More Info | Instructions | Get Quote

MODBUS User Guide

SDI-12/MODBUS Interface Cable

February 12, 2025

Get Quote

Solinst®

Table of Contents

1 Introduction

MODBUS is a serial communications protocol designed to allow a number of different devices to communicate with a single MODBUS controller.

The Solinst Model Levelogger 5, Barologger 5, Levelogger 5 Junior, Levelogger 5 LTC, Rainlogger 5, LevelVent 5, and Levelogger Edge Series dataloggers are able to act as sensors in a MODBUS network simply by using the Solinst SDI-12/MODBUS Interface Cable.

This User Guide focuses on configuration, communication, installation and description of the MODBUS protocol. See the separate User Guide if you are configuring your datalogger with SDI-12.

Note: The Interface Cable may be configured as an SDI-12 device, or as a MODBUS device, but never both concurrently.

1.1 Compatibility

The Solinst SDI-12/MODBUS Interface Cable is compatible with latest Solinst datalogger firmware:

Datalogger	Firmware Version
Levelogger 5	1.006
Barologger 5	1.006
Levelogger 5 Junior	1.006
Levelogger 5 LTC	1.006
Rainlogger 5	1.006
LevelVent 5	1.006
Levelogger Edge	3.004
Barologger Edge	3.004
Levelogger Junior Edge	3.004
LTC Levelogger Edge	1.003
Rainlogger Edge	3.001
LevelVent	1.000

Table 1-1 Compatible Datalogger firmware

Note: The SDI-12/MODBUS Interface Cables have firmware that is upgradable, see Section 6.2.

1.2 MODBUS Interface Cable Overview

Solinst 2-wire signals are converted to MODBUS signals by the Solinst SDI-12/MODBUS Interface Cable. One end of this cable has wires that connect to a MODBUS controller (See Section 3.2). The other end of the cable terminates in a Direct Read connection. This connects to the top of a Direct Read Cable for connection to a Solinst Levelogger, or the Wellhead connection of a LevelVent.

The Solinst datalogger has a self-contained battery and does not use +12V DC power. The MODBUS Interface Cable circuitry requires a 12V connection, which is to be powered by the customer equipment.

Note: The power may be cycled off and on, and normally, no energy will be used for an LED indication.

MODBUS communication uses the RTU mode over a half-duplex, aka 2-wire (plus ground) RS-485 interface. It uses the standard MODBUS default settings of 8-e-1 (8 bit, even parity, 1 stop bit), but also supports 8-o-1 and 8-n-2. The default device address is 1, but it supports all up to 247.

This interface uses 19,200 baud by default, but also supports 1200, 2400, 4800, 9600, 38,400, and 57,600 baud. Note that the idle current consumption of the cable will be slightly elevated while using MODBUS with baud rates above 9600.

The MODBUS interface applies under 1/256 of an RS-485 load unit to the bus, allowing a large number of devices. The MODBUS transmitter is current limited, but drives up to 5V differential, suitable for long distances.

The MODBUS receiver applies an internal "fail-safe" voltage bias, which causes an open or short circuit bus connection to appear "idle", from the perspective of the interface cable. While this causes the receiver to be less sensitive than what the RS-485 standard calls for (about 400mV differential instead of 200mV), this "fail-safe" biasing technique is a very common and desirable practice within the industry, and makes no practical difference to system performance in all but extreme cases.

In the standard product configuration, the MODBUS interface, does NOT contain a bus terminator. It is the responsibility of the installer to determine if, where, and what types of bus terminators may be needed in the system as a whole. Similarly, while the MODBUS Interface Cable is supplied with some length of wire, it is up to the installer to suitably limit the length of any wire configured as a stub, anywhere along their bus infrastructure.

Note: When powering on, provided that the cable is not in Off-line mode, there will be a slight delay before the cable goes on-line with the MODBUS controller.

Solinst communication between the Interface Cable and the Solinst datalogger always runs the Solinst protocol at 9600 baud, using the device address 1. Solinst Levelogger PC Software is used to specify the device address.

MODBUS Interface Cable Technical Specifications	
Standard Compliance	RTU over 2-wire RS-485, Non-terminated, Rx fail-safe bias for open/ short, < 1/256 unit load.
+12V Input	7V to 30V DC
Average Current Draw While Taking Readings	30 mA
Idle Power Draw	Baud < 9600 is 17uA @ 12V Baud >= 9600 (only applicable to MODBUS operation) is 25uA @ 12V
Direct Read Cable – Max Length	450 m (1500 ft)
Vented Cable – Max Length	150 m (500 ft)
MODBUS Cable Length – Max Length	60 m (200 ft)
Operating Temperature	-40°C to 80°C
Ingress Protection Level	IP65

