

# MODBUS User Guide

## AquaVent

July 1, 2020



**Solinst**<sup>®</sup>



|  |           |
|--|-----------|
| <b>1 Introduction</b>                            | <b>5</b>  |
| 1.1 MODBUS Interface Overview                    | 5         |
| 1.2 LED Status                                   | 6         |
| <b>2 AquaVent Independent Recording Option</b>   | <b>7</b>  |
| 2.1 Data Downloading and Programming in Field    | 7         |
| <b>3 AquaVent MODBUS Installation</b>            | <b>8</b>  |
| 3.1 Setting the MODBUS Device Address            | 8         |
| 3.2 Wiring Overview                              | 10        |
| 3.2.1 RS-232 Connection                          | 10        |
| 3.2.2 RS-485 Connection                          | 11        |
| 3.3 AquaVent Setup and Startup                   | 12        |
| 3.3.1 AquaVent Setup                             | 12        |
| 3.3.2 AquaVent Startup Notes                     | 13        |
| <b>4 AquaVent MODBUS Operation</b>               | <b>14</b> |
| 4.1 AquaVent MODBUS Configuration Overview       | 14        |
| 4.2 AquaVent MODBUS Data Types                   | 15        |
| 4.3 AquaVent Supported MODBUS Functions          | 16        |
| 4.4 AquaVent MODBUS Register Map                 | 16        |
| 4.5 AquaVent MODBUS Status and Diagnostic Codes  | 18        |
| 4.5.1 Device Status                              | 18        |
| 4.5.2 Diagnostic Tests                           | 18        |
| 4.5.3 Exception Response Codes                   | 19        |
| <b>5 Maintenance and Troubleshooting</b>         | <b>21</b> |
| 5.1 Maintenance                                  | 21        |
| 5.1.1 Changing or Updating AquaVent Loggers      | 21        |
| 5.1.2 Changing SPX Wellhead Batteries            | 21        |
| 5.1.3 Firmware Upgrades                          | 21        |
| 5.2 Troubleshooting                              | 22        |
| <b>6 Appendix: Additional MODBUS Information</b> | <b>23</b> |
| 6.1 Endian Order and Byte Packing                | 23        |
| 6.2 AquaVent Register Offset Assumptions         | 24        |
| <b>7.0 References</b>                            | <b>25</b> |



**! NOTE**

The user is expected to have working knowledge of the MODBUS protocol. For more information on MODBUS, go to: [www.modbus.org](http://www.modbus.org), where you can also find the references listed in [Section 7](#).

**! NOTE**

For more information about the AquaVent loggers, Wellheads and Vented Cables, see the [Vented Dataloggers User Guide](#).

## **1 Introduction**

MODBUS is a serial communications protocol designed to allow a number of different devices to communicate with a single MODBUS master. The current AquaVent 5 logger and the previous AquaVent logger are able to communicate with a MODBUS master using the AquaVent 5 SPX Wellhead or AquaVent SPX Wellhead, and a RS-232/RS-485 Connector Cable.

**Note: As features differ slightly between models, it is recommended to use the AquaVent 5 Wellhead with an AquaVent 5 logger.**

The SPX Wellhead (translator) converts MODBUS commands to the Solinst communications protocol for the AquaVent. The SPX Wellhead and Connector Cable provide a hardware and protocol conversion interface between a Solinst AquaVent logger and a MODBUS network.

### **1.1 MODBUS Interface Overview**

There are two serial interface options for communicating with the AquaVent using MODBUS protocol; RS-232 and RS-485. Solinst provides RS-232 and RS-485 Connector Cables, which are plugged into the side of the SPX Wellhead.

To connect the Connector Cables to the MODBUS customer equipment, you must ensure correct pin-to-pin connection, including a continuous external power supply to the SPX Wellhead. Power supplied to the Wellhead must be between 10V and 24V DC. See [Section 3](#) for more information on physically installing the system.

The bottom of the SPX Wellhead connects to an AquaVent logger through a Vented Cable. The AquaVent logger contains a small battery, used only for memory and clock back-up. The Wellhead contains four user-replaceable 1.5V AA lithium batteries that power the AquaVent.

The communication settings for the SPX Wellhead always use Serial, RTU mode, 8 data bits, even parity, and one stop bit. The MODBUS registers do not provide a means of changing any of these communication settings.

As the Wellhead acquires its identity from the AquaVent logger, in the absence of a functional AquaVent logger, or without setting a specific address or Baud rate, the Wellhead will power up with a default MODBUS device address of 1, and a default Baud rate of 19,200. These settings can be modified by Solinst Levellogger PC Software (see [Section 3.1](#)).


The AquaVent only uses MODBUS functions that operate on registers (16 bits) and transfers data in Big Endian. The AquaVent only supports holding registers, and consequently only MODBUS functions that deal specifically with holding registers. The MODBUS master must be able to recognize floating point data types.

| Technical Specifications        |                                   |
|---------------------------------|-----------------------------------|
| Wellhead Power Input:           | 10V to 24V DC                     |
| Modbus Connector Cable Length:  | 4.5 m (15 ft)                     |
| Maximum Vented Cable Length:    | 150 m (500 ft)                    |
| Wellhead Operating Temperature: | -20°C to 80°C                     |
| Wellhead IP Rating:             | IP 64 (dust and splash resistant) |


Table 1-1 MODBUS Technical Specifications


## 1.2 LED Status


The AquaVent 3500 SPX Wellhead contains a multicolour LED.

 A properly addressed MODBUS command is acknowledged by a single Yellow flash. A normal MODBUS response is indicated by a single Yellow flash. A MODBUS Exception response is indicated by a single Red flash (see [Section 4.5.3](#) for Exception response codes).

The following will occur during power up:

 Tricolour spin (3 cycles) to indicate a power-on event.

