Introduction:

ModbusTest is an invaluable software testing tool that you may use to validate communications between your PC and a Watlow EZ-ZONE® family controller.

In the basic form, you will use this tool to either read or write to Modbus registers with in a controller equipped with the Modbus RTU protocol.

Each Modbus register in the controller contains 16-bits of information called one word. All 16-bit words are stored in unsigned integer format. Any information that pertains to precision data is stored in two consecutive Modbus registers such as 360 and 361 which together form a location to store a closed loop set point. Since each register is 16-bits wide, two registers will form 32-bits of information using two words.

Some information is encoded in 32-bit IEEE-754 floating point format. In this way, the controller can display process or temperatures to a precision of 0.001 units or degrees. Ever piece of information in the controller is referenced by a Modbus register or address location starting at an even number. You will never write to an odd numbered Modbus address (using relative addressing, see more detail below).

Protocol Background Information:

Gould Modicon, now called AEG Schneider, created the protocol referred to as "Modbus" and used it in process control systems. Modbus provides the advantage of being extremely reliable in exchanging information, a highly desirable feature for industrial data communications. This protocol works on the principle of packet exchanges. The packet contains the address of the controller to receive the information, a command field that says what is to be done with the information, and several fields of data. Reading from these registers retrieves all information in the controller. Each of these registers' address is listed in the user's manual (Operations, Setup, Profiling, & Factory Pages). You will need this list to determine where the data is located. The last item sent in the packet is a field to ensure the data is received intact. This is called a cyclical redundancy check-sum (CRC). This program creates the required packet for you and sends it to the controller. A packet response from the controller is automatically decoded and checked for accuracy.

Many parameter values within the controller are four bytes in length and require two Modbus registers. By default, the low register address contains the two lower bytes and the high register address contains the two higher bytes. If it makes your programming easier you may reverse this Modbus default where the low register address contains the two higher bytes and the high register address contains the two lower bytes. This program will let you select the word order to match the controllers' setting.

Modbus RTU is typically deployed over serial connections where Modbus TCP is deployed over the Ethernet physical layer. If it is desired to acquire more information on Modbus RTU or Modbus TCP protocols refer to http://www.modbus.org for the detailed specification.

Lastly, Modbus register addressing is sometimes referenced by a relative address or an absolute address. This program and the user's manual use the relative addressing concept. If your program or PLC uses absolute addressing, either add 40,001 or 400,001 to the listed relative address. Your PLC must use the 400,001 method to access register addresses higher than absolute address 49,998. The maximum relative address that may be specified is 65,565 or absolute address 465,566. Although this may be confusing, the details are not important to make this work. Mainly, you must know how to enter the correct register address for the program or PLC being used. Again, this program uses relative addressing which we list in the user's manual for every parameter.

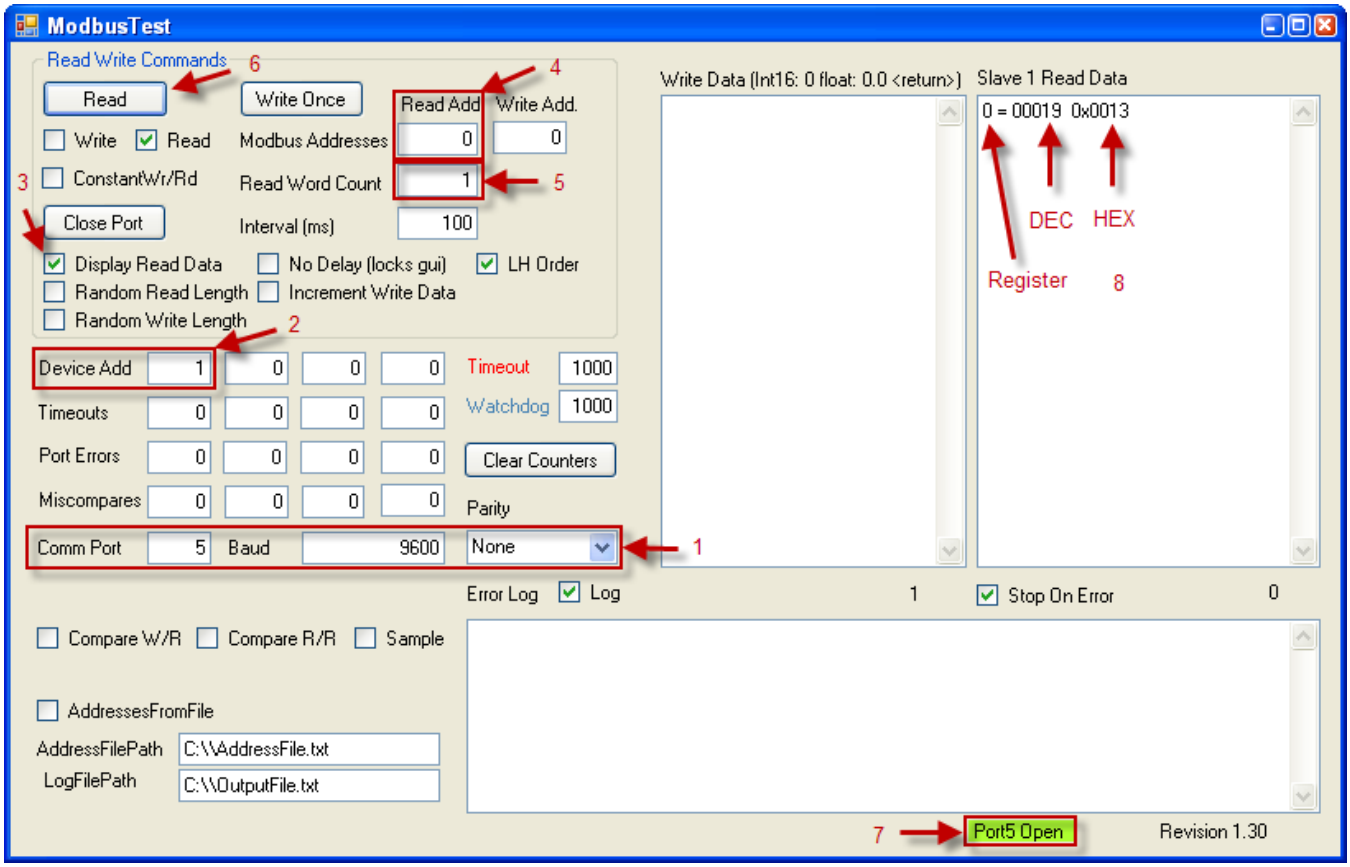
Watlow Winona
1241 Bundy Blvd
Winona, MN 55987
Telephone (507) 494-5656

To see a white paper called Sample EZ-ZONE PM 32-bit Modbus packet refer to
http://www.watlow.com/literature/prodtechinfo/files/controllers/sample%20ez-zone%20pm%2032-bit%20modbus%20packet.pdf

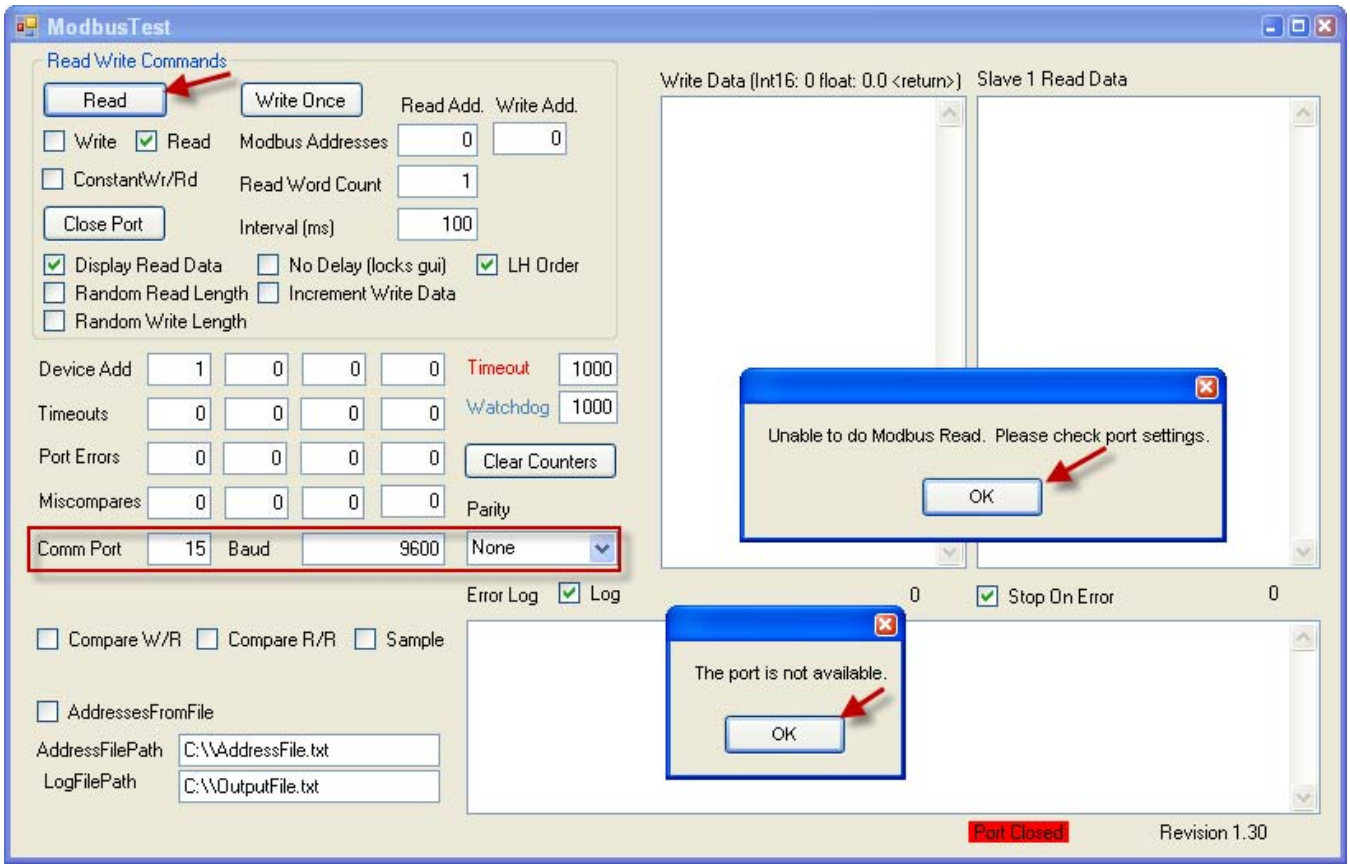
Using the program:

Example 1: To read a 16-bit value (1 register or word).

