User Guide

Models VP4-2310 and VP4-2810 ValuPoint 4 Programmable I/O for Modbus RTU

Rev. 1.0 – June 2013

VP4-2310/VP4-2810 User Guide Contents

User Guide

ValuPoint® 4-23 & VP4-28

Model VP4-2310 & Model VP4-2810
Programmable I/O for Modbus RTU Networks
Rev. 1.0 – June 2013

1 Overview

1.1 How to Use This Guide

Section 1 gives an overview of the VP4-2310 and VP4-2810 programmable I/O devices. Section 2 talks about installing the configuration software and connecting the VP4-2x10. Sections 3 through 11 are guides for each of the tabs found on the screen of the configuration software. Sections 12 through 14, and Appendix A through C, are reference material.

1.2 Important Safety Notice

Proper system design is required for reliable and safe operation of distributed control systems incorporating any Control Solutions product. It is extremely important for the user and system designer to consider the effects of loss of power, loss of communications, and failure of components in the design of any monitoring or control application. This is especially important where the potential for property damage, personal injury, or loss of life may exist. By using ANY Control Solutions, Inc., product, the user has agreed to assume all risk and responsibility for proper system design as

well as any consequence for improper system design.

1.3 Overview of the VP4-2310 and VP4-2810

Control Solutions' Model VP4-2310 ValuPoint® Programmable I/O for Modbus RTU operates as a master or slave for I/O expansion, or as a controller with soft PLC capability. This 4th generation ValuPoint® platform features 14 universal context sensitive inputs. When configured as an analog input, the continuously self-calibrating sigma-delta converter produces 15-bit resolution with high noise immunity. When configured as a discrete input, the converter switches to 10-bit resolution at 1,000 samples per second for fast response. ValuPoint 4-23 also includes 3 analog outputs, 6 Form A relay outputs, and pulse counting capability on all inputs.

Hardware Features of Model VP4-2310

- 14 Analog/universal inputs, software selectable types
  - 0-10VDC, thermistor, discrete, dry contact, pulse
  - 0.1% reference, up to 16-bit resolution
  - Continuously self-calibrating sigma-delta converter
  - Non-volatile totalizing count inputs
  (to 10Hz on all channels, to 1kHz on 2 channels)
  - 10-bit fast scan mode
- 3 Analog outputs
  - 4-20mA (0-20mA), 8-bit resolution, 500 ohm load max.
- 6 Discrete outputs
  - Form A relay
  - 2A @ 120VAC
  - 2A @ 30VDC
- Battery backed real time clock/calendar
- Modbus RTU at 4800 to 38400 baud
- Hardened RS-485 port
- 128KB non-volatile EEPROM configuration file capacity
- 64K Flash for User Program
- ARM 32-bit processor, 512K Flash, 56K RAM
- Powered by 18-30VDC or 24VAC 50/60 Hz
- 0.3A @ 24VDC
- DIN rail mounting, 100mm H x 105mm W x 60mm D
- -40C to +70C, 5%-95% RH non-condensing
- Certifications: FCC, CE, UL 916 Listed

Control Solutions' Model VP4-2810 ValuPoint® Programmable I/O for Modbus RTU operates as a master or slave for I/O expansion, or as a controller with soft PLC capability. This 4th generation ValuPoint® platform features 12 discrete inputs configurable as voltage input or dry contact closure to ground. The discrete inputs feature pulse count capability and non-volatile totalizing counter capability. The VP4-2810 also includes 16 open drain FET outputs capable of sinking 1A at up to 30VDC.

Hardware Features of Model VP4-2810

- 12 Discrete Inputs
  - Voltage sensor or dry contact closure to ground
  - Fast (1kHz) pulse count capability on 2 channels
  - Slow (10Hz) pulse count capability on all channels
  - Non-volatile totalizing count inputs
- 16 Discrete outputs
  - 30VDC 1A open drain FET

- Battery backed real time clock/calendar
- Modbus RTU at 4800 to 38400 baud
- Hardened RS-485 port
- 128KB non-volatile EEPROM configuration file capacity
- 64K Flash for User Program
- ARM 32-bit processor, 512K Flash, 56K RAM
- Powered by 18-30VDC or 24VAC 50/60 Hz
    - 0.3A @ 24VDC
- DIN rail mounting, 100mm H x 105mm W x 60mm D
- -40C to +85C, 5%-95% RH non-condensing
- Certifications: FCC, CE

The VP4-2310 and VP4-2810 are configured by writing various special Modbus registers accessible as holding registers. The VP4 configuration tool simplifies this process by providing a graphical tool for automatically writing all of the necessary registers by selecting various options on the screen and then clicking the 'Write' button. Configuration is saved in non-volatile memory. Once configured, you can save your entire configuration to an XML format file on your PC for later re-use. To replicate the same configuration, simply load the XML file back into the configuration tool and click the 'Write All' button.

1.4 Warranty

This software and documentation is provided "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fi tness or merchantability for a particular purpose. Control Solutions may make improvements and/or changes in this documentation or in the product(s) and/or the program(s) described in this documentation at any time. This product could include software bugs, technical inaccuracies, typographical errors, and the like. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the software.

1.5 Required License Information

The VP4-2310/VP4-2810 configuration and line programming tools include the SmartWin library (http://smartwinlib.org) under the following terms:

License agreement for SmartWin++ (BSD license)

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

i.CanDrawIt includes, licensed under LGPL, TinyCAD, Copyright 1994-2009 Matt Pyne. Source code is available at http://tinycad.sourceforge.net. Open source products included under either GPL or LGPL include TinyCAD v2.70; Unicode/Font Conversions: iconv.dll version 1.9.0.0; PNG Image Support: libpng13.dll version 1.2.8.0; Image compression support: zlib1.dll version 1.2.1.0.

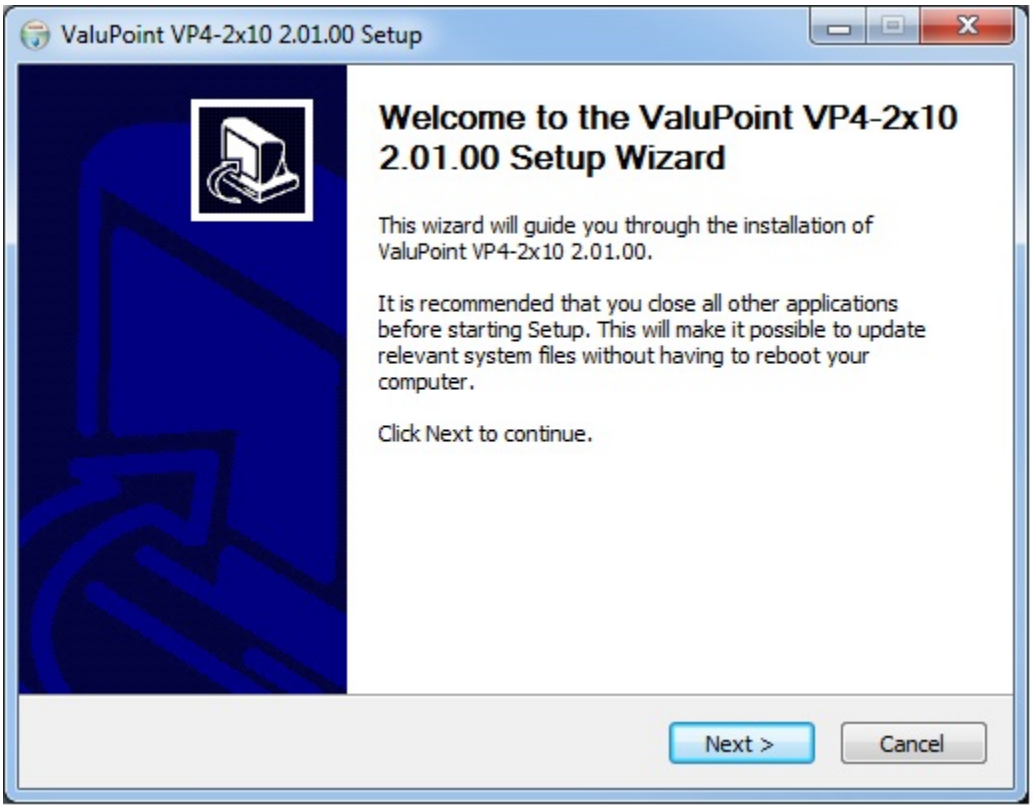
2 Installation and Connections

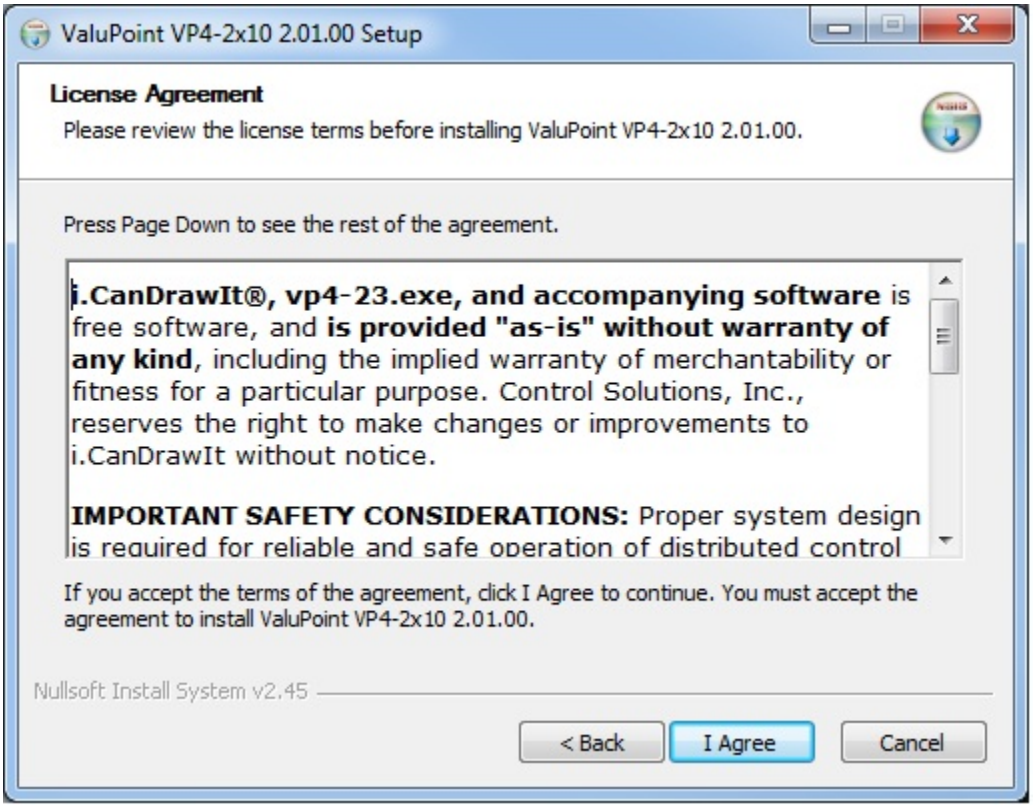
2.1 Installing the configuration software

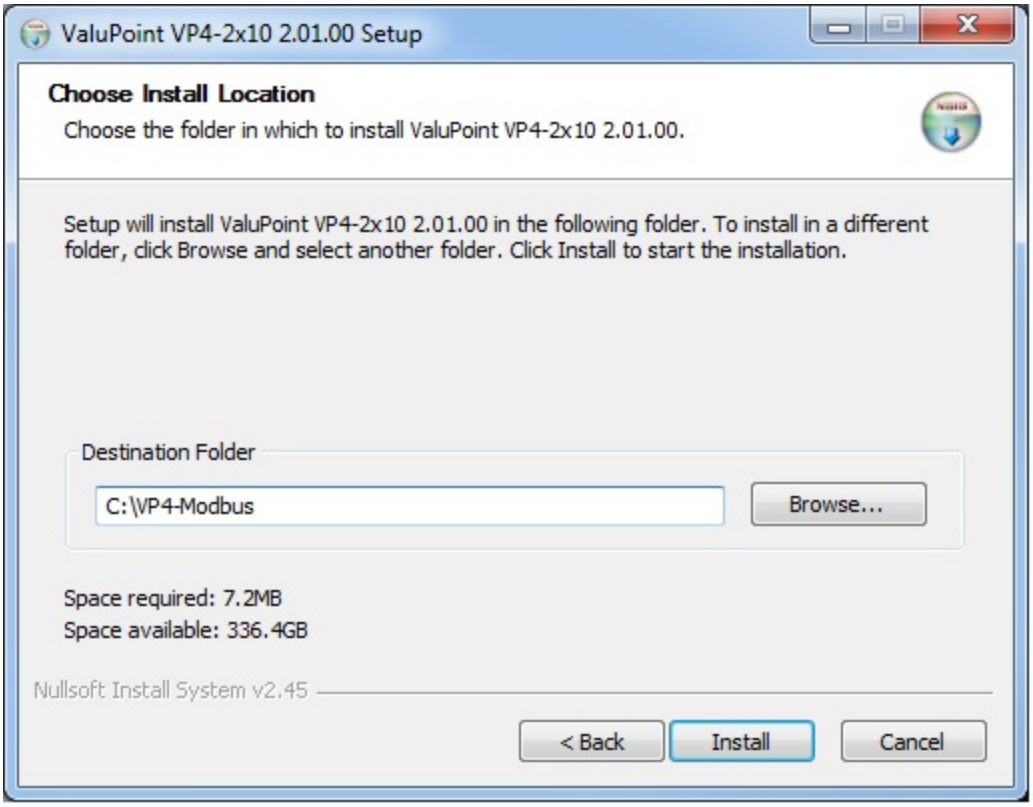
Look for the installer icons in the directory where you unzipped the download that got you to this document. The installer icons look like this:

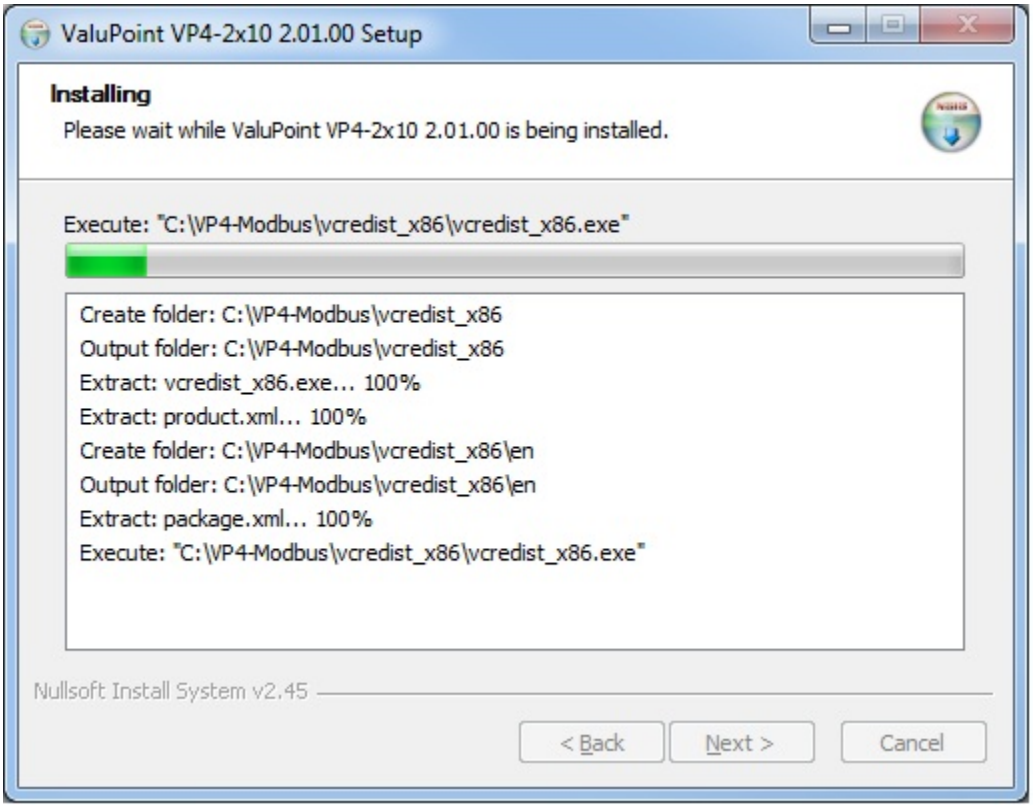
The installation is a 2-step installation. Install VP4-Modbus first. Then install i.CanDrawIt second.