 Upon entering Field mode, a rapid 10 cycle alternation of Yellow/Green colour indicates that communication has been established with the AquaVent logger, and is running with the Baud rate and MODBUS device address prescribed by the AquaVent settings (which can only be changed by Solinst Levellogger PC Software).

 Upon entering Field mode, a rapid 10 cycle alternation of Yellow/Red colour indicates that the Wellhead has come online WITHOUT establishing communication with the AquaVent logger, and is running with the default Baud rate and MODBUS device address. (Check the batteries and the connections to your AquaVent logger).

**! NOTE**

See the [Vented Dataloggers User Guide](#) for detailed AquaVent operating instructions.

**! NOTE**

Once you are finished programming your AquaVent, unplug the USB Connector Cable from the Wellhead. The Wellhead only communicates with the MODBUS network when just the RS-232 or RS-485 Connector Cable is connected.

**! NOTE**

The Wellhead batteries will drain more quickly if the AquaVent logger is also set to record independently.

**! NOTE**

Other downloading options are available using the [Solinst Levelogger App](#) and [DataGrabber](#). See separate operating instructions.

## 2 AquaVent Independent Recording Option

The AquaVent logger has the ability to record and store readings in internal memory, independent from the MODBUS network. Before connecting the AquaVent to the MODBUS network, it can be programmed and started using Solinst Levelogger PC Software (see [Figure 3-1](#)).

All standard sampling options provided by Solinst Levelogger PC Software are available while the AquaVent is operating in a MODBUS network. The AquaVent logger can be set to record at a user-defined sampling rate; event, linear, and scheduled sampling modes are available. This allows the AquaVent logger to provide back-up data if the MODBUS network fails. The AquaVent stores the data in its internal memory, until it is downloaded.

Each time the MODBUS master asks the AquaVent for a current reading, the Wellhead batteries are used to transmit the information from the logger to the Wellhead. The AquaVent logger will also be using the batteries if programmed to record independently – draining the batteries more quickly.

### 2.1 Data Downloading and Programming in Field

If the AquaVent has been programmed to record on its own independent schedule, data can be downloaded from the AquaVent using a laptop and USB Connector Cable in the field (see [Figure 3-1](#)). Temporarily disconnect the Wellhead from the RS-232 or RS-485 Connector Cable, preferably in between recordings by the MODBUS master, as the master will not retry for any missed recordings. Connect a laptop and download the independently recorded data.

If the AquaVent is disconnected from the MODBUS network, and the MODBUS master tries to communicate with the AquaVent, the MODBUS master and the rest of the sensors in the network are not disrupted. After the data download is complete, the AquaVent is easily connected back into the MODBUS network without disruption.

### ! NOTE

For information on downloading and starting the [Levellogger PC Software](#), see the [Vented Dataloggers User Guide](#).

### ! NOTE

The RS-232 or RS-485 Connector Cable can remain in place while programming the AquaVent.

## 3 AquaVent MODBUS Installation

### 3.1 Setting the MODBUS Device Address

In the absence of a functional AquaVent logger, or without setting a specific address or Baud rate, the Wellhead will power up with a default MODBUS device address of 1, and a default Baud rate of 19,200.

To set the device address for the [AquaVent logger](#), it must be connected to Solinst Levellogger PC Software. The AquaVent communicates with the Software using a USB Connector Cable connected to the SPX Wellhead. The USB Connector Cable connects to the 10-pin (Solinst Protocol) connection on the Wellhead.

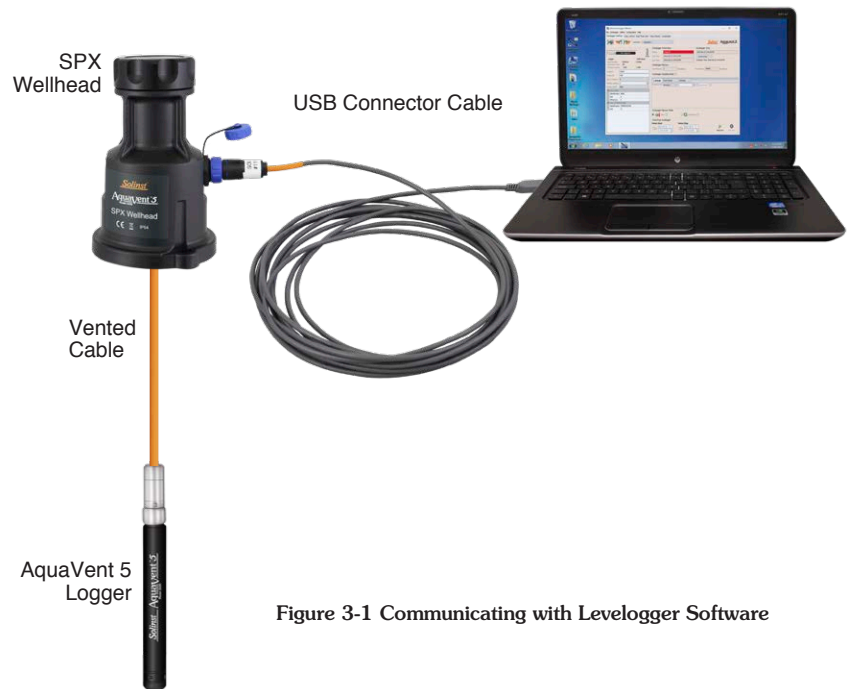


Figure 3-1 Communicating with Levellogger Software



After you start the Levelogger Software, the Main Window will appear, with the Datalogger Settings tab open.