1. Enter Comm Port address of PC, Baud and Parity of controller's settings
2. Enter 'Device Add' which is controller address on network (1 to 247)
3. Check 'Display Read Data'
4. Enter Modbus register to read in controller using relative addressing. The user's manual show relative addresses for all parameters. Register 0 is the hardware ID of EZ-ZONE® products.
5. Enter 'Read Word Count' (1 = 16 bits)
6. Click 'Read' button
7. If valid port was entered, display will show 'Portx Open'
8. Value read from register is displayed in decimal and hexadecimal formats.

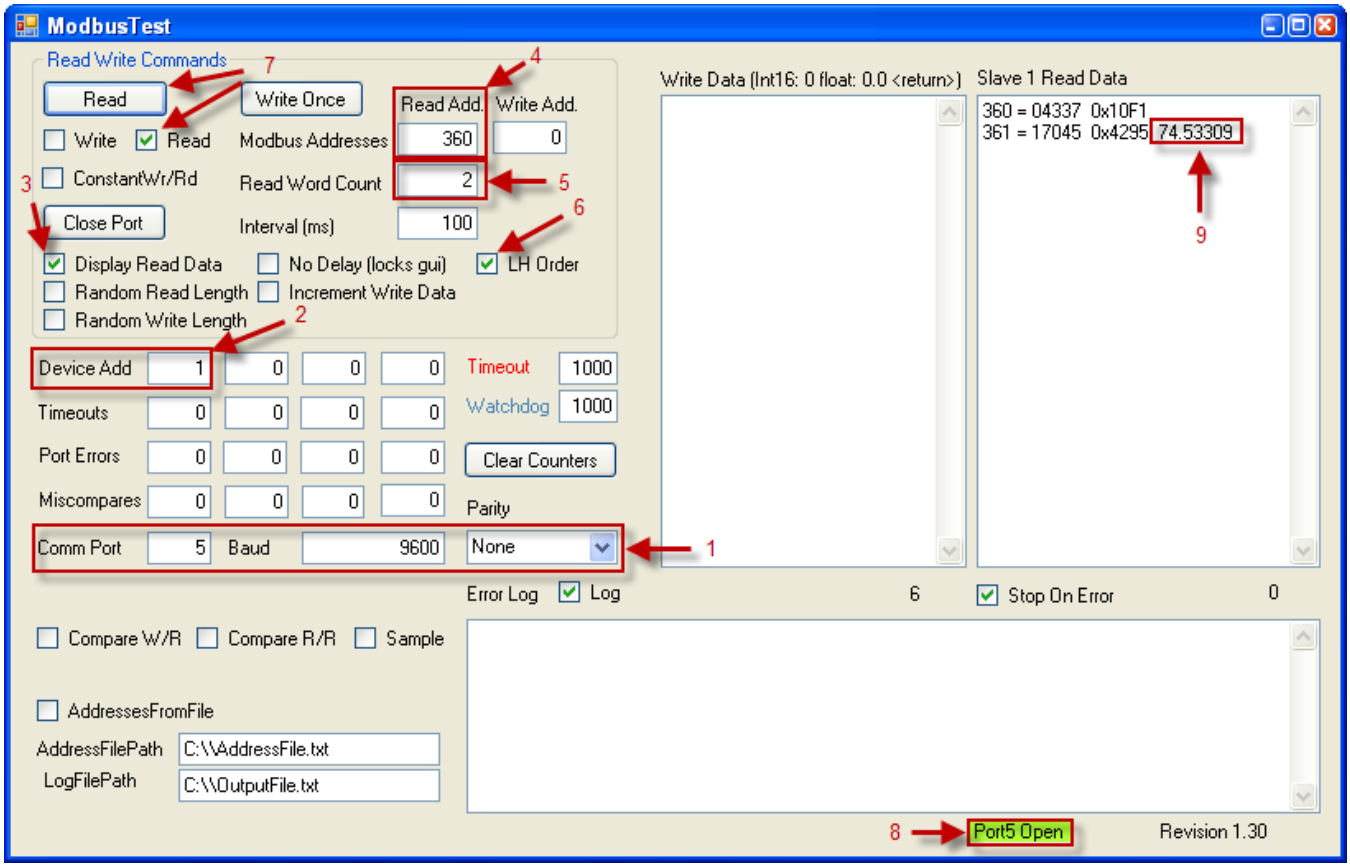


If Comm Port is not valid, you will receive error messages.



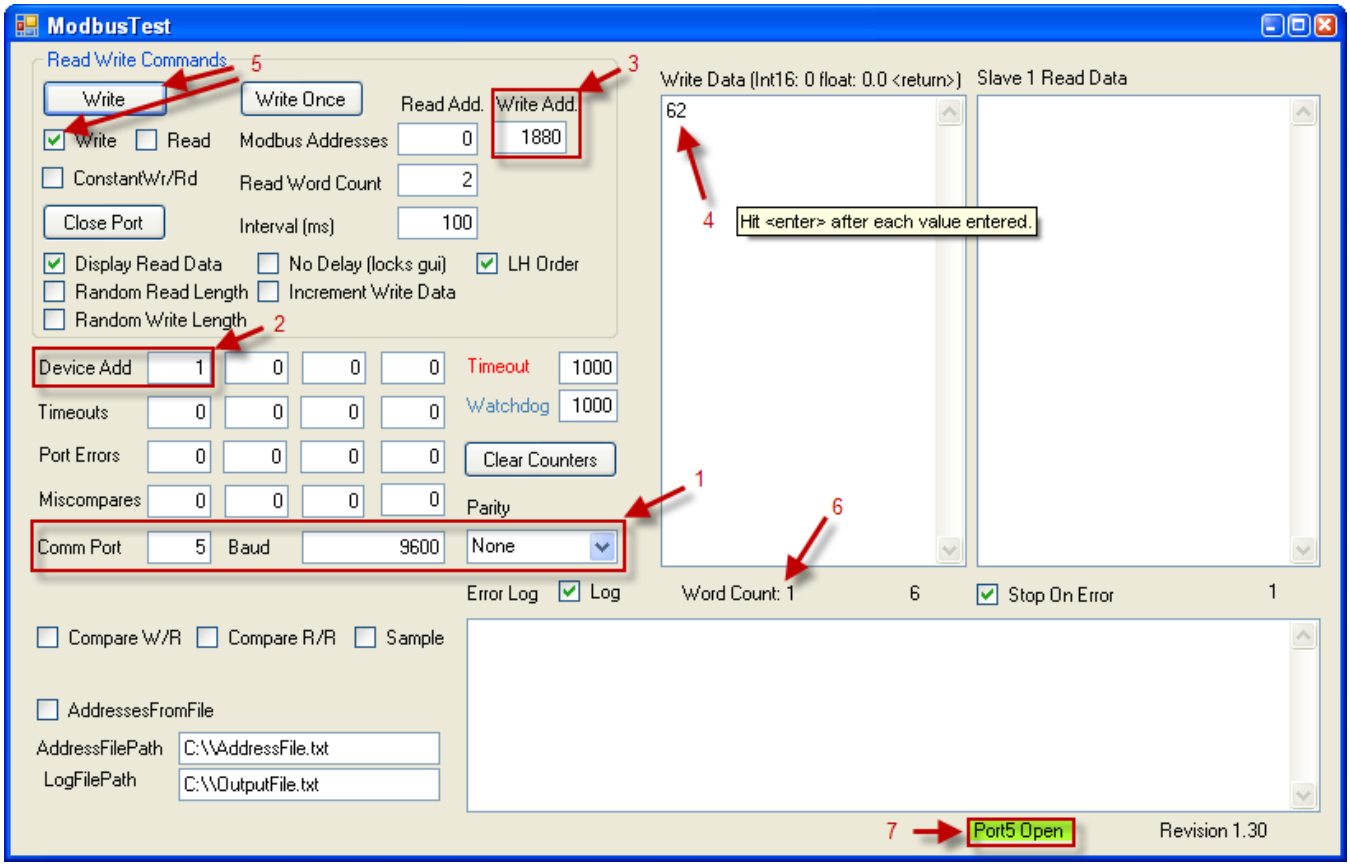
Example 2: To read a 32-bit value (2 registers or words).

