



# User Guide Supplement

**Model VP6-1460**
**ValuPoint IoT**
**Edge Server for Modbus**

**Rev. 1.0 – Feb. 2024**



**VP6-1460 User Guide Supplement Contents**

1 Introduction
1.1 How to Use This Guide
1.2 Important Safety Notice
1.3 Overview of the VP6-1460
1.4 Warranty
1.5 Required License Information

2 Installation
2.1 Installing the configuration software

3 Connect
3.1 Connect to Target

4 Read/Write
4.1 Read Registers
4.2 Write Registers
4.3 Errors

5 Programming Page
5.1 Program Loading and Execution
5.2 Program Editing and Debugging
5.3 Program Capacity
5.4 Program States and Error Codes

6 Diagnostic Log
6.1 Using the Diagnostic Log


# 1 Introduction

## 1.1 How to Use This Guide

Section 2 talks about installing the programming software for the VP6-1460. Section 3 tells you how to connect the programming tool to the VP6-1460. Sections 4 and 6 are for diagnostics. Section 5 gives an overview of programming the VP6-1460 using the i.CanDrawIt graphical programming tool. This type of programming and use of this programming tool are optional.

## 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.**

**CAUTION: The lithium battery contained in this device may explode if mistreated. DO NOT recharge, disassemble, or dispose of in fire.**

**No action is required of the user to activate the battery that backs up the real time clock. Important: Replace battery with BR1225A only. Use of another battery may present a risk of fire or explosion.**

## 1.3 Overview of the VP6-1460

This programming tool is a supplement to the main user guide which walks you through the Web User Interface. All configuration of the VP6-1460 is done through the Web UI. Instructions and additional detail are provided in the main user guide. Please refer to that document as this document is only a supplement.

## 1.4 Warranty

**This configuration 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 fitness 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.

**Warranty:** All Control Solutions products are warranted against defects in materials and workmanship for a period of time from date of shipment from factory as follows: Two years on non-mechanical parts, one year on mechanical parts (e.g. relays). Defective units will be repaired or replaced, at manufacturer's discretion, at no cost to user except when negligence or improper use has resulted in damage. The express warranty stated herein is in lieu of all other warranties, express or implied, including without limitation any warranties of merchantability or fitness for a particular purpose and all other warranties are hereby disclaimed and excluded by Control Solutions, Inc.

Configuration errors made by customer are not covered under warranty. Damage caused by incorrect electrical connection is not covered under warranty. Removing circuit boards from their enclosures will void the warranty - the complete product with all of its original circuit boards and components must be returned for warranty consideration.

## 1.5 Required License Information

The VP6-1460 programming tools include the SmartWin library (http://smartwinlib.org) under the following terms:

License agreement for SmartWin++ (BSD license)

Copyright (c) 2005, Thomas Hansen All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the SmartWin++ nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (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

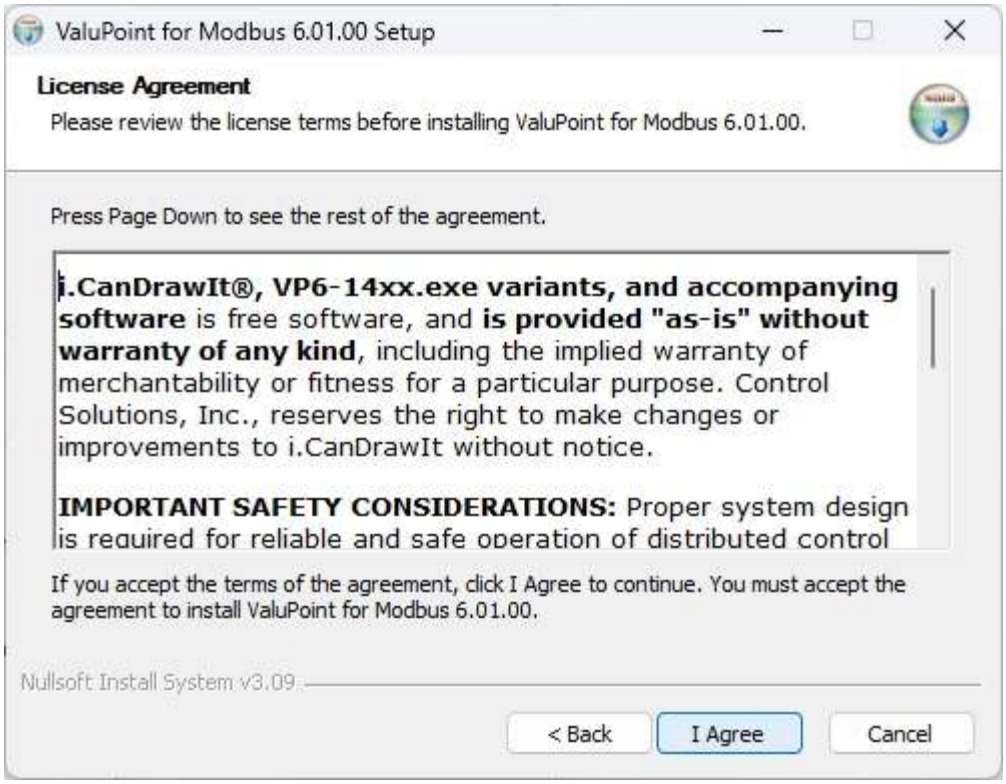
## 2.1 Installing the software

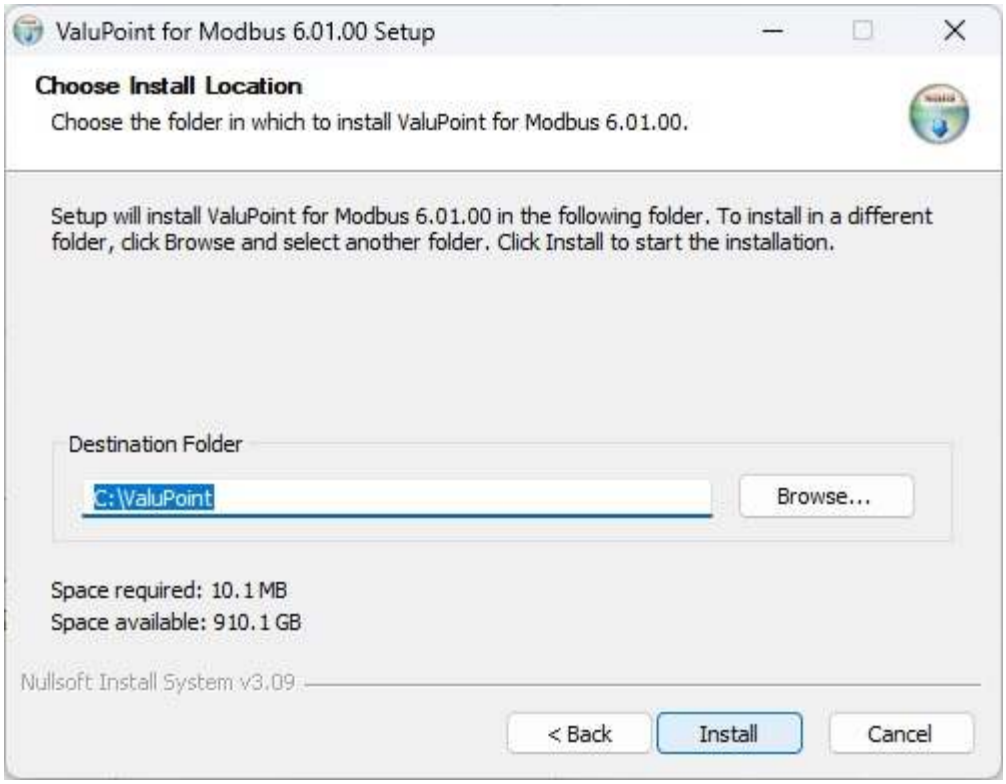
Look for the installer icons in the directory where you unzipped the download that got you to this document. The installer icons look like this:

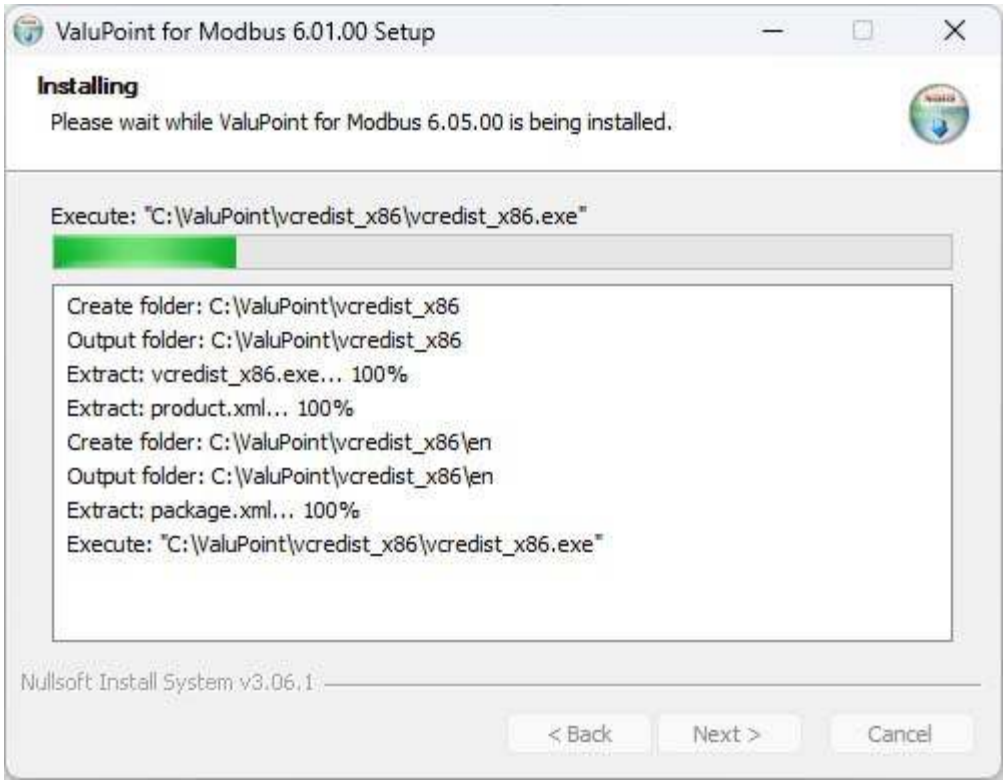

The installation is a 2-step installation. Install VP6-1460 first. Then install i.CanDrawIt second.

