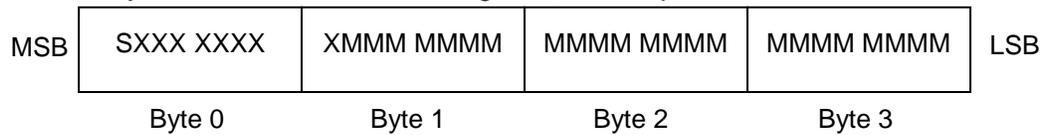
8.11.4 Unsigned Long

A 4-byte unsigned integer contained in two register data address's. IEEE standard.



8.11.5 Float

IEEE 4-byte numeric standard – 1 sign bit, 8-bit exponent, 23-bit mantissa.



Where S = sign bit, X = exponent bits and M = mantissa bits.

8.11.6 Double

IEEE 8-byte numeric standard – 1 sign bit, 11-bit exponent, 64-bit mantissa.



Where S = sign bit, X = exponent bits and M = mantissa bits.

8.11.7 Character

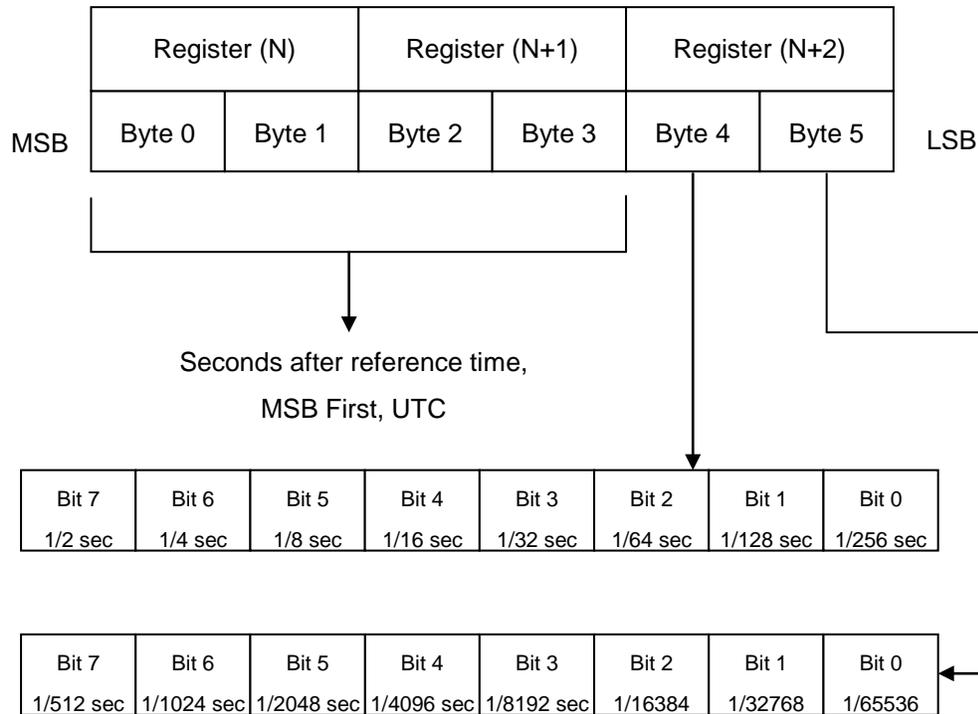
A 2 byte Unicode character contained with a single register data address.

8.11.8 String

The number of consecutive registers in the register map will represent the maximum string length in Unicode characters excluding any termination characters. For example, a 6-register string can have 6 Unicode characters. When reading/writing a string, all register values must be transmitted. If the string to be written does not require the full defined field length, the extra bytes must be padded with the value 0x0000 to reach full length. A string termination character is not required. All registers may contain a character. Failure of the Master device to transmit or request all registers of a string field will be rejected by the Slave device with the Modbus exception error code 0x80.