1. Enter Comm Port address of PC, Baud and Parity of controller's settings
2. Enter 'Device Add' which is controller address on network (1 to 247)
3. Check 'Display Read Data'
4. Enter Modbus register to read in controller using relative addressing. The user's manual show relative addresses for all parameters. Registers 360 & 361 is the analog input 1 process value of EZ-ZONE® PM product.
5. Enter 'Read Word Count' (2 = 32 bits)
6. Check the word order for 32-bit values. The default setting in the controller is Low word – High word order.
7. Check only 'Read' box and Click 'Read' button
8. If valid port was entered, display will show 'Portx Open'
9. Values read from register are displayed in decimal and hexadecimal formats. The converted 32-bit floating point value is displayed in degrees Fahrenheit by default.



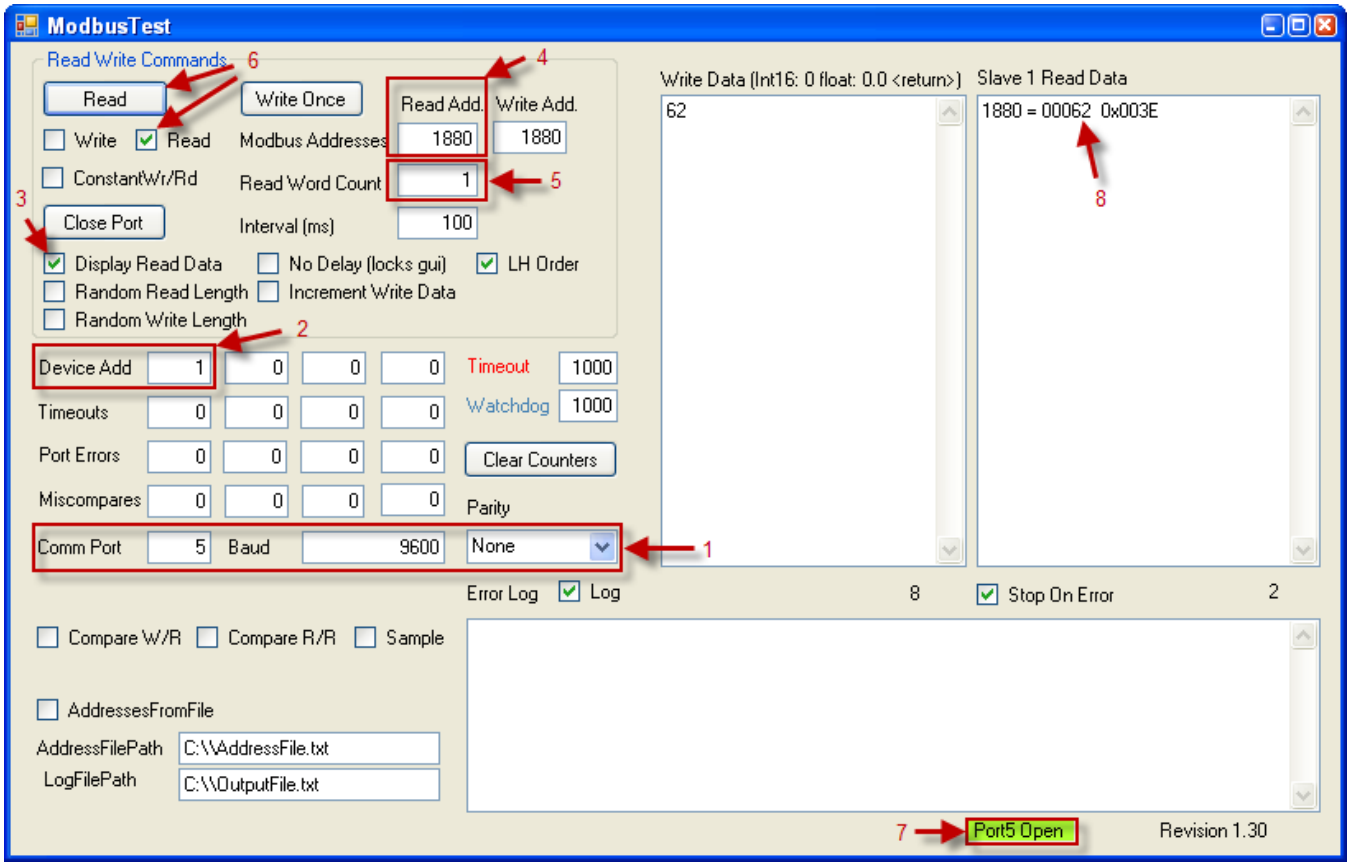
Example 3: To write a 16-bit value (1 register or word).

1. Enter Comm Port address of PC, Baud and Parity of controller's settings
2. Enter 'Device Add' which is controller address on network (1 to 247)
3. Enter Modbus register to read in controller using relative addressing. The user's manual show relative addresses for all parameters. Register 1880 is the control mode 1 value of EZ-ZONE® PM product.
4. Enter data value to write. 62 = Off, 10 = Auto and 54 = Manual. After entering integer value in this field, be sure to hit the 'Enter key'
5. Check only 'Write' box and Click 'Write' button or Click the 'Write Once' button
6. Word Count indicates number of words written
7. If valid port was entered, display will show 'Portx Open'



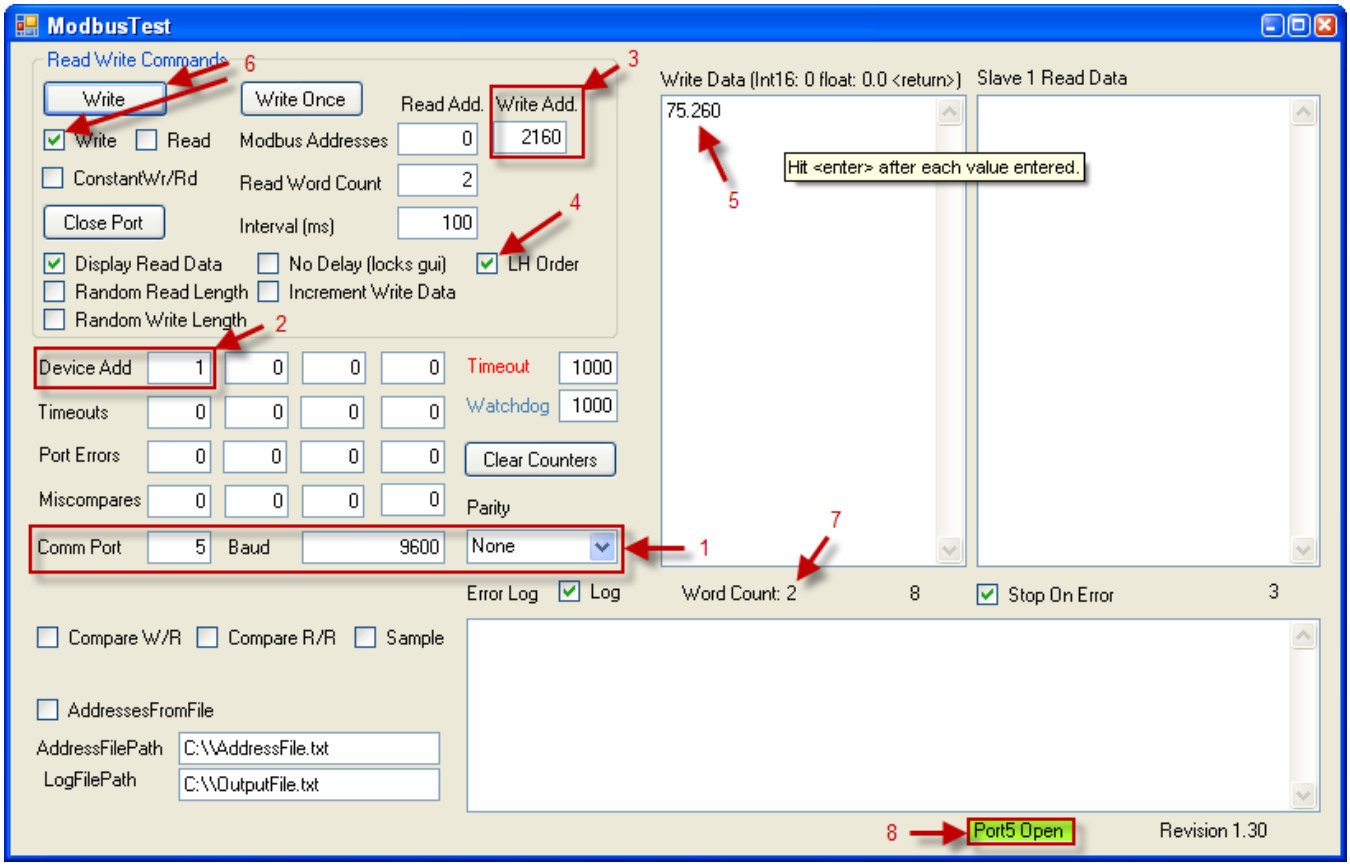
Example 4: To validate previous write of a 16-bit value (1 register or word).