Double click the icon to run the VP6-1460_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 11:

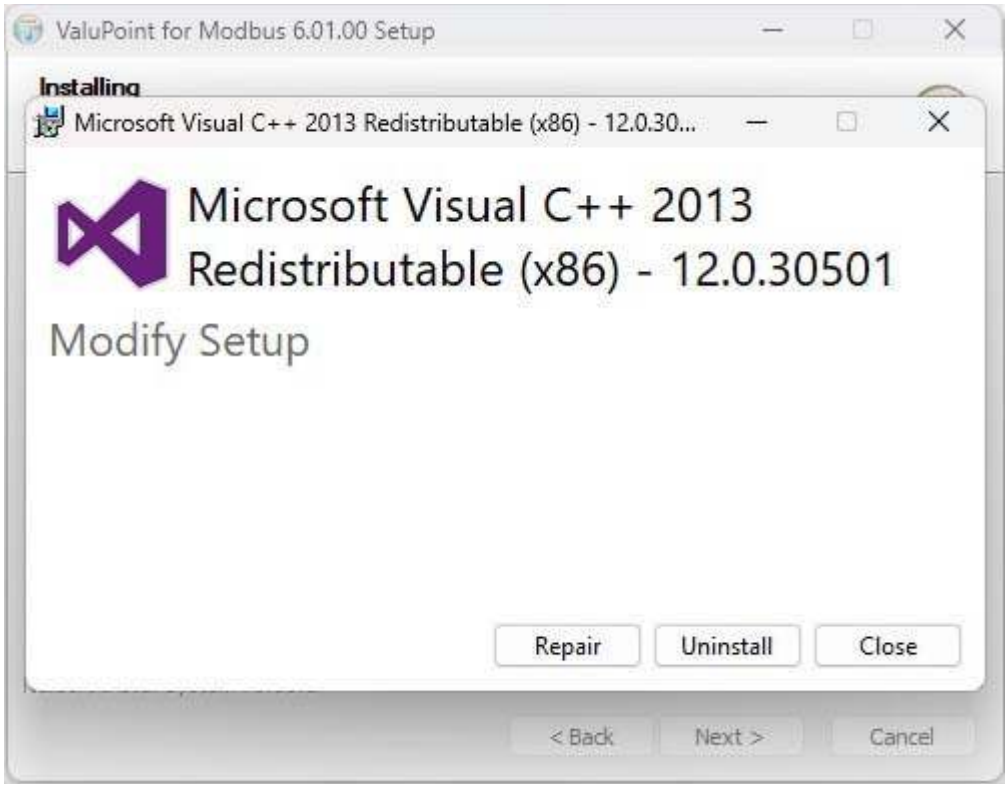

ValuPoint for Modbus 6.01.00 Setup
Welcome to ValuPoint for Modbus 6.01.00 Setup
Setup will guide you through the installation of ValuPoint for Modbus 6.01.00.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel


ValuPoint for Modbus 6.01.00 Setup
License Agreement
Please review the license terms before installing ValuPoint for Modbus 6.01.00.
Press Page Down to see the rest of the agreement.
i.CanDrawIt®, VP6-14xx.exe variants, 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 for Modbus 6.01.00.
Nullsoft Install System v3.09
< Back
I Agree
Cancel

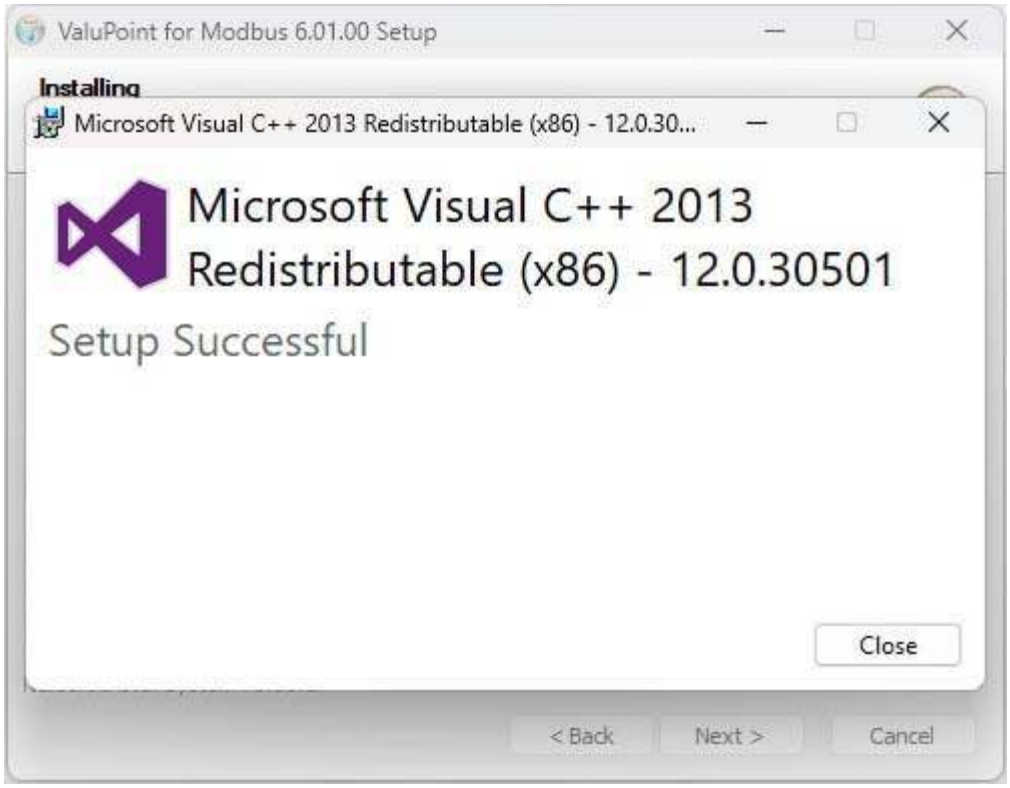


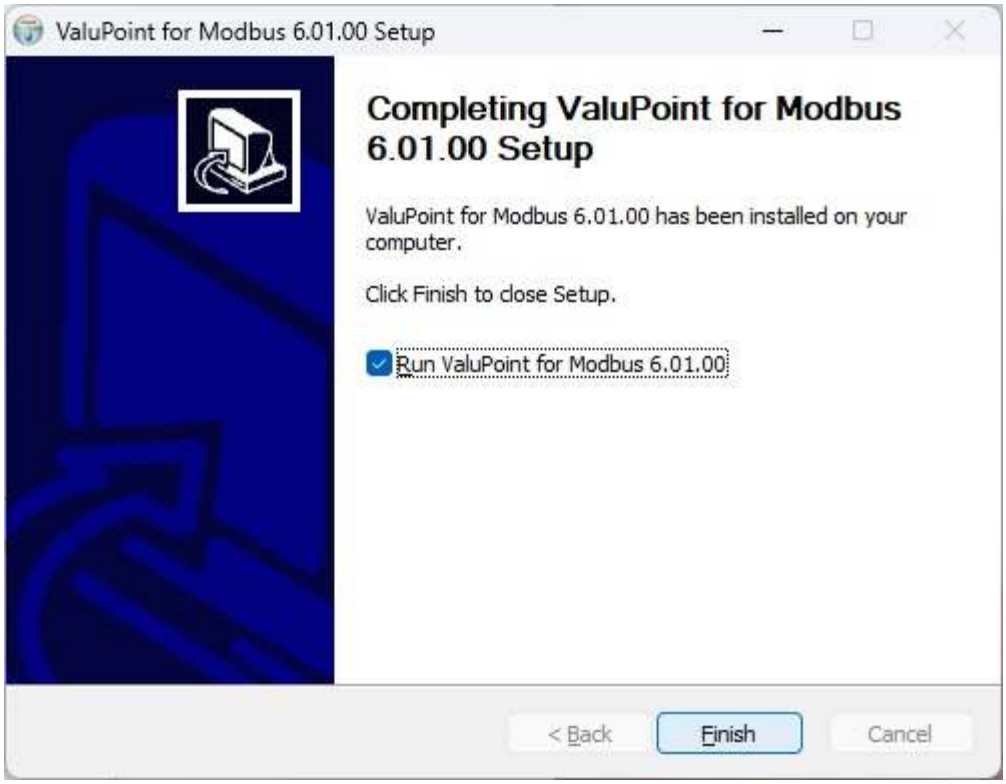

The installer will check to see whether Visual C++ support is already installed on your system, and install it if not. This is standard software provided by Microsoft. If it is already there, it will give you the option to "Repair". Select Repair (which in most cases will not really do anything other than verify the installation).