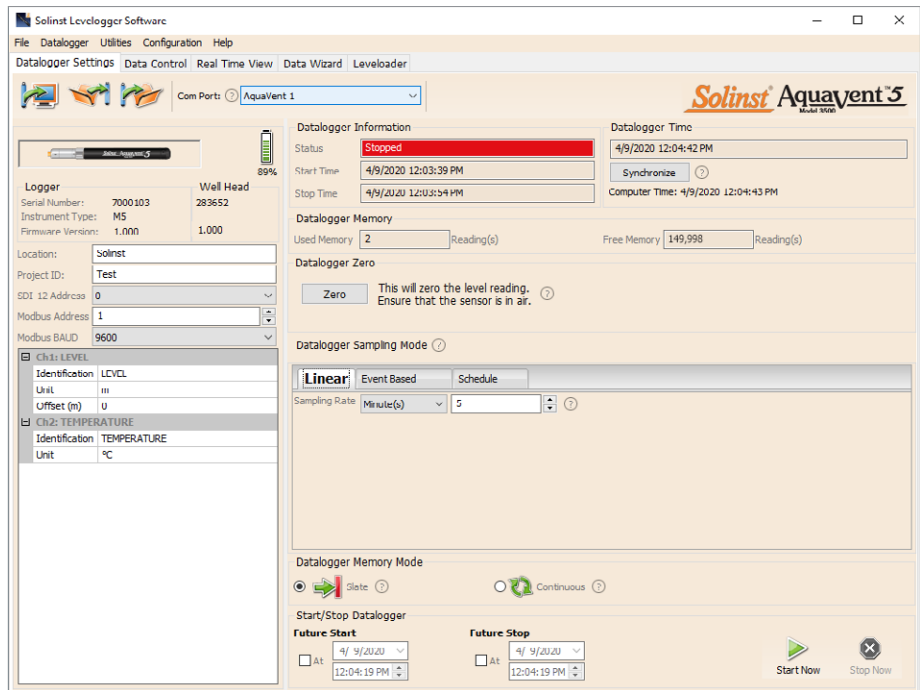


Figure 3-2 Dataloggers Settings Tab


Select the appropriate device from the centre Com Port drop-down menu.

Click  to retrieve the current settings from the connected AquaVent logger.

After you have retrieved the settings of the connected AquaVent, the Datalogger Settings tab will identify the Instrument Type, Serial Numbers, Firmware Versions, Project ID, Location, the Channel Settings and Battery Level.

Here, you can set the MODBUS device address and the Baud rate. Valid addresses range between 1 and 247.

You can also change the units that the AquaVent will be measuring in for the Level and Temperature Channels.

Once you have set the address, Baud rate, and any other desired settings, click the start button . This applies the address and settings to the logger, **AND starts the AquaVent logging internally.**

**If you do not want your AquaVent to log internally,** independent of MODBUS operation, click the stop button .

(See [Section 2](#) for more information on the independent recording option).

## NOTE

Once you are finished programming your AquaVent logger, unplug the USB Connector Cable from the Wellhead. The Wellhead only communicates with the MODBUS network when just the RS-232 or RS-485 Connector Cable is connected.

### NOTE

Solinst only supplies the Connector Cables which connect to the SPX Wellhead. The customer must supply all wiring to connect to the MODBUS equipment, and must ensure correct pin-to-pin connection with the Connector Cables according to the pin-out diagrams provided.



Figure 3-3 RS-232 Connector Cable

### NOTE

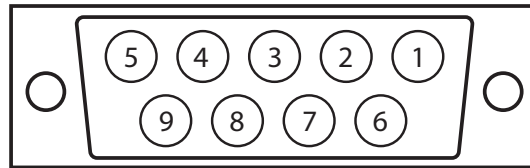
Solinst recommends that the RI signal of the customer equipment NOT be connected.

## 3.2 Wiring Overview

Refer to the wiring diagrams below for correct connection of the RS-232 and RS-484 Connector Cables to the MODBUS master, including external power to the Wellhead (10 - 24V).

### 3.2.1 RS-232 Connection

The MODBUS over RS-232 Connector Cable is equipped with a female DE-9 connector. The other end of the Connector Cable connects to the 12-Pin connection on the SPX Wellhead.



Shell: Cable Shield Drain

1: N.C.

2: RxD of RS-232 DTE      Industry Standard

3: TxD of RS-232 DTE      Industry Standard

4: N.C.

5: Circuit Ground          Industry Standard

6: N.C.

7: N.C.

8: N.C.

**9: (+) Power to Wellhead (Note: This pin would normally be RI (Ring Indicator))**

Figure 3-4 RS-232 Female DE-9 Connector (front view)

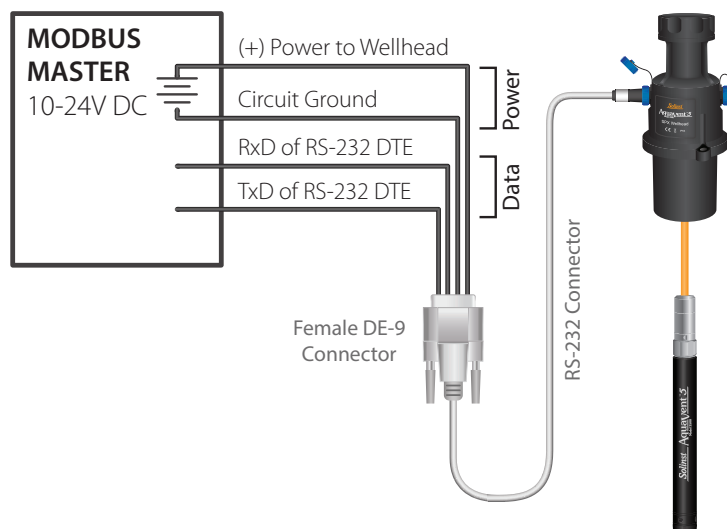


Figure 3-5 RS-232 Wiring Overview

### ! NOTE

Solinst only supplies the Connector Cables which connect to the SPX Wellhead. The customer must supply all wiring to connect to the MODBUS equipment, and must ensure correct pin-to-pin connection with the Connector Cables according to the pin-out diagrams provided.