8.11.9 Time

Time is represented by a 6-byte (3 register) number. The first 4 bytes represent the number of seconds since 00:00:00 January 1, 1970 UTC, MSB first, not adjusted for DST. The 5th and 6th bytes are fractions of a second represented by the bits in powers of 2 starting with the MSB. If a device does not have the ability to support the full fractions of a second resolution available in the time format, unused bits must be set to 0.



Time Example: For a time value of 0x001A5E00C000, the bytes 0x001A5E00 represent the whole number of seconds from the reference time. The bytes 0xC000 represent the additional fractional number of seconds as shown in the diagram above. In this example, the whole number of seconds represents 20 days and the fractional seconds represents 750 ms.

8.12 Exception Codes

The instrument manufacturer supports the standard Modbus exception codes but also provides additional exception codes to assist with troubleshooting problems.

| Code | Name | Description |
|------|----------------------|--|
| 1 | Illegal Function | The function code received in the query is not an allowable action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it. |
| 2 | Illegal Data Address | The data address received in the query is not an allowable address for the slave. |
| 3 | Illegal Data Value | A value contained in the query data field is not an allowable value for the slave. |
| 4 | Slave Device Failure | An unrecoverable error occurred while the slave was attempting to perform the requested action. |
| 5 | Acknowledge | The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed. This exception code may not be supported by the devices |
| 6 | Slave Device Busy | The slave is engaged in processing a long-duration program command. The master should retransmit the message later when the slave is free. This exception code may not be supported by the devices |
| 8 | Memory Parity Error | Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server (or slave) attempted to read record file, but detected a parity error in the memory. The client (or master) can retry the request, but service may be required on the server (or slave) device. This exception code may not be supported by the devices |

| Code | Name | Description |
|------|---|---|
| 0x0A | Gateway Path Unavailable | Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded. This exception code may not be supported by the devices. |
| 0x0B | Gateway Target Device Failed To Respond | Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network. This exception code may be used by controllers, but is not used by probes. |