Double click the icon to run the setup.exe. You will be questioned about whether to continue because Windows cannot verify the publisher of the software. Permit installation to continue. The sequence of installer screens include the following on Windows 7:



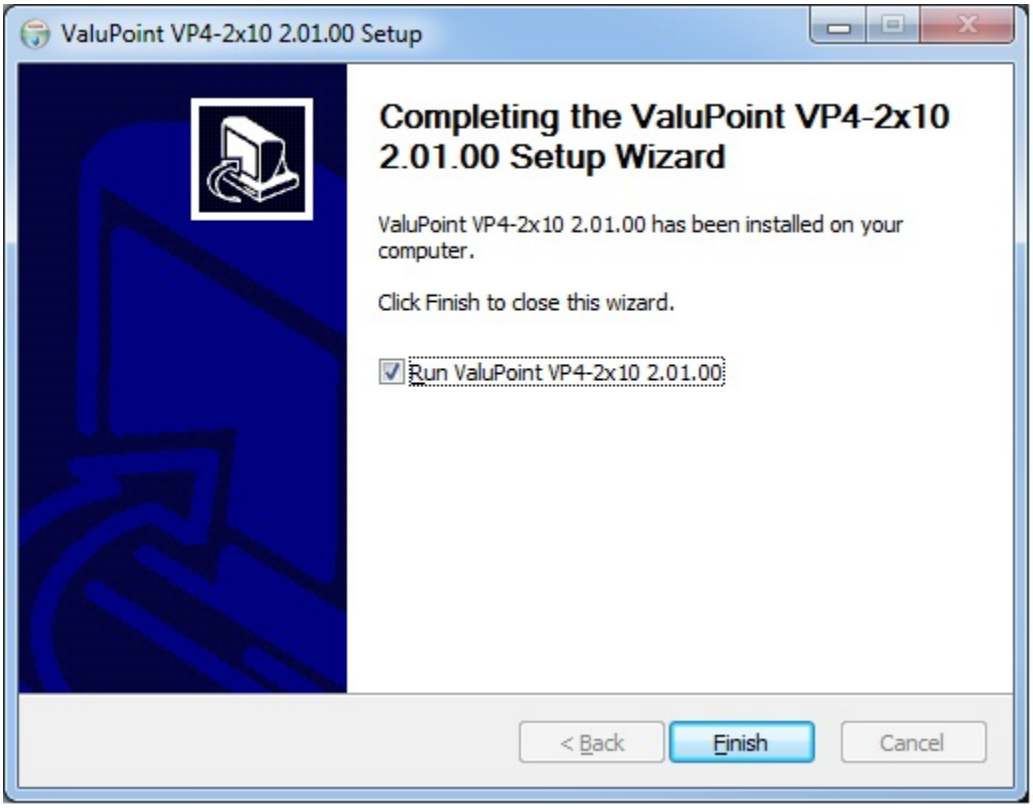

ValuPoint VP4-2x10 2.01.00 Setup
License Agreement
Please review the license terms before installing ValuPoint VP4-2x10 2.01.00.
Press Page Down to see the rest of the agreement.
i.CanDrawIt®, vp4-23.exe, and accompanying software is free software, and is provided "as-is" without warranty of any kind, including the implied warranty of merchantability or fitness for a particular purpose. Control Solutions, Inc., reserves the right to make changes or improvements to i.CanDrawIt without notice.
IMPORTANT SAFETY CONSIDERATIONS: Proper system design is required for reliable and safe operation of distributed control
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install ValuPoint VP4-2x10 2.01.00.
Nullsoft Install System v2.45
< Back
I Agree
Cancel


ValuPoint VP4-2x10 2.01.00 Setup
Choose Install Location
Choose the folder in which to install ValuPoint VP4-2x10 2.01.00.
Setup will install ValuPoint VP4-2x10 2.01.00 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\VP4-Modbus
Browse...
Space required: 7.2MB
Space available: 336.4GB
Nullsoft Install System v2.45
< Back
Install
Cancel

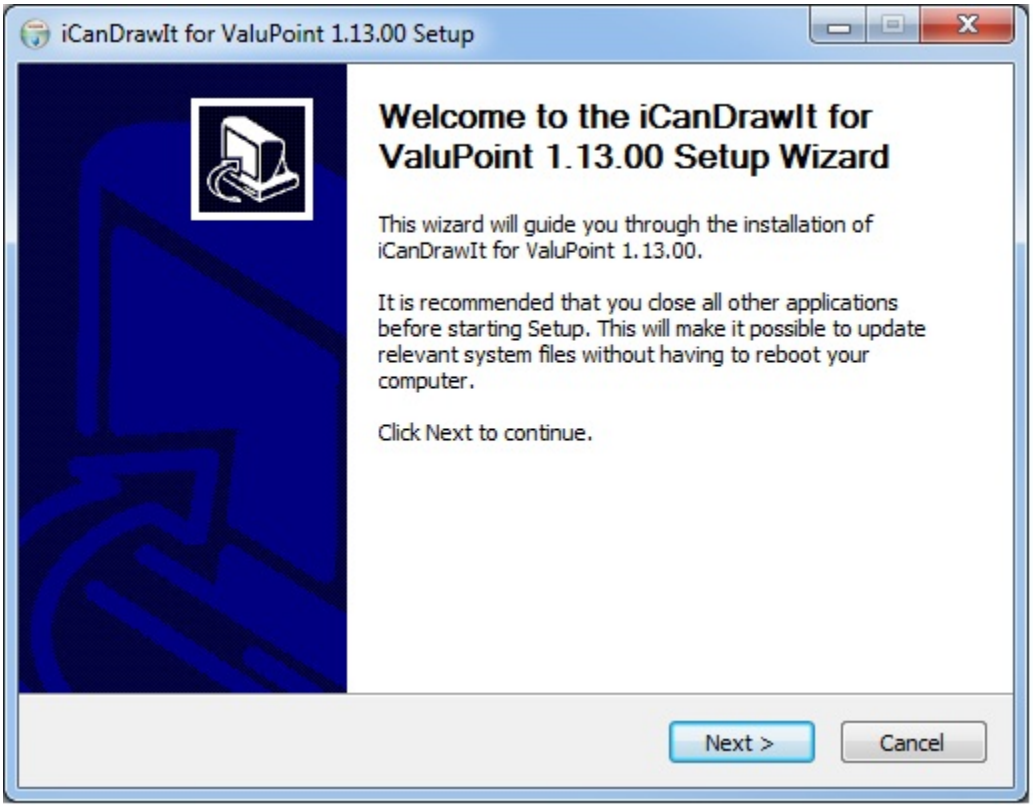


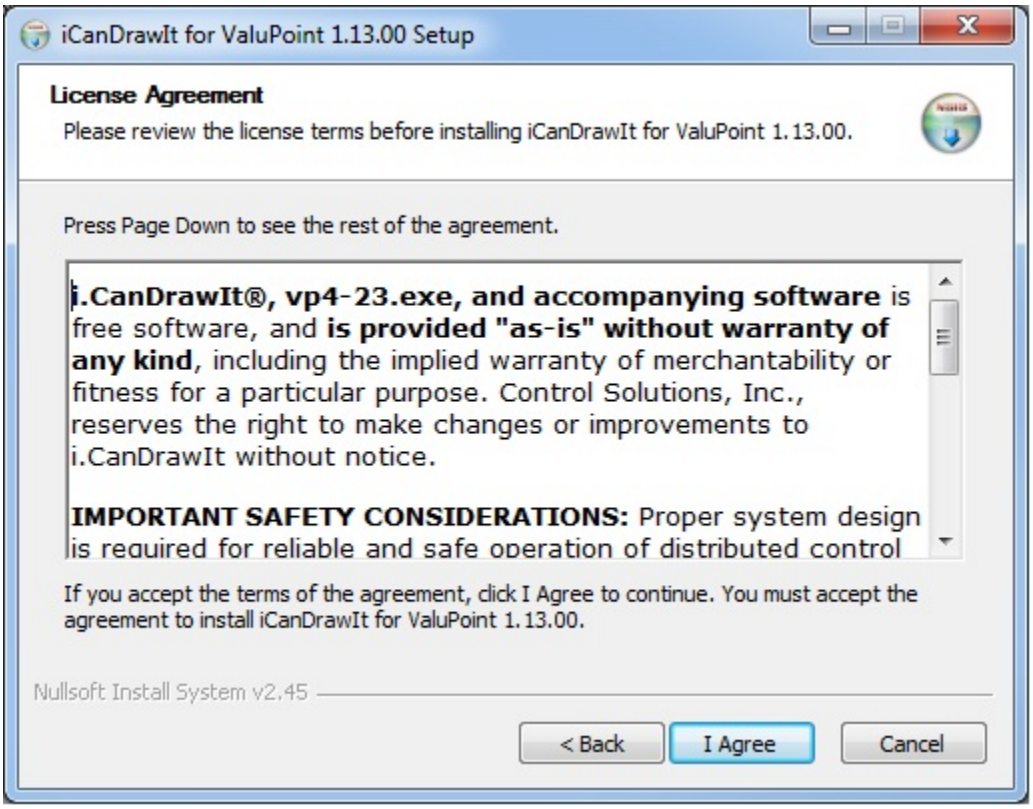
The installer will check to see whether Visual C++ support is already installed on your system, and install it if not. This is standard software provided by Microsoft.

When you get to the "Finish" screen, you are ready to go.

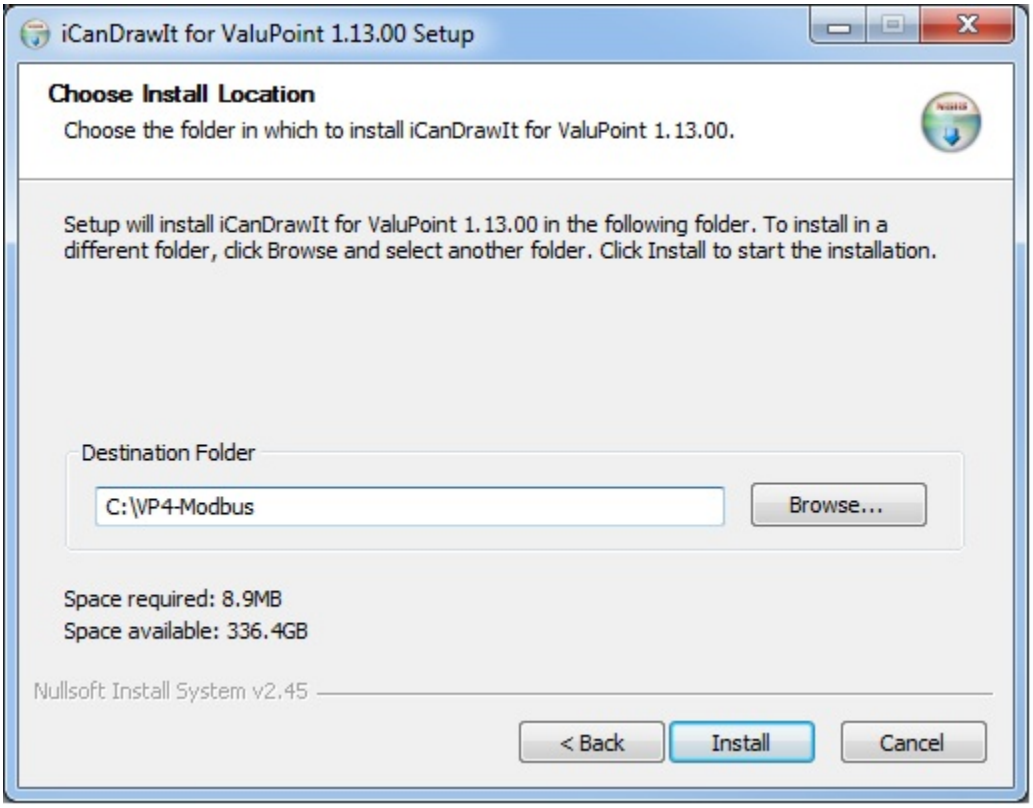


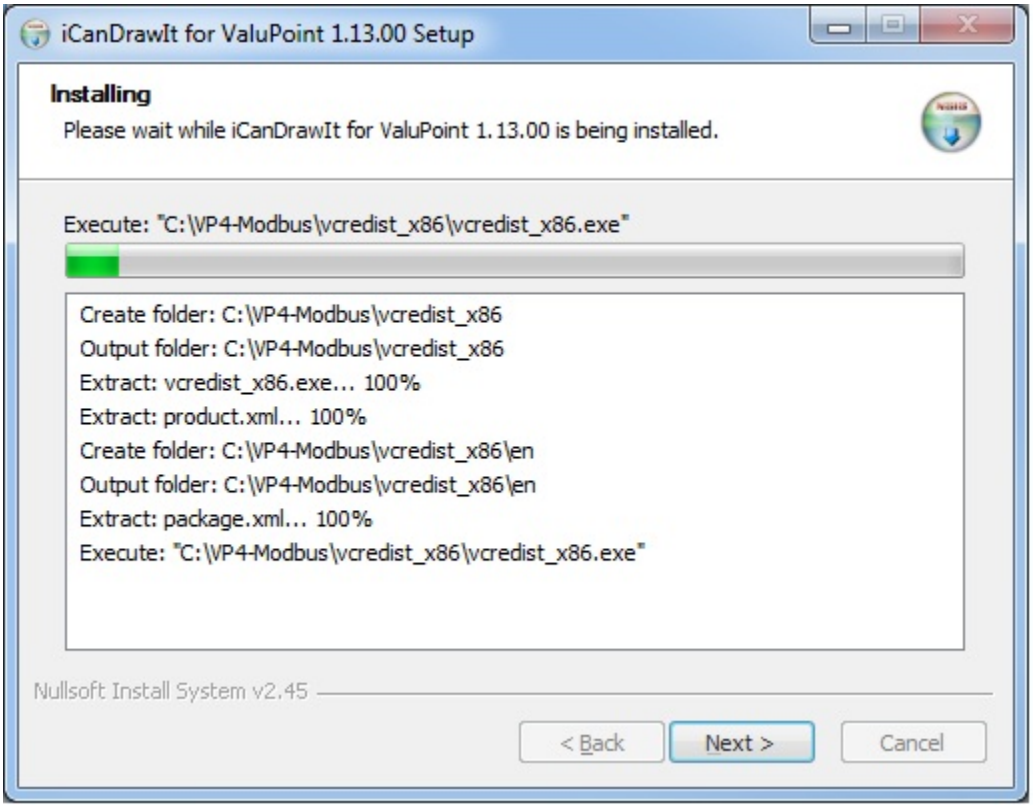
Next, proceed to install i.CanDrawIt. This part is optional. If you will not be using the VP4-2310/VP4-2810 as a programmable controller, you can skip this step. The first installer screen looks like this:



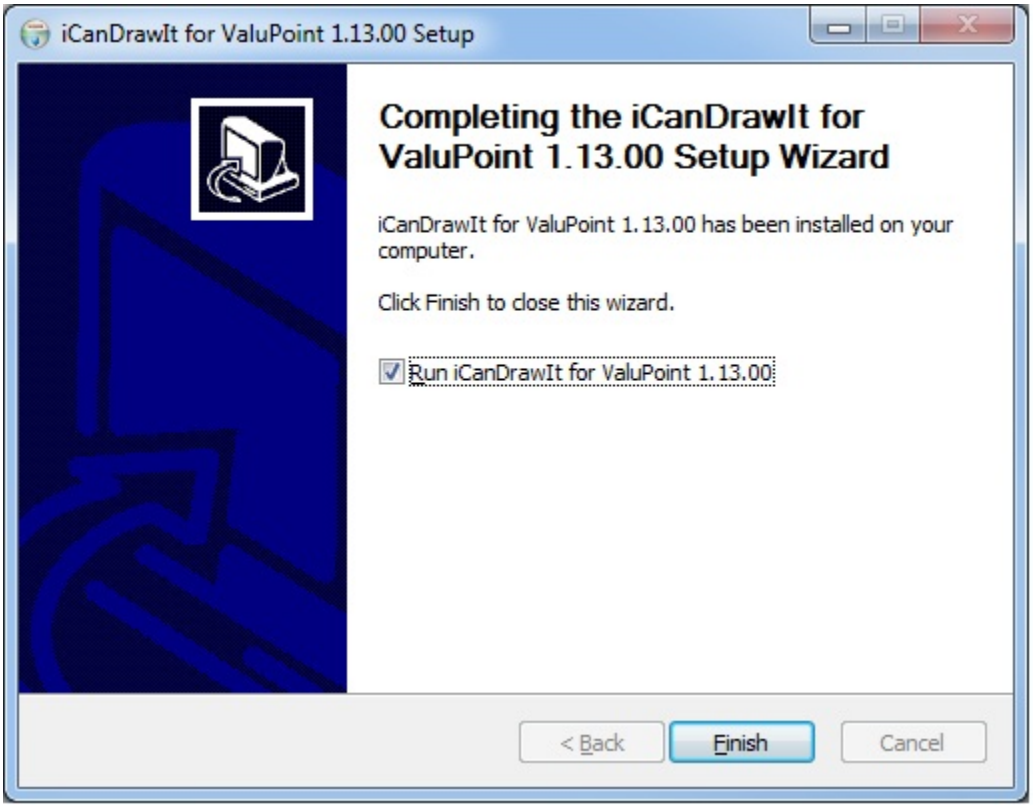


The installation directory should be the same directory that VP4-Modbus was installed into.





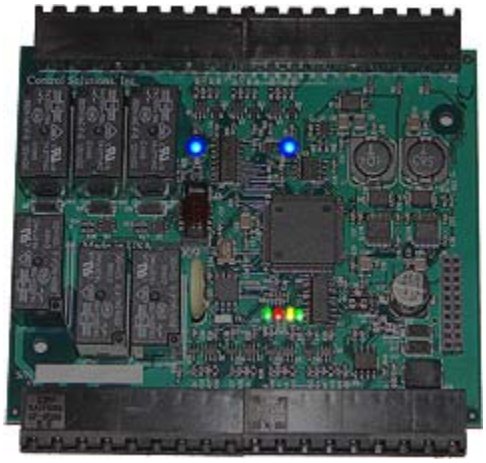
After a few more screens, you will get the familiar 'done' screen.



2.2 Serial Port Connection

The configuration and programming tools communicate with the ValuPoint using Modbus RTU via any COM port on your PC, with an RS232 to RS485 adapter (unless you have a native RS485 COM port on your PC, which is unlikely). If you don't have such an adapter, you can purchase one at www.csimn.com. You do not need to install any special drivers to use an RS485 adapter on your COM port. The supercom.dll that gets installed with your tools takes care of connecting the tools to your COM port.

2.3 Indicators on the VP4-2310 Programmable I/O



The LED indicators for the VP4-2310 are mounted on the circuit board inside the device, and are viewable through the vent slots in the case. The two blue LEDs are power indicators. There are two power supplies in the VP4-2310, and both are necessary for proper operation.

The four LEDs toward the bottom are status and communications - primarily used for communications. The green LED to the left is a system heartbeat that simply indicates the device is running. The heartbeat flashes about once every three seconds. If the device is configured to be Modbus master (normally it is slave) but is in configuration mode, then the heartbeat LED flashes faster - about once a second.

The other three LEDs are Modbus communications indicators. The yellow LED flashes when a packet is received (as slave) or sent (as master). The red LED flashes if there is an error. The error can be CRC error or request for something illegal when ValuPoint is a slave. The red LED will also indicate response timeout when operating as master, or indicate that an exception (error) code was returned by the slave. The green LED flashes upon successful completion of a good message.

The red LED is a communication error indicator most of the time. However, during restart it will be on solid. Immediately following startup, it may flash a fault code if a serious error has occurred.

2.4 Indicators on the VP4-2810 Programmable I/O

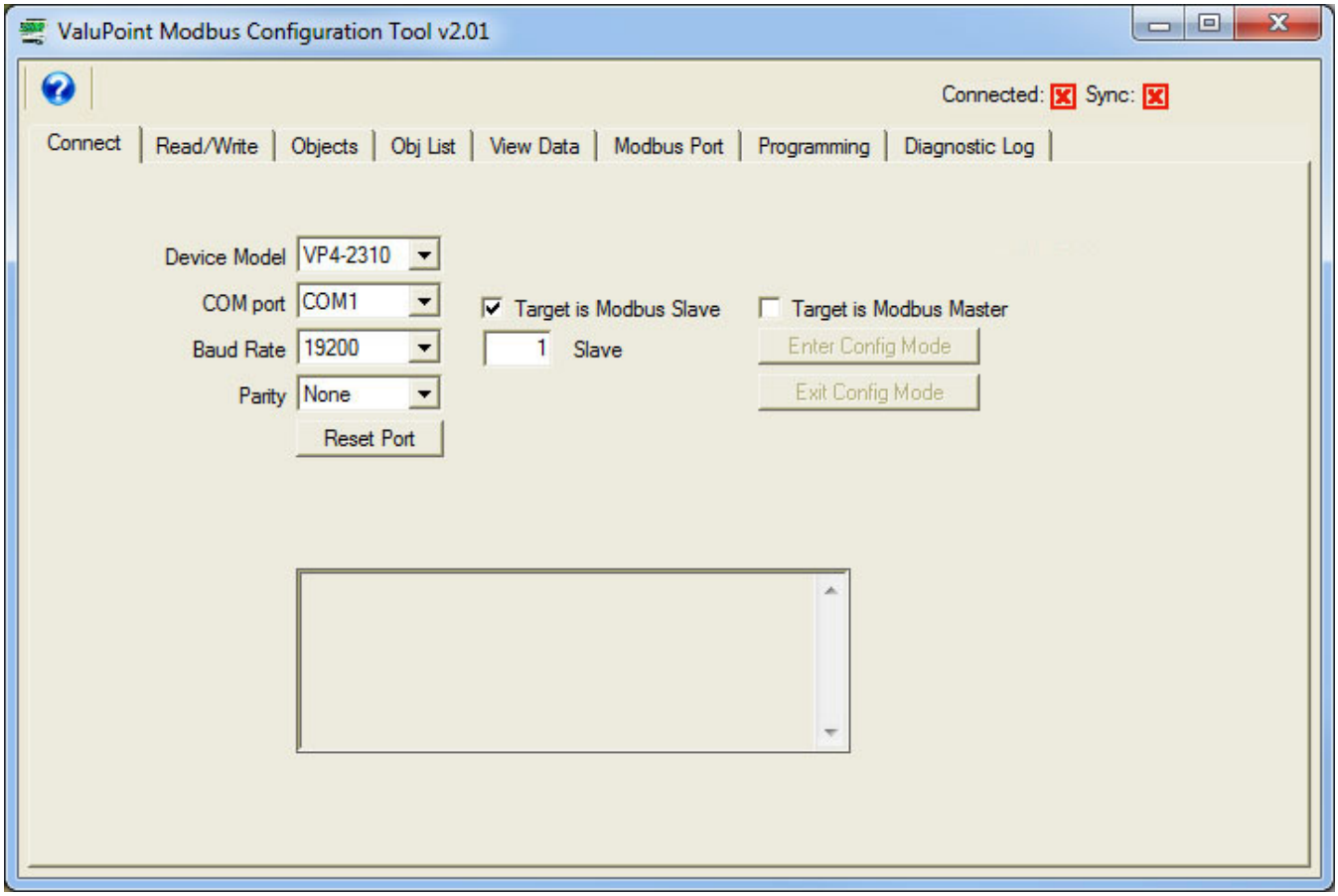
The LED indicators for the VP4-2810 are mounted on the circuit board inside the device, and are viewable through the vent slots in the case. The blue LED is the power indicator. There is one power supply in the VP4-2810.