1. Enter Comm Port address of PC, Baud and Parity of controller's settings
2. Enter 'Device Add' which is controller address on network (1 to 247)
3. Check 'Display Read Data' if not already checked
4. Enter Modbus register to read in controller using relative addressing. The user's manual show relative addresses for all parameters. Register 1880 is Control Mode 1 of EZ-ZONE® PM products.
5. Enter 'Read Word Count' (1 = 16 bits)
6. Check only 'Read' box and Click 'Read' button
7. If valid port was entered, display will show 'Portx Open'
8. Values read from register are displayed in decimal and hexadecimal formats.



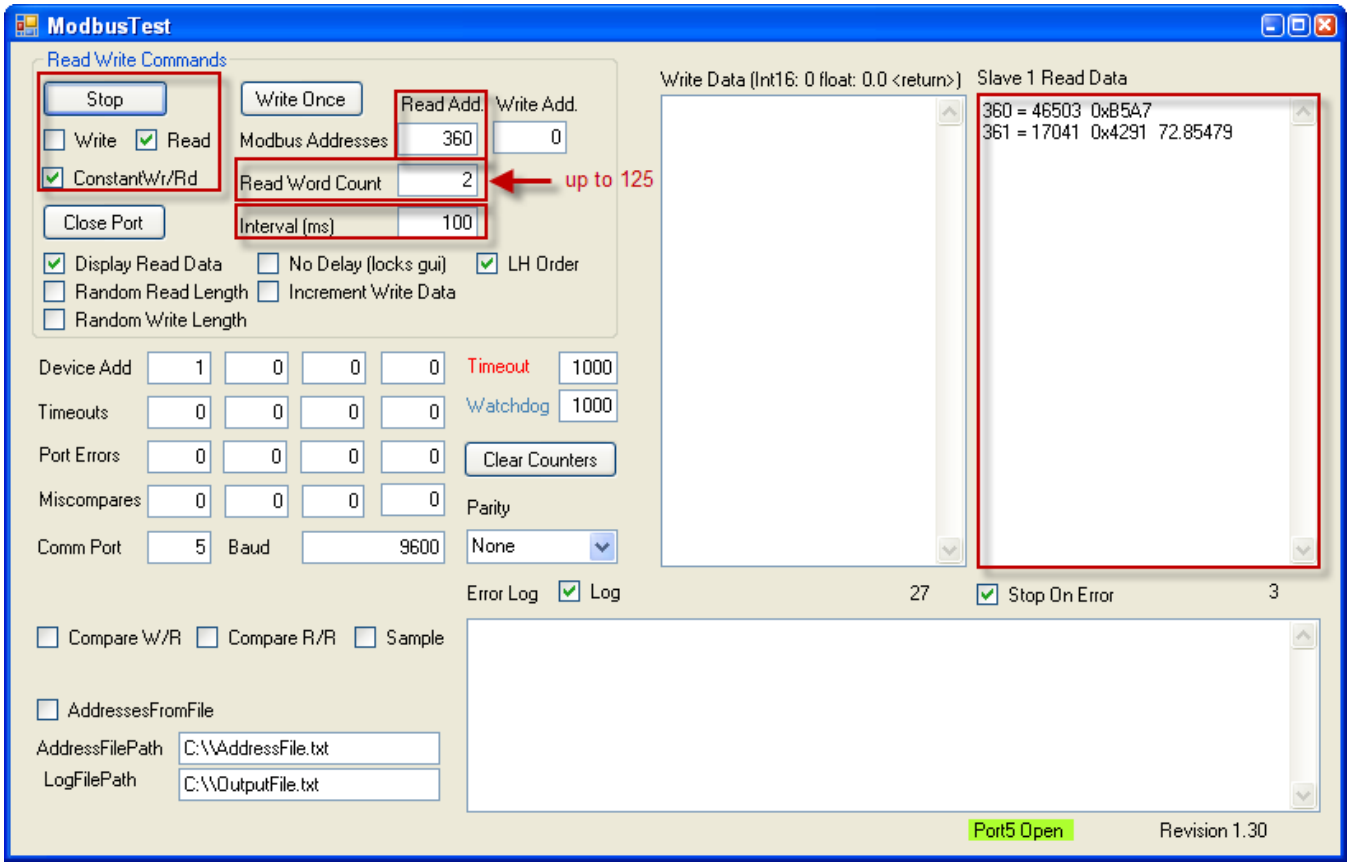
Example 5: To write a 32-bit value (2 registers or words).

1. Enter Comm Port address of PC, Baud and Parity of controller's settings
2. Enter 'Device Add' which is controller address on network (1 to 247)
3. Enter Modbus first register to write in controller using relative addressing. The user's manual show relative addresses for all parameters. Registers 2160 & 2161 is the closed loop set point 1 value of EZ-ZONE® PM product.
4. Match the word order for 32-bit values. The default setting in the controller is LH Order.
5. Enter data value to write. For 32-bit floating point values, enter the decimal point followed by the precision required then hit enter key.
6. Check only 'Write' box and Click 'Write' button or Click the 'Write Once' button
7. Word Count indicates number of words written
8. If valid port was entered, display will show 'Portx Open'

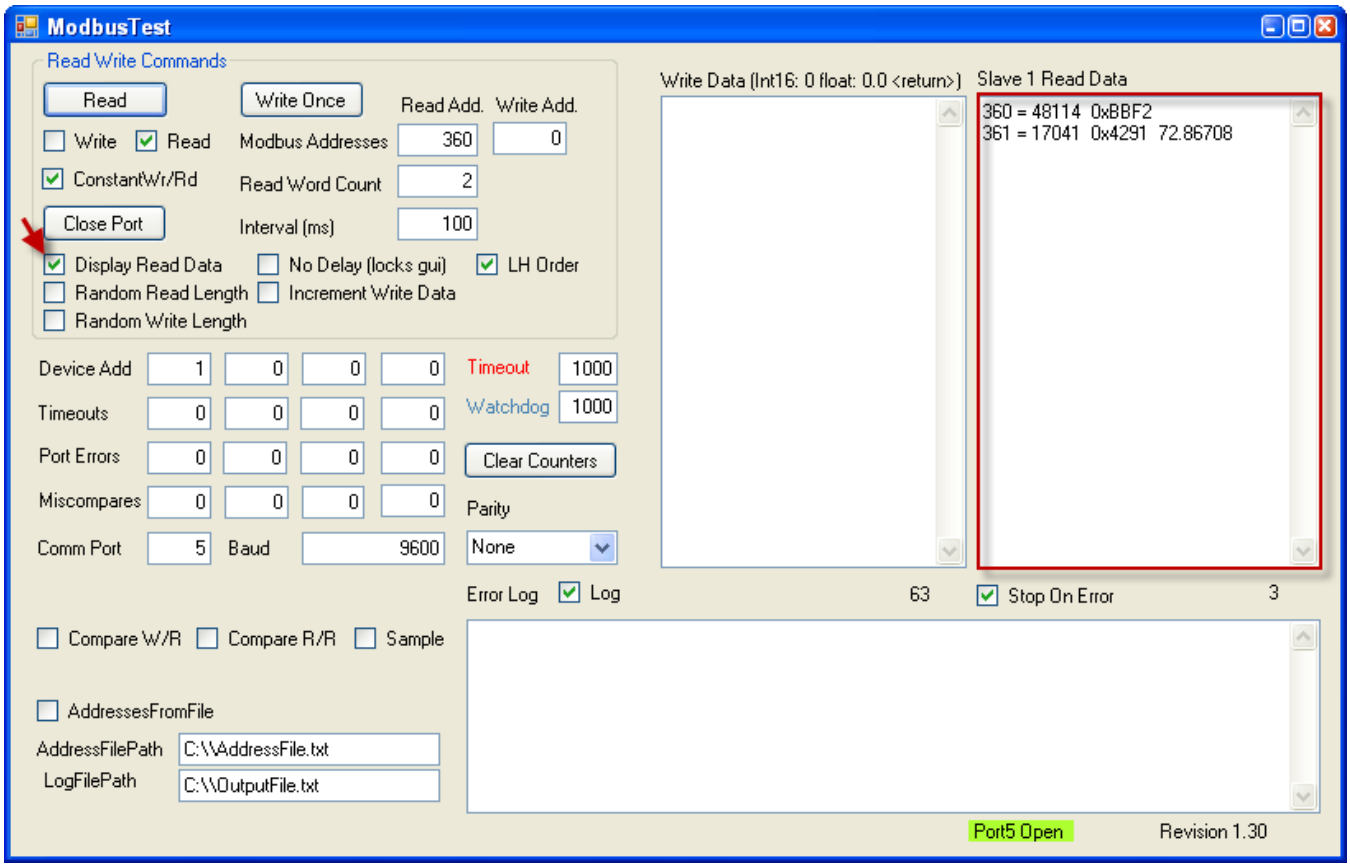


Tip: The word count for write data is displayed below the write data text box if the user wants to validate the number of words actually written.