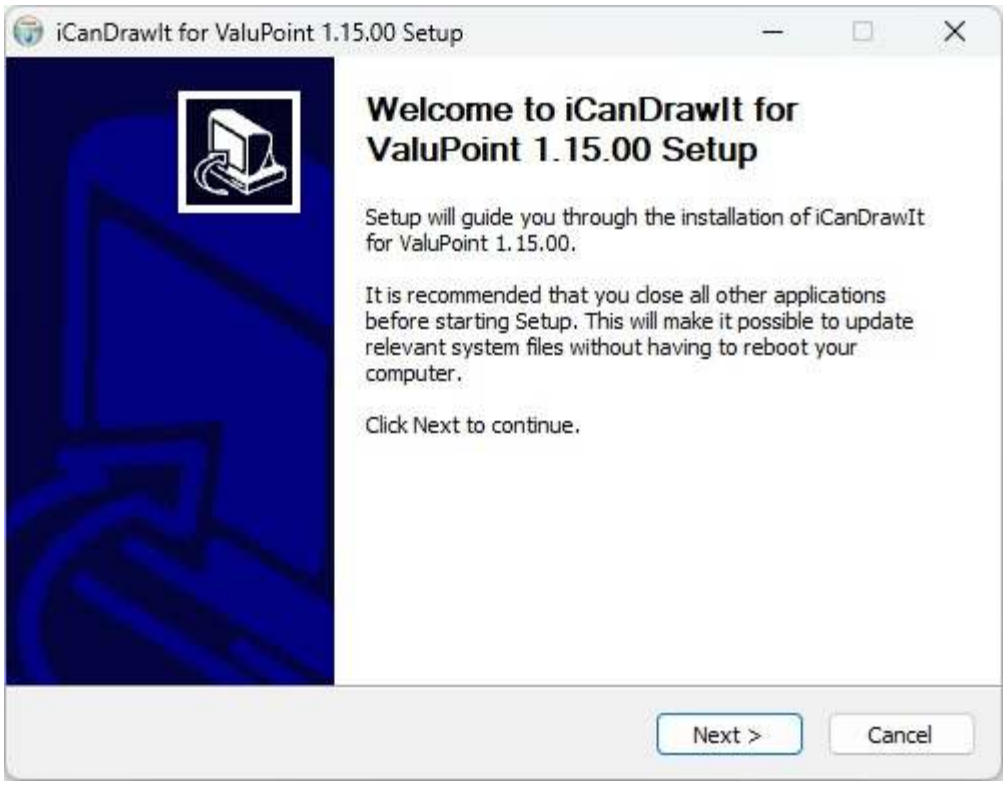

Click Close to continue the installation.

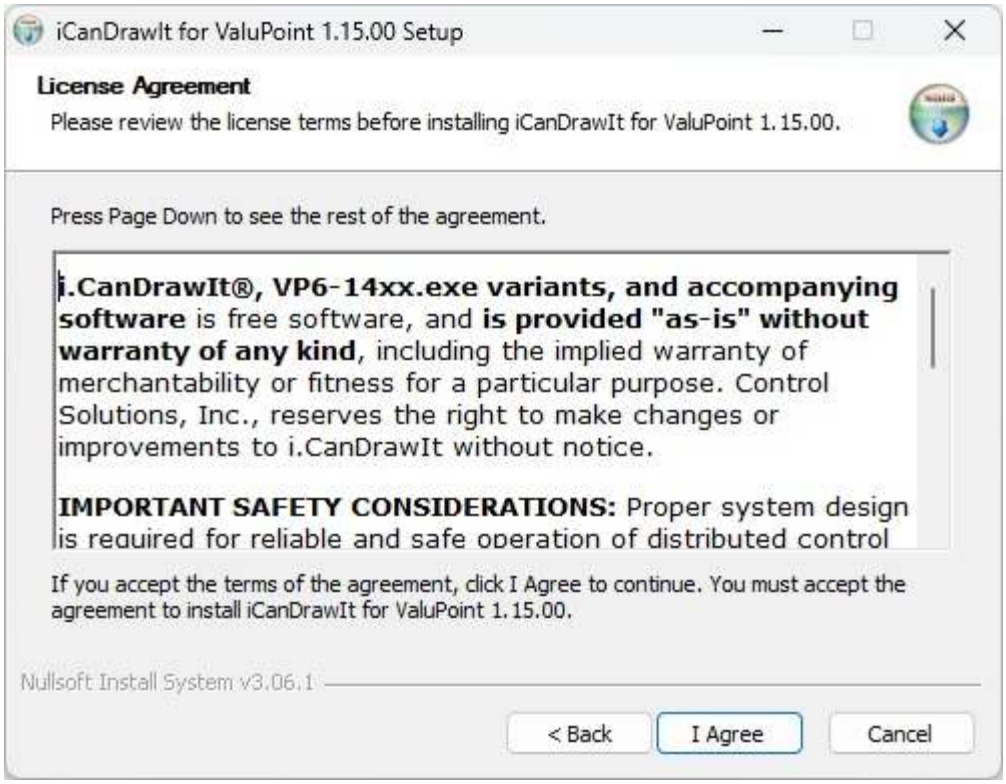



When you get to the "Finish" screen, you are ready to go.

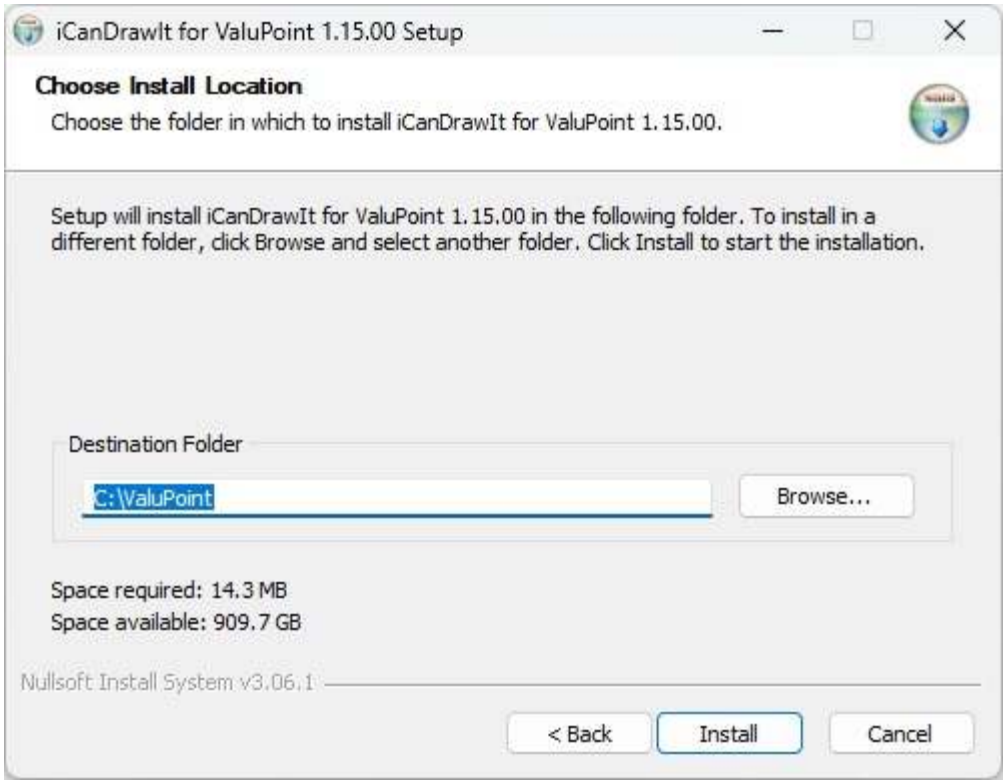



Next, proceed to install i.CanDrawIt. The first installer screens look like this:

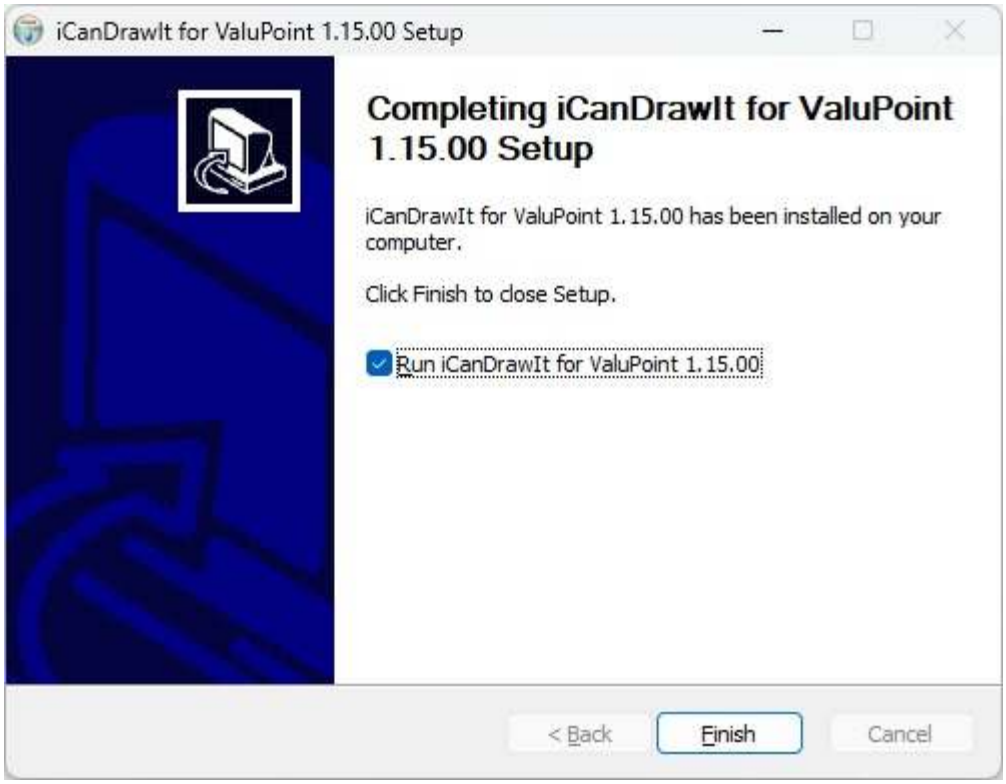





The installation directory should be the same directory that VP6-1410 was installed into.




After a few more screens that look much like the VP6-1410 screens above, you will get the familiar 'done' screen.