The other four LEDs toward the bottom are status and communications - primarily used for communications. The green LED to the left (next to power) is a system heartbeat that simply indicates the device is running. The heartbeat flashes about once every three seconds. If the device is configured to be Modbus master (normally it is slave) but is in configuration mode, then the heartbeat LED flashes faster - about once a second.

The other three LEDs are Modbus communications indicators. The yellow LED flashes when a packet is received (as slave) or sent (as master). The red LED flashes if there is an error. The error can be CRC error or request for something illegal when ValuPoint is a slave. The red LED will also indicate response timeout when operating as master, or indicate that an exception (error) code was returned by the slave. The green LED flashes upon successful completion of a good message.

The red LED is a communication error indicator most of the time. However, during restart it will be on solid. Immediately following startup, it may flash a fault code if a serious error has occurred.

3 Connect



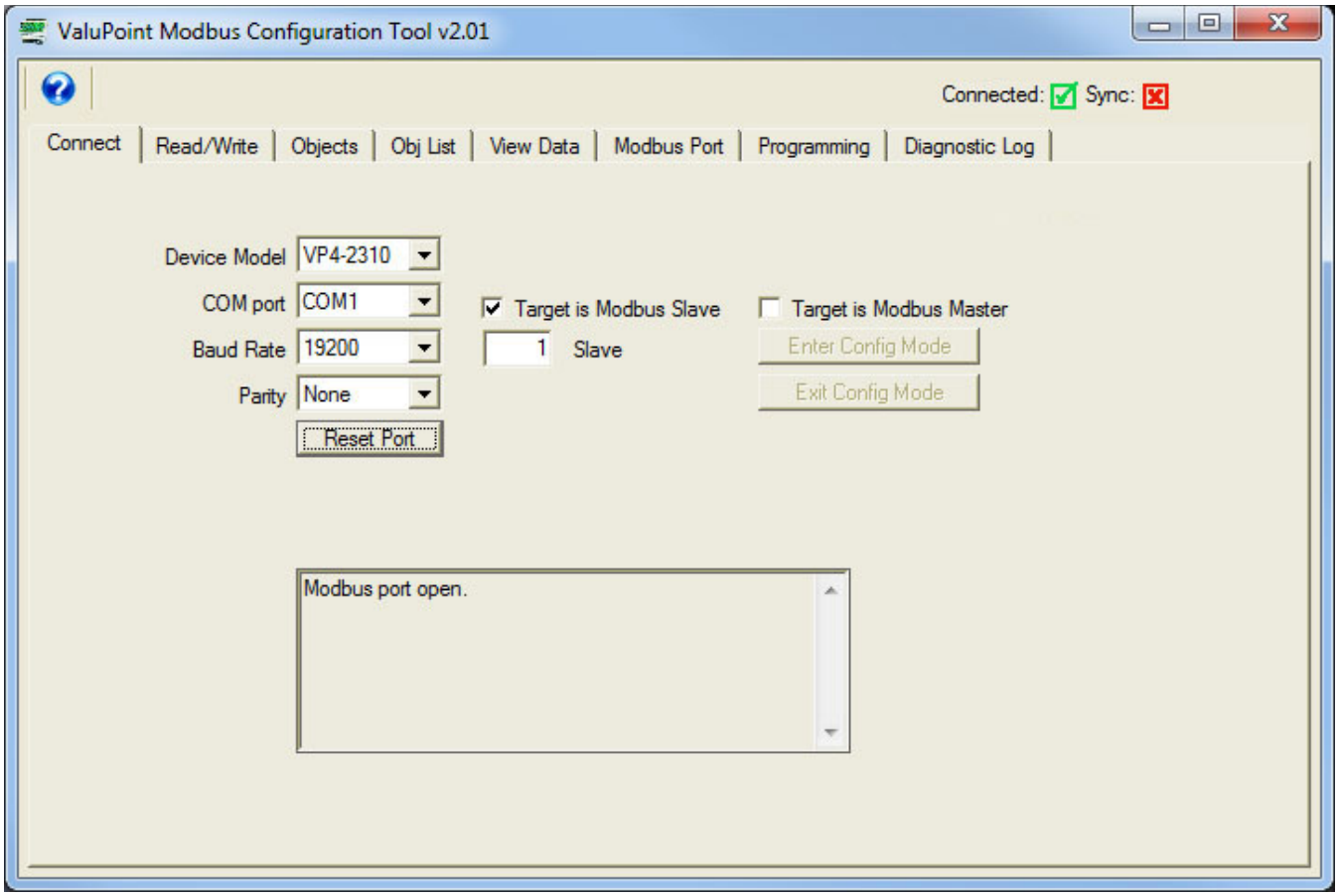
3.0 Do This for First Time Startup

Your Modbus ValuPoint will be shipped as a Modbus slave. Its default slave address is 1, and baud rate 19,200 (N81). The default settings are shown above, except for COM port. You will need to locate the COM port that is appropriate for your PC. You will need an RS485 adapter if connecting to an RS232 port. You can also use an adapter that connects to USB on your PC and provides a serial RS485 port. You cannot connect to ValuPoint directly with RS232.

For first time startup, you should check 'Target is Modbus Slave'. The device is automatically always in configuration mode when running as slave, so do not be concerned about the disabled 'Config Mode' buttons. Those are only used to get the device's attention when it is configured as a Master.

3.1 Connect to Target Running as Slave

To connect to a device configured as Modbus slave, simply select device model, COM port, baud rate, parity, and slave address and enter these on the Connect page of the configuration tool. You may click Reset Port to open the COM port. If you don't do this, the port will be automatically opened the first time you try to communicate with the device. Simply resetting the port does not communicate with the device, therefore when the Connected indicator in the upper right corner changes to a green check mark, this only means the COM port was successfully opened. To test actual communications with the device, go to the Read/Write tab and simply test reading a single holding register.

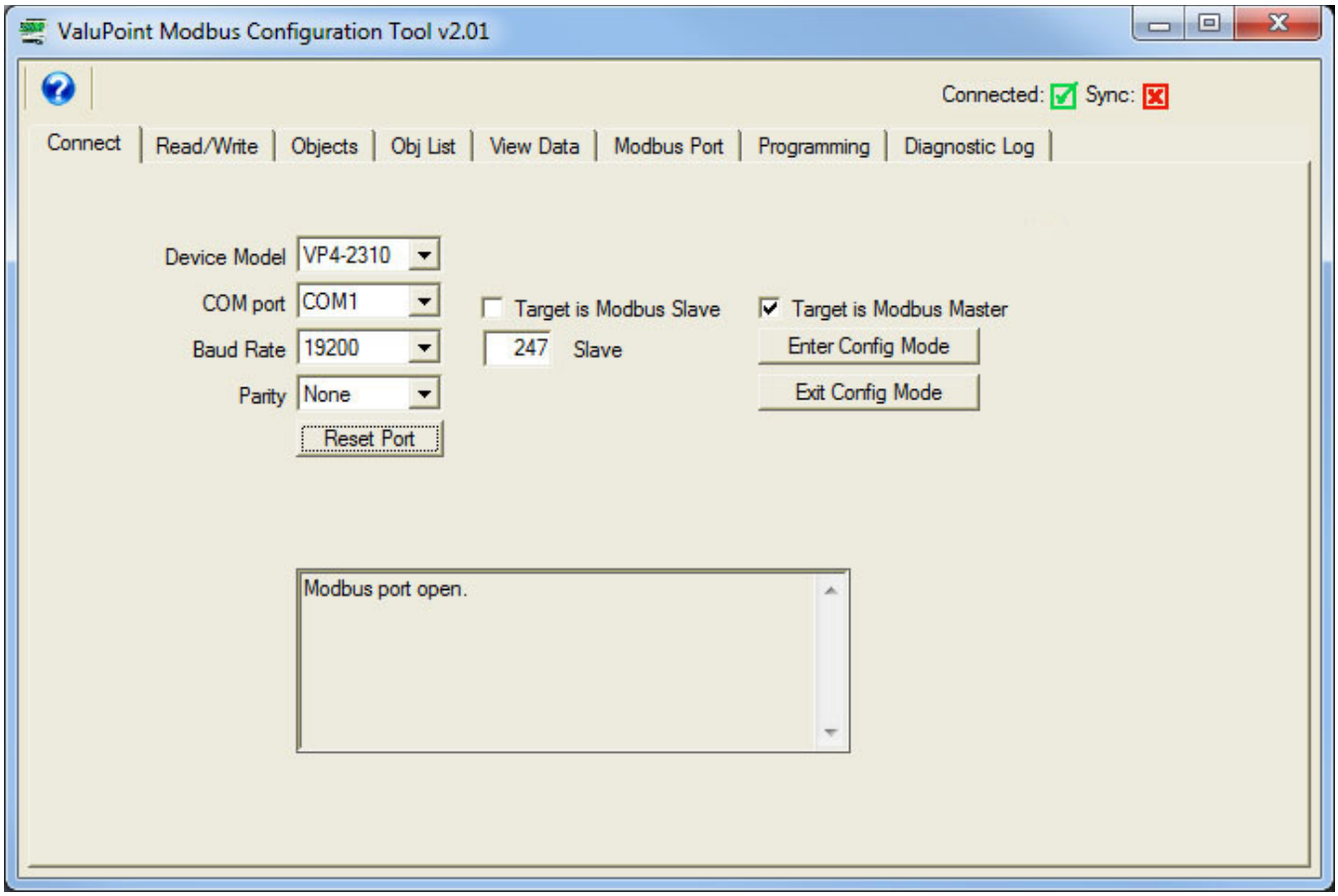


3.2 Connect to Target Running as Master

Connecting to a device configured to be Modbus master is a little tricky because the device is already a master, but the configuration tool also wants to be master. You can never have more than one master on a Modbus RTU (RS485) link. To begin the process of getting the device's attention if it is running as master, check the 'Target is Modbus Master' box. The Enter/Exit Config Mode buttons will now come alive.

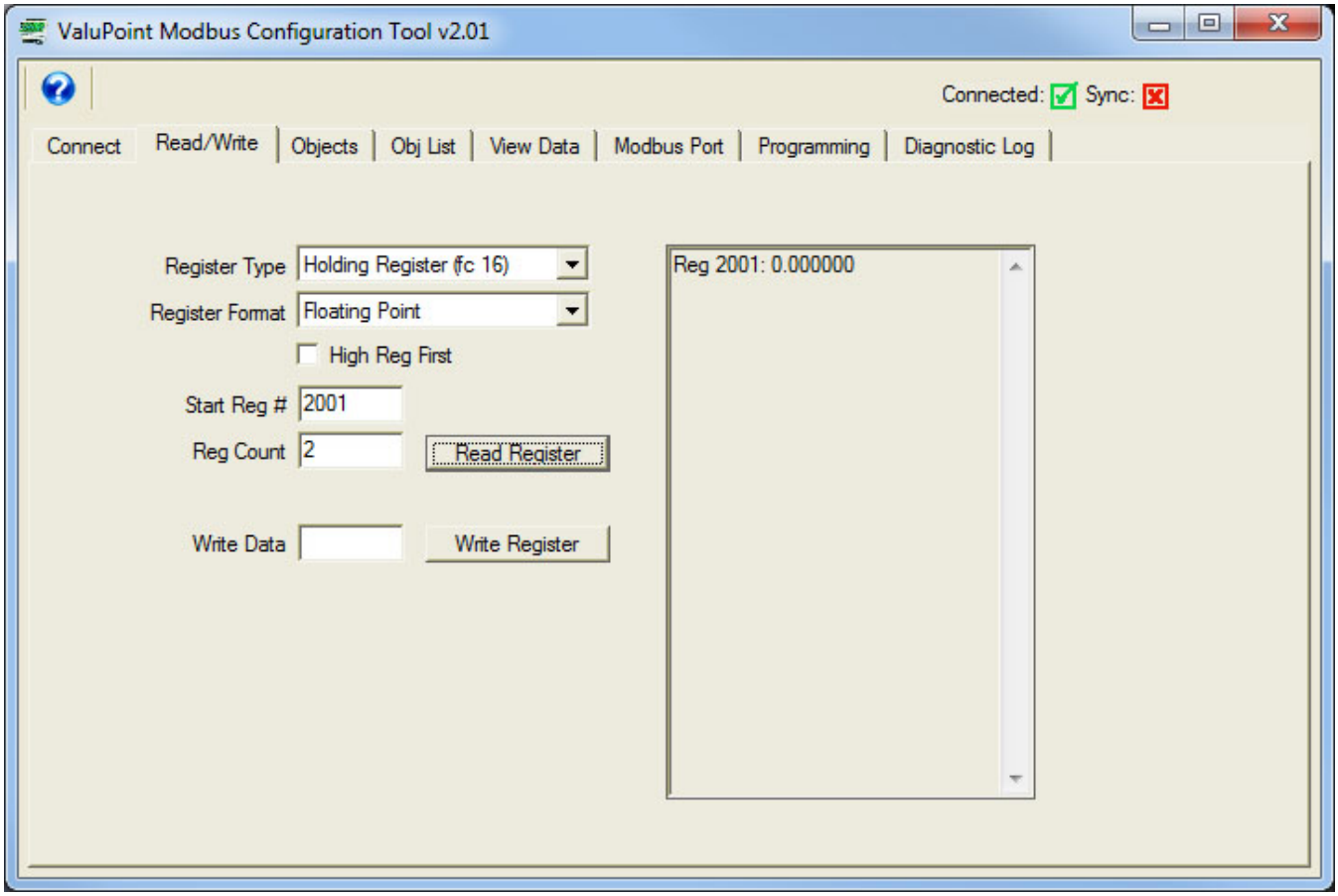
Click 'Enter Config Mode'. During configuration mode, the ValuPoint normally acting as master will become a slave at address 247. When you click the 'Enter Config Mode' button, the first thing that happens is that the tool will wait for the ValuPoint to query the tool for the mode change. The ValuPoint, when running as master, will query a register at slave address 247 every few seconds. If the configuration tool responds to that request with the appropriate configuration mode code, then the ValuPoint will 'switch gears' and become a slave temporarily. Once the ValuPoint has switched to slave mode, you can proceed to configure it. You cannot configure a ValuPoint Modbus device that is not functioning as a slave at least temporarily. To end the temporary slave operation, click 'Exit Config Mode'.

You will know you are in configuration mode when the green 'system health' LED inside the VP4 flashes more rapidly than normal. Normally it is mostly on with a brief flash off about once every 2 seconds. In configuration mode, it flashes off about every half second.



IMPORTANT: Do not change device mode to Modbus master until AFTER you have created some Modbus register mappings for the master to act upon. These are set up on the Objects page. If you switch to master mode without anything for the master to do, you will have trouble entering configuration mode. See Modbus Port page for more about this.

4 Read / Write



4.0 Do This for First Time Startup

Although you don't really have to do anything on this page for first time startup, it is a handy way to confirm that you are communicating with the ValuPoint Modus device. If you can read any valid register here, you will be able to do all necessary configuration. The example above shows reading input 1 as floating point. The default (and most simple) instance is to read a single register at address 1. This should always work regardless of ValuPoint Modbus model.

4.1 Read Registers

Select 'Register Type'. This determines the Modbus function code that will be sent to the Modbus device. Selecting 'Register Format' only tells the configuration tool how to interpret the data and format for you to read. The format has nothing to do with construction the query sent to the device. If reading a 32-bit data value, it will occupy two consecutive Modbus registers. The order is not standardized from one manufacturer to the next, so the option of swapping them is provided here. Check the 'High Reg First' box if the most significant data is found in the first of the two registers (must check with manufacturer of device). Again, this is only used to interpret the data and has nothing to do with constucting the query to the device.

Select starting register number and count. Note that 32-bit values will always occupy 2 registers each. The starting register and register count are important pieces of information used to construct the query to the Modbus slave. Click 'Read Register' to cause the query to be created and transmitted over the RTU network. The results of the query will be displayed in the log window on the right.

The Read/Write functionality provided here is completely generic, and can be used to read/write any

Modbus slave connected to the same RS485 network that your PC is connected to.

NOTE: When reading data from some other manufacturer's device, you need to check with that manufacturer to see what order the registers are in for reading 32-bit values. When reading/writing a ValuPoint, the 'High Reg First' box on the Read/Write page should match whatever was configured on the Modbus Port page. The default as initially shipped is to not check this box.

4.2 Write Register

Select all of the same parameters that you would for reading a register. In addition, enter a data value to be written, and then click 'Write Register'. The 'Register Format' will be used to determine how the data you enter is converted to raw binary form to be transmitted to the device. The result of your 'Write Register' attempt will be displayed in the log window.

4.3 Errors

One of the most common Modbus RTU errors is simply 'Timeout' or 'No Response'. This means the slave device did not respond for any of the following reasions: (a) Slave not powered up; (b) slave not connected correctly; (c) baud rate mismatch. In the case of a poorly behaved Modbus slave, it may ignore a request it does not like, but Modbus protocol says it is supposed to return an exception error if it received a query but does not like it for some reason.