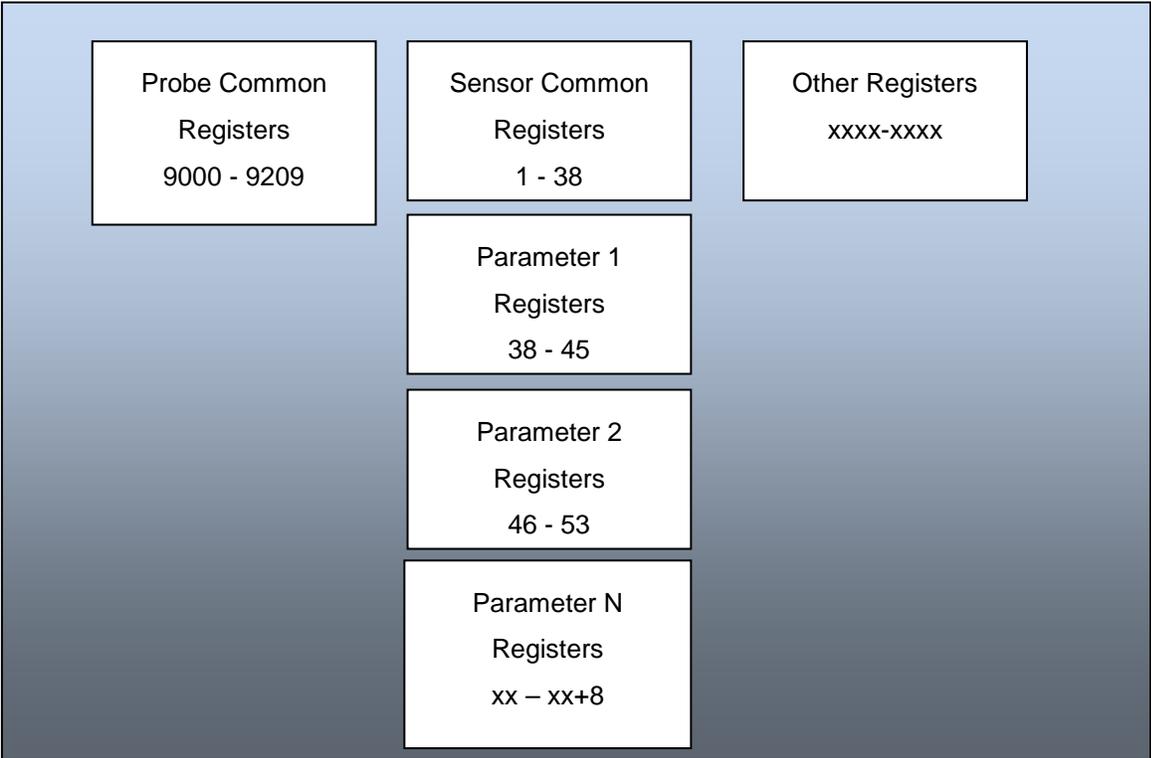
8.13 Extended Exception Codes

| Code | Name | Description |
|------|---------------------|---|
| 0x80 | Field Mismatch | Mismatch between register number, count and field size. |
| 0x81 | Write Only Register | Attempting to read a Write Only register. |
| 0x82 | Read Only Register | Attempting to write a Read Only register. |
| 0x83 | Access Level | Attempting to Read/Write a register with invalid access level. |
| 0x84 | Write Value | Attempting to write an illegal field value. |
| 0x85 | Command Sequence | Invalid device command register sequence. |
| 0x86 | File Sequence | Invalid file command register sequence. |
| 0x87 | File Command | Invalid file command. |
| 0x88 | File Number | Invalid file number. |
| 0x89 | File Size | Invalid file size. |
| 0x8A | File Data | The file data block transferred to device is invalid. |
| 0x8B | File Interval | Invalid file interval. |
| 0x90 | Gateway Error | Invalid probe command on controller gateway. |
| 0x91 | Sensor Sequence | Invalid sensor command register sequence. |
| 0x92 | Sensor Mode | Invalid change to sensor mode, or attempting to read/write with an invalid sensor mode set. |
| 0x93 | Sensor Config | Attempting a sensor operation on a sensor port that is not part of the current configuration. |

| Code | Name | Description |
|------|--------------------|---|
| 0x94 | Sensor Missing | Attempting a sensor operation on a sensor port with no sensor connected. |
| 0x95 | Sensor Invalid | Attempting a sensor operation on a sensor port that has a sensor that is not compatible with the sensor that was originally configured. |
| 0x96 | Sensor Firmware | Attempting a sensor operation on a sensor with no application code. |
| 0xA0 | Data Log Register | Attempting to write a register that is read-only during logging. |
| 0xA1 | Data Log Memory | Data log memory is full. |
| 0xA2 | Data Log Directory | Data log directory is full. |
| 0xA3 | Data Log Edit | Log configuration cannot be edited. |
| 0xA4 | Data Log Sequence | Invalid data log command sequence. |

8.14 Probe Register Map Layout

Each probe and sensor will follow a common layout pattern which provides consistency and improves code reuse between platforms. **All Instrument registers are Holding Registers.** The common probe register map layout is as follows:



All registers in this document are 1-based. This means the actual packets sent to the devices must have a data address 1 less than what the register number is in this document.