Table 1-3 SDI-12/MODBUS Interface Cable Technical Specifications

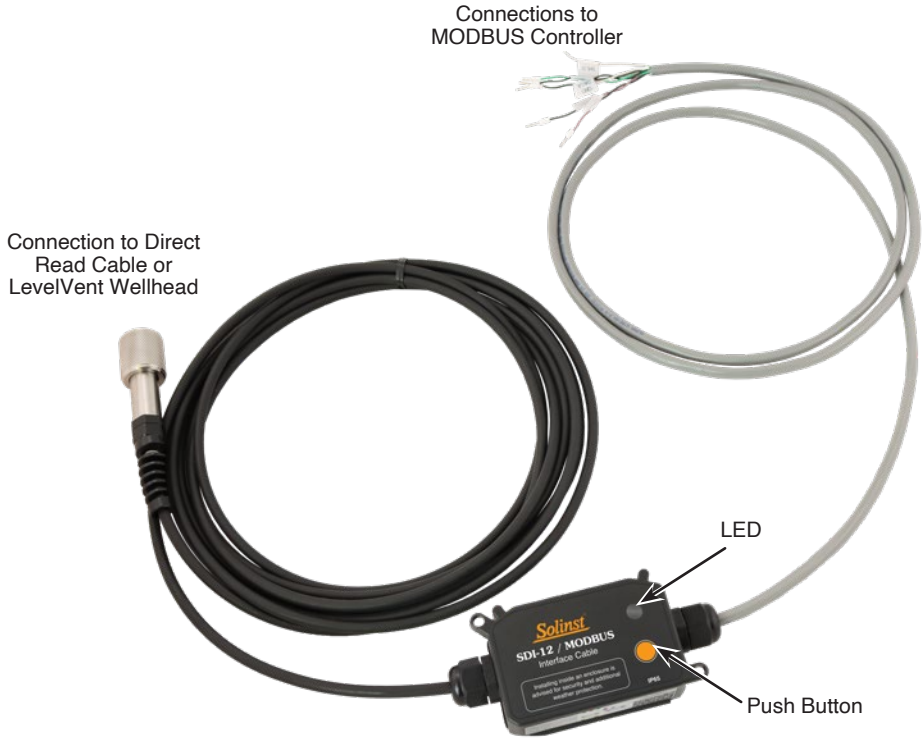

Figure 1-1 SDI-12/MODBUS Interface Cable

1.3 MODBUS Interface Cable Push Button and LED Status

Note: See the label of the SDI-12/MODBUS Interface Cable for a summary of LED status definitions.

Push Button Control and LED Signals for Commissioning and Firmware Updates

The SDI-12/MODBUS Interface Cable may operate in Firmware Upgrade mode, SDI-12 mode, MODBUS mode, or Off-line.

Solid White – Button is pressed, ready to select BIT (Built-In Test) and datalogger synchronization, by releasing button.

Blinking White – Button has been released while LED was White. BIT and datalogger synchronization will begin after a time delay.

Solid Yellow – Button is still pressed, waiting for Firmware Upgrade (press well over 20 seconds), or release, whichever comes first.

Blinking Yellow – Button has been released with no action selected. Waiting for a time delay before resuming Normal operation.

Note: The button press will not respond correctly until after Normal operation resumes.

Communication (Normal Operation) LED signals:

Short blink Yellow – Command (includes broadcast) accepted from bus (and sending SDI-12 ack when appropriate).

Short blink Green – Normal response sent to commanding bus.

Short blink Magenta – Exception response sent to MODBUS.

Short blink Cyan – Message sent to datalogger.

Short blink Blue – Message received from datalogger.

BIT (Built-In Test) LED signals:

Cycling Blue/Red/Green – BIT (Built-In Test) in progress.

Long blink Green – BIT success.

Long blink White – BIT failure(s) - Followed by a series of Short Red blinks, of the following counts, with a pause in between:

1 - Logger communication and type recognition test.

2 - Cable voltage out of range test.

3 – Contact Solinst.

After blinking out the failure code(s), there will be a long pause before sending other signals.

Datalogger Synchronization (settings configure bus communications) LED signals:

Long blink Yellow – datalogger type un-recognized, communication is Off-line. Try a Firmware Upgrade, to accommodate newer datalogger types.

Long blink Red – datalogger not found, communication is Off-line.

Long blink Green – Configured for SDI-12 operation.

Long blink Blue – Configured for MODBUS operation.

Power-Up LED signals:

Long blink Red – Internal problem with Cable.

Firmware Upgrade (BSL - Boot Strap Loader) LED signals:

Solid Blue – Firmware Upgrade Mode.

Short blink Cyan – Msg out from BSL to upgrader host.

Short blink Green – Msg into BSL from upgrader host.

Other LED signals:

Solid Yellow – Low voltage lockout – May be followed by an attempted re-boot, which possibly may repeat, causing the appearance of a yellow LED flashing very slowly. Note that when power is shut off from the cable (as in the common practice of power cycling to save energy between samples), the yellow LED can only expend the small internally stored energy, not energy from the external battery.