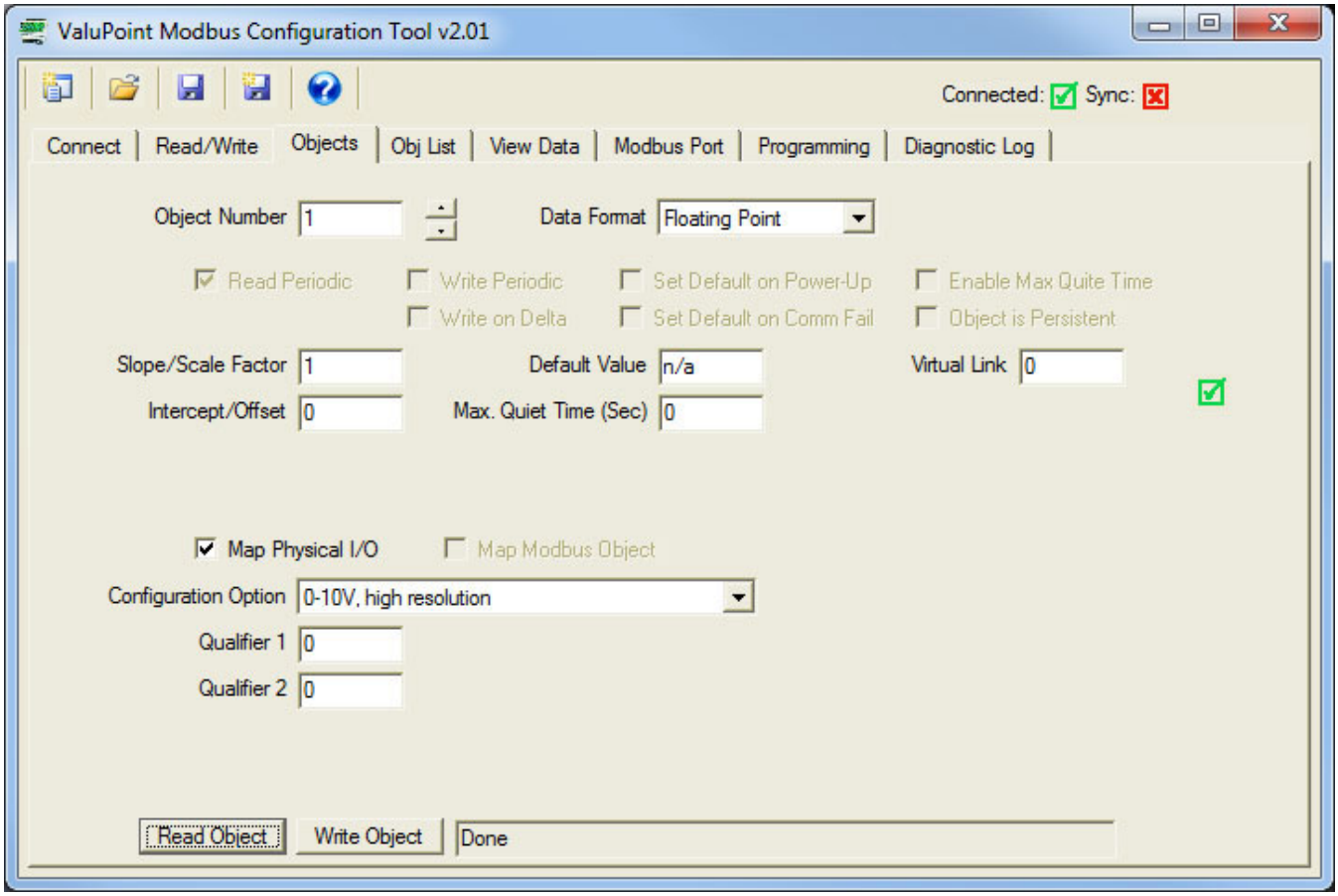
If communication is successful, but the Modbus slave returns an exception error, the most common errors are listed below. If you get an error other than these, it will be unusual and you will need to consult both the Modbus protocol definition of error codes as well as the manufacturers data to see what caused it.

The most common error will be 'illegal data address'. This simply means you requested a register that does not exist in the slave device, usually due to a typo or a misinterpretation of the manufacturers data. Illegal function will happen only if you attempt to send a request that is not at all supported by the Modbus slave. If this happens, it will most likely be associated with a Write request, and most often happens trying to use 'write multiple' when only 'write single' is supported, or vice versa, as it applies to coils or holding registers.

1	Illegal Function	The function code received in the query is not recognized by the slave or is not allowed by the slave.
2	Illegal Data Address	The data address (register number) received in the query is not an allowed address for the slave, i.e., the register does not exist. If multiple registers were requested, at least one was not permitted.
3	Illegal Data Value	The value contained in the query's data field is not acceptable to the slave.

If the error you are getting is CRC and this happens frequently, the problem might be parity setting, but is most often related to wiring problems such that excessive noise is getting onto the data lines. Often the CRC errors are mixed in with frequent occurrences of 'no response', but the meaning is the same. Check port settings and wiring.

5 Object Map Page



5.0 Do This for First Time Startup

When connecting and syncing for the first time, you should go to the 'Obj List' page and read all object maps with one click of the mouse. You can read all properties here, but you must click once per object (as many as 320 times for a fully loaded device).

Once you have read all the objects from the device, come back to this page to configure individual objects (aka I/O points or Modbus registers).

5.1 Common Parameters

Object Number – The ValuPoint has a collection of data "objects" and each of these may be configured using the Objects page. The object number is not necessarily the Modbus register number. If you do use the object number as holding register number, you will read and write the object as a 16-bit integer even if the object's native data format is floating point. You need to reference the object using the correct floating point register number to access it as floating point.

Data Format – Select the native data format for this data object. This determines how the data is maintained internally. The format in which you access the object depends on the Modbus register number you use. There are multiple Modbus register numbers that access each data object.

Slope/Scale Factor – Scaling applies the formula y=mx+b. When reading from the slave, the raw data as read is multiplied by the scale factor, then the offset is added to produce the resulting Present Value. When writing to the slave, the offset is first subtracted from Present Value, and that result is divided by the scale factor to produce the raw data actually written to the slave device. NOTE: If no scale factor is

given (zero is entered), no scaling will be done, as if slope=1 and intercept=0.

Intercept/Offset – The offset portion of the scaling as noted above.

Default Value – This is the value that should become the Present Value upon power-up or upon communications failure, if either of these options are selected by the appropriate check boxes above.

Max. Quiet Time (Sec) – This is applicable to an object which is set to Write on Delta. The result is that if there has been no change within this amount of time, the slave device will be re-written anyway. In addition to entering a nonzero value here, you must check the 'Enable Max Quiet Time' box.

CHECK BOXES will be enabled when applicable to the data object being configured, and these are as follows:

Read Periodic (check box) – Check this box if the data object will be periodically reading from a slave Modbus device. The box is always checked (selected/enabled) for physical hardware inputs.

Write Periodic (check box) – Check this box if the data object will be periodically writing to a slave Modbus device. This box is always checked (selected/enabled) for physical hardware outputs.

Write on Delta (check box) – Check this box if the data object will be writing to the slave device only when the object's value changes by a specified 'Delta'. It is valid to check both Write Periodic and Write on Delta at the same time, or check just one or the other.

Set Default on Power-Up (check box) – Check this box if the object should assume the default value every time the ValuPoint powers up.

Set Default on Comm Fail (check box) – Check this box if the object should assume the default value when communication with the slave device has failed some number of times.

Enable Max Quiet Time (check box) – Check this box to enable maximum quiet time. This is applicable to an object which is set to Write on Delta. The result is that if there has been no change within the max quite time, the slave device will be re-written anyway.

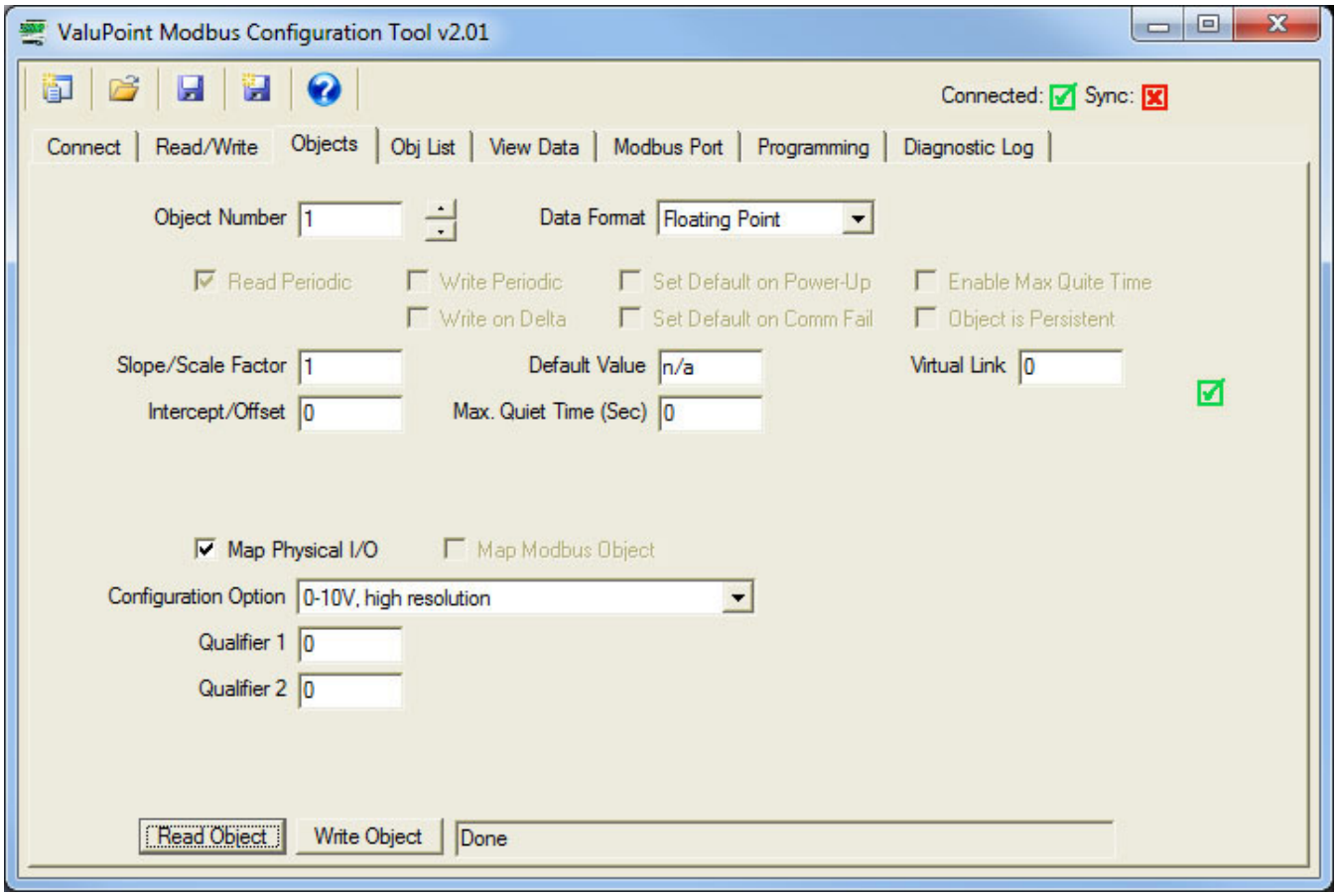
Object is Persistent (check box) – Checking this box means the data value of the object will be retained through restart or power cycle, rather than reset to zero. This is applicable to only certain objects. Data objects mapped to physical inputs, for example, will not honor the Persistent flag - an Analog Input will always reflect the actual input value. Analog Outputs will also not honor the Persistent flag - if a specific power-up value is desired, use the Default Value and check 'Set Default on Power-Up'.

Virtual Link - Not normally used in conjunction with physical I/O or mapped remote Modbus registers, these represent "virtual" inputs found in the system. These may be used as the source of an object's data. Example of usage: If you want to read via Modbus how many times the relay output DO 1 on the VP4-2310 (object 18) has turned on (cycle count), you would enter virtual link number 1018 in some otherwise unused object. Now read this object to see how many times the output has turned on. Both cycle count and on-time are retained in non-volatile memory which is written each time the output makes a transition, or every 15 minutes (for on time) if the output remains on for an extended period of time. These virtual link numbers are NOT Modbus register numbers, they are virtual input numbers, and cannot be read directly via Modbus. They must be used as the virtual link in an ordinary data object which may then be read/written via Modbus.

Normally the data objects representing cycle count or on-time would be "read only", but you can reset the counts to zero by writing zero to the object linked to those properties.

Link number	Virtual Object Linked
402	Virtual link to system up time in seconds, 32-bit value
1001-1320	Virtual link to cycle counter for objects 1-320, 32-bit value (automatically persistent)

2001-2247	Virtual link to Modbus error code by slave address, 16-bit value
3001-3320	Virtual link to on-time counter in seconds, objects 1-320 (automatically persistent)



5.2 Physical Hardware I/O Parameters

Map Physical I/O (check box) – This box is forced to the selected/enabled state for those objects that map to physical I/O, as determined by the I/O features of the ValuPoint model.

Configuration Option – Select the I/O configuration desired for the physical point. This only applies to A/UI analog/universal input points, and DI discrete input points. The A/UI inputs may function as any of the following (and DI inputs may function as a subset of these):

0-10V, high resolution
0-10V, low resolution
4-20mA, high res with resistor
4-20mA, low res with resistor
Discrete, active high
Discrete, active low
Dry contact, active open
Dry contact, active closed
Pulse, high speed
Pulse, low speed (no hi-res)
Resistance, high resolution
Resistance, low resolution
Position pot, hi-res, 1K-30K
Position pot, lo-res, 1K-30K

Thermistor, 10K type III hi-res, F

Thermistor, 10K type II hi-res, F
Thermistor, 3K type II hi-res, F
Thermistor, 20K type IV hi-res, F
Thermistor, 5K type II hi-res, F
Thermistor, 10K type III hi-res, C
Thermistor, 10K type II hi-res, C
Thermistor, 3K type II hi-res, C
Thermistor, 20K type IV hi-res, C
Thermistor, 5K type II hi-res, C

Thermistor, 10K type III lo-res, F
Thermistor, 10K type II lo-res, F
Thermistor, 3K type II lo-res, F
Thermistor, 20K type IV lo-res, F
Thermistor, 5K type II lo-res, F
Thermistor, 10K type III lo-res, C
Thermistor, 10K type II lo-res, C
Thermistor, 3K type II lo-res, C
Thermistor, 20K type IV lo-res, C
Thermistor, 5K type II lo-res, C

The designation hi-res and lo-res refer to the A/D resolution. There are two types of A/D in the ValuPoint. The 'hi-res' A/D is a delta-sigma integrating converter that produces 15-bit results (16-bit including sign, which is always positive with the input circuitry implemented). The 'lo-res' A/D is a fast successive approximation converter which produces 10-bit results.

Due to the fact that 'hi-res' conversion is relatively slow (about 1/4 second compared to a few microseconds for lo-res), some combinations of I/O are not possible. Low speed pulse counting to 10Hz is only possible when all inputs are in lo-res mode. All discrete or dry contact modes are lo-res. High speed pulse counting (to 1kHz) is possible with any combination of modes; however high speed counting uses dedicated hardware and is only available on two specific channels that vary by model number.

Low speed pulse counting in combination with hi-res input conversions is possible, but input pulse rate is limited to about 2Hz. To increase this limit to 10Hz, all analog inputs must be using lo-res modes.

Qualifier 1 – Enter the configuration qualifier value, if applicable, for the selected configuration. Qualifiers are required only in the following modes:

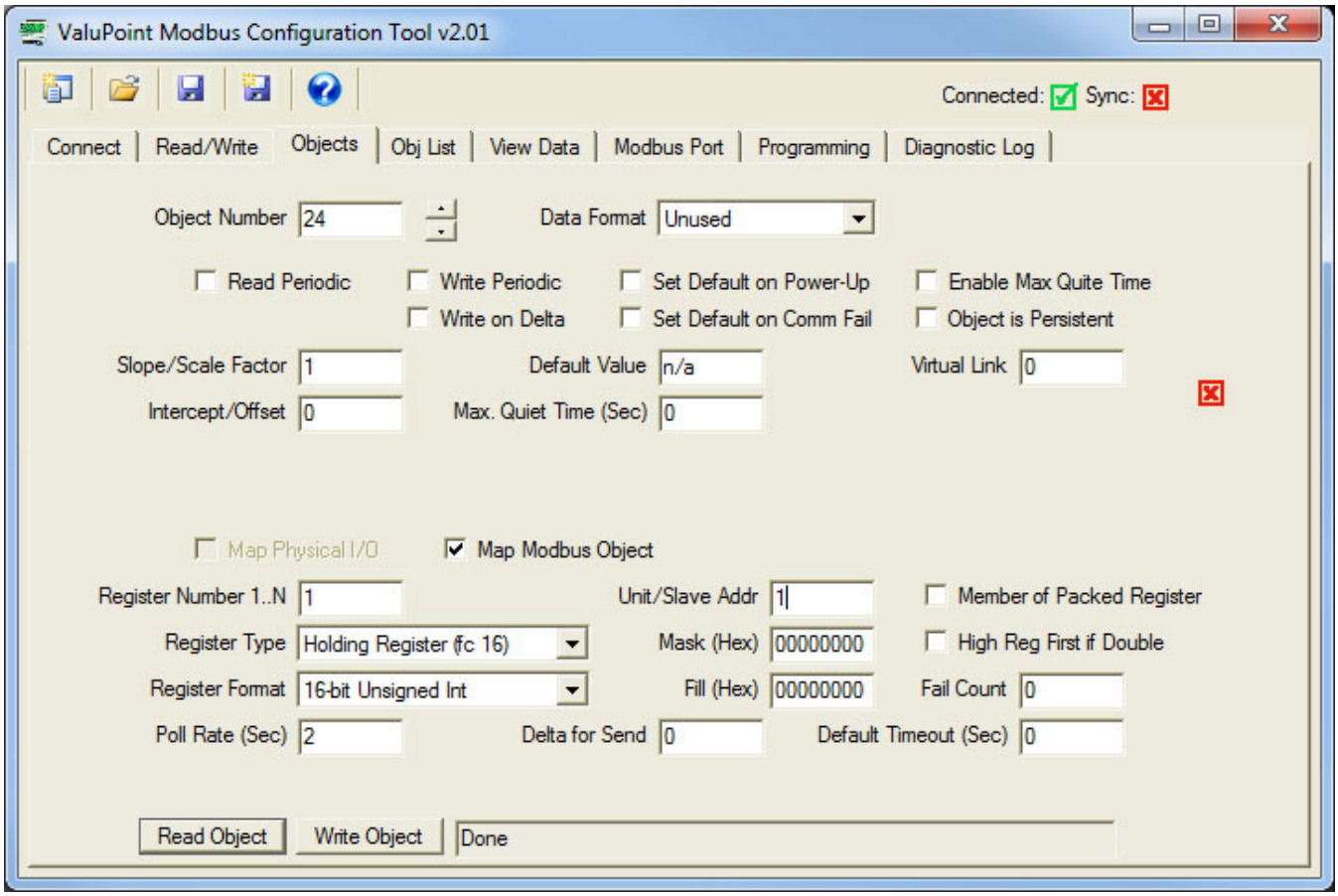
4-20mA modes: The qualifier is the resistance in ohms of the dropping resistor used to convert the current to voltage. An external resistor must be provided, connected between the A/UI input and ground/common. The resistor needs to be 1/2 watt (2 watt to withstand 24V power), and is left external simply because miswiring the 4-20mA sensor can easily apply 24V power directly to the input and cause the dropping resistor to heat up and possibly fail. The external resistor is simple to replace, whereas an internal resistor on a circuit board would be more trouble to replace.