8.15 Probe Common Registers

| Register XXXX | Size (registers) | Mode (R/W) | Data Type | Description |
|------------------|---------------------|---------------|--------------|---|
| 9000 | 1 | R | ushort | Register Map Template Version (SIS Level) |
| 9001 | 1 | R/W | ushort | Device ID |
| 9002-003 | 2 | R/W | ulong | Device Serial Number |
| 9004-006 | 3 | R/W | time | Manufacture Date |
| 9007 | 1 | R | ushort | Firmware version * 100 |
| 9008 | 1 | R | ushort | Boot Code version * 100 |
| 9009 | 1 | R | ushort | Hardware version |
| 9010 | 1 | R | ushort | Max Data Logs |
| 9011-012 | 2 | R | ulong | Total Data Log Memory (bytes) |
| 9013-014 | 2 | R | ulong | Total Battery Ticks |
| 9015-017 | 3 | R/W | time | Last Battery Change |
| 9018 | 1 | | | Reserved |
| 9019-050 | 32 | R/W | string | Device Name |
| 9051-082 | 32 | R/W | string | Site Name |
| 9083-086 | 4 | R/W | double | Latitude Coordinate (degrees) (negative = east) |
| 9087-090 | 4 | R/W | double | Longitude Coordinate (degrees) (negative = south) |
| 9091-094 | 4 | R/W | double | Altitude Coordinate (meters) |
| 9095-096 | 2 | | | Reserved |
| 9097-099 | 3 | R/W | time | Current time (UTC) |
| 9100-101 | 2 | R | 32 bits | Device Status |
| 9102-103 | 2 | R/W | ulong | Used Battery Ticks |
| 9104-105 | 2 | R | ulong | Used Data Log Memory (bytes) |

8.16 Communication Control Registers

| Register XXXX | Size (registers) | Mode (R/W) | Data Type | Description |
|------------------|---------------------|---------------|--------------|---|
| 9200 | 1 | R/W | ushort | Device Address (1-247, default = 1) |
| 9201 | 1 | R/W | ushort | Serial Communication Configuration |
| 9202 | 1 | R/W | ushort | EOM timeout (1000-15000 ms, default = 1000) |
| 9203 | 1 | R/W | ushort | EOS timeout (5000-60000 ms, default = 5000) |
| 9204 | 1 | R | ushort | Max allowed baud rate id (0-7) |
| 9205 | 1 | R | ushort | Max Message/Response size (bytes) |
| 9206-207 | 2 | R/W | ulong | Good message counter |
| 9208 | 1 | R/W | ushort | Bad message counter |
| 9209 | 1 | R/W | ushort | Exception response counter |

8.17 Sensor Connection Registers

| Register XXXX | Size (registers) | Mode (R/W) | Data Type | Description |
|-----------------------------|---------------------|---------------|--------------|---|
| 9300 | 1 | R1 | ushort | Max Sensor Connections (1-32) |
| 9301-302 | 2 | R1 | 32 bits | Sensor Connection Status |
| Sensor Map Registers | | | | |
| RDO Sensor | | | | |
| 9303 | 1 | R1 | ushort | Connection 1 sensor id |
| 9304 | 1 | R1 | 16 bits | Connection 1 sensor status |
| 9305 | 1 | R1/W2 | ushort | Connection 1 sensor command |
| 9306 | 1 | R1 | ushort | Connection 1 sensor data register map version |
| 9307 | 1 | R1 | ushort | Connection 1 sensor data register offset |
| Conductivity Sensor | | | | |
| 9308 | 1 | R1 | ushort | Connection 2 sensor id |
| 9309 | 1 | R1 | ushort | Connection 2 sensor status |
| 9310 | 1 | R1 | ushort | Connection 2 sensor command |

| Register XXXX | Size (registers) | Mode (R/W) | Data Type | Description |
|-------------------------|---------------------|---------------|--------------|---|
| 9311 | 1 | R1 | ushort | Connection 2 sensor data register map version |
| 9312 | 1 | R1 | ushort | Connection 2 sensor data register offset |
| Level Sensor | | | | |
| 9313 | 1 | R1 | ushort | Connection 3 sensor id |
| 9314 | 1 | R1 | ushort | Connection 3 sensor status |
| 9315 | 1 | R1 | ushort | Connection 3 sensor command |
| 9316 | 1 | R1 | ushort | Connection 3 sensor data register map version |
| 9317 | 1 | R1 | ushort | Connection 3 sensor data register offset |
| pH/ORP Sensor | | | | |
| 9318 | 1 | R1 | ushort | Connection 4 sensor id |
| 9319 | 1 | R1 | ushort | Connection 4 sensor status |
| 9320 | 1 | R1 | ushort | Connection 4 sensor command |
| 9321 | 1 | R1 | ushort | Connection 4 sensor data register map version |
| 9322 | 1 | R1 | ushort | Connection 4 sensor data register offset |
| Connections 2-32 | | | | |
| 9323-462 | 5 x 31 | | | Connections 2-32 patterned after connection 1 |
| 9463 | 1 | R1/W3 | ushort | Sensor Data Cache Timeout (0-60000 msec) |