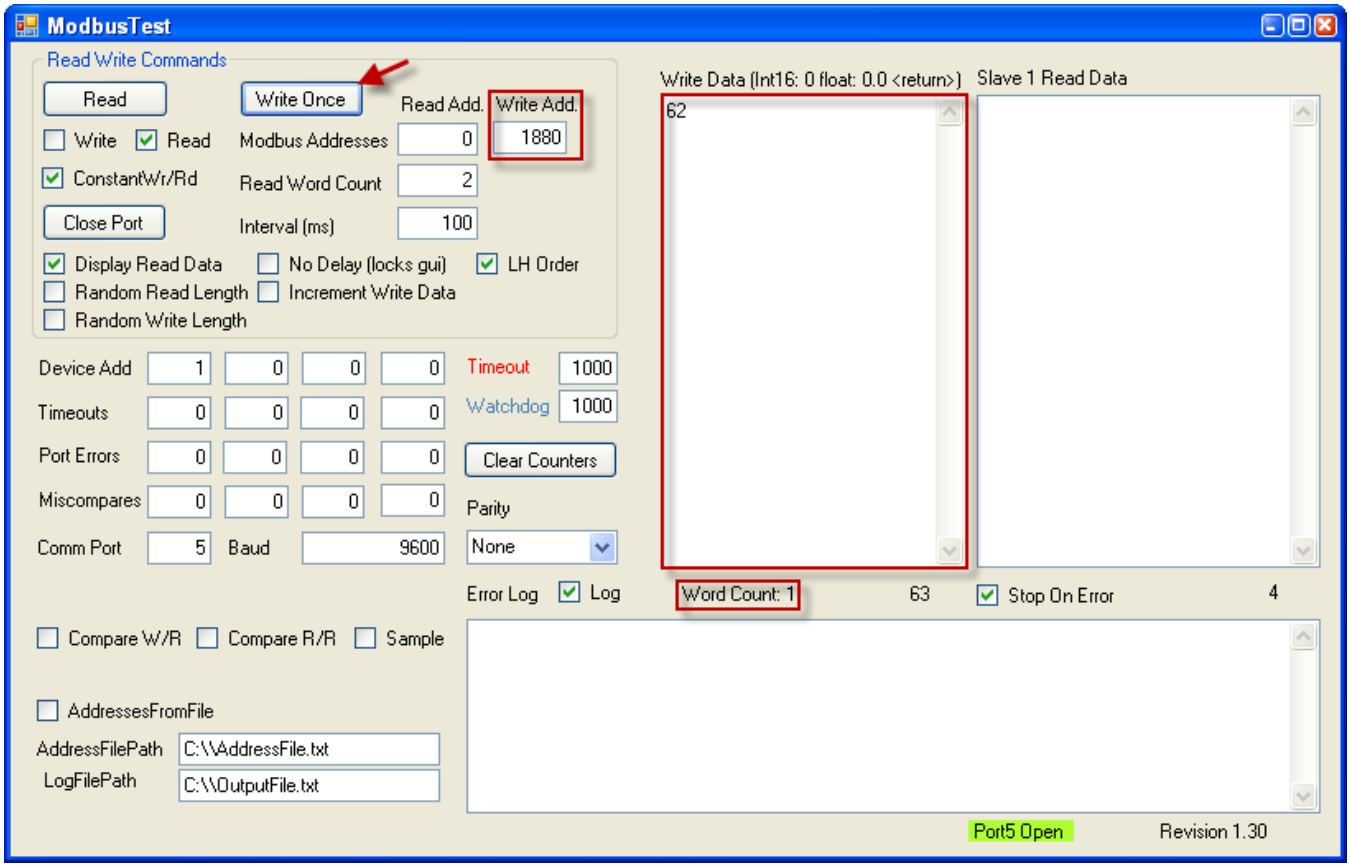
Use the ‘Read’ button to read registers one time or repeatedly. Port is automatically opened when selected. Register entered is read and displayed in panel to right. User may read 1 to 125 registers with one read command by entering the number of registers to read in the ‘Read Word Count’ box. The check box for ‘ConstantWr/Rd’ will repeatedly gather data at Interval in milliseconds selected. Read data is displayed in decimal and hexadecimal format. When pairs of registers are read, the 32-bit floating point value is interpreted and displayed. EZ-ZONE products always start at even numbers like 2, 360 or 2200. Use ‘Stop’ button to terminate Constant Write/Read commands.



The check box for ‘Display Read Data’ determines if data is returned to the panel on right. Typically, this is always selected.

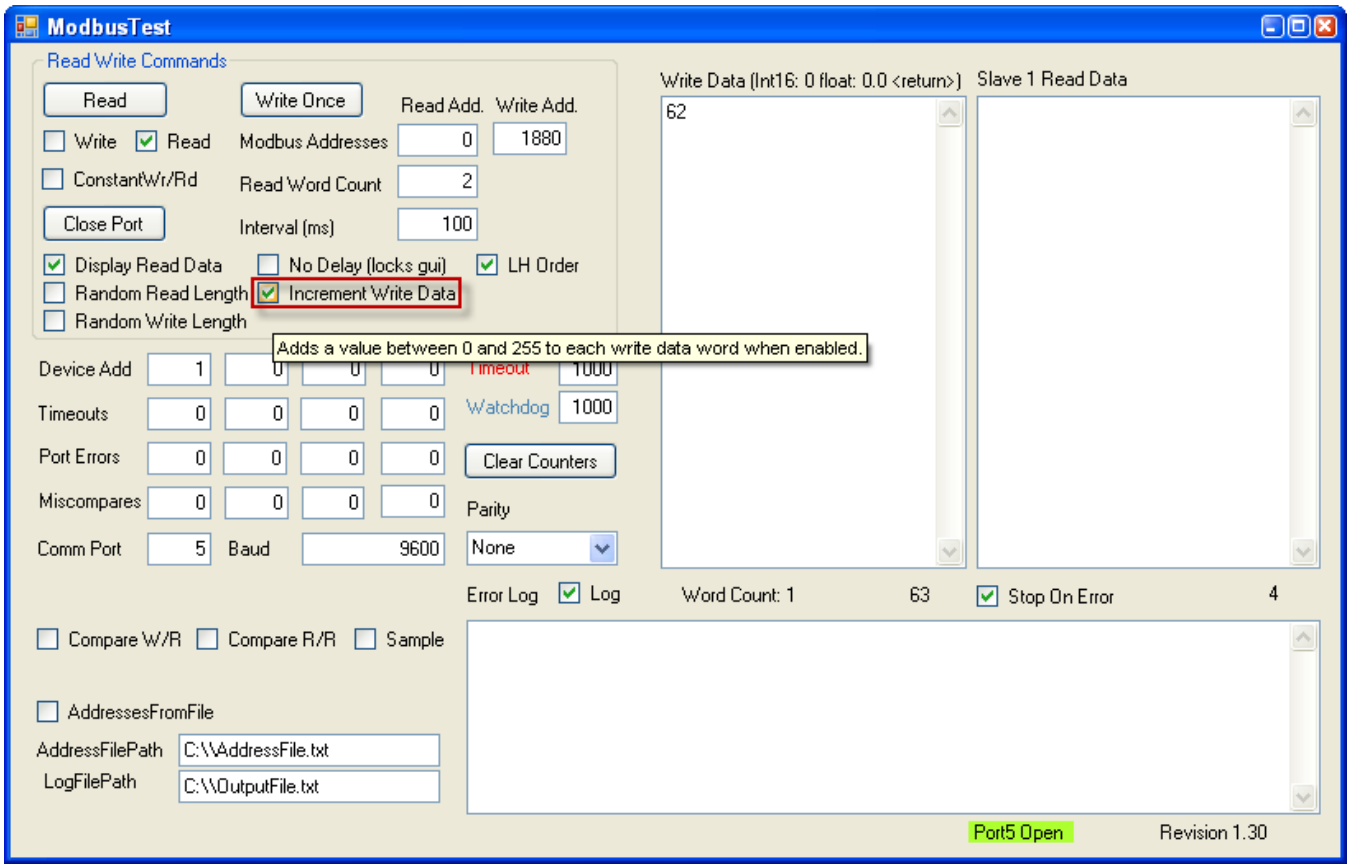


Use the ‘Write Once’ button with the ‘Write Data’ pane to right. Write Data will be written one time starting at register address entered in ‘Write Add.’ box. It is suggested that a read command follows to validate correct writes. The program uses the multi-write function code 0x10. If you wish to write multiple ‘Write Address’ registers with ‘Write Data’ values, enter each data value followed by an ‘Enter key’ on a separate line in the ‘Write Data’ panel. Each ‘Write Data’ value will be written to incremental Modbus registers starting at entered register ‘Write Add.’