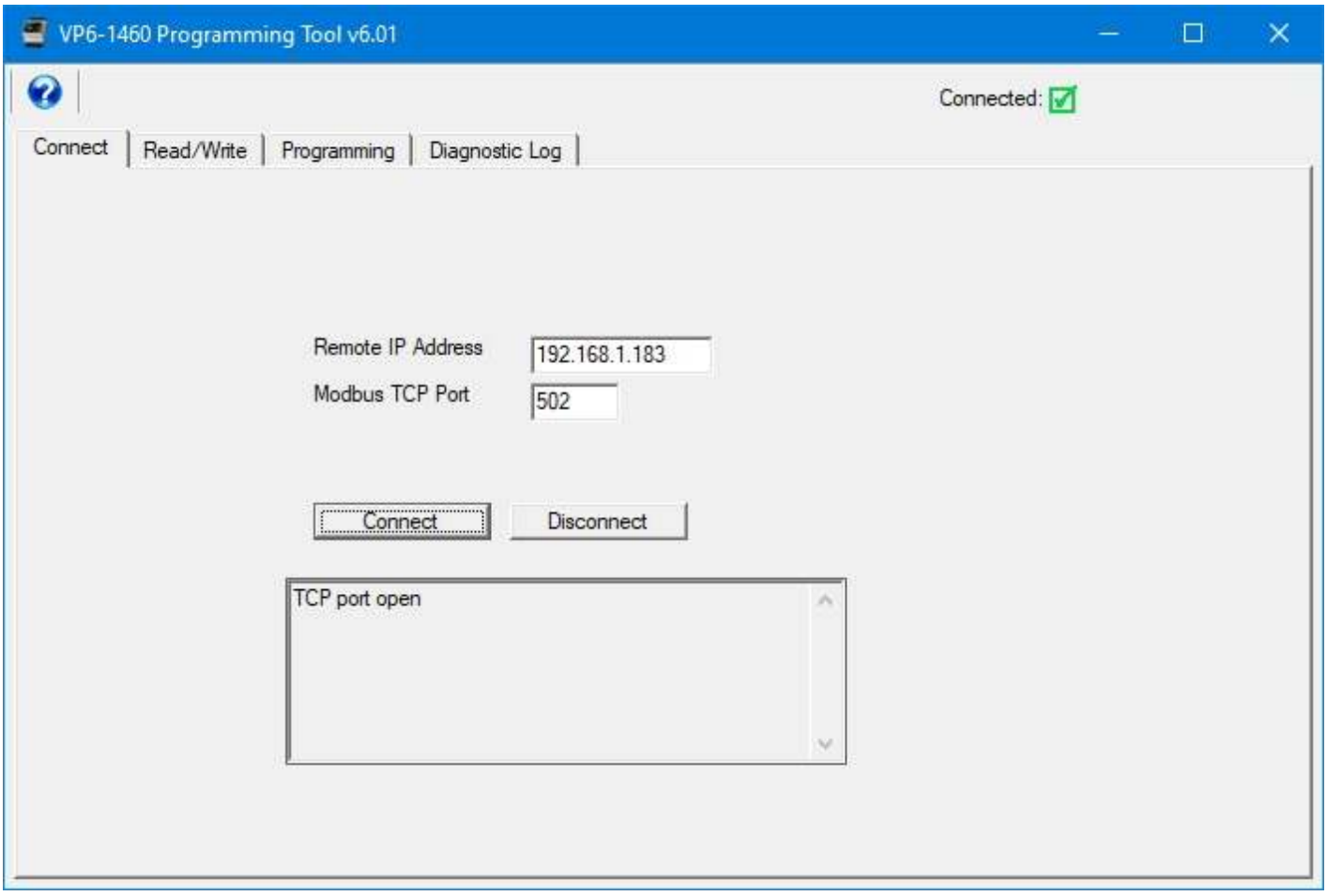

iCanDrawIt for ValuPoint 1.15.00 Setup
Completing iCanDrawIt for ValuPoint 1.15.00 Setup
iCanDrawIt for ValuPoint 1.15.00 has been installed on your computer.
Click Finish to close Setup.
Run iCanDrawIt for ValuPoint 1.15.00
< Back
Finish
Cancel


# 3 Connect

The primary purpose of this programming tool is to create a bridge between your VP6-1460 and the i.CanDrawIt graphical programming tool. The i.CanDrawIt tool cannot talk directly to the VP6-1460 by itself. This is because the same i.CanDrawIt tool is used for both BACnet and Modbus, and for both IP and RS485 versions of devices. This programming tool provides translation between i.CanDrawIt messages and Modbus TCP (in this case).

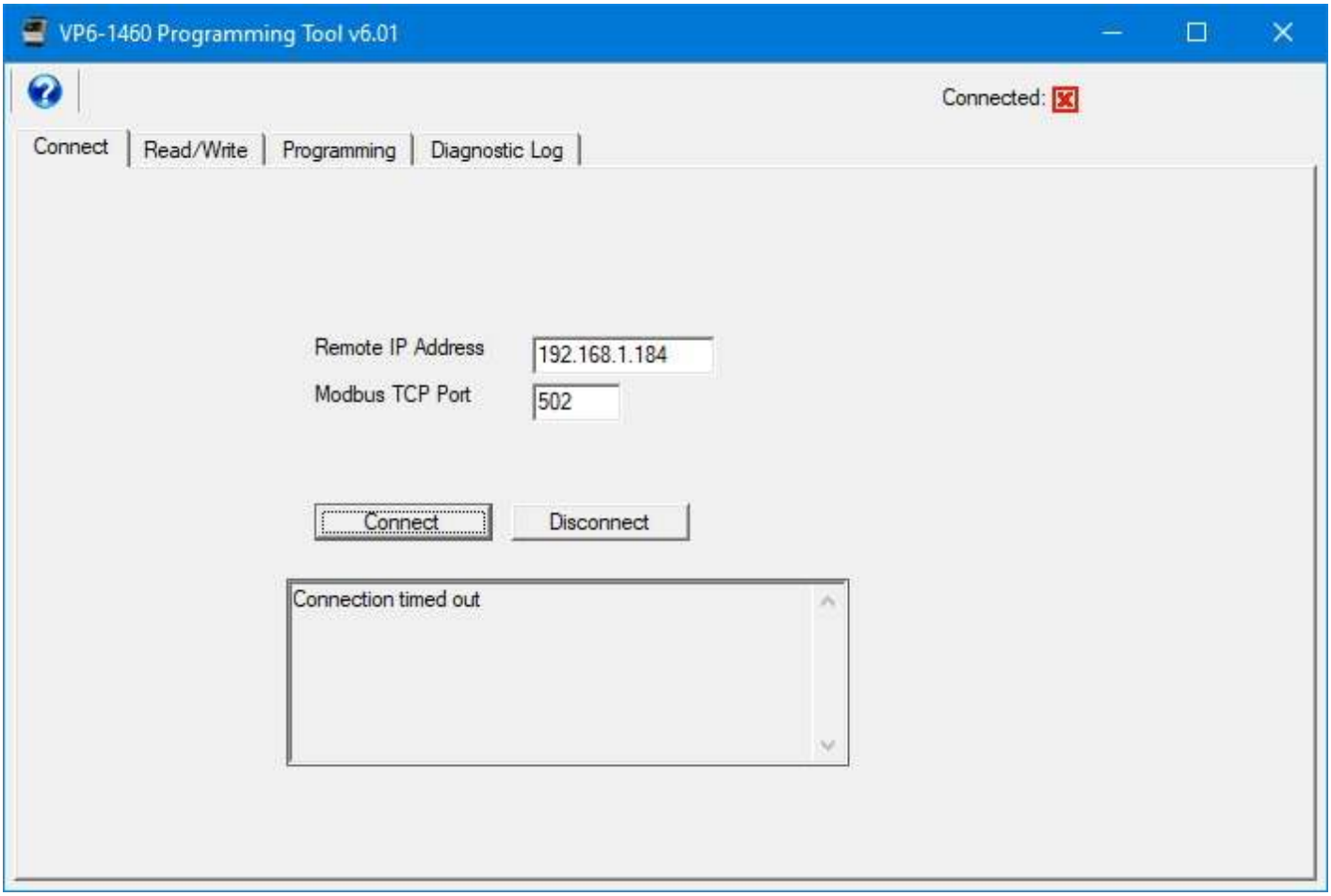
## 3.1 Connect to Target

This tool connects to the VP6-1460 via Modbus TCP. Simply enter the IP address of your VP6-1460, and set the port if it was changed from the standard Modbus 502 port. Click Connect.




You should almost immediately get the "TCP port open" message if the connection was

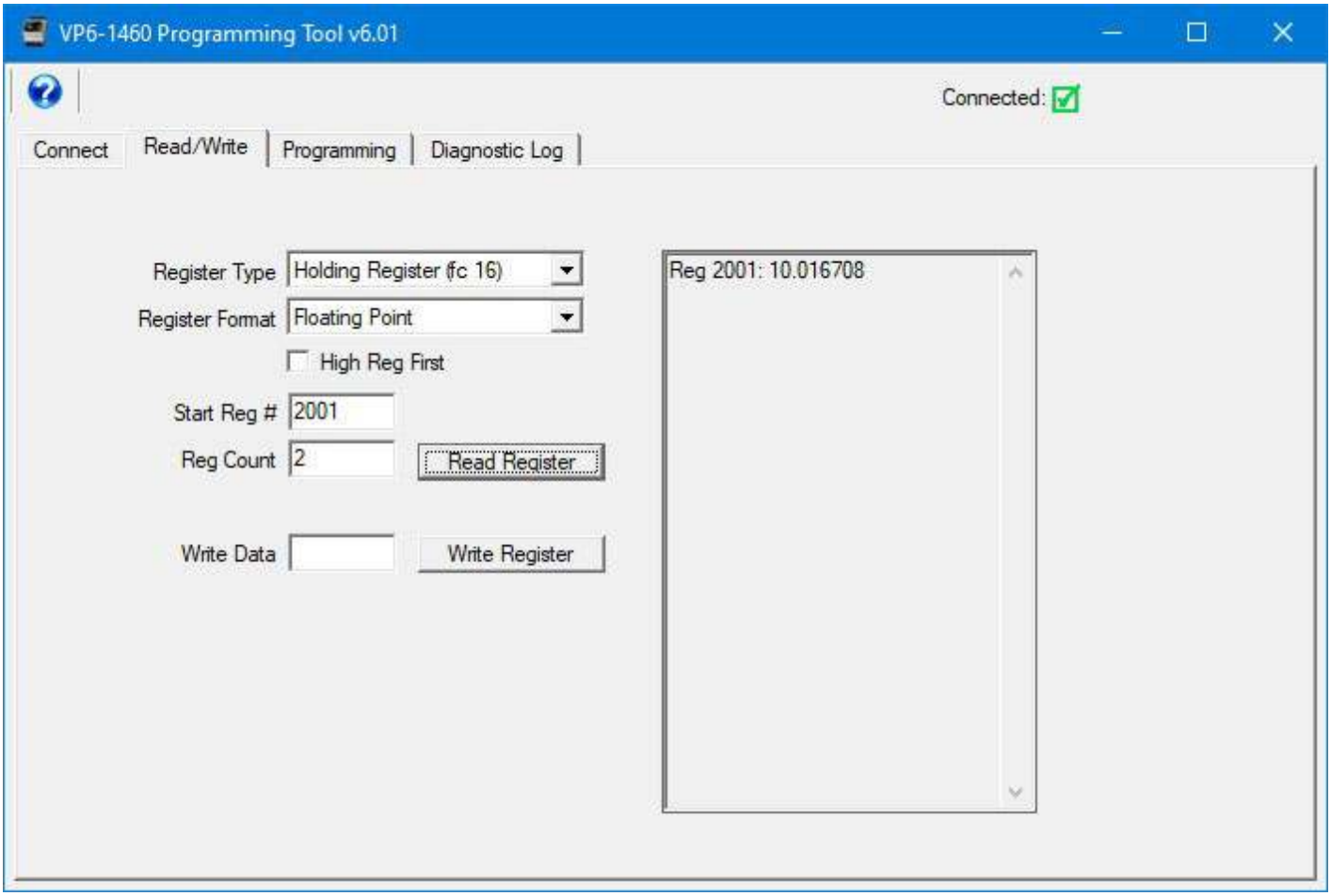
successful. If the given IP address is not accessible, you will get a timeout message as illustrated below.