8.18 Current Loop Configuration Registers

| Register XXXX | Size (registers) | Mode (R/W) | Data Type | Description |
|------------------|---------------------|---------------|--------------|--|
| 9501 | 1 | R1/W3 | ushort | Current loop sensor number |
| 9502 | 1 | R1/W3 | ushort | Current loop parameter number |
| 9503-504 | 2 | R1/W3 | float | Current loop 20 mA setpoint (I_{20}) |
| 9505-506 | 2 | R1/W3 | float | Current loop 4 mA setpoint (I_4) |
| 9507 | 1 | R1/W3 | ushort | Current loop enable (0 = off, 1 = on) |

8.19 Logged Record Registers

| Register XXXX | Size (registers) | Mode (R/W) | Data Type | Description |
|------------------|---------------------|---------------|--------------|--|
| 9600-601 | 2 | R1 | ulong | Number of logged records |
| 9602-603 | 2 | R1/W3 | ulong | Requested log record number |
| 9604-606 | 3 | R1 | time | Time stamp for record |
| 9607-608 | 2 | R1 | float | Parameter 1 measured value |
| 9609 | 1 | R1 | ushort | Parameter 1 data quality ID |
| 96010-696 | 29 x 3 | R1 | --- | Parameters 2 – 30 (30 is max allowed #log parameters) |

8.20 Register Map Template ID

This is a number that represents the version of the reserved register map supported by the device. This provides for the modification of the reserved register map at a future date and detection of this difference by the Master. This register is commonly referred to as the System Interface Specification Level or (SIS Level).

8.21 Device ID

Device ID for Multi-PRO 400 = 18

8.22 Device Serial Number

This is a 6-digit serial number engraved on the device. Serial numbers for devices in this system will range from 000001 to 999999.

8.23 Manufacture Date

This is the date and time of manufacture in the Time format.

8.24 Firmware, Boot Code, Hardware Versions

The firmware and boot code versions will be the floating point version multiplied by 100 to create an integer. For example, version 1.32 will be stored as 132. The hardware version will be a non-scaled integer that represents the circuit board version. The hardware version will be calculated by the firmware based on parameters determined from the circuit board components.

8.25 Device Name

This is a general-purpose 32-character string representing a user-defined device name or identifier.

8.26 Site Name

This is a 32-character string that represents the location where the instrument is recording data.

8.27 Coordinates

These registers are used by the computer to store the coordinates of a device when taking measurements. The device expects coordinates in meters and degrees.

- A negative latitude value represents south, a negative longitude value represents east.
- If an attempt is made to change the coordinates while logging, the device will return an exception response with error code 0xA0 (illegal write with running log).

8.28 Current Time

This represents the time in the device in UTC.

8.29 Device Status

The device status register holds general status information. Each set bit represents a status value. There are a limited number of standardized predefined status values that all devices will support. These predefined status values are contained in the lower register. The upper register is reserved for device specific status values.

| Bit | Category | Description |
|-------|----------|--|
| 0 | Alarm | Sensor high alarm |
| 1 | Warning | Sensor high warning |
| 2 | Warning | Sensor low warning |
| 3 | Alarm | Sensor low alarm |
| 4 | Warning | Sensor calibration warning |
| 5 | Alarm | Sensor malfunction |
| 6-7 | N/A | Reserved |
| 8 | Status | Power management disabled |
| 9 | Status | Device off line |
| 10 | Alarm | Device hardware reset occurred |
| 11 | Alarm | Device malfunction |
| 12 | Status | No external power. |
| 13 | Warning | Low battery – battery capacity < 5% |
| 14 | Warning | Low memory – data log memory capacity < 5% |
| 15 | N/A | Reserved |
| 16-31 | N/A | Available for device-specific status |