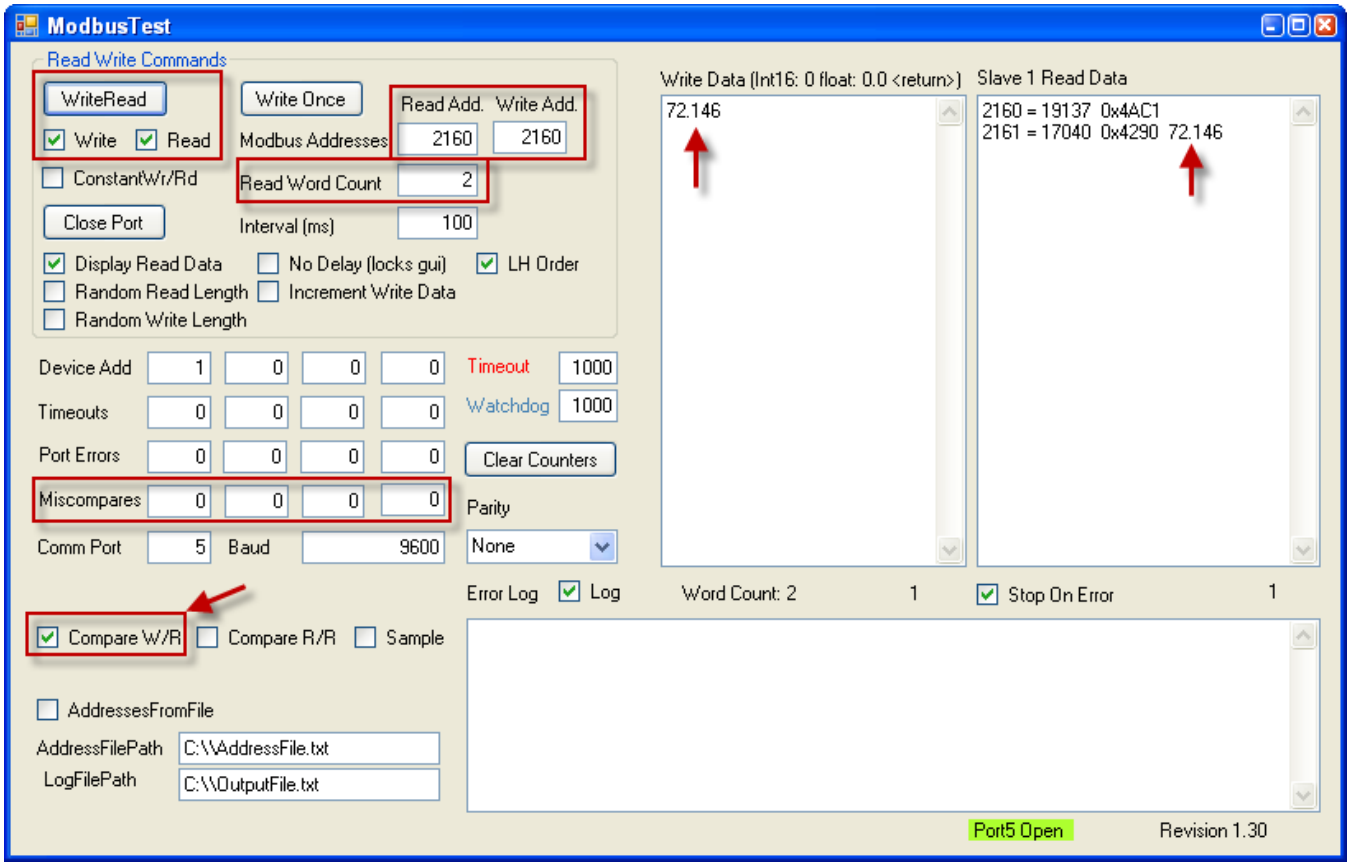
Tip: Floating point values must be entered with a decimal and integer values are entered with no decimal in the ‘Write Data’ pane. Each data value must be followed by a carriage return. The Word Count: indicated the number of words written. A word is 16-bits. Be sure to write the correct data type to the appropriate registers.

The check box ‘Increment Write Data’ is for product testing purposes only and should be avoided. This will increment the ‘Write Data’ value written to a register automatically from 0 to 255 with each execution of the write command.

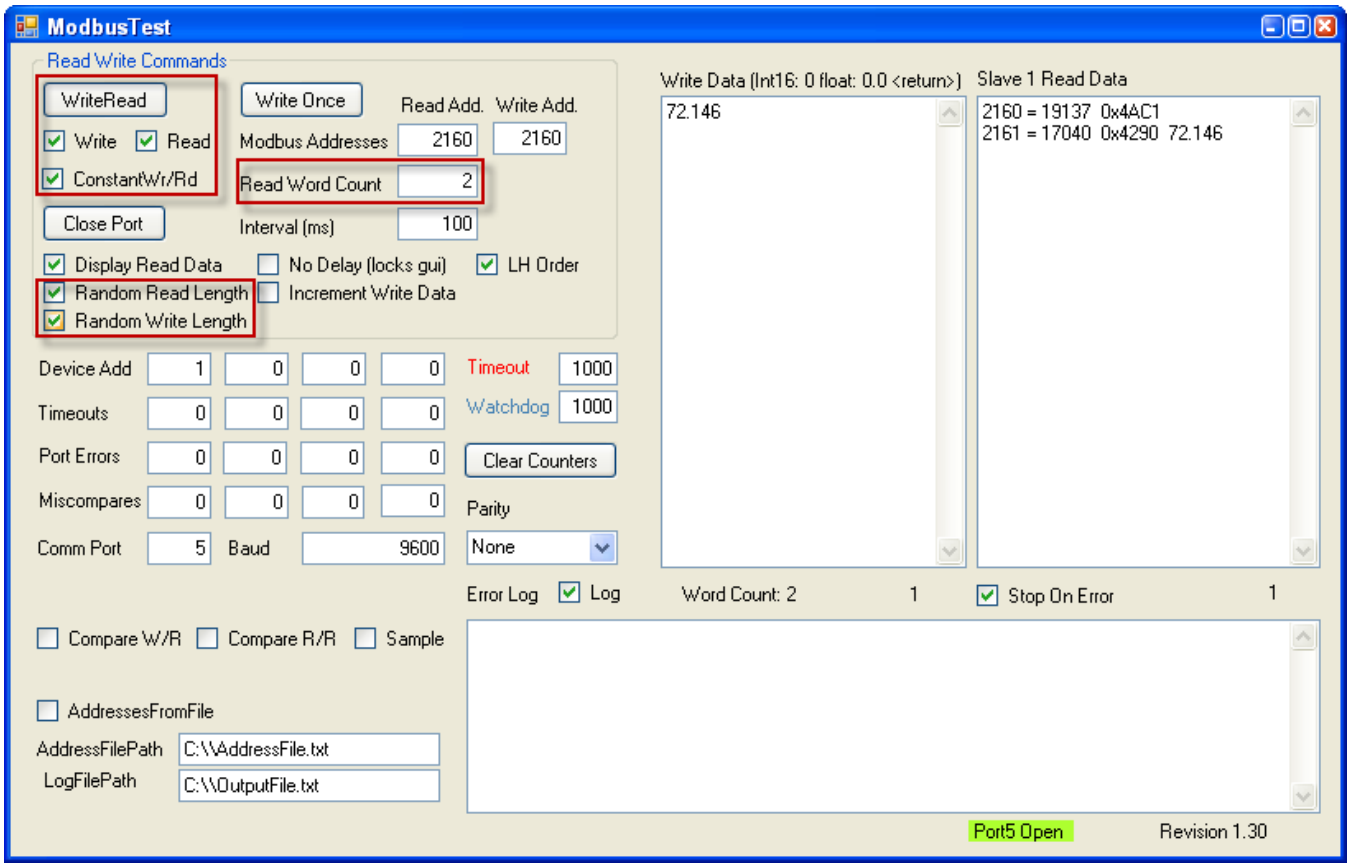


Tip: The increment write data function starts at the write value entered, and adds between 0 and 255 to each word transmitted. So if you send 1 word, it will write values of 1,2,3,4,5.

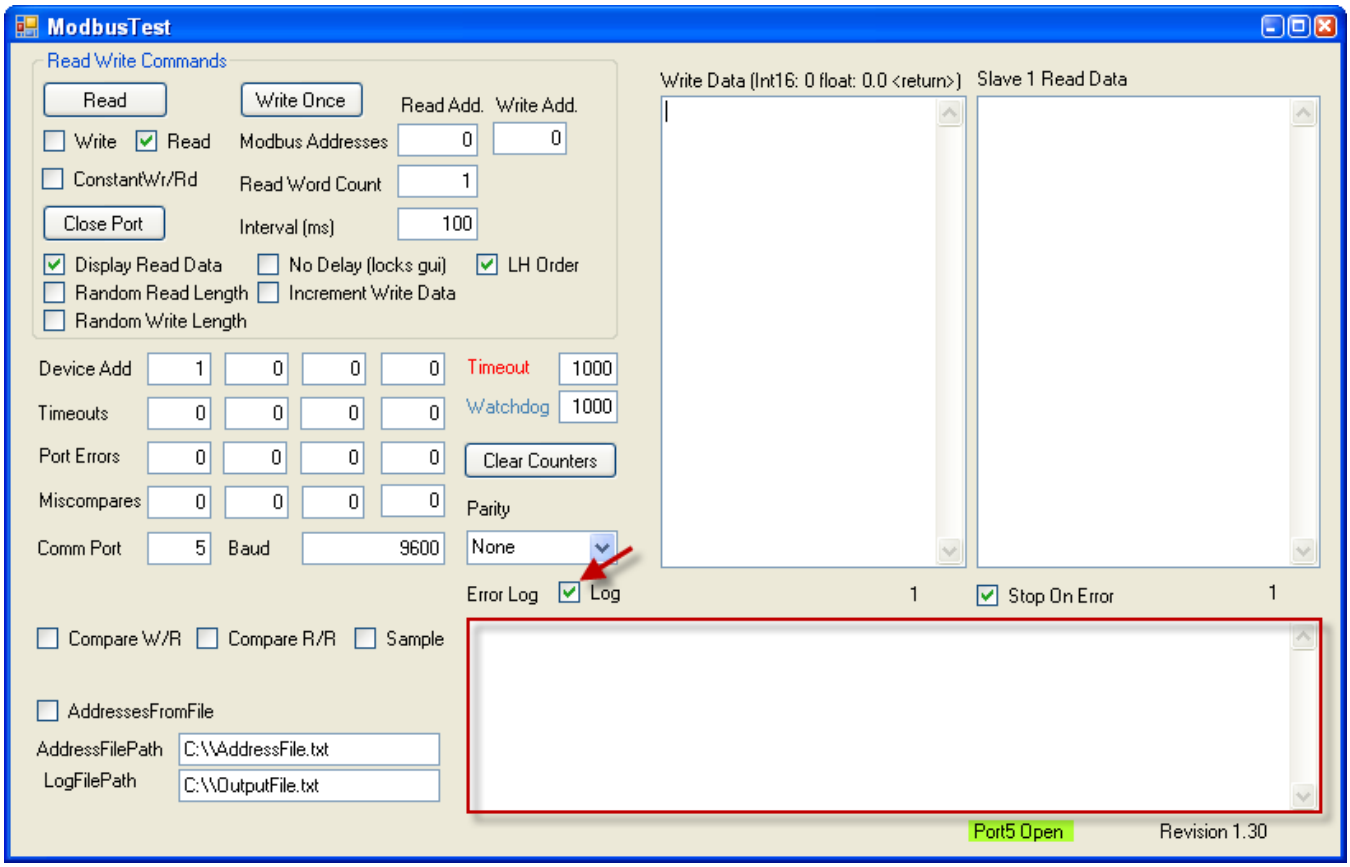
The check box ‘Compare W/R’ feature is intended to be used when both the ‘Write’ and ‘Read’ check boxes are enabled. The ‘Modbus Addresses’ registers for ‘Read Add.’ and ‘Write Add.’ must be the same. The ‘Read Word Count’ must be correct for the data type. The program will initiate a write and then a read to validate that the values match. If the values do not match, the ‘Miscompares’ error counter will increment.