1.4 MODBUS System Components

1.4.1 Levelogger 5 Components

The Solinst Model 3001 Levelogger 5 requires the following components to complete a MODBUS monitoring system:

- Levelogger 5
- MODBUS Interface Cable
- Direct Read Cable
- PC Interface Cable (for programming the Levelogger) (Desktop Reader 5 or Field Reader 5 can also be used)
- Levelogger PC Software (free download on solinst.com)
- (User supplied equipment)

Note: For more information about the Levelogger, PC Interface Cable and Direct Read Cable, see the Levelogger User Guide.

Figure 1-2 Levelogger MODBUS System Components

1.4.2 LevelVent 5 Components

The Solinst Model 3250 LevelVent 5 requires the following components to complete a MODBUS monitoring system:

- LevelVent 5
- MODBUS Interface Cable
- Vented Cable
- LevelVent 5 Wellhead
- PC Interface Cable (for programming the LevelVent)
- Levelogger PC Software (free download on solinst.com)
- (User supplied equipment)

Note: For more information about the LevelVent, Wellhead, PC Interface Cable, and Vented Cables, see the LevelVent User Guide.

Figure 1-3 LevelVent 5 MODBUS System Components

2 Solinst Datalogger Independent Recording Option

Note: For best operation, have the Solinst dataloggers not in logging mode, so all measurements are controlled and initiated by the MODBUS controller. The datalogger can be used in logging mode in the MODBUS network, though it may require an occasional measurement retry by the controller if the datalogger happens to be busy at that moment.

Solinst dataloggers have the ability to record and store readings in their internal memory, independent from the MODBUS network, while connected to an MODBUS controller. Before connecting the Solinst datalogger to the MODBUS Interface Cable and controller, it can be programmed and started using Solinst Levelogger Software.

Note: See Levelogger and LevelVent User Guides for detailed operating instructions.

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

Note: The Solinst datalogger uses its internal battery each time the MODBUS controller requests a reading. The internal battery will drain more quickly if the datalogger is also set to record independently.

Each time the MODBUS controller asks the datalogger for a current reading, the internal datalogger battery is used to transmit the information to the MODBUS datalogger. The datalogger will also be using its internal battery if programmed to record independently – draining the battery more quickly. If the datalogger independent recording option is not used, the internal battery will only be used for communication with the MODBUS network.

2.1 Data Downloading and Programming in Field

If the datalogger has been programmed to record on its own independent schedule, data can be downloaded using a laptop PC and PC Interface Cable, Levelogger 5 App Interface, SRU or DataGrabber 5 in the field (see separate operating instructions). Temporarily disconnect the datalogger from the MODBUS Interface Cable, preferably in between recordings by the MODBUS controller. Connect a laptop, Levelogger 5 App Interface, SRU or DataGrabber 5 to download the independently recorded data.

If the datalogger is disconnected from the MODBUS network, and the MODBUS controller tries to communicate with the datalogger, the Red LED will flash as described in Section 1.3. This does not disrupt the MODBUS controller or the rest of the sensors in the network. The Red LED will stop flashing once the datalogger is reconnected.

After the data download is complete, the datalogger is easily connected back into the MODBUS network without disruption. See Section 6.1.

3 MODBUS Setup and Installation

3.1 Setting the MODBUS Device Address

To set the device address for the Solinst datalogger, it must be connected to the Levelogger PC Software. A Levelogger communicates to the PC using a PC Interface Cable connected to a Direct Read Cable (or use a Desktop Reader 5/Field Reader 5). A LevelVent communicates with the PC using a PC Interface Cable connected to the LevelVent Wellhead.

Note: For information on downloading and starting the Levelogger PC Software, see the Levelogger User Guide.