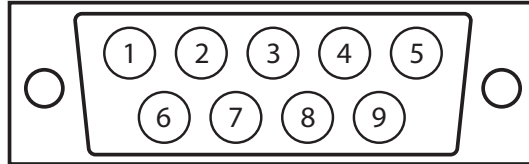
Figure 3-6 RS-485 Connector Cable

### ! NOTE

The AquaVent uses 2-wire Modbus. For 4-wire equipment, connect customer wires as labelled.

### 3.2.2 RS-485 Connection

Two-wire MODBUS over RS-485 Connector Cable is equipped with a male DE-9 connector. The other end of the Connector Cable connects to the 12-Pin connection on the SPX Wellhead.



Shell: Cable Shield Drain

1: Circuit Ground

2: (+) Power to Wellhead

3: N.C.

4: N.C.

5: RS-485 B/B'/Z/+, MODBUS D1

6: N.C.

7: N.C.

8: N.C.

9: RS-485 A/A'/Y/-, MODBUS D0

Figure 3-7 RS-485 Male DE-9 Connector (front view)

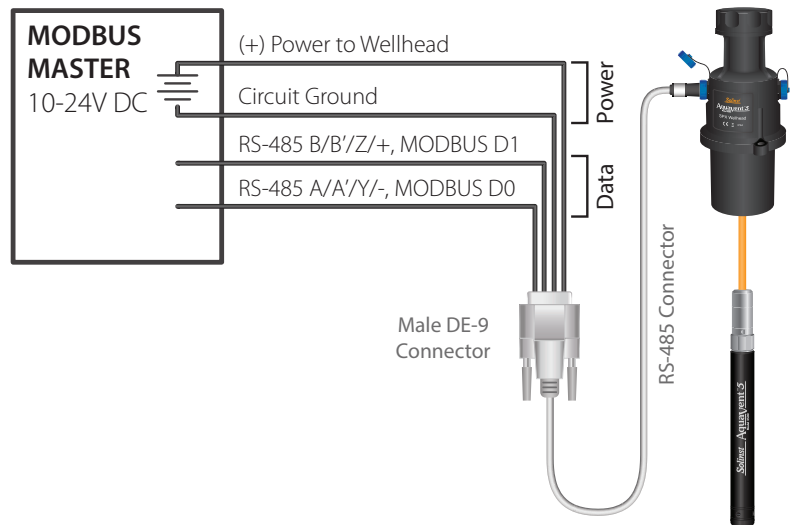


Figure 3-8 RS-485 Wiring Overview

### ! NOTE

See the [Vented Dataloggers User Guide](#) for detailed AquaVent installation Instructions.

### 3.3 AquaVent Setup and Startup

#### 3.3.1 AquaVent Setup

- 1) Be sure that the AquaVent logger is properly connected to the SPX Wellhead using a Vented Cable. See the [Vented Dataloggers User Guide](#) for detailed AquaVent installation Instructions.
- 2) Be sure the RS-232/RS-485 Connector Cable is connected to the 12-pin connection on the SPX Wellhead (Ensure you have unplugged the USB Connector Cable after using the AquaVent with Levellogger Software).
- 3) Using the figures in [Section 3.2](#), connect the RS-232 or RS-485 Connector Cable to the MODBUS equipment. Ensure you have correct pin-to-pin connection and external power supplied to the Wellhead.
- 4) Turn on power to the SPX Wellhead. The SPX Wellhead LED should emit the following signals in order:
  - Tricolour spin (3 cycles) to indicate a power-on event.
  - Upon entering Field mode, a rapid 10 cycle alternation of Yellow/ Green colour indicates that communication has been established with the AquaVent logger, and is running with the Baud rate and MODBUS device address prescribed by the AquaVent settings (which can only be changed by Solinst Levellogger PC Software).
  - Upon entering Field mode, a rapid 10 cycle alternation of Yellow/ Red colour indicates that the Wellhead has come online WITHOUT establishing communication with the AquaVent logger, and is running with the default Baud rate and MODBUS device address.
- 5) The SPX Wellhead and the AquaVent logger are physically and properly hooked up to the MODBUS network.



Figure 3-9 AquaVent MODBUS Installation

### 3.3.2 AquaVent Startup Notes

#### Wellhead Clock

The SPX Wellhead maintains its own RTC (Real Time Clock). The RTC is synchronized to the AquaVent logger on power-up (defaults to a start-up time of 0:00:00.0, Jan. 01, 1970). When the time is requested, the RTC will be re-synchronized to the AquaVent logger, if it has not been synchronized recently.

If an AquaVent logger is changed, it is recommended to sync the Wellhead clock to the new logger. This can be done by disconnecting the RS232/RS485 Connector Cable from the Wellhead, then re-connecting after 30 seconds.

Since the Wellhead clock is used to provide time-stamps with real-time data, and may be less accurate than the AquaVent logger clock, it is recommended to periodically force a re-sync for best accuracy. This can be done by programming the MODBUS master to read the "current time" register set, about once per day.

#### Power Usage

Upon initial power-up, or after receiving any properly addressed MODBUS command over the RS-232 interface, the MODBUS RS-232 interface is powered up, waiting to receive the first command. With the MODBUS RS-232 interface in standby, the Wellhead draws about 3mA while connected to the customer equipment, depending on how much that equipment loads the data line.

If a properly addressed command is received through RS-485, then the RS-232 interface enters power saving mode, in which it no longer transmits any voltage (not a valid line condition) onto the data line. At this time the Wellhead idles with a supply draw of about 30uA, while still listening to the RS-232 interface.

## **4 AquaVent MODBUS Operation**

Once the AquaVent system has been installed and powered up, you can start configuring it.

The user is expected to have working knowledge of the MODBUS protocol. For more information on MODBUS, visit: [www.modbus.org](http://www.modbus.org), where you can also find the references listed in [Section 7](#).