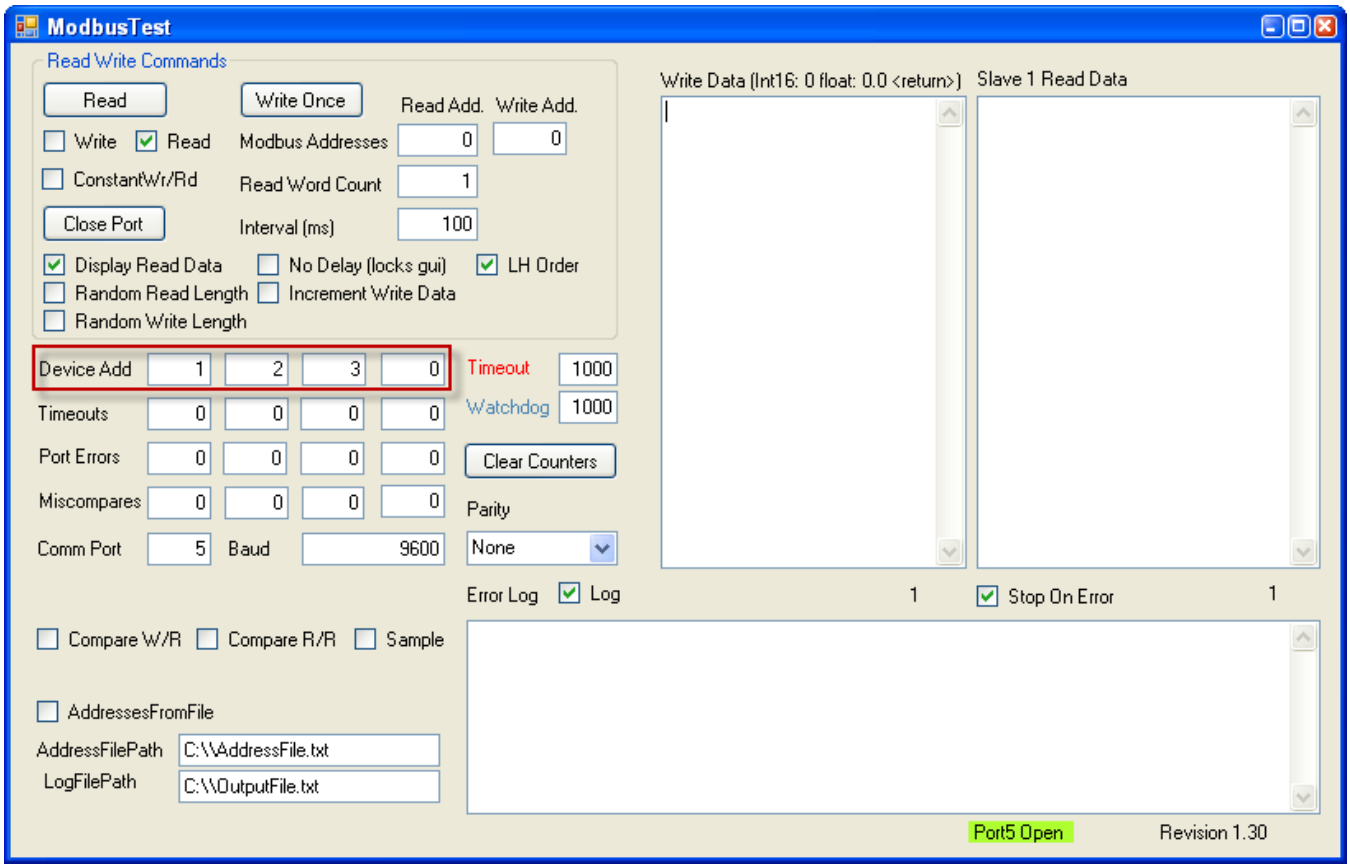
The check boxes ‘Random Read Length’ and ‘Random Write Length’ are for product testing purposes only and should be avoided. This feature will randomize the quantity of read or write registers in conjunction with ‘ConstantWr/Rd’.



The check box for ‘Log Errors’ enables any errors to be logged in ‘Error Log’ panel.

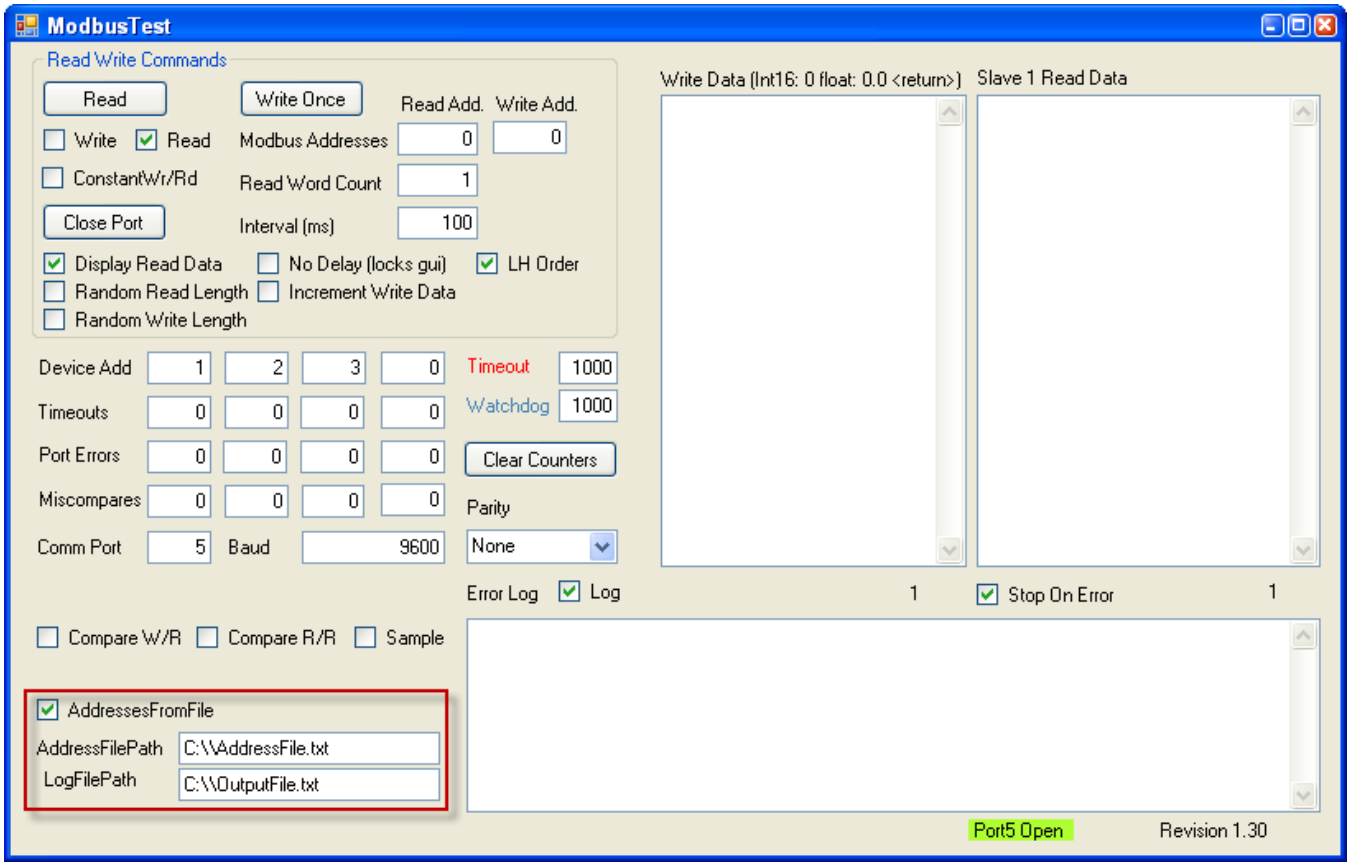


The ‘Device Add’ boxes allow up to 4 controllers to be accessed on the network. The program will read from each controller in round robin sequence unless the ‘Device Add’ field is 0.