# 4 Read/Write

This page is primarily for diagnostic use. If you are questioning whether your VP6-1460 is responding to Modbus TCP, this page will answer that question. Simply read, or optionally write, a Modbus register in the VP6-1460 from this page.

## 4.1 Read Registers



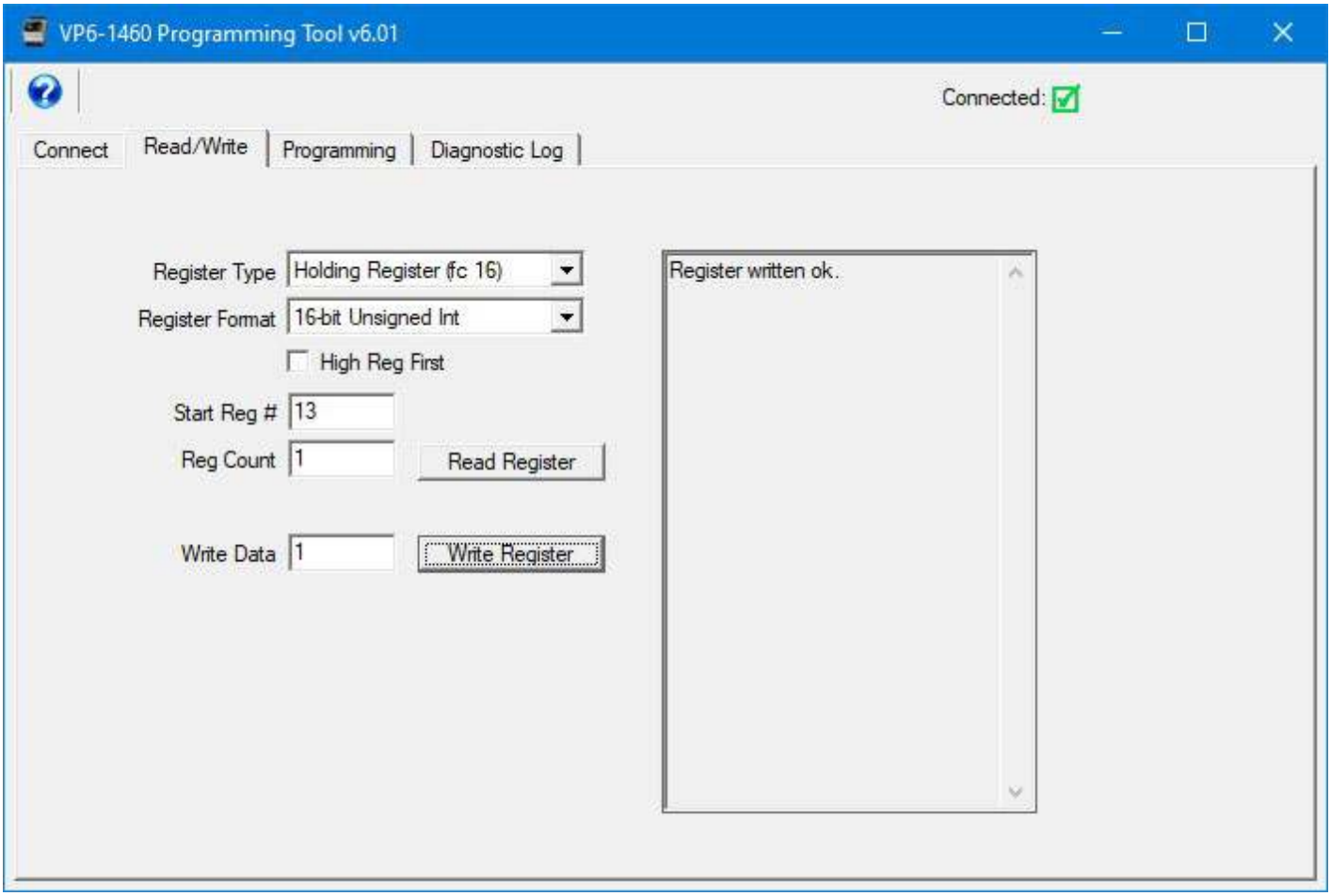

Select 'Register Type'. This determines the Modbus function code that will be sent to the Modbus device. Selecting 'Register Format' only tells the configuration tool how to interpret the data and format for you to read. The format has nothing to do with construction the query sent to the device. If 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 device. Click 'Read Register' to cause the query to be created and transmitted over the TCP 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 device connected to the same network that your PC is connected to.

## 4.2 Write Registers




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

If communication is successful, but the Modbus server 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 server 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 server. 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 server or is not allowed by the server. |
| 2 | Illegal Data Address | The data address (register number) received in the query is not an allowed address for the server, 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 server. |


# 5 Programming Page

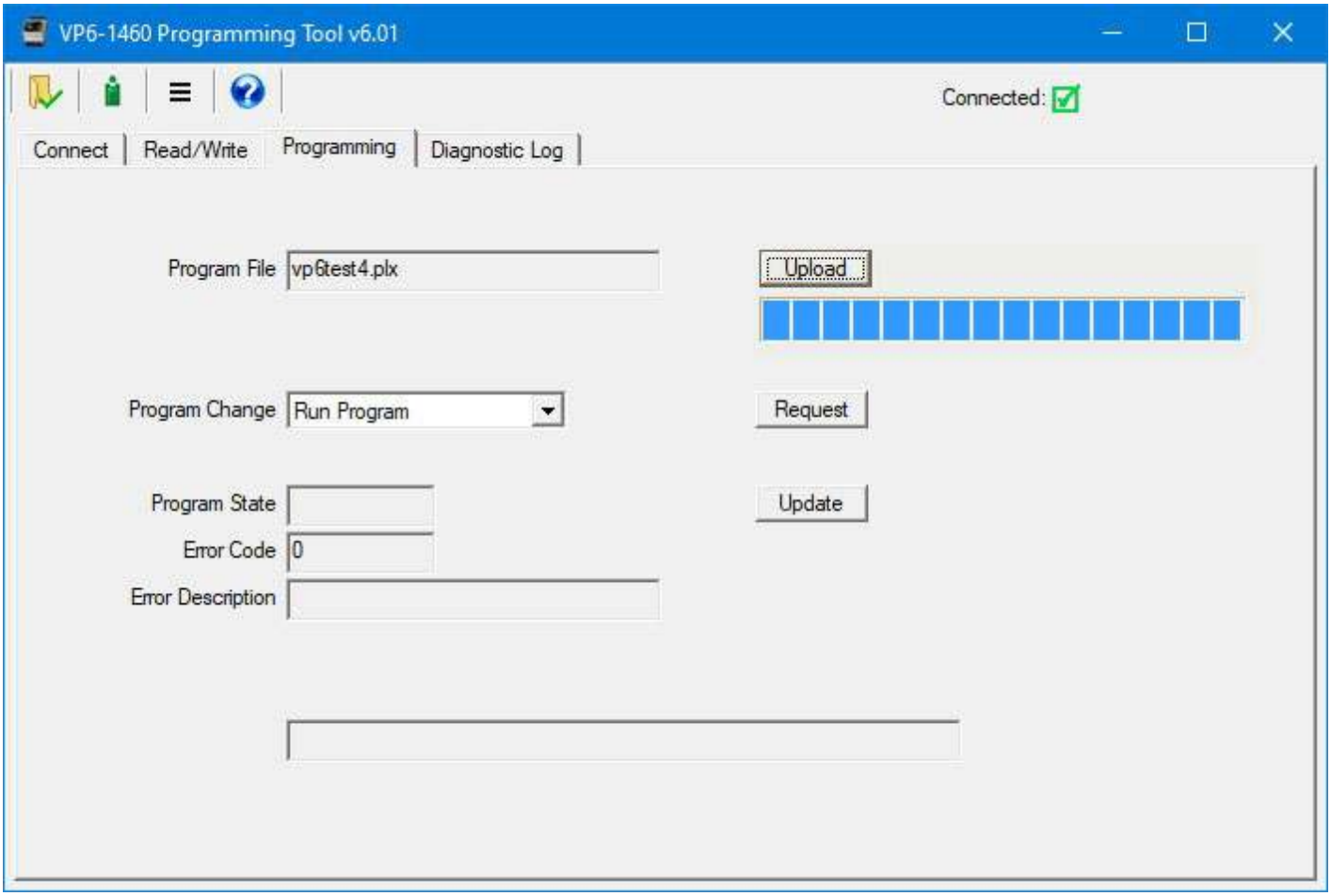
This Programming Page refers to this page in this PC based tool for using the i.CanDrawIt graphical programming tool. The web UI in the VP6-1460 contains additional pages under the heading "Programming". The programming environment found in the web UI is for Script Basic and runs independently of the i.CanDrawIt program environment. There are two independent programming environments in the VP6-1460.