### **! NOTE**

Each MODBUS controller will have specific requirements for configuration. You will need to refer to the documentation specific to the equipment you are using.

### **4.1 AquaVent MODBUS Configuration Overview**

Apply the settings according to the requirements of the MODBUS controller.

The communication settings for the SPX Wellhead always use Serial, RTU mode, 8 data bits, even parity, and one stop bit, and transfers data in Big Endian. The MODBUS registers do not provide a means of changing any of these communication settings.

Using the Register Map in [Section 4.4](#), enter the desired register addresses.

Once you have entered the register map, you can start issuing function codes or set up a polling schedule. See [Section 4.3](#) for supported MODBUS functions.

**! NOTE**

The MODBUS master must be able to recognize floating point data types.

**! NOTE**

See [Appendix 6.1](#) for more information on Endian Order and Byte Packing.

**! NOTE**

A MODBUS exception response will be returned unless the MODBUS request matches the exact number of registers associated with that data type. See [Section 4.5.3](#) for Exception Codes.

## 4.2 AquaVent MODBUS Data Types

The following data types are used by the AquaVent 5 and AquaVent:

| Name   | Registers  | Comment  |
|--|------------|--|
| int16  | 1          | 16 bits, 2's compliment binary   |
| uint16   | 1          | 16 bits, natural binary  |
| bits16   | 1          | 16 bit field, typecast as uint16   |
| int32  | 2          | big endian, 32 bits, 2's compliment binary   |
| uint32   | 2          | big endian, 32 bits, natural binary  |
| bits32   | 2          | big endian, 32 bit field, typecast as uint32   |
| single   | 2          | big endian, 32 bits, IEEE-754 single precision floating point number   |
| nString  | int(n+1)/2 | Byte-packed (first byte in Msb) UTF-8 (variable length) Unicode; this string type has a fixed length (specified in the register map) that is expressed in units of bytes, not characters or registers. Beyond the end of the text, any unused bytes must be padded with null characters. Strings that are too long will generate a MODBUS exception code.            |
| timeGMT<br>(six uint16 values packed together) | 6          | Method of reading/writing current time. Universal time is presented in the familiar GMT format, which used ephemeris seconds and no leap seconds. This time format uses 24 hour time (no AM/PM), and assumes no DST, and no time zones. From Msb to Lsb, the registers (uint16) contain: Year, Month, Day, Hour, Minute, Seconds, and all must be accessed together. |
| Array  |            | An array contains elements, all of the same data type, such that each element occupies the same number of MODBUS registers. Any number of elements may be accessed together, starting at any element.  |

### 4.3 AquaVent Supported MODBUS Functions

The AquaVent only uses MODBUS functions that operate on registers (16 bits) and transfers data in Big Endian. The AquaVent only supports holding registers, and consequently only MODBUS functions that deal specifically with holding registers.

The following are the supported MODBUS functions:

**03: Read Holding Registers (code 0x03):** this command reads one or more registers.

**06: Write Holding Register (code 0x06):** this command sets a single register.

**10: Write Holding Registers (code 0x10):** this command sets one or more registers.

**16: Mask Write Register (code 0x16):** this command will set and/or clear one or more bits in a single register (see [Section 4.5.1](#) for example of where used).

*It is recommended to refer to the document: MODBUS Application Protocol Specification V1.1b, December 28, 2006, found at [modbus.org](http://modbus.org), for more information on the specific function codes.*

### 4.4 AquaVent MODBUS Register Map

The following MODBUS registers are specific to the AquaVent 5 and AquaVent.

Legend: Data Held in Logger Data Held in Wellhead Constant Data R/W=Read/Write

| Register      | Size | R/W | Type    | Example            | Description   |
|---------------|------|-----|---------|--------------------|---|
| <b>Header</b> |      |     |         |                    |   |
| 40001         | 1    | R   | uint16  | 2                  | Register Map Version (1=AquaVent, *2=AquaVent 5)                          |
| 40021-022     | 2    | R   | uint32  | 1234567            | Serial number   |
| 40041-052     | 12   | R   | nString | AquaVent<br>Logger | Product ID (24 characters)  |
| 40061         | 1    | R   | uint16  | 1                  | Hardware version: 1   |
| *40071-72     | 2    | R   | nString | XV                 | Hardware Identification (4 characters)                                    |
| 40081-084     | 4    | R   | nString | "M10"              | Model ID (8 characters)   |
| 40121         | 1    | R   | uint16  | 770                | Software Version: Major*256+Minor<br>E.g. Version 3.002 reads 3*256+2=770 |
| 40161-176     | 16   | R   | nString | "000123"           | Inst. Num./Proj. ID (32 bytes)  |
| 40177-200     | 24   |     |         |                    | Reserved  |
| 40201-216     | 16   | R   | nString | "WELL 15"          | Location name (32 bytes)  |
| 40217-240     | 24   |     |         |                    | Reserved  |
| 40301         | 1    |     |         |                    | Reserved  |
| 40241-242     | 2    | R   | single  | 43.6736            | Latitude Coordinate (degrees, neg.=S)                                     |
| 40261-262     | 2    | R   | single  | -79.4144           | Longitude Coordinate (degrees, neg.=W)                                    |
| 40321-322     | 2    | R   | single  | 987.123            | Current Battery Charge (mAh)  |
| 40341         | 1    | R   | uint16  | 73%                | Battery %   |
| 40361-362     | 2    | R   | single  | 2.96433            | Battery Voltage (V)   |
| 40401-402     | 2    | W   | bits32  | 0...01             | Test Vector (see <a href="#">Section 4.5.2</a> )                          |
| 40421-422     | 2    | R   | bits32  | 0...01             | Test Results (see <a href="#">Section 4.5.2</a> )                         |
| 40441         | 1    | R/W | bits16  | 0...00000000       | Device Status (see <a href="#">Section 4.5.1</a> )                        |
| 40442-580     | 138  |     |         |                    | Reserved  |
| 40581         | 1    | R   | uint16  | 2                  | Num. of parameters (2 for AquaVent)                                       |

\*Only available with AquaVent 5 Wellheads

#### ! NOTE

See [Appendix 6.2](#) for more information on register offset assumptions.