- Bits 0-7 of the device status register are reserved for sensor status. These bits are the logical OR of bits 0-7 of the sensor status register in each sensor connection.
- Bits 8-15 of the device status register are reserved for common device status. Any bit in this range that is not applicable to a device will be set to zero.
- Bits 16-31 of the device status register are available for device-specific status. Any bit in this range that is not utilized by a device will be set to zero.

8.30 Serial Communication Configuration

The 16 bits in this register are mapped to the communication parameters. The bits are mapped as follows:

| Bits | Description |
|---------|---|
| 0 | Modbus Transmission Mode 0 = RTU (default) 1=ASCII |
| 1,2 & 3 | Baud Rate Id 0 = 9600 (mandatory) 1 = 19200 (default) 2 = 38400 3 = 57600 4 = 115200 5 = 128000 6 = 230400 7 = 256000 |
| 4 | Data Bits 0 = 7 data bits 1 = 8 data bits (default) |
| 5,6 | Parity Bits 0 = Even (default) 1 = Odd 2 = None |
| 7 | Stop Bits 0 = 1 Stop Bit (default) 1 = 2 Stop Bits |
| 8-15 | Unassigned |

When the communication configuration register is changed, the Modbus response will be sent to the Master at the current configuration (mode, baud, parity, data bits...etc). After the response has been sent to the Master, the device will switch to the new settings.

The Master software must switch communications to the new settings after receiving a positive Modbus response to the write message. The Master software should confirm the new settings by reading back the device address and communication configuration register.

8.31 Baud Rates

The instrument supports 9600 through 57600 baud rates. A device will support all baud rates from 9600 up to and including the maximum baud rate as specified by the Max Baud Rate ID register.

- If the Master attempts to set the baud rate of a device to a non-supported value, the device will respond with a Modbus exception error code 3 (Illegal Data Value).
- Baud rates will be referenced in this document by the ID 0-7.

8.32 RTU Settings

Every device supporting serial communications will implement Modbus RTU. The device will at a minimum support the standard settings defined below.

- 1 Start Bit
- 8 Data Bits
- 1 Parity Bit
- 1 Stop Bit
- Even Parity

Note: If an attempt is made to write RTU mode with 7 data bits, the device will return an exception with error code 0x84 (Write Illegal Value). RTU communication does not support 7-data-bit settings.

8.33 ASCII Settings

When implementing Modbus ASCII, the device will at a minimum support the standard settings defined below.

- 1 Start Bit
- 7 Data Bits
- 1 Parity Bit
- 1 Stop Bit
- Even Parity

8.34 Max Message/Response Size

This register indicates to the Master the largest message or response the device can accept. This may vary based on the hardware configuration of the device.

8.35 Message Counters

The message counter registers are to provide diagnostic information for troubleshooting communication problems. The counters will not roll over and can only be cleared using the configuration software.

There are 3 message counter registers allocated as follows:

- Good Message Counter – count of number of properly formatted messages received that are addressed to this device.

- Bad Message Counter – count of number of improperly formatted messages received (i.e. bad CRC). Bad messages might not be associated with this device because it is impossible to determine if a bad message was addressed to the device or not.
- Exception Response Counter – count of the number of messages received that were rejected with a Modbus exception response.

8.36 Sensor Connection Registers

8.36.1 Max Sensor Connections

Each probe or controller can present up to 32 sensor images to the outside world. These sensors may be real sensors plugged into physical ports, sensors on the internal PCB or it may be virtual sensors created from derived information. This register tells how many sensor images are being presented by the probe or controller at the time it is read.

8.36.2 Sensor Connection Status

Each bit, when set, indicates that the probe or controller is actively presenting a sensor image on the corresponding port. The bit position 0-31 is used to determine the sensor map register group to use to control the sensor or access its data block.