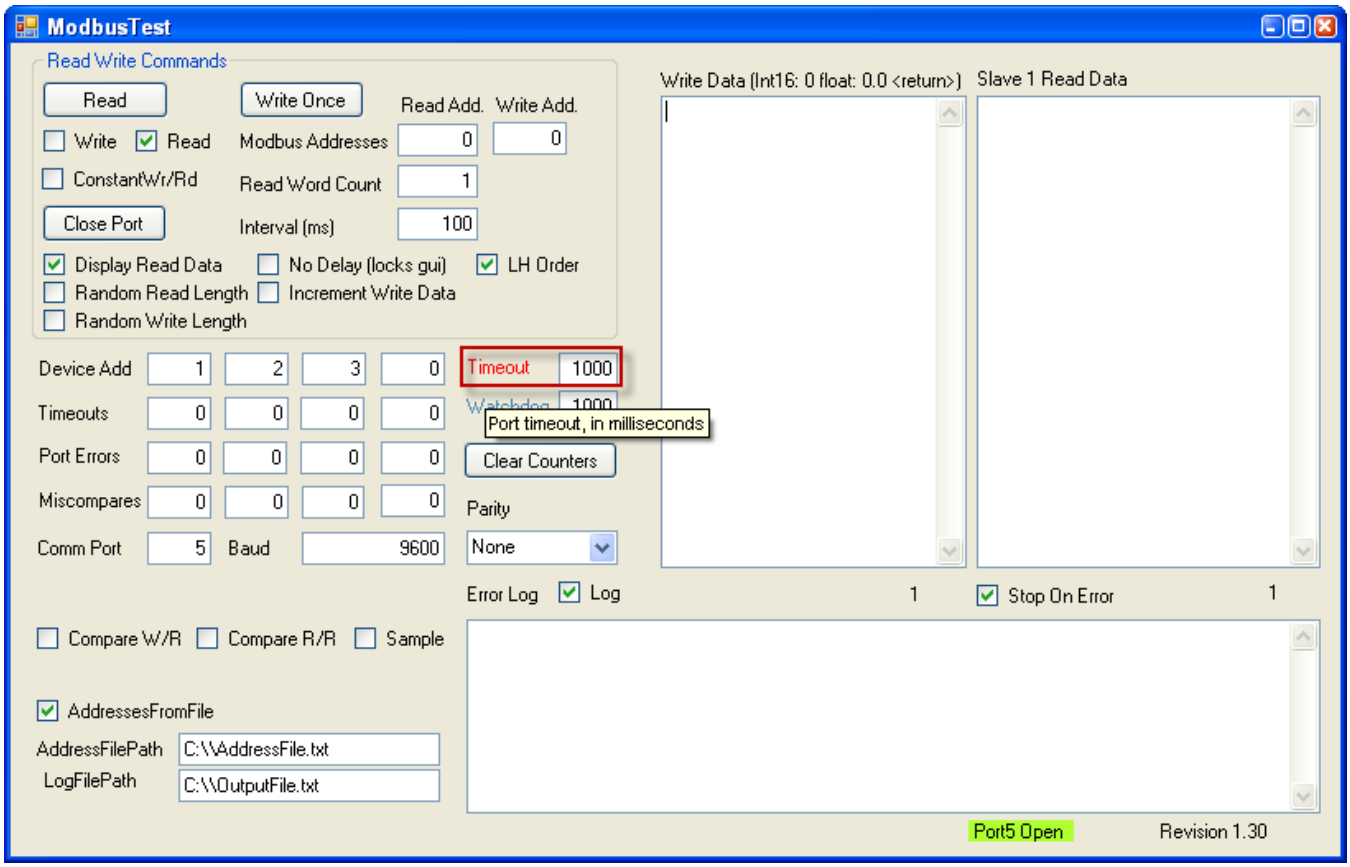
Tip: The rotating function on the 4 device addresses will round robin until it hits a device address of 0, so if you put a 0 in the third field, it will not check the fourth, etc...

The check box ‘AddressesFromFile’ feature will read file ‘AddressFilePath’ to determine which registers to read from controller and write values read to file “LogFilePath’. This feature allows a given set of registers to be read and the results written to a file for comparisons. This happens once when the read button is clicked. This is not meant to be a data log file and does not use time/date stamping.

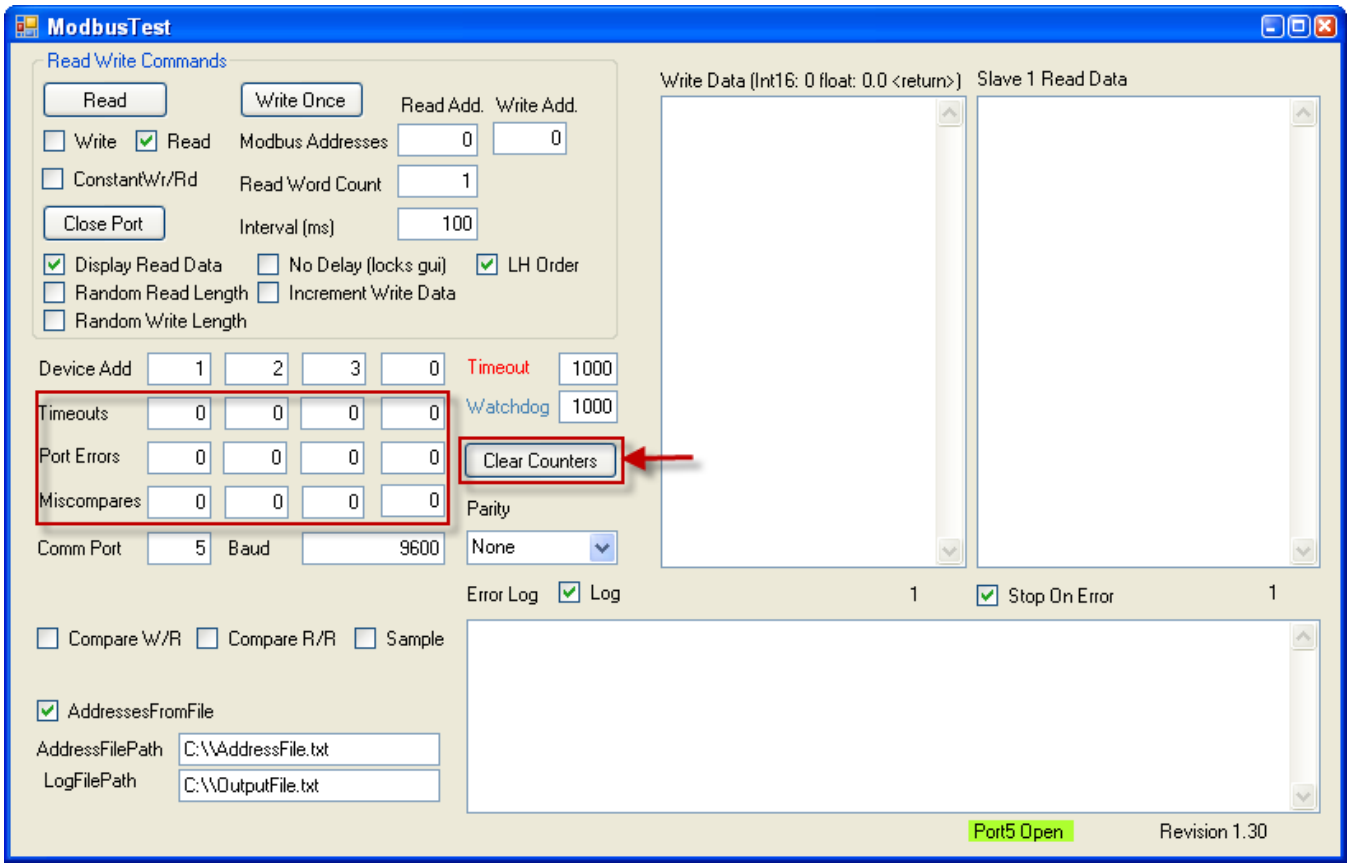


Tip: You can use the ‘AddressFromFile’ function as a data logger if you want to repeat reads of a single or multiple addresses. You just need to create a large file with the addresses repeated for the number of times you want to read the address, or addresses.

The field ‘Timeout’ specifies in milliseconds the wait for a response time before a communications port timeout is generated. The timeout field monitors all comms activity including connect, and other comms and application overhead. Typical values are between 1000 to 5000 milliseconds. Enter higher numbers if occasional errors are generated.



The button ‘Clear Counters’ resets the error counts accumulated to left. Also, the Error Log panel will be cleared.



The check box ‘No Delay (locks gui)’ is for testing purposes only and should be avoided. When used with ‘ConstantWr/Rd’, this feature maximizes the speed of read/writes to stress the communications capabilities of the controller. The ModbusTest program must be terminated via Windows Task Manager.