#### ! NOTE

See [Section 3.1](#) to change the units that the AquaVent is using for measuring level and temperature.

#### ! NOTE

Longitude and latitude coordinates will only be available if the AquaVent was previously used with the Solinst Levellogger App, or if you have written them to the logger.



| Register  | Size | R/W | Type    | Example       | Description   |
|---|------|-----|---------|---------------|---|
| <b>Array of process variables – duplicated in the parameter block registers.</b>                  |      |     |         |               |   |
| 40601-602   | 2    | R   | single  | (-)yyymmdd    | DateStamp (neg means yr=19xx)<br>Note: If the year is wrong, check to see if you need to reset the AquaVent time. |
| 40603-604   | 2    | R   | single  | hhmmss        | TimeStamp (military format)   |
| 40605-606   | 2    | R   | single  | 30.5          | Ch1 Value   |
| 40607-608   | 2    | R   | single  | 676.4         | Ch2 Value   |
| 40609-700   | 92   |     |         |               | Reserved for another 46 process variables   |
| <b>Modbus Communications</b>  |      |     |         |               |   |
| 40701   | 1    | R   | uint16  | 199           | Device Address (1-247, default=1)   |
| 40721   | 1    | R   | uint16  | 18            | Serial Communications Configuration   |
| 40741   | 1    | R   | uint16  | 1000          | EOM (End Of Message for ASCII mode) timeout (1000-15000ms, default=1000)  |
| 40761   | 1    | R   | uint16  | 1             | Max allowed baud rate id (0-7)  |
| 40781   | 1    | R   | uint16  | 256           | Max PDU (bytes of payload, not characters)  |
| 40801-802   | 2    | R/W | uint32  | 74585638      | Good message counter  |
| 40821   | 1    | R/W | uint16  | 1             | Bad message counter   |
| 40841   | 1    | R/W | uint16  | 5             | Exception response counter  |
| <b>timeGMT - Current time, Universal time, (no DST), non-leaping ephemeris seconds.</b>           |      |     |         |               |   |
| 40901-906   | 6    | R/W | uint16  |               | Year (0~65535)  |
|   |      |     | uint16  |               | Month (1~12)  |
|   |      |     | uint16  |               | Day (1~31, depending on month)  |
|   |      |     | uint16  |               | Hour (0~23)   |
|   |      |     | uint16  |               | Minute (0~59)   |
|   |      |     | uint16  |               | Second (0~59)   |
| <b>Parameter blocks – also see array of process variables, data duplicated for easier polling</b> |      |     |         |               |   |
| <b>Channel 1 Parameter Block</b>  |      |     |         |               |   |
| 41001-002   | 2    | R   | single  | 30.5          | Ch1 Value   |
| 41021-023   | 3    | R   | nString | "degC"        | Ch1 Units (6 bytes)   |
| 41041-056   | 16   | R   | nString | "Temperature" | Ch1 Parameter (32 bytes)  |
| 41057-200   | 144  |     |         |               | Reserved  |
| <b>Channel 2 Parameter Block</b>  |      |     |         |               |   |
| 41201-202   | 2    | R   | single  | 676.4         | Ch2 Value   |
| 41221-223   | 3    | R   | nString | "Cm"          | Ch2 Units (6 bytes)   |
| 41241-256   | 16   | R   | nString | "Level"       | Ch2 Parameter (32 bytes)  |
| 41261-262   | 2    | R   | single  | 200.7         | Ch2 Offset  |
| *41281-283  | 3    | R   | nString | "mm"          | Ch2 OffsetUnits (6 bytes)   |
| 41263-400   | 138  |     |         |               | Reserved  |
| 41401-46000   | 4600 |     |         |               | Reserved for another 23 parameter blocks  |

\*Only available with AquaVent 5 Wellheads

## 4.5 AquaVent MODBUS Status and Diagnostic Codes

### 4.5.1 Device Status

There are two types of Device Status bits. Both types of bits remain in the cleared state during normal operation, resulting in a register value of zero. The “State” type bits simply indicate a current condition, and will be cleared only when that condition is resolved. The “Event” type bits are set by an event, and will remain set until the client writes to the register, clearing the bit, as a way of acknowledging the event. It is suggested that you make use of the MODBUS function 0x16 (Mask Write Register). Keep in mind that an “Event” bit only represents device history as far back as the last time it was cleared.

Some events are normal, and only require acknowledgement to clear them. These are called “Event-Acknowledge” bits. Other events require intervention, which is enacted when you clear the bit. These are called “Event-Retry” bits.

Device Status bit map:

| Bit  | Type        | Description  |
|------|-------------|--|
| 0    | Event -Ack. | Power cycle (power-up event) detected.             |
| 1    | State       | Communication setting not synchronized with probe. |
| 2    |             | Reserved   |
| 3    |             | Reserved   |
| 4    |             | Reserved   |
| 5    |             | Reserved   |
| 6    |             | Reserved   |
| 7    | State       | BIT test in progress - some registers unavailable  |
| 8-15 |             | Reserved   |

### 4.5.2 Diagnostic Tests

Diagnostics are performed on the AquaVent logger by writing any non-zero value to the “Probe Test Vector” registers. Bits that are set in the “Probe Test Vector” registers determine which routines will be executed. The results will eventually be available for reading from the “Probe Test Results” registers, where the same bit positions are used, and a set bit indicates a failed test, while clear bits either were not selected for testing, or passed the test successfully.

Some of these tests will take a long time to process, and while waiting for completion, any MODBUS request involving the AquaVent logger will generate a MODBUS exception response with code 1, “Illegal Function”. To avoid this uncertainty, you may poll the “Device Status” register, and wait for the “Waiting for completion of logger diagnostic test(s)” bit to clear.