Figure 3-1 Communicating with Levelogger Software

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

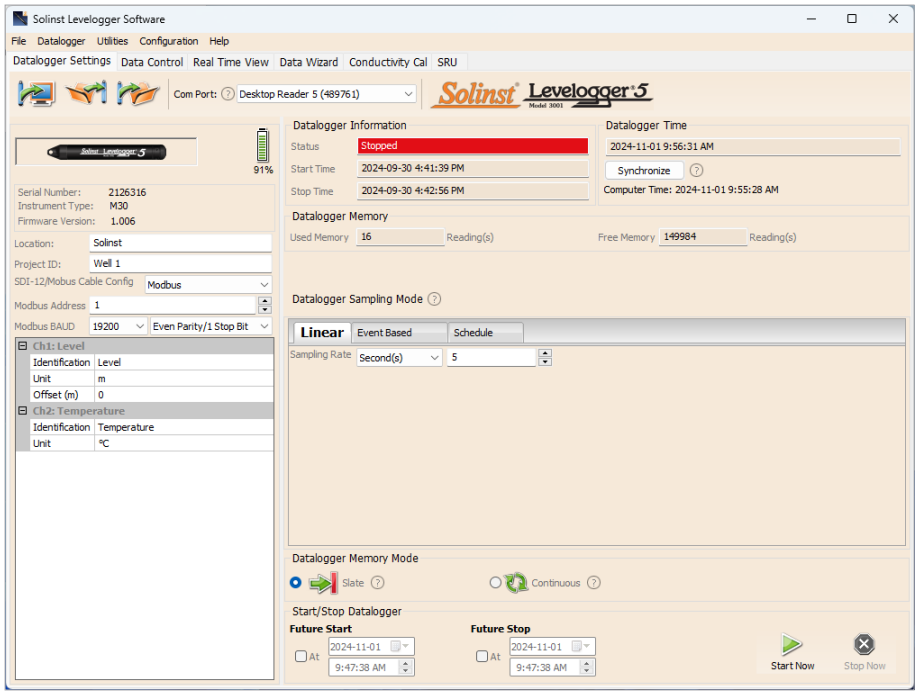

Figure 3-2 Levelogger 5 Dataloggers Settings Tab

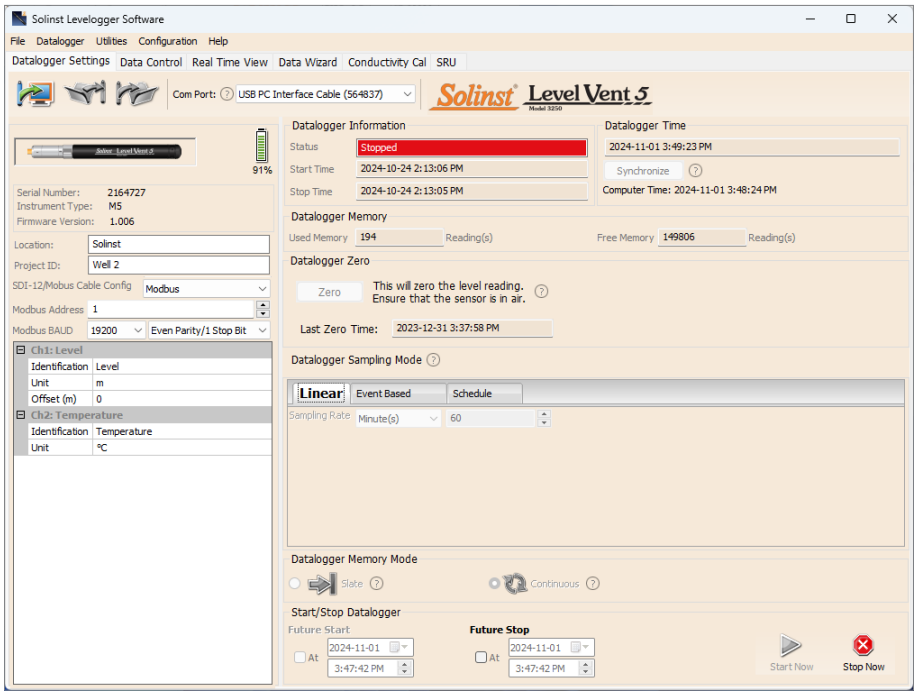

Figure 3-3 LevelVent 5 Dataloggers Settings Tab

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

Click  to retrieve the current settings from the connected datalogger.

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

Select MODBUS from the "SDI-12/MODBUS Cable Config" drop-down menu.

Select the desired address from the "MODBUS Address" drop-down menu.

Select the desired settings from the "MODBUS BAUD" drop-down menus.

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

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

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

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

The settings will be transferred from the Solinst datalogger to the MODBUS Interface Cable using the push button on the MODBUS Interface Cable once all appropriate connections are made. See Section 3.2.

3.2 MODBUS Wiring Installation

Note: Installing the SDI-12/MODBUS Interface Cable inside an enclosure is advised for security and additional water protection.

1) Be sure that the Levelogger is properly connected to the Direct Read Cable, or the LevelVent properly connected to the Vented Cable and Wellhead. See separate User Guides for more information.
2) Refer to the wiring diagram below for correct connection of the MODBUS Interface Cable to the MODBUS controller, including external power (12V).

   To avoid possible damage in case of a live (powered) network, always prevent the bare metal areas of wires and connectors from unintentionally touching each other, and use the following sequence:

   - When connecting, Ground first, Data next, Positive power last.
   - When disconnecting, Positive power first, Data next, Ground last.

   The network end of the MODBUS Interface Cable contains 6 wires, plus a shield connection. Use only the wires needed for MODBUS connection. You may leave the unused wire(s) open (securely insulated with tape) or grounded.

Note: The SDI-12/MODBUS Cable Power (supplied by the customer equipment) must be between 7V and 30V DC.