Discrete and Dry Contact modes: The qualifier is a threshold between 1% and 99% at which the input should trip from off to on or vice versa. Although the A/UI inputs are specified as 0-10V inputs, they will actually measure up to 11.4V accurately in hi-res mode, and up to 15.0V in lo-res mode. Therefore, since discrete inputs are sampled as lo-res analog values and compared to a threshold, the qualifier here is a percentage of 15V for the trip point. A value of 50% will mean a threshold of around 7.5V.

Pulse, Low Speed mode: Because low speed pulse input is really a different interpretation of discrete input, a threshold setting is required for pulse sensing. The same threshold as used for discrete input applies here. Without any threshold (or with value of zero), no pulse counting will occur.

Position Pot: Position is simply a different interpretation of resistance measurement. The value resulting from position pot measurement will be a percentage from 0% to 100%, but this percentage will be a ratio based on the resistance value in ohms provided as the qualifier.

Qualifier 2 – Enter a second configuration qualifier value if instructed to do so. This qualifier is reserved for future use.



5.3 Modbus RTU Master Parameters

Map Modbus Object (check box) – Check this box when the ValuPoint should look for another Modbus device as a slave device to query for data. When this box is checked, the remaining boxes required for Modbus map setup will appear.

Register Number 1..N – Enter the Modbus register number between 1 and 65535. The register number is raw address plus one. Therefore if your Modbus device's documentation indicates that the first register is at address zero, add one to every address to get the register 'number'.

Register Type – Select a Modbus register type from the list, such as Holding Register, etc. Coils and Holding Registers can be written as single or multiple, and some Modbus devices only recognize one function code or the other. If you are having trouble writing, check the device's documentation and see if you are using the correct function code. Both possible codes for each register type are included in the list, denoted '(fc x)'.

Register Format – Select a register format from the list. If the data format is 32-bit integer or floating point, the BB2-3010 will automatically read/write two consecutive registers to get the entire value. The designation 'high reg first' means the most significant part of a multi-register value will be in the first, or lowest numbered, Modbus register.

Format designations:

CSV	Description

Notation	
SIGN	Signed 16-bit
UNSI	Unsigned 16-bit
SDBE	Signed 32-bit (high reg first)
UDBE	Unsigned 32-bit (high reg first)
FPBE	Floating Point (high reg first)
BBIT	Bit (coil/discrete)
SDLE	Signed 32-bit (low reg first)
UDLE	Unsigned 32-bit (low reg first)
FPLE	Floating Point (low reg first)

Poll Rate (Sec) – This rate applies to periodic reading or writing, and simply specifies how often the slave device is read from or written to. (This parameter does not apply to physical I/O.)

Unit/Slave Addr – Enter the Modbus device's slave address here. This number is also often referred to as unit number or slave ID.

Mask (Hex) – When extracting a single bit or set of bits from a packed Modbus register, this mask specifies where in the register the desired bits are found. This mask is entered as a 4-digit hexadecimal number representing 16 bits. After masking data read from Modbus by performing a bit-wise And between the data and this mask, the resulting data is right shifted so that the least significant bit of interest becomes the LSB of data. When writing, this process is reversed.

Fill (Hex) – Applicable only when writing, this value is optionally used to always set specified bits. This value is entered as a 4-digit hexadecimal number representing 16 bits. The data to be written to the Modbus slave is bit-wise logical Or-ed with this value just before being written to the device.

Member of Packed Register (check box) – Check this box if the BB2-3010 should look at the next consecutive object map, or is included in the group started by a previous object map, to combine this and at least one other object map to collect multiple BACnet objects into a single Modbus register (or vice versa when reading).

High Reg First if Double (check box) - Used only when reading 32-bit data, check this box if the first register will contain the most significant half of the data.

Fail Count – When 'Set Default on Comm Fail' is checked, this entry will allow you to disregard a small number of spurious errors. At least this many errors must occur consecutively before a fault will actually be flagged. Causing a fault as a result of a single instance of spurious noise on a communication line can be a nuisance. This setting allows quieting the nuisance fault notifications. (This parameter does not apply to physical I/O.)

Delta for Send – Enter the threshold for writing when configured to Write on Delta. If a value of 5.0 is entered, the value must change by more than 5.0 before the slave device will be written to. Delta applies only to Analog objects. Binary and Multi-state objects will send on any change when Write on Delta is enabled.

Default Timeout (Sec) – Timeout for Modbus slaves is specified on the Modbus page. When the slave device is another BACnet device, the timeout given here specifies the amount of time the VP4-2330 will wait for a response before calling it an error.

5.4 Insert, Add, Delete

Clicking 'Insert' will insert a blank map before the currently displayed object map. Clicking 'Add' will insert a blank map after the currently displayed object map. Clicking 'Delete' will remove the currently displayed

map and slide the remainder of the list up by one slot on the list.

The only time position matters much in the list is when configuring a packed register that maps a single remote Modbus register to multiple local data objects. The members of the packed register must appear in consecutive maps.

5.5 Read/Write Device

When changes are made, the in-sync icon changes to a red X on the right hand side of this page. When properties are either read or written, the device is now in sync, and the red X changes to a green check mark. This icon will change on an object by object basis indicating those objects that are in sync.

When all aspects of device configuration are in sync, between device and tool, the 'In Sync' icon in the top right corner of the screen will change to a green check box. This is the master In-Sync icon, and will be green only when all other in-sync icons are also green. This icon is effectively a summary icon.

5.6 File Read/Write (Object Map XML)

Click the 'new' icon to erase all object maps.

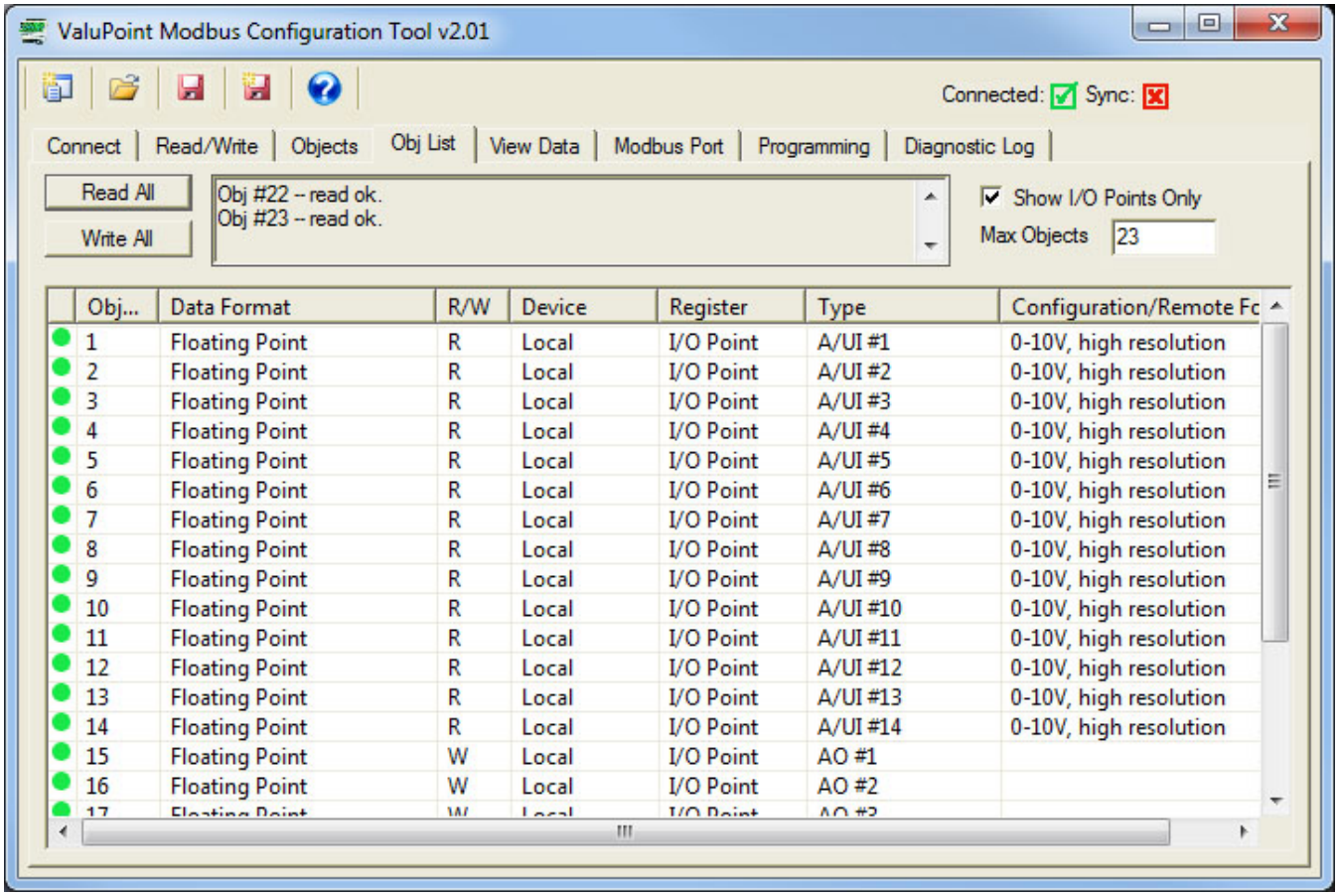
Click the 'open file' icon to read a set of object maps from an XML file. If you will be replicating the same configuration multiple times, you will save time by creating a file that can be reloaded. The currently open file will be indicated in the title bar of the program.

Click the 'save' icon to re-save the previously opened file, or to save for the first time if creating a new file. The presently displayed device configuration will be saved.

Click the 'save as' icon to create a copy of a previously opened file. A file dialog will appear allowing you to browse a selected directory and rename the file.

The file icons at the top of the tool screen will be different colors on different pages, and will be absent on some pages. These icons only appear where a file save is applicable. The different colors are meant to suggest different file types are used with different pages. There are 2 file types: (i) Object Map XML, (ii) Object Map CSV.

6 Object Map List



Obj...	Data Format	R/W	Device	Register	Type	Configuration/Remote Fc
1	Floating Point	R	Local	I/O Point	A/UI #1	0-10V, high resolution
2	Floating Point	R	Local	I/O Point	A/UI #2	0-10V, high resolution
3	Floating Point	R	Local	I/O Point	A/UI #3	0-10V, high resolution
4	Floating Point	R	Local	I/O Point	A/UI #4	0-10V, high resolution
5	Floating Point	R	Local	I/O Point	A/UI #5	0-10V, high resolution
6	Floating Point	R	Local	I/O Point	A/UI #6	0-10V, high resolution
7	Floating Point	R	Local	I/O Point	A/UI #7	0-10V, high resolution
8	Floating Point	R	Local	I/O Point	A/UI #8	0-10V, high resolution
9	Floating Point	R	Local	I/O Point	A/UI #9	0-10V, high resolution
10	Floating Point	R	Local	I/O Point	A/UI #10	0-10V, high resolution
11	Floating Point	R	Local	I/O Point	A/UI #11	0-10V, high resolution
12	Floating Point	R	Local	I/O Point	A/UI #12	0-10V, high resolution
13	Floating Point	R	Local	I/O Point	A/UI #13	0-10V, high resolution
14	Floating Point	R	Local	I/O Point	A/UI #14	0-10V, high resolution
15	Floating Point	W	Local	I/O Point	AO #1	
16	Floating Point	W	Local	I/O Point	AO #2	

6.0 Do This for First Time Startup

Click 'Read All' to read the entire object map list from the device. Initially the icons in the first column will all be read. This means the copy of configuration found on your PC has not been confirmed to match what is actually in the device. After reading configuration from the device, the icon will turn green, indicating your configuration tool is in sync with the device. Once every aspect of configuration is in sync, the global sync icon in the top right corner will also switch to green.

6.1 Read All, Write All

Click 'Read All' to begin the process of reading all of the available object maps from the device. If the tool was not in sync with the device previously, the red dots in the first column will turn green as the map on that line is updated and synced with the device.

Click 'Write All' to begin the process of writing all object maps to the device. It would generally be assumed that you will only click this button after opening a file containing a previously defined configuration.

6.2 Select for Edit

Click any line in the list of objects to be taken directly to the object map editing page for that object.

6.3 File Read / Write (Object Map CSV)

Click the ‘new’ icon to erase all object maps.

Click the ‘open file’ icon to read a set of object maps from a CSV file. If you will be replicating the same configuration multiple times, you will save time by creating a file that can be reloaded. The currently open file will be indicated in the title bar of the program.

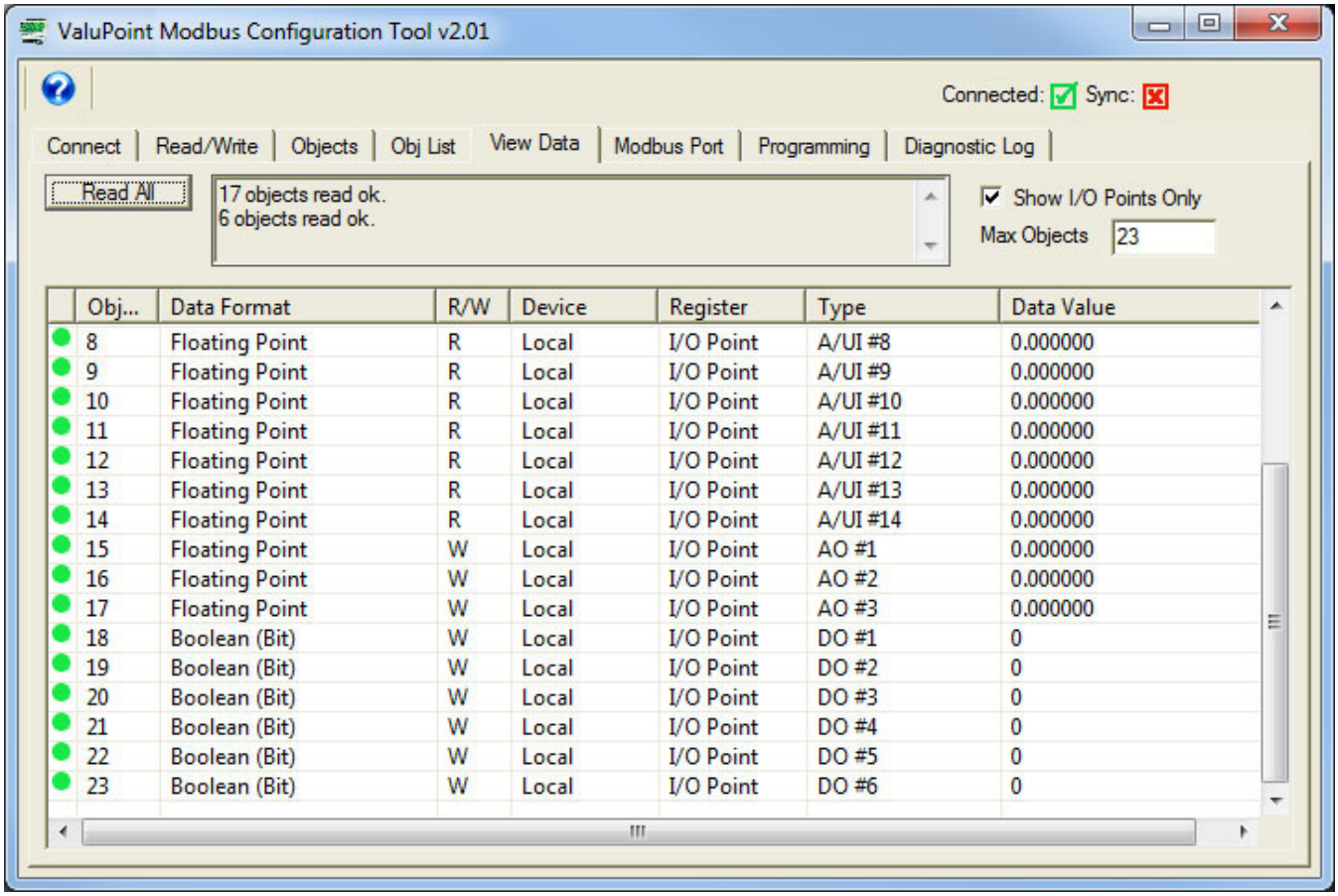
Click the ‘save’ icon to re-save the previously opened file, or to save for the first time if creating a new file. The presently displayed device configuration will be saved.

Click the ‘save as’ icon to create a copy of a previously opened file. A file dialog will appear allowing you to browse a selected directory and rename the file.

The file icons at the top of the tool screen will be different colors on different pages, and will be absent on some pages. These icons only appear where a file save is applicable. The different colors are meant to suggest different file types are used with different pages. There are 2 file types: (i) Object Map XML, (i) Object Map CSV.

NOTE: The CSV file must be in a specific format. The CSV file can be edited with any standard spread sheet program such as Microsoft Excel®. When creating or manipulating a large configuration, this can be a time saving approach to creating the configuration. However, some practice will be required to learn proper formatting, etc, before it will become a time saver. Refer to the section ‘Editing a CSV Object Map File’ that follows later.

7 Data List



Obj...	Data Format	R/W	Device	Register	Type	Data Value
8	Floating Point	R	Local	I/O Point	A/UI #8	0.000000
9	Floating Point	R	Local	I/O Point	A/UI #9	0.000000
10	Floating Point	R	Local	I/O Point	A/UI #10	0.000000
11	Floating Point	R	Local	I/O Point	A/UI #11	0.000000
12	Floating Point	R	Local	I/O Point	A/UI #12	0.000000
13	Floating Point	R	Local	I/O Point	A/UI #13	0.000000
14	Floating Point	R	Local	I/O Point	A/UI #14	0.000000
15	Floating Point	W	Local	I/O Point	AO #1	0.000000
16	Floating Point	W	Local	I/O Point	AO #2	0.000000
17	Floating Point	W	Local	I/O Point	AO #3	0.000000
18	Boolean (Bit)	W	Local	I/O Point	DO #1	0
19	Boolean (Bit)	W	Local	I/O Point	DO #2	0
20	Boolean (Bit)	W	Local	I/O Point	DO #3	0
21	Boolean (Bit)	W	Local	I/O Point	DO #4	0
22	Boolean (Bit)	W	Local	I/O Point	DO #5	0
23	Boolean (Bit)	W	Local	I/O Point	DO #6	0

7.0 Skip This for First Time Startup

Nothing needs to be done on this page to get the tool initially synced to the device.

7.1 Read All

Click ‘Read All’ to cause the tool to query all data objects in the ValuPoint device. A brief summary of the object’s mapping is displayed along with its present data value. Progress of the ‘Read All’ process will scroll through the log window at the top.