You do have the ability to use both Script Basic and i.CanDrawIt at the same time. If you do this, just be aware that neither program has the ability to block the other from writing to registers. It will be up to you to make sure you avoid conflicts.

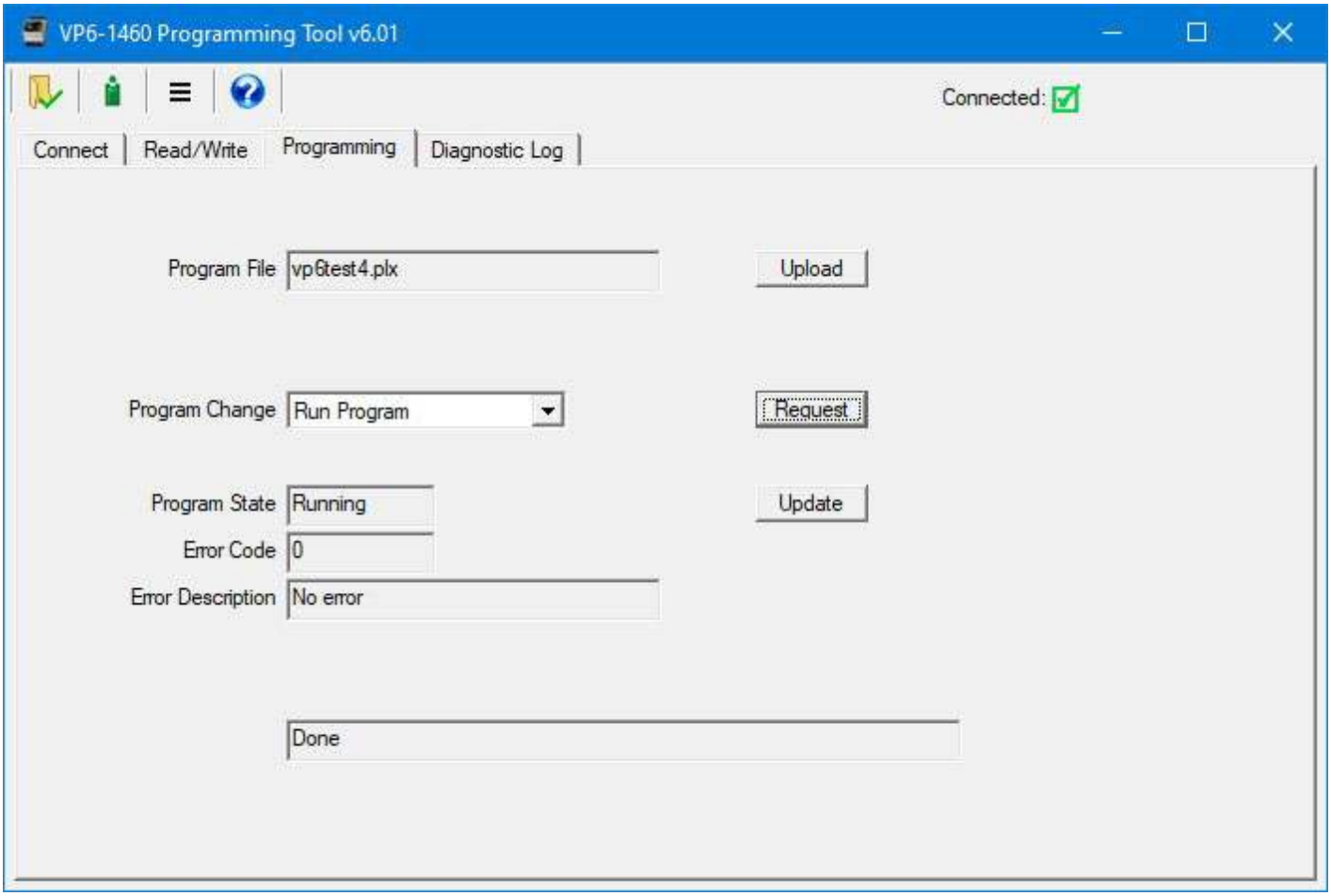
## 5.1 Program Loading and Execution

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.

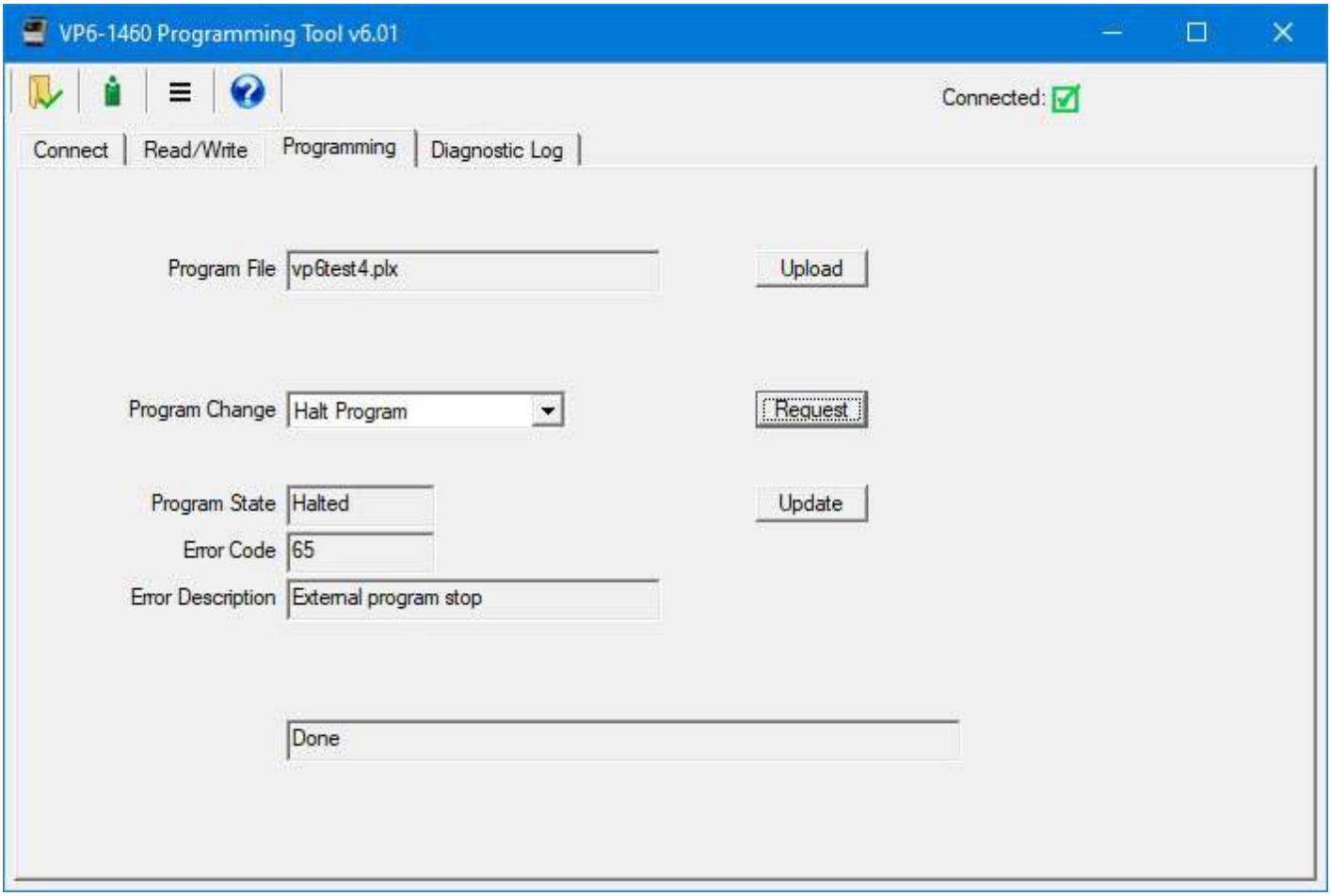
After a program (.plx file) has been opened, click the Upload button to send that program to the ValuPoint. A progress bar will indicate program loading progress.