3) Connect the MODBUS Interface Cable to the Levelogger's Direct Read Cable, or the LevelVent Wellhead.
4) Press the button on the MODBUS Interface Cable for two seconds to bring it on-line.

Note: There will be a small delay between powering up the cable, and having the cable responsive to a MODBUS command.

Figure 3-1 Datalogger SDI-12 Wiring Overview

Note: Wires are arranged inside the cable as twisted pairs, where each pair has one black wire and one wire of a colour that is unique among all the pairs. In case the labels are missing from the wire, the colour scheme can be followed as shown in Fig. 3-1.

When MODBUS is being used, you need to consider if a bus terminator should be added between the MODBUS D0 and D1 wires (normally 120 Ohm resistors, 1/4 Watt minimum, on each end of the bus).

Other equipment manufacturers often mix up their A & B labelling on RS-485 data lines. The labelling of - & + seem to be more often in agreement between equipment manufacturers. When the data lines are at idle levels, + has a positive voltage with respect to -.

MODBUS data wires (D0 & D1, aka - & +, or A & B) may be swapped without risk of hardware damage, so if the bus signal polarity is in doubt, go ahead and try swapping to see if it works then.

Note: The Shield wire connection is not required for operation, but may be used to enhance protection against surges. Ideally, it should be connected to a chassis, earth ground, power ground, or the shield of the cable it is connecting to.

Note: Power Ground and Data Ground are connected internally. Solinst recommends also connecting them together at the terminal block, when it is convenient to do so, for maximum range and noise immunity.

It is also important (if not already in place) to have a connection between one of these Grounds and the reference ground or power ground of the equipment used as the MODBUS Controller.

The RS-485 standard requires 2 data + 1 common ground for each bus. If this is not considered during the installation, then the system may experience intermittent failures after installing Solinst equipment.

4 MODBUS Operation

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

Once the MODBUS cable has been installed, powered up, and commissioned with a datalogger set up for MODBUS protocol, and its communication settings and device address set, you can start issuing MODBUS commands to it. Apply the settings according to the requirements of the MODBUS controller. Using the Register Map (Section 5.5), enter the desired register addresses. Once you have entered the register map, you can start issuing function codes or set up a polling schedule.

Note that the MODBUS Device Address of this SDI-12/MODBUS Interface Cable must be unique among all devices on the MODBUS network, and the other MODBUS settings must be identical to other devices.

It can be helpful to have the LED of the SDI-12/MODBUS Interface Cable in view during the early parts of basic testing, which may be accomplished by having the MODBUS controller send out periodic commands to the Device Address of the SDI-12/MODBUS Interface Cable/datalogger, so that the LED will show a response as soon as it is hooked up and started successfully.

4.1 Activation and Verification

Testing should begin with the simplest case, which is reading a single MODBUS register, containing constant data (predictable), not relying on other communication (data is not from datalogger). The ideal test case is to Read the First (address 0) Holding Register, which is represented in this document as register number 400001 (your register numbering system may differ). The result will be a successful read, with the value of 3 (since at the time of writing this document, the production firmware uses Solinst MODBUS Register Map Version 3). If the number seems to be multiplied by 256 (for example, reading 768, while expecting 3), then the MODBUS controller needs to have its Byte-Swap control toggled, in order to correct the register reading values. If no value can be read, a good cable must have a problem with communication settings, commissioning, physical hookup, or power.

After that, by reading "Serial number", the Byte-Swap and Word-Swap controls of the MODBUS controller can be adjusted for reading Integer type data that spans over multiple registers. This also relies on successful communications between the SDI-12/MODBUS Interface Cable and the datalogger.

Similarly, by reading "Model ID (8 characters)" from 4 registers, the Byte-Swap and Word-Swap controls of the MODBUS controller can be adjusted for reading String type data.

Finally, by reading "Ch1 Value" from 2 registers, the Byte-Swap and Word-Swap controls of the MODBUS controller can be adjusted for reading Single-precision Floating point type data.

For best operation, have the Solinst dataloggers not in logging mode, so all measurements are controlled and initiated by the MODBUS controller. The datalogger can also be used in logging mode in the MODBUS network, though it may require an occasional measurement retry by the controller if the datalogger happens to be busy at that time. To change the logging mode of the datalogger, use the Solinst Levelogger PC Software (see Sections 2 and 3.1).

5 Supported MODBUS Functions

Read Holding Registers (code 0x03)

Write Holding Register (code 0x06)

Write Holding Registers (code 0x10)

Mask Write Register (code 0x16)

5.1 MODBUS Exception Response Codes

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	Slave Device Failure	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
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 re-try 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
0x82	Read only register	Attempting to write to a read only register
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 Logger	The translator doesn't recognize the datalogger 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	Logger Timed-out	Solinst bus response timed-out
0xB4	Bad logger CRC returning	Solinst Bus CRC error received.
0xB5	Bad logger CRC sending	Solinst Bus CRC error received by datalogger (returning BCC offset by+ 7).
0xB6	Logger Exception	Solinst datalogger Other fault (Probe returns BCC offset by +56).