7.2 Data Definitions

The Object column indicates the data object number.

The Data Format column indicates the native data format configured for this object. The object may be read in any of multiple formats depending on which Modbus holding register range you use to access the object.

The R/W column indicates the following: R means periodic read, W means periodic write, W+ means write on delta. RW+ means a combination of those, etc.

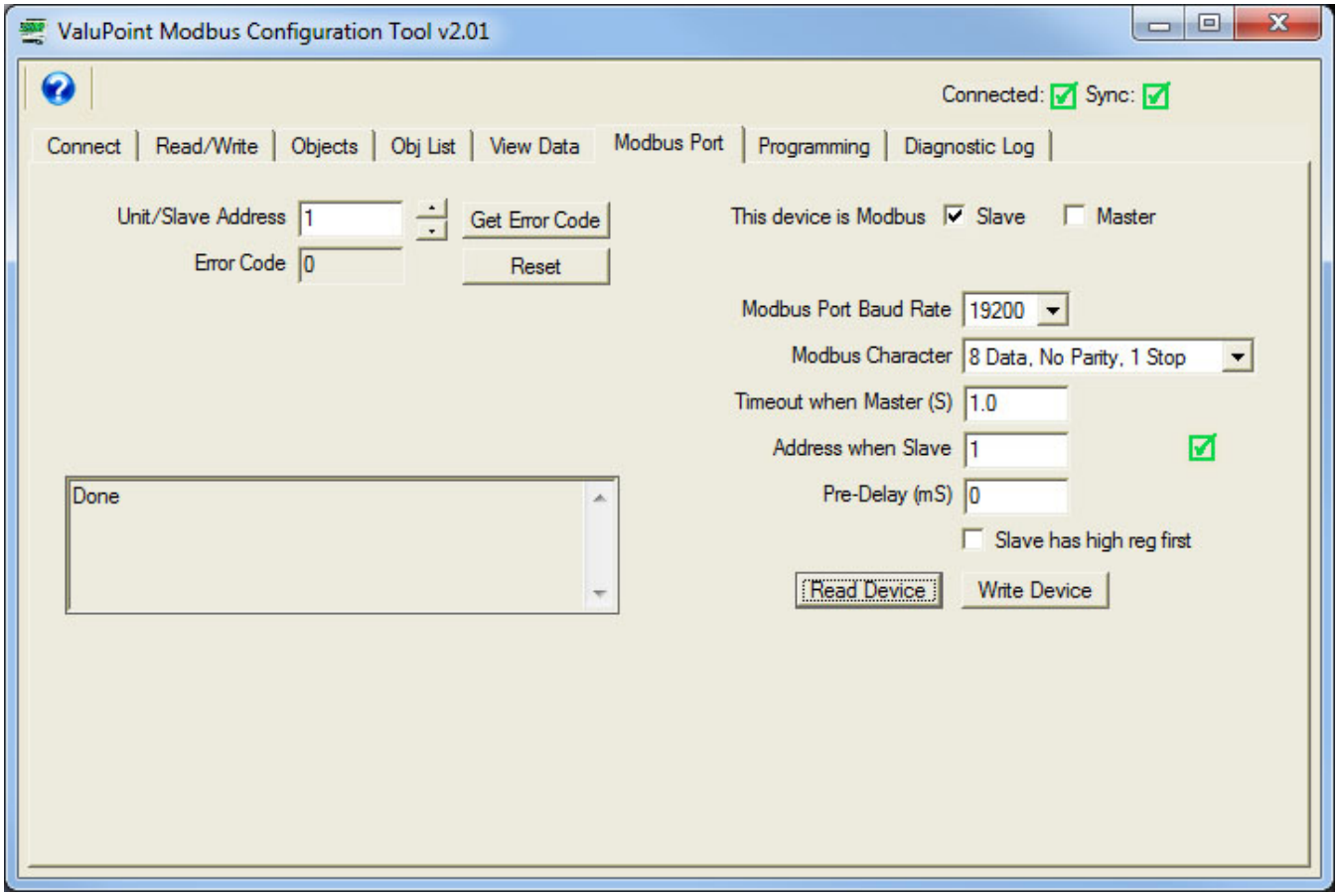
The Device column will show 'Local', or show what remote slave device address this object is mapped to. 'Local' means a physical I/O point on the local ValuPoint.

The Register column is significant only when the object is mapped via the Modbus RTU master to a remote slave device. The register number that will be queried in the remote slave device is listed here..

Type shows the physical I/O point locally, or the remote register type, such as "Holding Register".

Data Value shows a representation of the present value of the data object in the ValuPoint.

8 Modbus Page



8.0 Do This for First Time Startup

Click 'Read Device' to get the tool synced to the device. Then set parameters as applicable.

8.1 Set Modbus Port Parameters

Select whether the ValuPoint will be a Modbus Master or Slave. Remember that there can be only one master on a Modbus RTU network. If you switch from slave to master, upon clicking 'Write Device', you may need to go back to the Connect page and follow the process for entering configuration mode when device is configured as master. You are automatically always effectively in configuration mode when the device is configured to be a slave, but you will exit configuration mode when switching to master for the first time.

Select the Modbus port communication parameters, including baud rate, character format, and timeout. The timeout is the amount of time the ValuPoint will wait for a Modbus slave to respond if configured as master.

The 'Address when Slave' only applies if Slave is checked and the ValuPoint is going to be a Modbus slave responding to another master on the Modbus RTU network.

Pre-Delay (in milliseconds) specifies an amount of time the ValuPoint will wait before transmitting. The ValuPoint is fast enough to overrun some Modbus devices by either sending the next request too soon after the previous response, or by responding too fast as a slave. The packet overrun will manifest itself as no-response errors and/or CRC errors on the Modbus link. If you are seeing such errors and you have Pre-Delay set to zero, set it to at least 50 mS and continue testing.

IMPORTANT: Do not change device mode to Modbus master until AFTER you have created some Modbus register mappings for the master to act upon. These are set up on the Objects page. If you switch to master mode without anything for the master to do, you will have trouble entering configuration mode.

If you are having trouble communicating, and you suspect configuration parameters may be something other than what you expect, it should be noted that the device will ALWAYS be Modbus slave, communicating at 19200 baud N81, with slave address 247, for about 3 seconds after power up. The LED "lamp check" will flash all LED indicators about 3 times, then the heartbeat indicator (green) will remain on solid for about 3 seconds. During this time, you can click Read on the Modbus page to see what the port parameters in the device are. If they are not what you want, you can power cycle one more time and write new Modbus port parameters during that 3 second window. Once you have switched the device back to slave mode, you will be able to fully access the device.

8.2 Check / Reset Errors

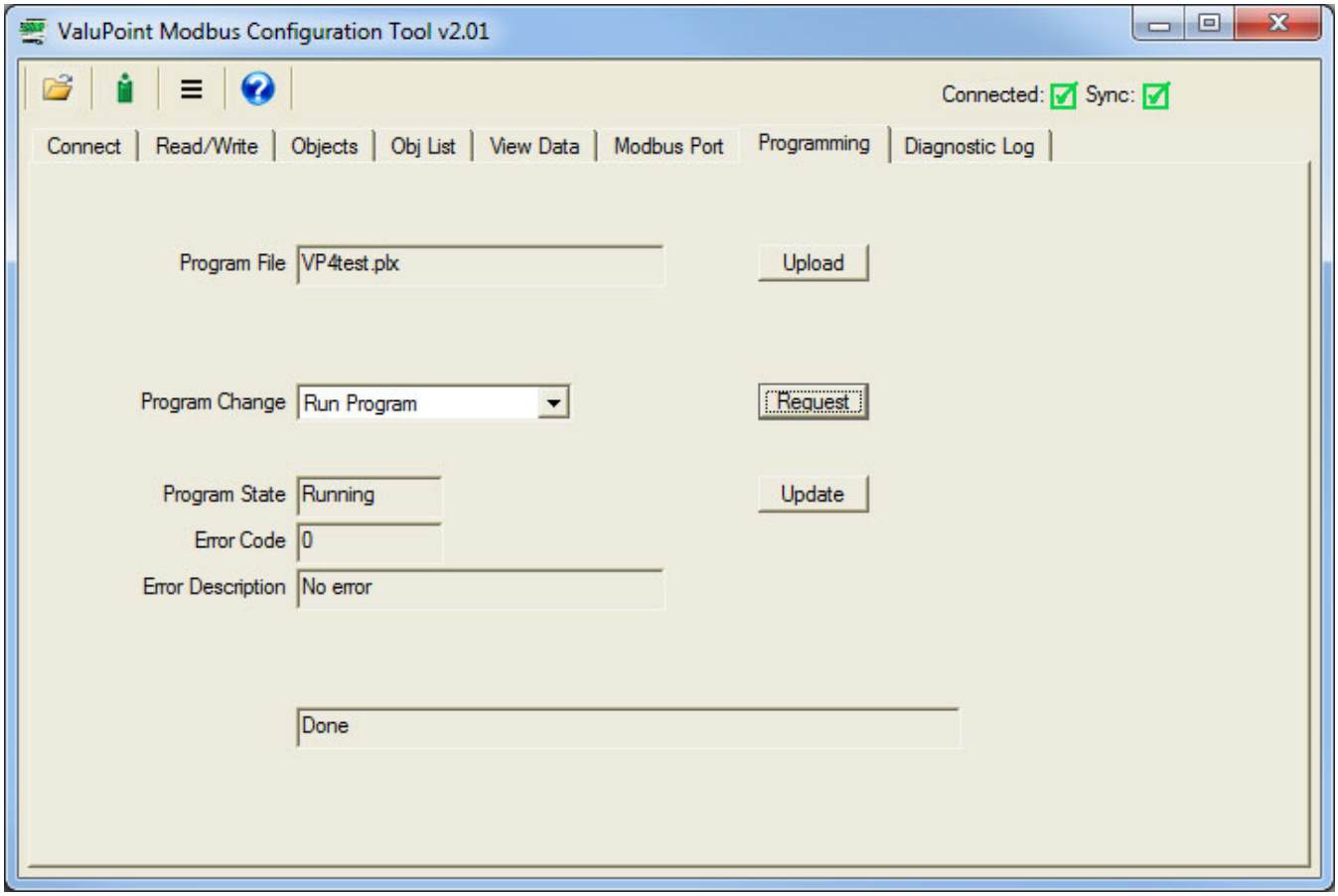
Enter or scroll to a desired Modbus slave address and click ‘Get Errors’ to see the error counts recorded by ValuPoint for that device. When configured as slave, errors will be logged as slave address 1 for purposes of error counting regardless of what the actual slave address is. When configured as master, counts are kept for each slave address that the master is querying; however, the counts can only be read while in configuration mode. This means you can only observe error counts after the ValuPoint has operated as master for some time, and then temporarily placed in slave mode by entering configuration mode from the Connect page.

Click ‘Reset’ to clear the error counts and message count for the Modbus device whose address is currently displayed.

Error codes will be as follows, with lower numbered codes being the standard Modbus exception codes as defined by Modbus protocol, and the higher numbered codes being indicators of non-exception type errorsl.

Standard Modbus Exception Codes		
1	Illegal Function	The function code received in the query is not recognized by the slave or is not allowed by the slave.
2	Illegal Data Address	The data address (register number) received in the query is not an allowed address for the slave, i.e., the register does not exist. If multiple registers were requested, at least one was not permitted.
3	Illegal Data Value	The value contained in the query's data field is not acceptable to the slave.
4	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action
6	Slave Device Busy	The slave is engaged in processing a long-duration command. The master should try again later.
10	Gateway Path Unavailable	Specialized use in conjunction with gateways, usually means the gateway is misconfigured or overloaded
11	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways, indicates no response was received from the target device.
ValuPoint Specific Codes (indicating non-exception errors)		
129	No Response	Valid only if ValuPoint is Modbus master, indicates the addressed slave has failed to respond 1 or more times.
130	CRC Error	Valid only if ValuPoint is Modbus master, indicates that a CRC error in slave's reply has been found 1 or more times.
131	No Response/CRC	Simply indicates both of the above (129, 130) are true.

9 PL/i Programming for Control



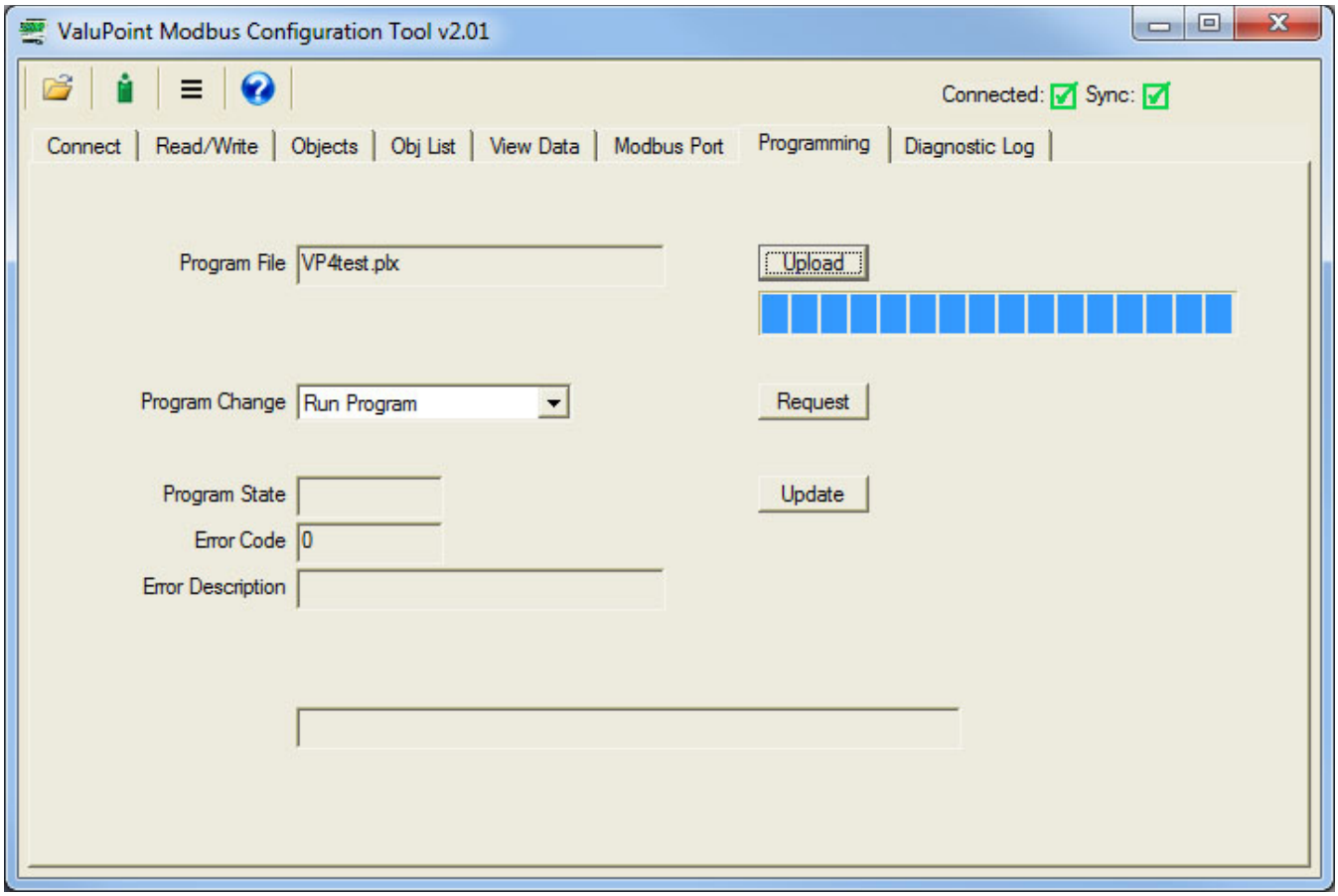
9.0 Skip This for First Time Startup

There is nothing that needs to be done on this page for initial startup. To use the ValuPoint as a simple slave I/O device, you will not need this page at all. If you want to turn the ValuPoint into a programmable controller, you will start here.

9.1 Program Loading and Execution

Click on the file folder icon at the top left to open a file. The file open dialog will appear. Select a .plx file from the list. If you do not yet have any programs compiled, you will need to use the program editing tools to create and compile a program.

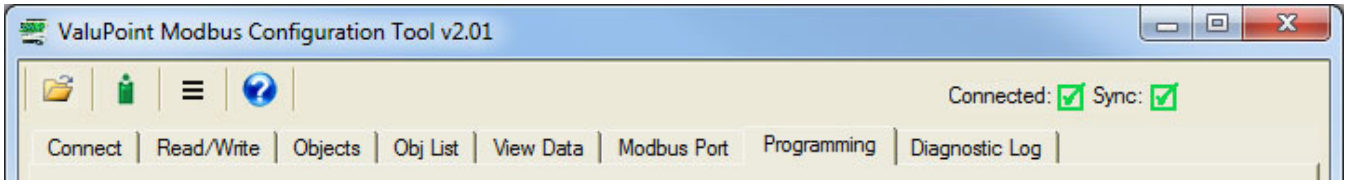
After a program (.plx file) has been opened, click the Upload button to send that program to the ValuPoint. A progress bar will indicate program loading progress.