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

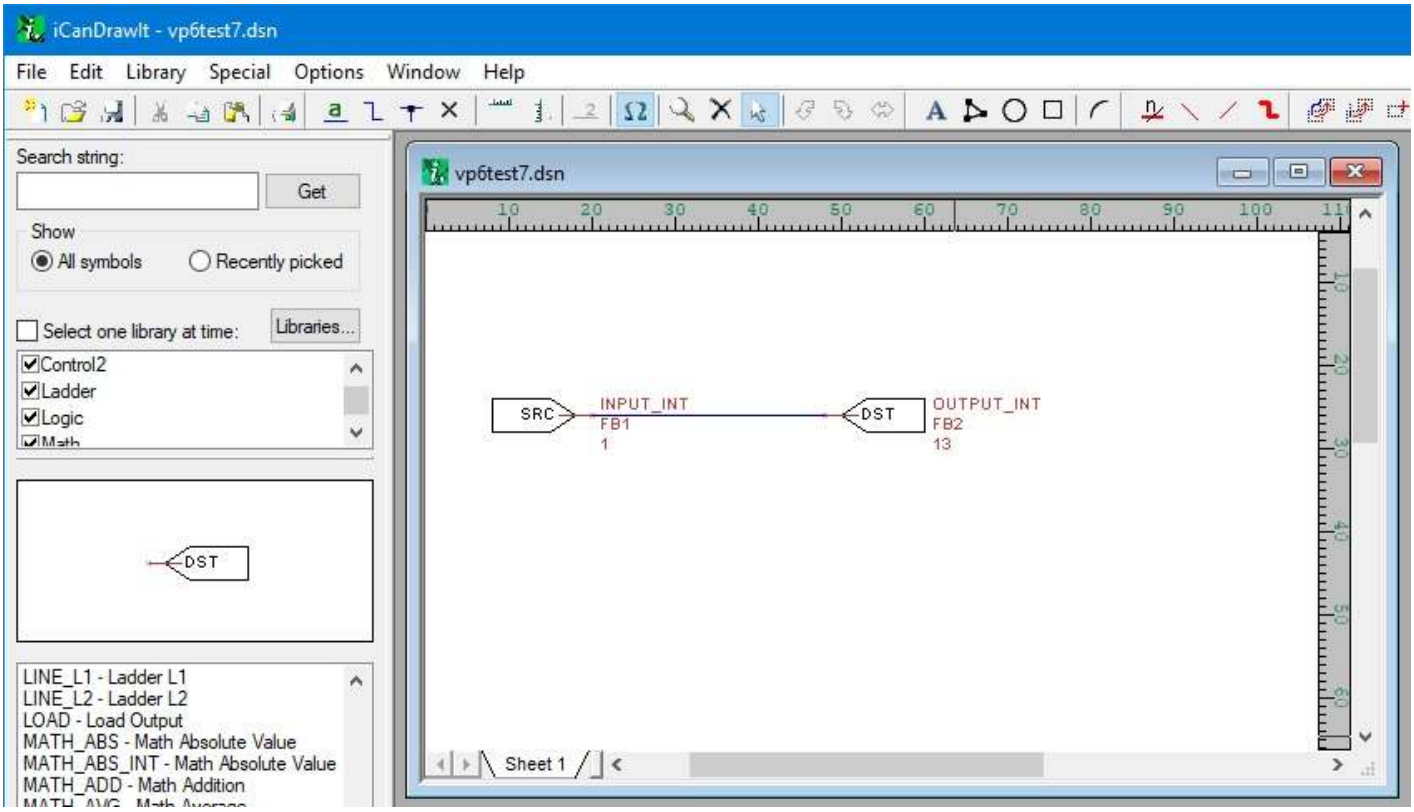



To stop the program, select Halt Program and click Request.




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.

## 5.2 Program Editing and Debugging



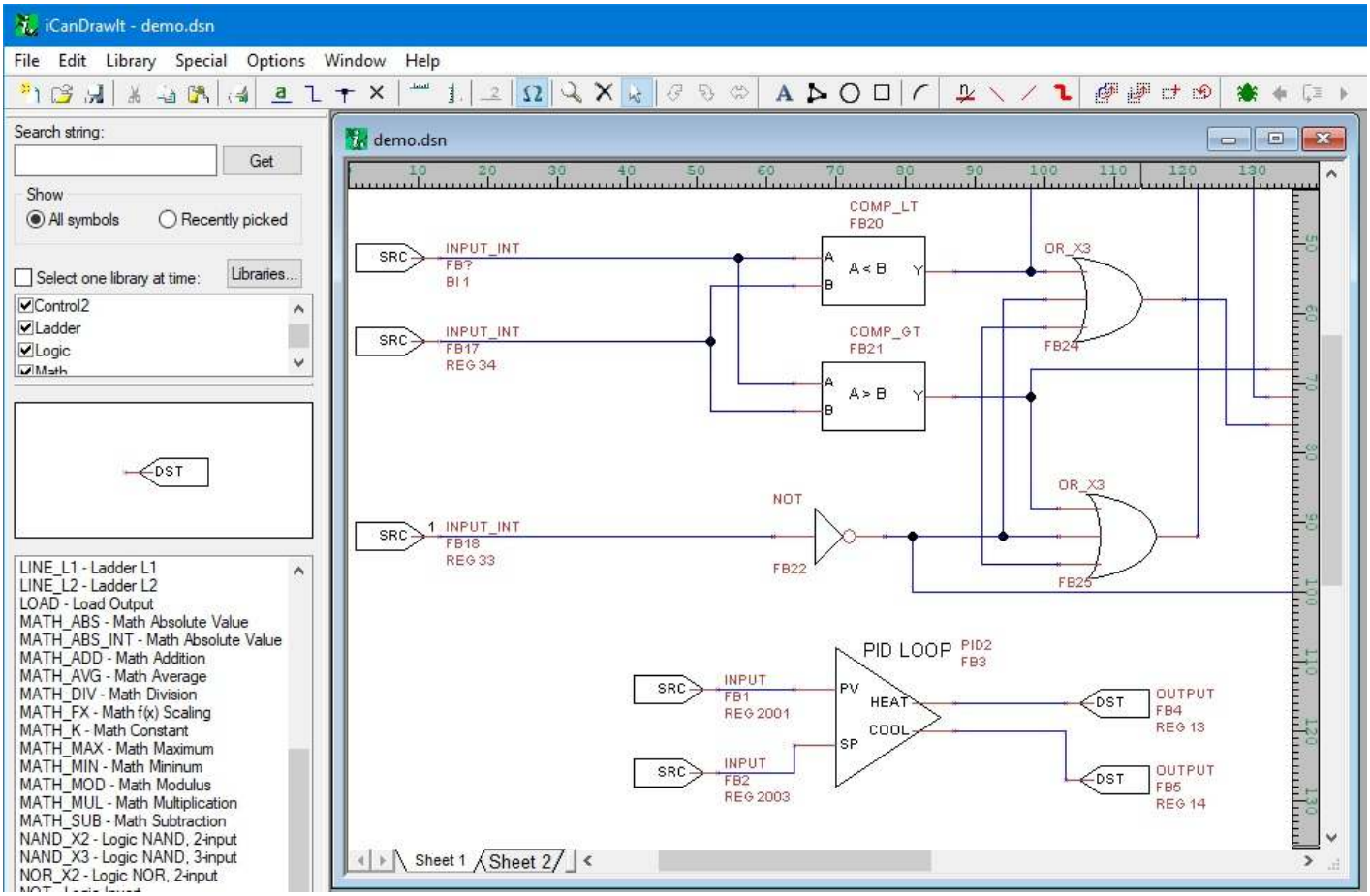

Click the green "i" icon next to the folder icon to open the i.CanDrawIt graphical programming tool that is illustrated below. It has its own set of help pages. Click on the "?" icon in that tool for more about programming. (Note that i.CanDrawIt is a second software package installed after the ValuPoint 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.

The above simple example program shows operating the relay output DO #1 (object 13) from input A/UI #1. To configure A/UI #1 for use in the above example, go to the I/O Configu page in the web UI and configure A/UI #1 for dry contact (or other applicable discrete type).

A more complex example is illustrated here:




## 5.3 Program Capacity

Maximum compiled program (.plx file) size: 65,280 bytes
Maximum RAM available for program variable and stack space: 16,384 bytes
Maximum EEPROM available: 256 bytes

## 5.4 Program States and Error Codes

Program State codes:

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

Reason for Halt codes (Indicated as Error Code):

- 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

## 5.5 Register Access

The i.CanDrawIt program has no way of looking at your Modbus register configuration in the server. Therefore, some restrictions are imposed so that configuration can be implied. All calls to geti or seti must reference a 32-bit integer Modbus register. All calls to getf or setf must reference a 32-bit floating point Modbus register. High order data must be in first register. Also note that only register numbers up to 16383 are accessible to i.CanDrawIt as Modbus registers created in the server.

The following special registers are available for access to the battery backed real time clock/calendar from the i.CanDrawIt program. Registers 128001-128007 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 128001-128007 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 128011-128017, and be sure to read 128011 first. Reading register 128011 (year) locks the rest of the time stamp and the remaining registers will return whatever the time/date was when register 128011 was read.

To set the clock/calendar, write all of registers 128011-128017, then write any value to register 128018 to trigger the write. Nothing is done with the content of register 128018 - it is only the trigger to tell ValuPoint to store the content of registers 128011-128017 into the clock/calendar hardware.

| Holding Reg. No. | Writeable | Description |
|---|---|---|
| 128001 | No | Year |
| 128002 | No | Month |
| 128003 | No | Day of Month |

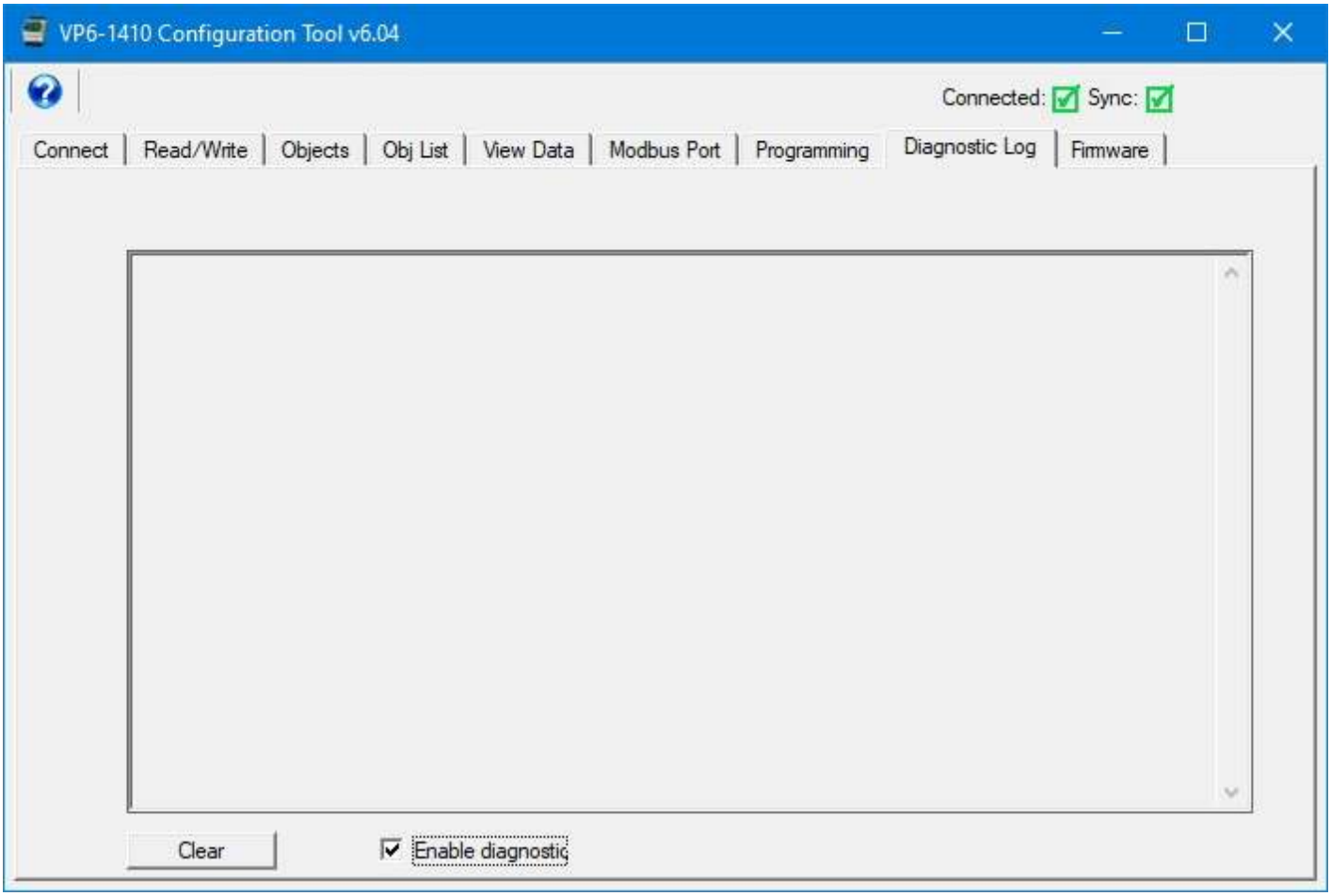
| | | |
|---|---|---|
| 128004 | No | Hour (0..23) |
| 128005 | No | Minute |
| 128006 | No | Second |
| 128007 | No | Day of Week (1=Monday, 2=Tuesday, etc) |
| 128011 | Yes | Year (is also lock trigger for read) |
| 128012 | Yes | Month |
| 128013 | Yes | Day of Month |
| 128014 | Yes | Hour (0..23) |
| 128015 | Yes | Minute |
| 128016 | Yes | Second |
| 128017 | Yes | Day of Week (1=Monday, 2=Tuesday, etc) |
| 128018 | Yes | Lock trigger for write |
| 128021 | No | Minutes since midnight |
| 128022 | No | Day of year (Jan 1 = 1, Dec. 31 = 366 if leap year, else 365) |