8.37 Sensor Map Registers

8.37.1 Sensor ID Registers

These registers duplicate the Sensor ID register provided in each sensor's Sensor Data Header Register block. If a sensor connection is open, the probe returns a zero. If an attempt is made to access a Sensor ID register that is not mapped to a sensor (one that exceeds the maximum number of sensors supported by the probe), the probe will return an exception response with error code 0x02 (illegal data address).

8.37.2 Sensor Status Register

These registers duplicate the sensor status register provided in each sensor's Sensor Data Header Block. If a sensor connection is open, the probe returns a zero. If an attempt is made to access a sensor status register that is not mapped to a sensor (one that exceeds the maximum number of sensors supported by the probe), the probe will return an exception response with error code 0x02 (illegal data address).

8.37.3 Sensor Command Register

These registers are used to send commands to a sensor. The behavior of the sensor command register and the commands available is different from SIS-1&2 and SIS-3. The sensor command register is primarily used to calibrate the sensor. Failure of the command is returned in the response exception.

If an attempt is made to access a sensor command register that is not mapped to a sensor (one that exceeds the maximum number of sensors supported by the probe), the probe will return an exception response with error code 0x02 (illegal data address).

| ID | Name | Description |
|--------|-------------------------|--|
| 0xE000 | Calibration Mode On | Put sensor in the calibration mode. |
| 0xE001 | Calibration Update | Commit new calibration to memory. |
| 0xE002 | Calibration Mode Off | Put sensor in the normal operating mode. |
| 0xE003 | Restore Cal Defaults | Restore factory default calibration for sensor only. |
| 0xE004 | Restore sensor defaults | Restore factory defaults for calibration, parameters and units, sentinel values, etc. for sensor only. |

8.37.4 Sensor Data Register Map Version Registers

These registers specify the version of the sensor data register map pointed to by the Sensor Data Register Offset registers. If a sensor connection is open, the probe shall return a zero. If an attempt is made to access a sensor map version register that is not mapped to a sensor (one that exceeds the maximum number of sensors supported by the probe), the probe will return an exception response with error code 0x02 (illegal data address).

8.37.5 Sensor Data Register Map Offsets

These registers specify the register number of the first register in each sensor's Sensor Data Header Block. These registers assume the 4x (holding register) reference ID and therefore will not contain the reference ID as part of the value (i.e. 40001 will be stored as 1).

If an attempt is made to access a sensor status register that is not mapped to a sensor (one that exceeds the maximum number of sensors supported by the probe), the probe returns an exception response with error code 0x02 (illegal data address).

If a sensor connection is open (the sensor is not plugged in or is not part of the current configuration) the probe returns 0. This permits a valid block read of the mapped sensor status registers regardless of the current configuration of the probe and sensors.

If an attempt is made to access a sensor connection register that is not mapped to a sensor (one that exceeds the maximum number of sensors supported by the probe), the probe will return an exception response with error code 0x02 (illegal data address).

8.37.6 Sensor Data Cache Timeout

Sensors shall measure all of their parameters whenever a parameter value register is read. The parameters are recorded in a data cache and a cache timeout is started. If any subsequent parameter value from the sensor is read within the specified cache timeout, the device shall return the value recorded in the cache instead of making a new measurement. The default value of the cache timeout is device specific. If the cache timeout value is set to 0 milliseconds, each Modbus message to read one or more parameters will cause a new sensor reading to be taken. Logs running on a probe will also populate the data cache when a measurement is taken.

The end-of-session timeout supersedes the cache timeout. If an end of session timeout occurs, the cache for all sensors shall be cleared.

8.38 Current Loop Configuration

Attempting to access current loop configuration registers 9501 through 9507 will result in an exception response with exception code 0x02 (illegal data address).

8.39 Last Logged Record Registers

These registers are not supported by the device since data logging is not supported. Attempting to access these registers will result in an exception response with exception code 0x02 (illegal data address).