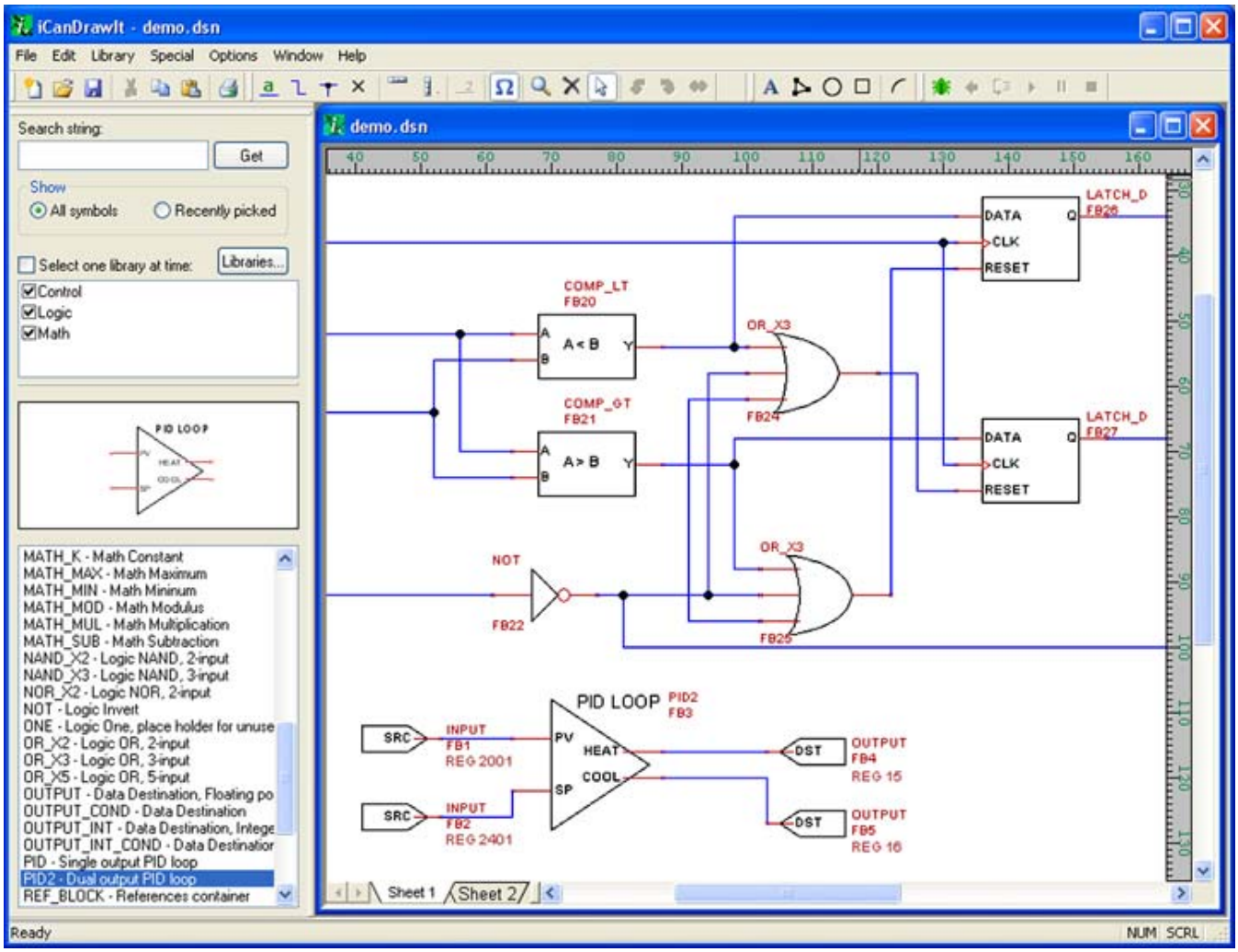
To invoke execution of the program, select 'Run Program' from the Program Change list, and then click Send.

If the program encounters a fatal error during execution, its error code and description will be displayed after clicking the Update button. The Update button causes the tool to query the ValuPoint device for status.

9.2 Program Editing and Debugging



Click the green "i" icon next to the folder icon to open the i.CanDrawIt graphical programming tool that is illustrated below. It has its own set of help pages. Click on the "?" icon in that tool for more about programming. (Note that i.CanDrawIt is a second software package installed after the ValuPoint configuration tool - if you have trouble starting up i.CanDrawIt, be sure it was installed.)



Click the "line" icon next to the "?" icon to open the line programming tool. If you do not want to "draw" a program, but would rather write a program using the native PL/i programming language, you can do this. The line programming tool also has its own set of help pages.

The PL/i programming language is a derivative of PL/1 but is not the same as PL/1. The language is referred to as PL/i with "i" as in i.CanDrawIt.

9.3 Program Capacity

Maximum compiled program (.plx file) size: 65,280 bytes
Maximum RAM available for program variable and stack space: 8192 bytes
Maximum EEPROM available: 2048 bytes

9.4 Program States and Error Codes

BACnet Program Change codes:

- 0 = ready
- 1 = load
- 2 = run
- 3 = halt
- 4 = restart
- 5 = unload

BACnet Program State codes:

0 = idle
1 = loading
2 = running
3 = waiting
4 = halted
5 = unloading

Reason for Halt codes (BACnet Program Error):

0 = no error (not halted)
1 = program load failed
64 = normal stop, end of program reached
65 = external stop via Program Change
66 = debug execution, suspended
+n = error code (400 or above)

Non-fatal runtime errors:

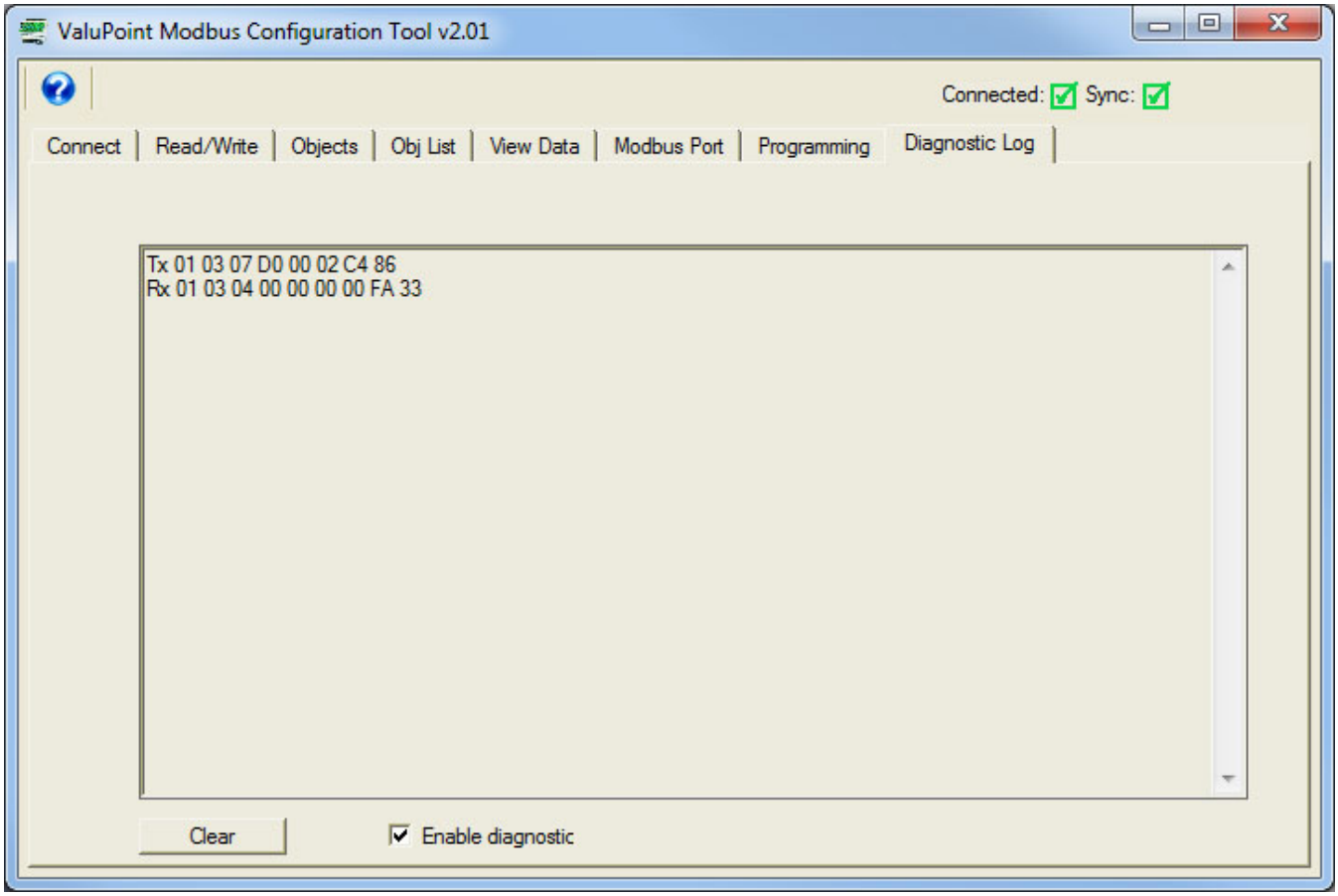
401: subscript out of bounds, non-fatal
402: divide by zero, non-fatal
406: EEPROM address out of range, operation skipped
407: object instance out of bounds, operation skipped

Note: Error codes will show up as "Reason for Halt" even if the error was not necessarily fatal. This is because "Reason for Halt" is the only available standard Program Object property whose purpose is to report errors. Check toe Program State to determine if the program is actually halted.

Fatal runtime errors:

451: unrecognized opcode, fatal
452: stack overflow, fatal
453: stack underflow, fatal
454: program pc out of bounds, fatal

10 Diagnostic Log



10.0 Skip This for First Time Startup

This page is not necessary for initially getting the device in sync with the tool. This page is also not necessary for performing any configuration functions. It is only a diagnostic tool useful in trouble shooting, provided you have a clear understanding of raw Modbus packets.

10.1 Using the Diagnostic Log

The diagnostic log is useful for verifying exactly what is going over the wire between the configuration tool and the device. The raw content of all Modbus packets sent to the ValuPoint device by the tool are displayed on a line labeled "Tx". The raw content of all Modbus packets received from the ValuPoint device by the tool are displayed on a line labeled "Rx". Each individual packet is displayed on a separate line.

Check the 'Enable diagnostic' box to enable the log. Click Clear to erase the contents of the window.

Note that the Read/Write page can be used to query any Modbus RTU device physically connected to the same network as the configuration tool. This means the ValuPoint tool can be a useful diagnostic tool for testing other Modbus devices. If you are using this tool to query a device other than a ValuPoint, all communication with that device will also show up here on the Diagnostic Log.

Note that only communication between the configuration tool and some other Modbus device are displayed here. Communication between two other Modbus devices will not be captured here. Only communication involving the tool is displayed.

11 Modbus Register Map

11.1 Modbus Registers - Full Map

The following chart shows the available Modbus registers. Data objects are accessible as holding registers, as well as input register, discrete input, and coil. Special registers such as configuration, etc, are only accessible as holding registers, as indicated by the chart below.

Modicon Std	Modicon Extd	Type	Std. Reg. No.	Description
Data object access registers (data registers)				
40001-40320	400001-400320	Holding	1-320	Objects 1-320 accessed as signed integer, 16-bit, single Modbus register VP4-2310: A/UI 1-14 = objects 1-14; AO 1-3 = objects 15-17; DO 1-6 = objects 18-23 VP4-2810: DI 1-12 = objects 1-12; DO 1-16 = objects 13-28
40801	400801	Holding	801	A/UI inputs accessed as packed register of bits, 1 bit per input, LSB=input 1 (if applicable)
40802	400802	Holding	802	DI inputs accessed as packed register of bits, 1 bit per input, LSB=input 1 (if applicable)
40803	400803	Holding	803	DO outputs accessed as packed register of bits, 1 bit per output, LSB=input 1 (if applicable)
41001-41320	401001-401320	Holding	1001-1320	Objects 1-320 accessed as unsigned integer, 16-bit, single Modbus register
42001-42640	402001-402640	Holding	2001-2640	Objects 1-320 accessed as IEEE754 floating point, 32-bit, double Modbus register
43001-43640	403001-403640	Holding	3001-3640	Objects 1-320 accessed as signed integer, 32-bit, double Modbus register
44001-44640	404001-404640	Holding	4001-4640	Objects 1-320 accessed as unsigned integer, 32-bit, double Modbus register
47001-47030	407001-407030	Holding	7001-7030	Real time clock/calendar registers
48001-48246	408001-408246	Holding	8001-8246	Modbus error codes for slaves 1-246 (8001 if VP4 is slave)
48401	408401	Holding	8401	Modbus port baud rate
48402	408402	Holding	8402	Modbus port slave address when operating as slave
48403	408403	Holding	8403	Modbus port timeout in tenths of seconds when operating as master
48404	408404	Holding	8404	Modbus port pre-delay
48405	408405	Holding	8405	Modbus port configuration bits (b0=set if big endian; b1=set for slave mode; b2=enable parity; b3=parity is odd; b4=two stop bits
49901	409901	Holding	9901	Configuration mode register
49902	409902	Holding	9902	Firmware revision (read only) (nxxyy where n=major, xx=minor, yy=build)

49903	409903	Holding	9903	Test mode register (write 9901 to test discrete, 9902 to test analog, 9900 for default I/O config)
---	410001-410320	Holding	10001-10320	Poll timers for objects 1-320, 16-bit unsigned integer, single Modbus register
---	410801	Holding	10801	Index register, write value 1-320 to index holding register 10001-10320
---	410802	Holding	10802	Value of register indexed by 10801
---	411001-411320	Holding	11001-11320	New data flags for objects 1-320, 16-bit unsigned integer, single Modbus register
---	411801	Holding	11801	Index register, write value 1-320 to index holding register 11001-11320
---	411802	Holding	11802	Value of register indexed by 11801
Configuration registers				
---	420001	Holding	20001	Index register, write 1-320 to index configuration register set for object 1-320
---	420002	Holding	20002	Object use configuration bit set (objectUse_t)
---	420003	Holding	20003	Poll time in seconds (uint16)
---	420004	Holding	20004	Default host timeout in seconds (uint16)
---	420005	Holding	20005	Max. quiet time in seconds (uint16)
---	420006-420007	Holding	20006-20007	Scale factor applied after mask (float)
---	420008-420009	Holding	20008-20009	Offset applied after scale (float)
---	420010-420011	Holding	20010-20011	Default value to apply upon read failure, POR, host timeout (u_data)
---	420012-420013	Holding	20012-20013	Send delta, change in local data needed to resent to remote (u_data)
---	420014	Holding	20014	Object link reference, link to virtual object
Modbus register definitions apply if 20001 defines this as a Modbus client map				
---	420015	Holding	20015	Modbus register number/point number (1-based index) (uint16)
---	420016H	Holding	20016H	Register type, 0=NONE, 1=REG_0X, 2=REG_1X, 3=REG_3X, 4=REG_4X
---	420016L	Holding	20016L	Format for registers
---	420017H	Holding	20017H	Remote unit # to query
---	420017L	Holding	20017L	Number of fails before calling it a fault (apply default value)
---	420018-420019	Holding	20018-20019	Bit mask to strip field out of 16-bit register (uint32)
---	420020-420021	Holding	20020-20021	Bit mask fill bits (uint32)
Hardware register definitions apply if 20001 defines this as a Hardware point				
---	420015	Holding	20015	Hardware channel code
---	420016	Holding	20016	Hardware channel qualifier 1
---	420017	Holding	20017	Hardware channel qualifier 2

---	420100	Holding	20100	Write value of 20100 to register 20100 to force NV memory write
Data mirror registers, data objects accessed via Modbus function codes other than holding registers				
00001-00320	000001-000320	Coil	1-320	Objects 1-320 accessed as single bit registers
10001-10320	100001-100320	Discrete Input	1-320	Objects 1-320 accessed as single bit registers
30001-30320	300001-300320	Input	1-320	Objects 1-320 accessed as unsigned integer, 16-bit, single Modbus register
30801	300801	Input	801	A/UI inputs accessed as packed register of bits, 1 bit per input, LSB=input 1 (if applicable)
30802	300802	Input	802	DI inputs accessed as packed register of bits, 1 bit per input, LSB=input 1 (if applicable)
30803	300803	Input	803	DO outputs accessed as packed register of bits, 1 bit per output, LSB=input 1 (if applicable)
31001-31320	301001-301320	Input	1001-1320	Objects 1-320 accessed as signed integer, 16-bit, single Modbus register
32001-32640	302001-302640	Input	2001-2640	Objects 1-320 accessed as unsigned integer, 32-bit, double Modbus register
33001-33640	303001-303640	Input	3001-3640	Objects 1-320 accessed as signed integer, 32-bit, double Modbus register
34001-34640	304001-304640	Input	4001-4640	Objects 1-320 accessed as IEEE754 floating point, 32-bit, double Modbus register

11.2 Modbus Registers - Real Time Clock Access

The following holding registers are available for access to the battery backed real time clock/calendar in the ValuPoint. Registers 7001-7007 will return the respective element of time as of the register read. The clock could roll over between successive reads, leading to an incorrect overall time stamp. Use the registers in the range of 7001-7007 only if you are basing an algorithm on whether day is the same as previous day, etc. To capture a complete correct timestamp, read registers 7011-7017, and be sure to read 7011 first. Reading register 7011 (year) locks the rest of the time stamp and the remaining registers will return whatever the time/date was when register 7011 was read.

To set the clock/calendar, write all of registers 7011-7017, then write any value to register 7018 to trigger the write. Nothing is done with the content of register 7018 - it is only the trigger to tell ValuPoint to store the content of registers 7011-7017 into the clock/calendar hardware.

Holding Reg. No.	Writeable	Description
7001	No	Year
7002	No	Month
7003	No	Day of Month
7004	No	Hour (0..23)
7005	No	Minute
7006	No	Second
7007	No	Day of Week (1=Sunday, 2=Monday, etc)
7011	Yes	Year (is also lock trigger for read)
7012	Yes	Month
7013	Yes	Day of Month

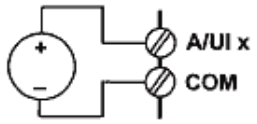
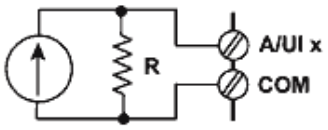
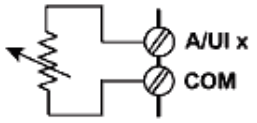
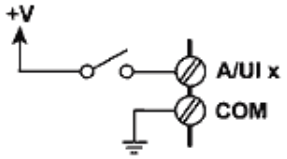
7014	Yes	Hour (0..23)
7015	Yes	Minute
7016	Yes	Second
7017	Yes	Day of Week (1=Sunday, 2=Monday, etc)
7018	Yes	Lock trigger for write
7021	No	Minutes since midnight
7022	No	Day of year (Jan 1 = 1, Dec. 31 = 366 if leap year, else 365)

12 Physical I/O Connections

12.1 Connection of Inputs

The VP4-2310 and VP4-2810 contain no configuration jumpers for configuring I/O points. There is no need to open the enclosure for configuration of I/O. Input types are switched under software control.

Input points should be connected as indicated in the various diagrams below. In addition to selecting a wiring diagram, the corresponding selections should be made on the I/O List page of the configuration tool.

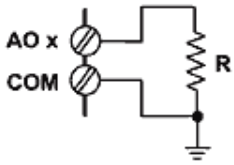
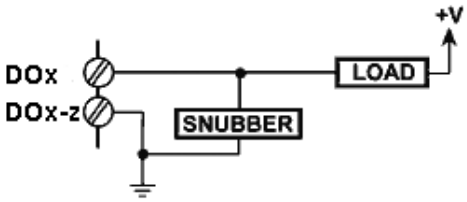
I/O Point Type	Wiring Guide	Additional Information
A/UI Analog Input: 0-10VDC Voltage Input		VP4-2310: A/UI inputs 1-14 will accept voltage inputs of up to 10VDC. Voltages to 12VDC will be measured. Voltages to 24VDC will be tolerated, but measurement is internally limited to a reading of 12VDC.
A/UI Analog Input: 0-20mA Current Input		VP4-2310: A/UI inputs 1-14 will accept current inputs of 0-20mA with the addition of a 500 ohm 1/2 watt external resistor. A 500 ohm resistor will produce a 0-10VDC signal. A 250 ohm resistor may also be used to produce a 0-5VDC signal.
A/UI Analog Input: Thermistor Input		VP4-2310: A/UI inputs 1-14 will accept thermistors of 3k, 10k, or 20k ohms. Linearization via interpolation of a 56-point table is performed internally.
A/UI or DI Discrete Input: Discrete Voltage		VP4-2310: A/UI inputs 1-14 may be connected as discrete voltage sensing inputs. Inputs up to 24VDC are tolerated, but threshold sensing only functions over the 0-10VDC range. VP4-2810: DI inputs 1-12 may be connected as discrete voltage sensing. Inputs up to 24VDC are tolderated. The switching threshold is around 3VDC. See note below.
A/UI or DI Discrete Input: Dry Contact Closure to Ground		VP4-2310: A/UI inputs 1-14 may be connected as discrete inputs sensing dry contact closure to ground. Internal excitation of 10mA is provided. VP4-2810: DI inputs 1-12 may be connected for sensing dry contact closure to ground. Internal excitation of 10mA is provided. NOTE: A/UI types are configured individually, but DI types are configured in groups. On the VP4-2810, DI 1-6 are configured by the selection for DI 1, and DI 7-12 are configured by the selection for DI 7. DI type refers to discrete voltage versus contact closure to ground.