5.2 Endian Order and Byte Packing

MODBUS transfers data in units of bits and registers. The MODBUS Interface Cable 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).

5.3 Data Types Used in MODBUS Registers

Note that a MODBUS exception response will be returned unless the MODBUS request matches the exact number of registers associated with the data type.

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
int64	4	big endian, 64 bits, 2's compliment binary
uint64	4	big endian, 64 bits, natural binary
single	2	big endian, 32 bits, IEEE-754 single precision floating point number
double	4	big endian, 64 bits, IEEE-754 double precision floating point number
nString	n/2	Solinst: Byte-packed (big endian) UTF-8 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.
timeY2K (single type for date and/ or time)	2	Alternate method of reading/writing current time. -Assumes 86,400 seconds every day, no time zones, no DST, and no leap seconds. -Presented as a (contiguous pair of) single data type value(s) in 2 or 4 registers: Date and/or Time of day (which should be accessed together). * Date uses 2 digits each (from left to right) for Year, Month, and Day. * Positive Date means yr=20xx, so add year 2000 to Date. * Negative Date means yr=19xx, so take absolute value, and add 1900. * Time of day uses 24 hour (no AM/PM) format, with 2 digits each (from left to right) for Hours, Minutes, and Seconds. Range 0 – 235959.
timeGMT (six unt16 values packed together)	6	Alternate method of reading/writing current time. -Presented in the familiar GMT format, which assumes 86,400 seconds every day, no time zones, no DST, and no leap seconds. * This format uses 24 hour time (no AM/PM). * From First to Last, the registers (uint16 type) contain: Year, Month, Day, Hour, Minute, Seconds. - All 6 registers must be accessed together.
Array		An array contains elements, all of the same data type, such thateach element occupies the same number of MODBUS registers. Any number of elements may be accessed together, starting at any element.

5.4 Register Offset Assumptions

- The MODBUS data model is specified as having of up to four tables (Coils, Contacts, Analogue Inputs, and Holding registers) that each contain elements numbered from 1 to n.
- These are accessed through addresses of 0 to n-1, within the PDU of a MODBUS message.
- When applying this specification to a product, the holding register numbers are traditionally offset from the data model, for example by 400000 (so the register numbers become 400001 through 465536), in order to avoid any confusion with the other tables. This has no effect on the address through which an element of any table is accessed.
- This product only supports holding registers, and consequently only MODBUS functions that deal specifically with holding registers. Since register 400001 represents element number 1 in the holding register table of the MODBUS data model, it is given the internal address of 0.
- Highlighting is as follows:
  - Data Held in Logger
  - Data Held in Cable
  - Constant Data

5.5 MODBUS Register Map

Register	Size	R/W	Type	Example	Description
Header					
40001	1	R	uint16	3	Register Map Version
400021-022	2	R	uint32	1234567	Serial number
400041-056	16	R	nString	LevelVent Logger	Product ID (32 characters)
400061	1	R	uint16	1	Hardware Version
400071-072	2	R	nString	XV	Hardware Identification (4 characters)
400081-084	4	R	nString	M10=>3	Model ID (8 characters)
400121	1	R	uint16	770	Software Version: Major*256+Minor E.g. Version 3.002 reads 3*256+2=770
400161-176	16	R	nString	"000123"	Inst. Num./Proj. ID (32 bytes)
400177-200	24				Reserved
400201-216	16	R	nString	"WELL 15"	Location name (32 bytes)
400217-240	24				Reserved
400241-242	2	R	single	43.6736	Latitude Coordinate (degrees, neg.=S)
400261-262	2	R	single	-79.4144	Longitude Coordinate (degrees, neg.=W)
400301	1				Reserved
400321-322	2	R	single	987.123	Current Battery Charge (mAh)
400341	1	R	uint16	73%	Battery %
400361-362	2	R	single	2.96433	Battery Voltage (V)
400401-580					Reserved
400581	1	R	uint16	2	Num. of parameters (Data Channels)
Logger Current time, Traditional (pre-UTC) standard (no DST) format, non-leaping seconds					
400591-94	4	R	timeY2K	241231235959	Date + Time, eg. 2024-12-31, 23:59:59