#### NOTE

During these tests the AquaVent logger will not be available for other functions.

Probe Test Vector & Probe Test Results bit map:

| Bit   | Test Description   |
|-------|--|
| 0     | Battery Voltage test (Bit 0). The external A/D's battery voltage will be used to generate a power supply fault whenever the voltage is below 3.1V.   |
| 1     | Program FLASH checksum test (Bit 1). The program memory computed checksum is compared to a 4-byte checksum stored in FLASH.  |
| 2     | Information FLASH checksum test (Bit 2). The information memory computed checksum is compared to a 2-byte checksum stored in FLASH.  |
| 3     | FRAM test (Bit 3). We will read/write various test patterns to the entire FRAM device. Any data in the FRAM is preserved. LTCG will only test portions of the FRAM.  |
| 4     | Logging Memory test 1 (Bit 4). Blocks of bytes are non-destructively written/verified for the first 512-byte segment of each 4096 byte logging memory page for the low-memory device.  |
| 5     | Logging Memory test 2 (Bit 5). As 5) except the high-memory device is tested.  |
| 6     | Temperature Sensor test (Bit 6). The raw temperature sensor A/D value is read. If this reading is between 740000H and E50000H, the test will pass.   |
| 7     | Pressure Sensor test (Bit 7). The raw pressure sensor A/D value is read. If this reading is between 810000H and B000000H, the test will pass.  |
| 8     | Full Logging memory test (Bit 8). Blocks of bytes are non-destructively written/verified for the entire logging memory range. A full memory test of all bytes typically takes about 5 to 10 seconds—depending on the amount of installed logging memory. |
| 9     | Reserved   |
| 10    | Bootloader FLASH checksum test (Bit 10). The bootloader memory computed checksum is compared to a 4-byte checksum stored in FLASH.   |
| 11-31 | Bits 11 to 31 are reserved for future use. These should be set to '0'.   |

### 4.5.3 Exception Response Codes

MODBUS Exception codes supported:

| Code                   | Name                | Comment  |
|------------------------|---------------------|--|
| Standard codes         |                     |  |
| 0x01                   | Illegal Function    | Unsupported MODBUS function code   |
| 0x02                   | Illegal Address     | Register range requested extends beyond supported register map   |
| 0x03                   | Illegal data value  | Request structure is invalid   |
| 0x04 - 0x05            | Reserved            |  |
| 0x06                   | Slave Device Busy   | The request could not be processed at this time, but is expected to be available immanently. The client may retry the request later. |
| 0x07-0x0B              | Reserved            |  |
| Solinst extended codes |                     |  |
| 0x80                   | Field Mismatch      | Address used doesn't match the beginning of a valid register field, or byte count doesn't match size of data type being addressed.   |
| 0x81                   | Write only register | Attempting to read from a write only register  |

| Code                                | Name                    | Comment   |
|-------------------------------------|-------------------------|---|
| Solinst extended codes continued... |                         |   |
| 0x83                                | Reserved                |   |
| 0x84                                | Write value             | Attempting to write data that is outside the valid range for the variable being addressed.    |
| 0x85-0xA4                           | Reserved                |   |
| 0xB0                                | Unknown Probe           | The wellhead translator doesn't recognize the probe type.                                     |
| 0xB1                                | Bad String              | Malformed string (incomplete/invalid Unicode/characters, incorrect termination/padding, etc.) |
| 0xB2                                | Long String             | String is too long to fit internally.   |
| 0xB3                                | Probe Timed-out         | Solinst bus response timed-out  |
| 0xB4                                | Bad probe CRC returning | Solinst Bus CRC error received at wellhead.   |
| 0xB5                                | Bad probe CRC sending   | Solinst Bus CRC error received by Probe (returning BCC offset by+ 7).                         |
| 0xB6                                | Probe exception         | Solinst Probe Other fault (Probe returns BCC offset by +56).                                  |

## **5 Maintenance and Troubleshooting**

### **5.1 Maintenance**

See the [Vented Dataloggers User Guide](#) for details on maintaining AquaVent loggers, Vented Cable and SPX Wellhead.

#### **5.1.1 Changing or Updating AquaVent Loggers**

If a new AquaVent logger is attached to the Wellhead, or you have changed the AquaVent settings, ensure that the Wellhead power is disconnected temporarily (30 seconds) by unplugging the RS-232/RS-485 Connector Cable from the side of the Wellhead. This is so the MODBUS translator will re-initialize, in order to cause the new MODBUS device address or settings to be retrieved from the AquaVent logger.

#### **5.1.2 Changing SPX Wellhead Batteries**

After replacing the batteries, it is recommended that the Solinst Levellogger PC Software Diagnostic Utility is used to reset the battery indicator. See the Vented Dataloggers User Guide for instructions to do this.

Power cycle the Wellhead by temporarily (30 seconds) unplugging the RS-232/RS-485 Connector Cable. This will cause the Wellhead to re-connect with the AquaVent logger.

Wait for the LED or status register to indicate a good connection. Try re-connecting if the first try fails.

Check or set the real time clock, through the "timeGMT" register.

#### **5.1.3 Firmware Upgrades**

See the Vented Dataloggers User Guide for details on firmware upgrades to the AquaVent logger and the SPX Wellhead.

The SPX Wellhead needs the power supply connected to the RS-232/RS-485 Connector Cable in order to perform the firmware upgrade.

While the firmware is being updated, the SPX Wellhead will ignore or provide time-out response to any requests from the MODBUS network.

## 5.2 Troubleshooting

The user is expected to have working knowledge of the MODBUS protocol. For more information on MODBUS, visit: [www.modbus.org](http://www.modbus.org).