12.2 Connection of Outputs

The VP4-2310 and VP4-2810 contain no configuration jumpers for configuring outputs. There is no need to open the enclosure.

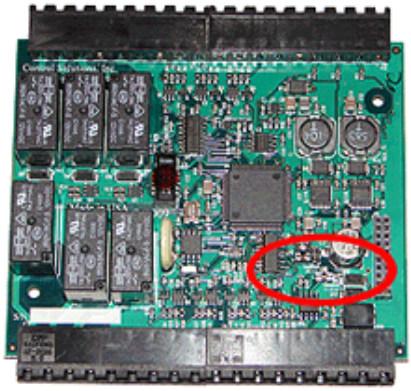
Output points should be connected as indicated in the various diagrams below. For VP4-2310, the discrete

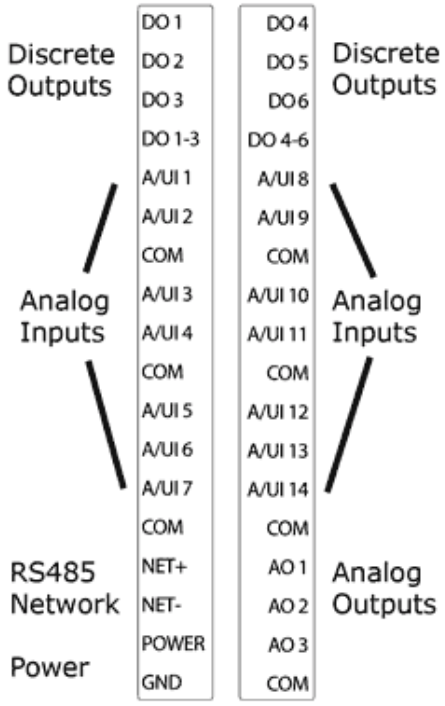
(relay) outputs are SPST N.O. with the common side banked in groups of 3. The DO1-3 and DO4-6 are not electrically common to each other or to COM/GND connections. For VP4-2810, all outputs are open drain FET outputs that sink up to 1A @ 30VDC when the output is "on". The FET outputs sink current to the COM/GND terminals which are all electrically common on VP4-2810.

I/O Point Type	Wiring Guide	Additional Information
VP4-2310 only: Analog Output: 0-20mA Current Output		AO outputs 1-3 produce outputs of 0-20mA. Use of an external 500 ohm resistor will produce a 0-10VDC voltage output. For best results, locate the resistor at the input of the receiving device.
Discrete Output: **VP4-2310 Form A Relay** **VP4-2810 Open Drain FET**		VP4-2310: DO outputs 1-6 are Form A dry contact relays rated for 2A @ 120VAC (resistive). Snubbers should be used with inductive loads. The relays are also rated for 2A @ 30VDC. Note: Relays are rated higher, but UL listing is for 2A. VP4-2810: DO outputs 1-16 are open drain FET outputs capable of sinking 1A @ 30VDC in the "on" state. FET outputs may NOT be used for AC loads.

12.3 Connection of Power and Communications, VP4-2310

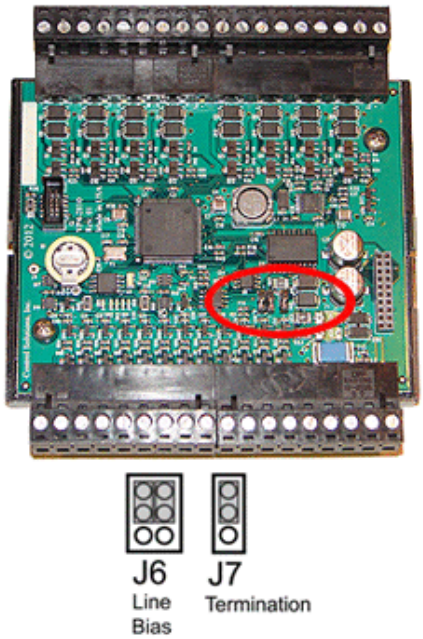
Power and communications should be connected as indicated below.

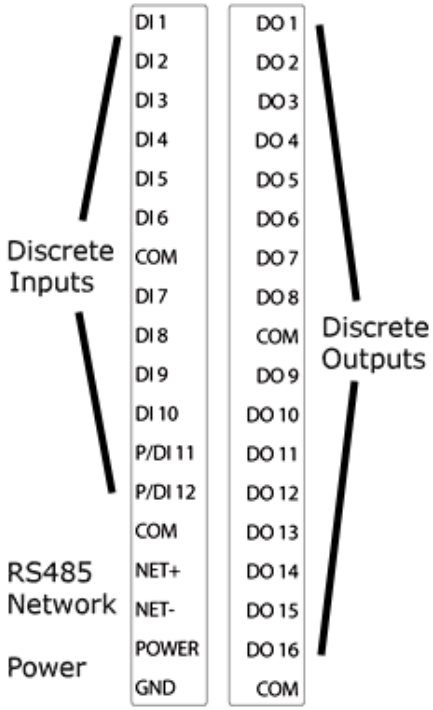
I/O Point Type	Wiring Guide	Additional Information
Power	Connect AC or +DC power to Power terminal. Connect common or -DC power to GND terminal. GND is common to all terminals labeled COM.	Nominal power consumption is 6 watts, or 0.25A @ 24VDC, with all relays on and all analog outputs at 20mA.
Communications	Connect MS/TP RS-485 network to NET+/- terminals.	Communication signals comply with EIA-485 standard.
Line Conditioning		The VP4-2310 is shipped with network line bias and termination disabled. If the VP4-2310 will be the last device at the end of the RS-485 link, then termination should be enabled. Often it is also necessary to have line bias enabled, but this must only be done at one point on the network (not both ends like termination). The bias and termination jumpers are located in the area of the circuit board highlighted. The bias and termination are enabled when the shunts are aligned with the white bars next to the header pins. Move the shunts to the opposite end to disable. It is not necessary to enable both at the same time. (Illustration shows both enabled.) To open the VP4-2310, use a very small screw driver to slide the cover up over the plastic tabs on the ends of the case. Do not push the tabs in as they are fragile. Slide the cover over the tabs by prying the cover slightly away from the tabs.

Wiring Terminals		Screw terminals ratings are substantially in excess of any I/O point ratings. Screw terminals are pluggable. They unplug from the unit in 9-position blocks. Any combination of smaller 5.00mm terminal blocks can optionally be used instead.

12.4 Connection of Power and Communications, VP4-2810

Power and communications should be connected as indicated below.

I/O Point Type	Wiring Guide	Additional Information
Power	Connect AC or +DC power to Power terminal. Connect common or -DC power to GND terminal. GND is common to all terminals labeled COM.	Nominal power consumption is 4 watts, or 0.15A @ 24VDC.
Communications	Connect MS/TP RS-485 network to NET+/- terminals.	Communication signals comply with EIA-485 standard.
Line Conditioning		The VP4-2810 is shipped with network line bias and termination disabled. If the VP4-2810 will be the last device at the end of the RS-485 link, then termination should be enabled. Often it is also necessary to have line bias enabled, but this must only be done at one point on the network (not both ends like termination). The bias and termination jumpers are located in the area of the circuit board highlighted. The bias and termination are enabled when the shunts are positioned on the upper pair of pins. Move the shunts to the lower pair to disable. It is not necessary to enable both at the same time. (Illustration shows both enabled.) To open the VP4-2810, use a very small screw driver to slide the cover up over the plastic tabs on the ends of the case. Do not push the tabs in as they are fragile. Slide the cover over the tabs by prying the cover slightly away from the tabs.

Wiring Terminals		Screw terminals ratings are substantially in excess of any I/O point ratings. Screw terminals are pluggable. They unplug from the unit in 9-position blocks. Any combination of smaller 5.00mm terminal blocks can optionally be used instead. Note that Discrete Inputs 11 and 12 are also the high speed pulse inputs.

13 Trouble Shooting

13.1 Modbus Trouble Shooting

There are multiple ways of observing errors. One is to check error codes on the Modbus tab of the configuration tool. Another is to check the error codes being reported by the Modbus master that is polling ValuPoint as a slave. The third is to look at the LED indicators inside the ValuPoint (visible through vent slots).

There can be a variety of reasons why you are not getting the data you expect in Modbus communications. No-response errors are probably the toughest because it means no activity is being recognized. CRC errors are marginal progress because it says the devices are at least seeing some bits on the line, even if the bits don't make sense yet. Exception errors are a good sign because it means you are successfully communication with the Modbus device. Receiving an exception error requires receiving a good packet with a good CRC check. This means communication is ok, but configuration is asking for something the Modbus device does not like.

No-response errors:
• Check to see that communication parameters are correct (baud rate, etc).
• Check to see that the slave address matches.
• Check to see that Pre-Delay is at least 50 mS.
• Check wiring and power.
• Check for reversed polarity on RS485 lines. If uncertain, just try swapping them.
• Check to see that slave device is enabled for Modbus communication (many devices default to disabled)

CRC errors:
• Check baud rate and character format.
• Check wiring – if everything else is correct, CRC errors mean noise on the line.
• Check for reversed polarity on RS485 lines. Reversed polarity often looks like just noise.
• Check to see that Pre-Delay is at least 50 mS.

Exception errors:
• Check configuration. You cannot receive an exception error report if you are not successfully communicating with the Modbus device. Wiring, etc, is not a problem. Configuration has something wrong (most often the register number requested is not available).

13.2 Modbus Exception (error) Codes

When a Modbus slave recognizes a packet, but determines that there is an error in the request, it will return an exception code reply instead of a data reply. The exception reply consists of the slave address or unit number, a copy of the function code with the high bit set, and an exception code. If the function code was 3, for example, the function code in the exception reply will be 0x83. The exception codes will be one of the following:

Standard Modbus Exception Codes		
1	Illegal Function	The function code received in the query is not recognized by the slave or is not allowed by the slave.
2	Illegal Data Address	The data address (register number) received in the query is not an allowed address for the slave, i.e., the register does not exist. If multiple registers were requested, at least one was not permitted.

3	Illegal Data Value	The value contained in the query's data field is not acceptable to the slave.
4	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action
6	Slave Device Busy	The slave is engaged in processing a long-duration command. The master should try again later.
10	Gateway Path Unavailable	Specialized use in conjunction with gateways, usually means the gateway is misconfigured or overloaded
11	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways, indicates no response was received from the target device.
ValuPoint Specific Codes (indicating non-exception errors)		
129	No Response	Valid only if ValuPoint is Modbus master, indicates the addressed slave has failed to respond 1 or more times.
130	CRC Error	Valid only if ValuPoint is Modbus master, indicates that a CRC error in slave's reply has been found 1 or more times.
131	No Response/CRC	Simply indicates both of the above (129, 130) are true.

13.3 LED Indicators in ValuPoint

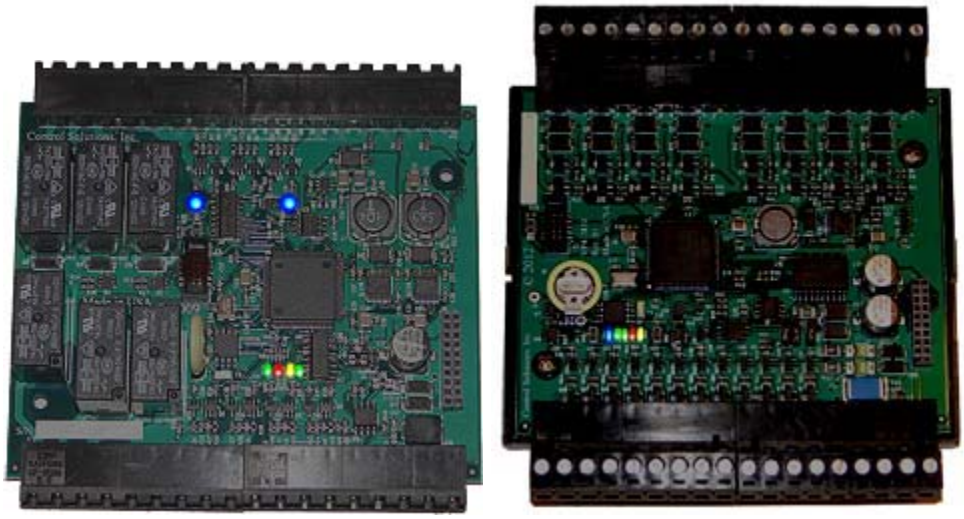
The LED indicators found on the VP4-2310 and VP4-2810 respectively are shown below. These can be viewed through the air vent slots in the case. The blue LEDs are power indicators. The row of LEDs that includes green, red, and yellow are the system and communication status indicators. The leftmost green LED is the system heartbeat which normally is on while flashing off briefly about once a second. It will flash more rapidly when the device is configured as Modbus master but is in configuration mode (meaning it is temporarily acting as a slave).

The remaining green, red, and yellow LEDs are normally the Modbus communication indicators. If there is a serious system error to report, the red LED temporarily acts as the system error indicators (see below section on system errors).

The communication indicators behave as follows:

While the ValuPoint is functioning as a Modbus slave, the yellow LED flashes every time a request is recognized as being addressed to this slave. The green will flash if a good reply is transmitted back to the master. The red will flash instead if an error was detected. If the error results in an exception code, that code will be transmitted back to the master. If the error was a CRC error, the request is discarded.

While the ValuePoint is functioning as Modbus master, the yellow LED flashes every time a request is sent out to a slave. The green will flash when a good reply is received from the addressed slave. The red will flash instead if no packet was received within the configured response timeout, or if the reply contained an error such as CRC check failed.



13.4 System Fault Indications

The red LED visible inside the VP4-2310 case, viewed through the vent slots, is the system fault indicator. It will be on during initial power-up boot mode operation, but should otherwise be off except for indicating communications errors.

During normal operation, a watchdog timer is always running to force a soft restart the system in the event of a software hang. Should the system restart as a result of watchdog timeout, the system fault LED will flicker at a very rapid rate for approximately 10 seconds.

System fault indication past the initial few seconds of boot mode followed by possible soft restart indication would consist of the red LED being mostly on, flashing off briefly some number of times, followed by a longer pause remaining on. Count the number of 'off' flashes. This is the fault code.

Fault codes are as follows:

(1-6) Processor abort codes
(7) System configuration CRC failed
(8) Bad configuration parameter found in system configuration
(9) EEPROM read failed
(10) EEPROM write failed
(11) EEPROM lock failed
(12) Object allocation failed

Report any of these to technical support. There are both a primary copy and backup copy of system configuration information. Both need to fail before the fault code will be indicated. These will generally indicate a hardware failure requiring factory attention. You should really never see any fault codes.

A processor abort will initially be indicated by the red LED on solid, and the yellow LED flashing a code from 1 to 6. However, the watchdog timer will normally restart the system sooner than you can observe this code and normal system fault indication will continue from that point. The abort cause is saved through soft restart. The term 'soft restart' means processor reset and complete reboot of the system as if power cycled. A power cycle is required for hard reset, and results in the same startup sequence except that processor abort codes are not retained.

14 Editing a CSV Object Map File

The easiest way to begin the process of configuring a device by editing a CSV file is to first create one using the configuration tool. Select the device type first. Then go to the Object Map List page. Even though there will be an essentially blank list here, click the Save button to create a file with place holders for the objects. Then use a spread sheet program to modify the various entries to your liking.

Column A: Object – Numeric object number, 1 to 320, not necessarily the Modbus register number - each object can be accessed via more than one Modbus register number.

Column B: DataFormat - numeric code defining internal data format for this object:
0 = null/unused
1 = floating point
2 = 32-bit unsigned int
3 = 32-bit signed int
4 = 16-bit unsigned int
5 = 16-bit signed int
6 = boolean/bit

Column C: IsHardware – T if mapped to physical I/O point, or F

Column D: IsModbus – T if mapped to external Modbus slave register, or F

Column E: IsPacked - T if object is member of multiple object packed Modbus register, or F

Column F: DefPOR – T if default on POR enabled, or F

Column G: DefNOK – T if default on comm. fail enabled, or F

Column H: ReadPoll – T if periodic Read, or F

Column I: WritePoll – T if periodic Write, or F

Column J: WriteDelta – T if Write on Delta, or F

Column K: HighRegFirst - T if most significant part of data is in first register of the pair, or F (applies to data >16 bits)

Column L: EnabMaxQuiet - T if maximum quiet time enabled, or F

Column M: IsPersistent - T if present value is preserved through power outage, or F

Column N: ObjIsLinked - T if object is linked to another object, or F

Column O: PollTime – Integer, Periodic poll time in seconds

Column P: Timeout – Integer, BACnet slave timeout in seconds

Column Q: MaxQuiet – Integer, max quiet time in seconds

Column R: Scale – Real, scale factor

Column S: Offset – Real, offset for scaling

Column T: DefaultValue – Real or Integer as applicable to object type, default value

Column U: SendDelta – Real, delta threshold for Write on Delta

Column V: LinkObj - Object number that this object is linked to as the source of data for this object.

Column W: HwCfg - Hardware configuration code if object is mapped to physical I/O (see Section 8 for codes)

Column X: HwQual1 - Hardware configuration qualifier value #1

Column Y: HwQual2 - Hardware configuration qualifier value #2

Column Z: RemoteRegNum – Integer, Modbus register number

Column AA: RemoteRegType – ASCII string, 2-character code, representing Modbus register type:

- 'NO' – none
- '0X' – coil(s) – uses FC15 to write
- '1X' – discrete input
- '3X' – input register
- '4X' – holding register(s) – uses FC16 to write
- '0S' – coil, use FC5 to write single
- '4S' – holding register, use FC6 to write single

Column AB: RemoteRegFormat – ASCII string, 4-character code, representing Modbus register format:

- 'NONE' – none
- 'SIGN' – signed integer (16-bit)
- 'UNSI' – unsigned integer (16-bit)
- 'SDBE' – signed double integer, big endian (32-bit, register pair)
- 'UDBE' – unsigned double integer, big endian (32-bit, register pair)
- 'FPBE' – floating point, big endian (register pair)
- 'BBIT' – bit

Column AC: SlaveId – Integer, Modbus slave address

Column AD: Mask – Integer, hexadecimal representation

Column AE: Fill – Integer, hexadecimal representation

Column AF: FailCount – Integer, count of comm. fails before Fault indicated