# 6 Diagnostic Log

## 6.1 Using the Diagnostic Log

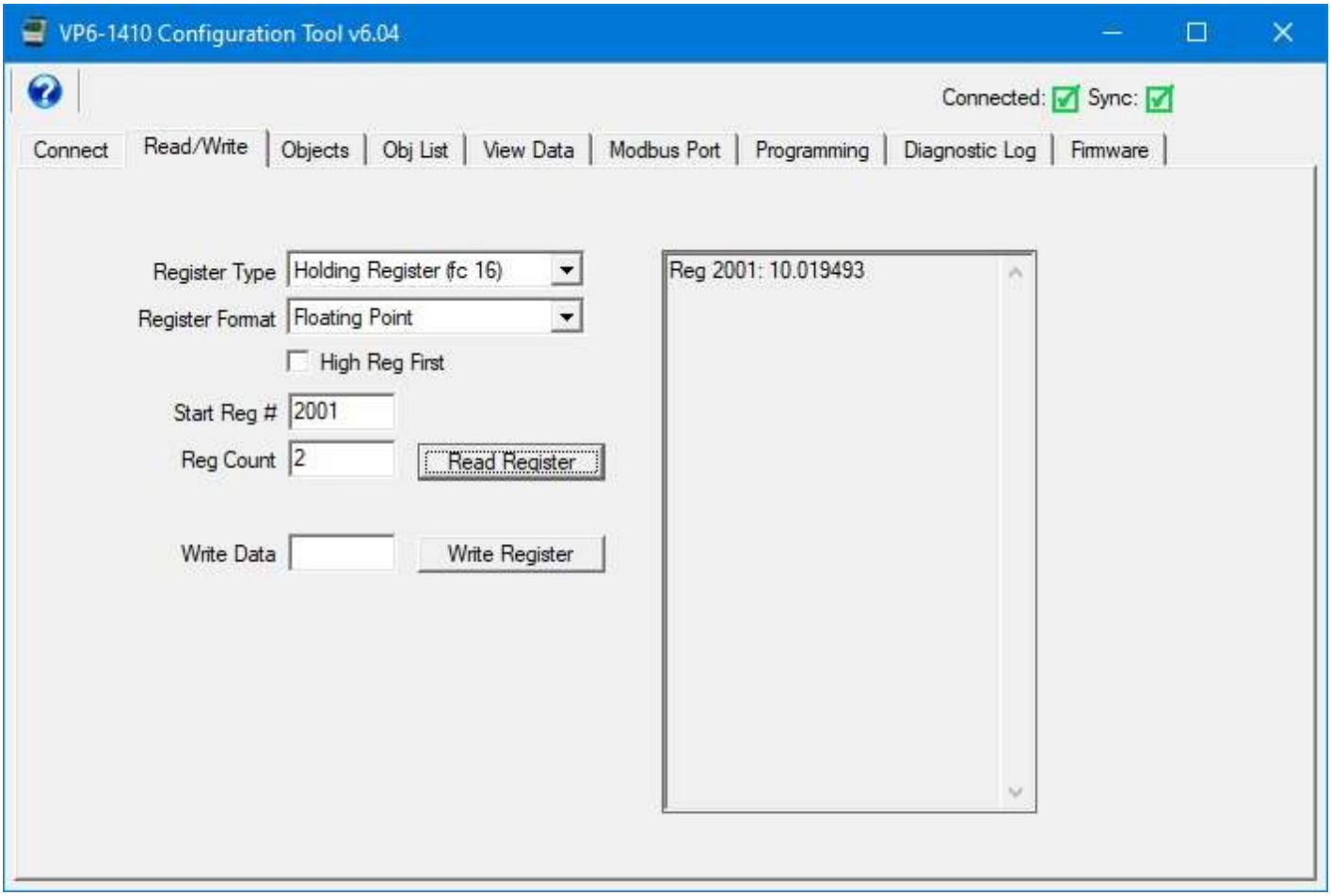
The diagnostic log is useful for verifying exactly what is going over the wire between the configuration tool and the device. The raw content of all Modbus packets sent to the ValuPoint device by the tool are displayed on a line labeled "Tx". The raw content of all Modbus packets received from the ValuPoint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

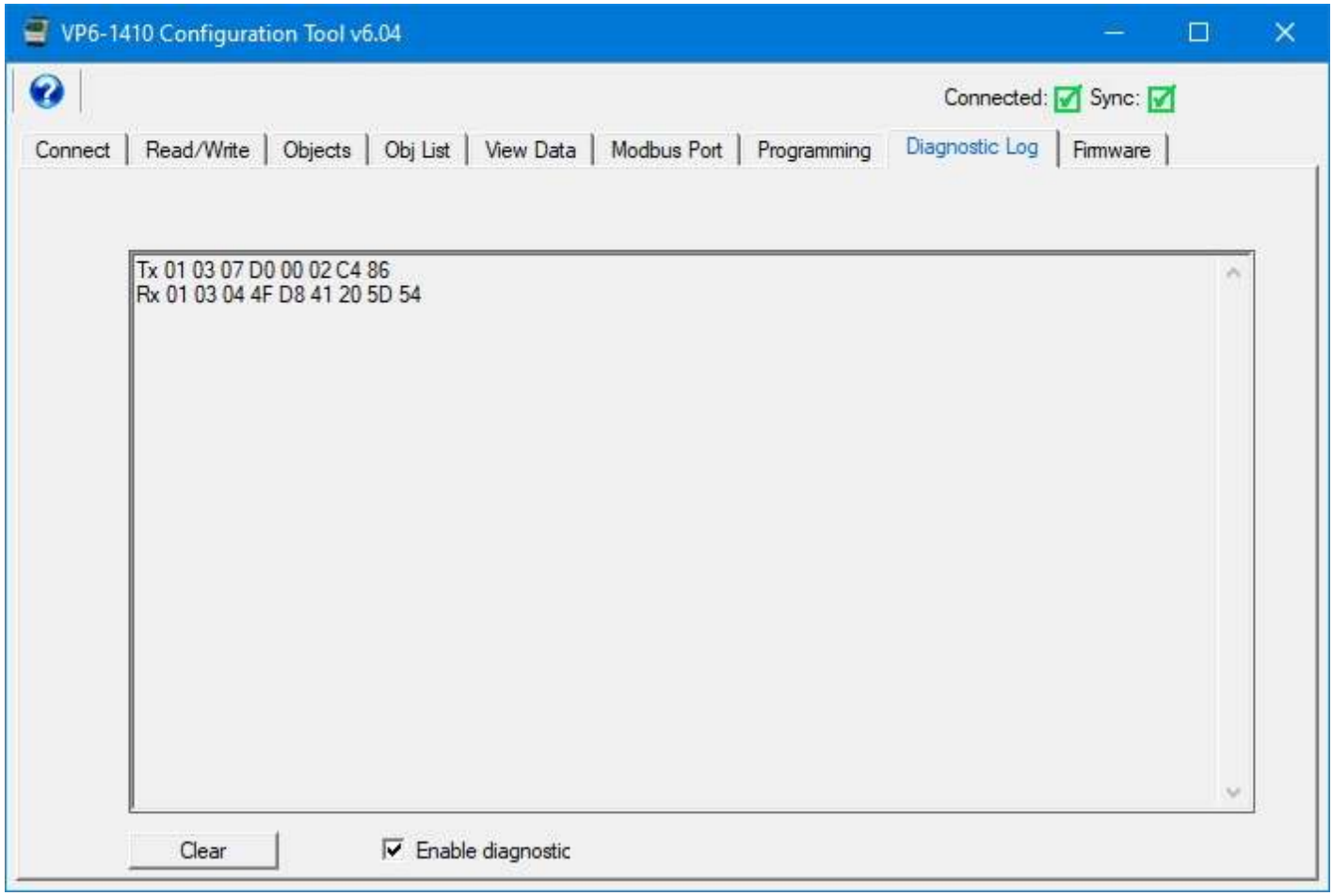



Note that the Read/Write page can be used to query any Modbus device physically connected to the same network as this tool. This means the ValuPoint tool can be a useful diagnostic tool for testing other Modbus devices. If you are using this tool to

query a device other than a ValuPoint, all communication with that device will also show up here on the Diagnostic Log.




The above register read operation produced the following 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.