### Compatibility

The AquaVent 5 is compatible with Solinst Levellogger PC Software Version 4.5 and up. The AquaVent is compatible with Solinst Levellogger PC Software Version 4.2 and up. However, Solinst always recommends using the latest software and firmware versions. See the Solinst Downloads page: <https://www.solinst.com/downloads/>

**Note: As features differ slightly between models, it is recommended to use the AquaVent 5 Wellhead with an AquaVent 5 logger.**

### AquaVent Logger Does Not Reply

The most common error is that MODBUS commands are being sent with an address which does not match the actual device address of the AquaVent logger being used, in which case the AquaVent will not reply. Try changing the MODBUS device address of the MODBUS command being sent.

Also ensure that you only have the RS-232 or RS-485 Connector Cable connected to the Wellhead. It will not work if the USB Connector Cable or a App/DataGrabber Connector Cable is also connected.

### **The recorder receives badly formatted replies from the MODBUS network of AquaVent loggers.**

Check that all AquaVent loggers on the network have different and unique MODBUS device addresses. Otherwise there will be data bus collisions and scrambled data will be returned on replies to the recorder or MODBUS datalogger. This is indicated if the yellow LED of two SPX Wellheads are flashing at the same time when only one MODBUS command has been sent. Change the device address of the AquaVent logger with the duplicate address.

Also ensure that you only have the RS-232 or RS-485 Connector Cable connected to the Wellhead. It will not work if the USB Connector Cable or a App/DataGrabber Connector Cable is also connected.

**LED flashes alternate Yellow/Red for 10 rapid cycles after powering up Wellhead.** (Indicates that the SPX Wellhead was unable to connect to the AquaVent logger.)

Check if the AquaVent logger is properly connected to the Wellhead through the Vented Cable. Check/replace the batteries in the Wellhead.

## 6 Appendix: Additional MODBUS Information

For more information on MODBUS, visit: [www.modbus.org](http://www.modbus.org).

### 6.1 Endian Order and Byte Packing

MODBUS transfers data in units of bits and registers. The AquaVent only uses MODBUS functions that operate on registers. A MODBUS register contains 16 bits.

MODBUS always transfers data in Big Endian, or Network order, regardless of the architecture of the system that is accessing the MODBUS. Normally the MODBUS protocol stack is compiled for the target system in such a way that it does whatever conversion is needed to move data properly between the application layer and the MODBUS. MODBUS always moves multiple registers over the communication line in ascending order of register address, and within each register the octets (bytes) are moved in order of descending arithmetic significance.

Floating point data types such as IEEE-754 will have their octets (bytes) ordered by significance in the same way as integers. This is also true in most modern systems (both big-endian and little-endian).

|   |      |                                   |   |                             |
|---|------|-----------------------------------|---|-----------------------------|
| Examples: (all three examples use the same register data).<br>Unicode String (UTF-8 encoding) "21"<br>or int32 0x3231C2B0,<br>or IEEE-754 sign*(mantissa/2 <sup>23</sup> +1)*2 <sup>(exp-127)</sup> = decimal +1.388753891*2 <sup>-27</sup> = +1.034702279e-8 |      |                                   |   |                             |
| Unicode string (UTF-8)  |      | 0x32 ('2')                        | 0x31 ('1')  | 0xC2B0 ('°')                |
| Byte Values   |      | 0x32                              | 0x31  | 0xC2 0xB0                   |
| Byte Significance (Transmission Order)  |      | High (First)                      | Low   | High Low (Last)             |
| Register Address (Transmission Order)   |      | n+0 (First)                       |   | n+1 (Last)                  |
| MODBUS Register Value   |      | 0x3231                            |   | 0xC2B0                      |
| Register Value (decimal)  |      | 12849                             |   | 49840 (or -15696 if signed) |
| Binary  |      | 0011 0010                         | 0011 0001   | 1100 0010 1011 0000         |
| IEEE-754 constituents   | Sign | Exponent (base 2, biased by +127) | Mantissa (parts per 2 <sup>23</sup> , 1<=norm.<2, less 1) |                             |
| IEEE-754 binary values  | 0    | 011 0010 0                        | 011 0001 1100 0010 1011 0000                              |                             |
| IEEE-754 decimal equiv.   | +    | 100                               | 3261104   |                             |
| IEEE-754 un-biased and/or de-normalized   | +    | -27                               | 1.388753891   |                             |

## 6.2 AquaVent Register Offset Assumptions

The MODBUS data model is specified as having up to four tables (Coils, Contacts, Analogue Inputs, and Holding registers) that each contain elements numbered from 1 to n, corresponding to the internal (within the PDU of a MODBUS message) addresses of 0 to n-1.

When applying this specification to a product, the holding registers are traditionally mapped to numbers from 40001 through 49999 (offset from the data model by 40000), in order to avoid any confusion with other tables, which traditionally use other offsets. The offset used depends on the element type that a MODBUS function is intended to deal with, and always results in mapping the first element of the table to an internal address of 0.

The AquaVent only supports holding registers, and consequently only MODBUS functions that deal specifically with holding registers. Since register 40001 represents element number 1 in the holding register table of the MODBUS data model, it is given the internal address of 0. See [Section 4.4](#) for the full AquaVent Register Map.



## 7 References

Modbus.org. *MODBUS over serial line specification and implementation guide V1.02, December 20, 2006*. Available [online]: <http://www.modbus.org/>

Modbus.org. *MODBUS Application Protocol Specification V1.1b, December 28, 2006*. Available [online]: <http://www.modbus.org/>



[www.solinst.com](http://www.solinst.com)

*High Quality Groundwater and Surface Water Monitoring Instrumentation*

Solinst Canada Ltd., 35 Todd Road, Georgetown, ON L7G 4R8  
Tel: +1 (905) 873-2255; (800) 661-2023 Fax: +1 (905) 873-1992  
[instruments@solinst.com](mailto:instruments@solinst.com)

**Solinst**<sup>®</sup>