Register	Size	R/W	Type	Example	Description
Array of process variables – duplicated in the parameter block registers.					
400605-606	2	R	single	30.5	Ch1 Value
400607-608	2	R	single or timeY2K	676.4	Ch2 Value (or DateStamp for Rainlogger)
400609-610	2	R	single or timeY2K	3657.8	Ch3 Value (or TimeStamp for Rainlogger)
400611-700	92	Reserved for another 45 process variables			
MODBUS Communications					
400761	1	R	uint16	57600	Max allowed baud rate id (0-7)
400781	1	R	uint16	256	Max PDU (bytes of payload, not characters)
400801-802	2	R/W	uint32	74585638	Good message counter
400821	1	R/W	uint16	1	Bad message counter
400841	1	R/W	uint16	5	Exception response counter
Logger Current time, Traditional (pre-UTC) standard (no DST) format, non-leaping seconds.					
400901-906	6	R/W	uint16		Year (1970~2099)
			uint16		Month (1~12)
			uint16		Day (1~31, depending on month)
			uint16		Hour (0~23)
			uint16		Minute (0~59)
			uint16		Second (0~59)
Parameter blocks – also see array of process variables, data duplicated for easier polling					
Channel 1 Parameter Block					
401001-002	2	R	single	30.5	Ch1 Value
401021-023	3	R	nString	"degC"	Ch1 Units (6 bytes)
401041-056	16	R	nString	"Temperature"	Ch1 Parameter (32 bytes)
401041-062	2	R	single	2.5906	Ch1 Calibration (Rainlogger only, Units/Tip)
401063-200	137				Reserved
Channel 2 Parameter Block					
401201-202	2	R	single or timeY2K	676.4	Ch2 Value (or DateStamp for Rainlogger)
401221-223	3	R	nString	"Cm"	Ch2 Units (6 bytes)
401241-256	16	R	nString	"Level"	Ch2 Parameter (32 bytes)
401261-262	2	R	single	2.7	Ch2 Calibration (Reference Offset) in Meters
401263-400	137				Reserved
Channel 3 Parameter Block					
401401-402	2	R	single or timeY2K	3657.8	Ch3 Value (or TimeStamp for Rainlogger)
401403-600	197				Reserved
401601-406000	4400	Reserved for another 22 parameter blocks			

6 Maintenance and Troubleshooting

6.1 Changing or Updating a Solinst Datalogger (Hot-Swapping)

Hot-swapping a datalogger (replacing one with another WITHOUT performing the start up (Commissioning) process) should normally never happen. When it does happen, the communication settings from the datalogger will NOT be updated in the cable, but acquisition of the NEW datalogger identity can be performed remotely. See Section 4.1.

This is done by reading a MODBUS register for "Hardware Version" or "Hardware Identification" or "BootLoader Version" or "Software Version" or "Software Development ID" or "Current Battery Charge" or "Battery %" or "Battery Voltage" or "Num. of parameters". After the NEW Solinst datalogger Identity is acquired, it is stored in the non-volatile memory of the Interface Cable, so all data will flow correctly thereafter.

Note that the MODBUS communication parameters and device address CANNOT be changed remotely, and it is possible for a hot-swapped datalogger to function perfectly while assuming (but not adopting) the device address setting from a previous (forgotten) datalogger, so its MODBUS communication parameters and device address may change unexpectedly next time the push button commissioning method is finally employed.

Note that due to the non-volatile memory of the Interface Cable, there is no reason to issue any command for acquiring the datalogger Identity, unless a datalogger has been hot-swapped. Knowing this, you can avoid unnecessary energy consumption due to unnecessary repetition of commands.

6.2 Firmware Updates

The SDI-12/MODBUS Interface Cable been designed with firmware that is easy to update whenever useful new functions or other improvements become available, as with software releases.

To update the firmware in your sensor, go to the Solinst Website at: https://downloads.solinst.com where you can sign-in or register to download the firmware upgrade file that is contained within a Zip Archive. Ensure you unzip the Archive to access the firmware *.ssf file.

Note: It is important that the communication between the PC and the sensor is not interrupted during a firmware upload, so please make sure to close any other running programs, including screen savers, and do not disconnect the sensor before the upload is finished.

This process will also require the use of a Desktop Reader 5 or Field Reader 5, and the Firmware Upgrade Adaptor included with the SDI-12/MODBUS Interface Cable.

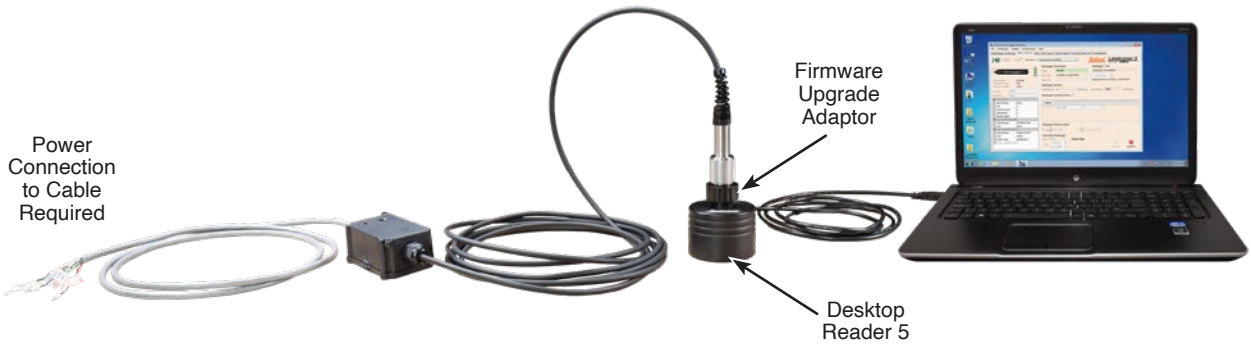

Figure 6-1 Firmware Upgrade Setup

To upload new firmware to a SDI-12/MODBUS Interface Cable, follow these steps:

1) Ensure your SDI-12/MODBUS Interface Cable is still connected to power.
2) Thread the SDI-12/MODBUS Cable to the Firmware Upgrade Adaptor, then place the optical end of the Adaptor in the Reader.
3) Open the Solinst Firmware Upgrade Utility that is downloaded along with Levelogger PC Software. Select the "Com Port" that the cable is connected to from the drop-down menu.
4) Click the 'Open' button, which should open a file dialog asking for the firmware file (*.ssf) to upload. Navigate to the directory where the firmware file was saved on your PC, then click on the file and click 'Open'.
5) Check the 'File Information' box to make sure the opened file is correct.
6) Press and hold the button on the SDI-12 Interface Cable until the LED turns Blue (in Firmware Upgrade mode).

Note: Even if the LED fails to respond to the push button, a very long (well over 20 second) press will still force entry into the Firmware Upgrade mode, at which time the LED should respond normally.

7) Click the 'Upload Firmware' button, to start the firmware upload process.
8) The LED now rapidly alternates between Green and Cyan colours, which means the firmware upgrade process is running. The green progress bar starts to fill up in the Firmware Upgrade Utility window.
9) When complete, the Cyan light turns off and the green progress bar fills the entire space in the Firmware Updgrade Utility window. A success message will also be displayed.

Note: When conducting a firmware upgrade, DO NOT interrupt the process prior to completion (This may take 2 to 4 minutes).

Note: If the above process fails, try starting the process again. If the LED cannot confirm entry into upgrade mode when the button is pressed, cycle the power, and try again. If the LED never lights up for the firmware upgrade process, verify that a suitable power source is being applied to the cable. If all the above fails, contact Solinst for assistance.

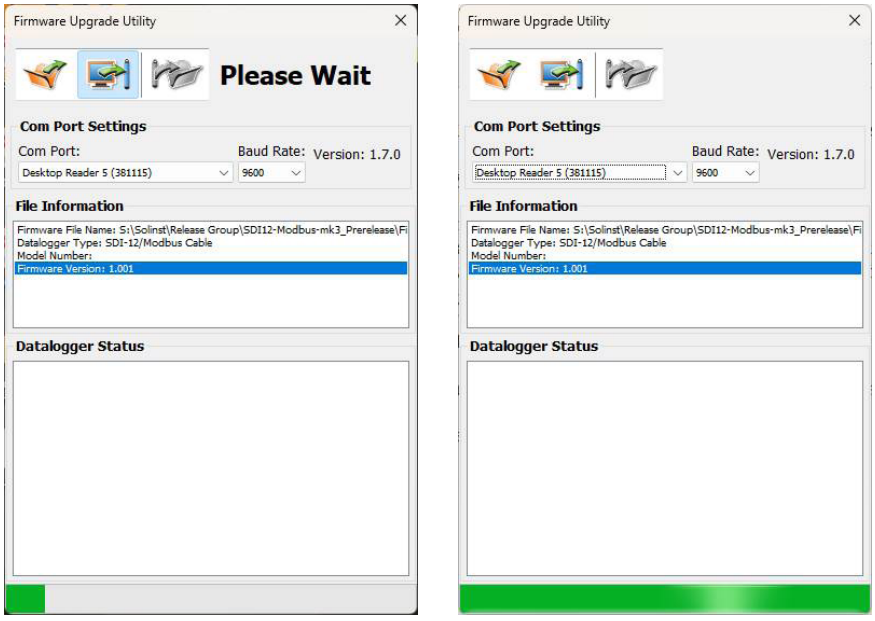

Figure 6-2 Firmware Upgrade Utility

6.3 Troubleshooting

Datalogger 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 datalogger being used, in which case the datalogger will not reply. Try changing the address of the MODBUS command being sent.

Another suggestion is to use a short press of the push button as a power indicator. If this triggers datalogger synchronization, be sure the resulting communication is On-line.

The controller receives badly formatted replies from the MODBUS network of datalogger.

Refer to MODBUS wiring installation section to ensure all wires are connected properly.

Check that all datalogger's on the network have different and unique device addresses. Otherwise there will be data bus collisions and scrambled data will be returned on replies to the MODBUS controller.

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/

Additional links:

https://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf

https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

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

Solinst®