To test SLATE Modbus communication it is simplest to use a Modbus utility. Baseblock ComTest Pro for Modbus Devices is one viable option. You can download this (free) program at www.baseblock.com/PRODUCTS/comtestpro.htm. ComTest Pro allows users to test and troubleshoot Modbus RTU or Modbus TCP communication. The software offers comprehensive data-logging, data-formatting and error checking features. It is available for Windows, OS X, iOS and Android.

In the following example we show how to use this program to make sure SLATE Modbus communication is enabled and working properly over Modbus/TCP.

Please keep in mind that SLATE is a Modbus "slave" and thus requires a Modbus "master" be present on the bus for Modbus communication to be successful.

Following are the steps for testing SLATE Modbus:

Enable Modbus on the SLATE Base Module:

At the SLATE Base…

Click the "Menu" button

Scroll to "Base setup". Click "OK" button.

Scroll to "Network". Click "OK" button.

Scroll to "Ethernet". Click "OK" button.

Scroll to "Modbus TCP". Click "OK" button.

Scroll to "Configuration" (note the port# 502). Click "OK" button.

Scroll to "None" and use the side arrows to select "Modbus/TCP".

Scroll to "OK" and click "OK" button.

The SLATE Base Module is now Modbus TCP/IP enabled and ready to test.

Now that the SLATE Base Module is ready to test, the next step is to access a register set to use. We used the Demo Case (DSP3983/U) program registers. You can use other programs you may have available.

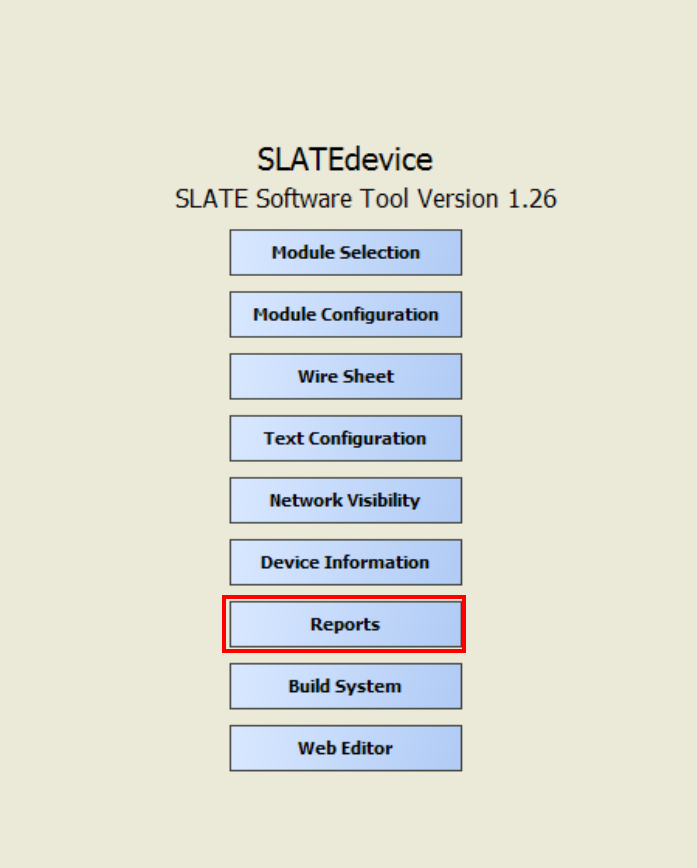
Following are the steps to get to the registers for your SLATE Device:

In Niagara AX, open your SLATE Device. Make sure this device has been built so the registers are visible in the report. Also make sure to unhide any (internal) registers you may want to access using the "Network Visibility" button.

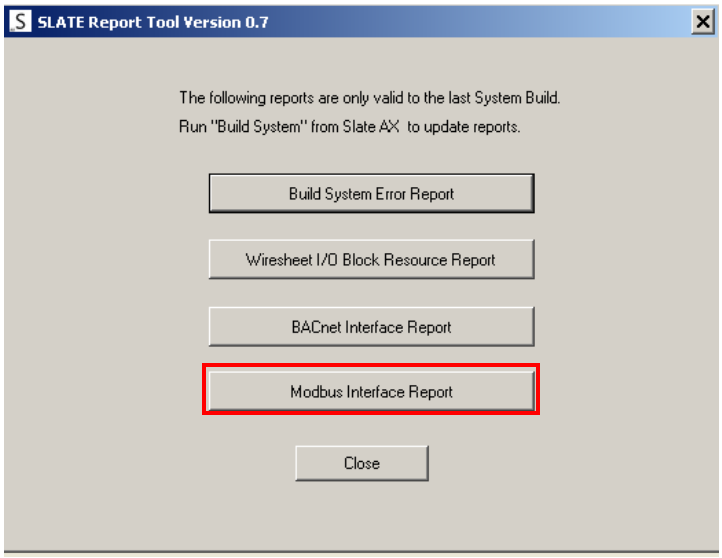
NOTE: All SLATE registers' default state is "hidden". All Designer registers (inputs and outputs created in the wire sheet using "NetworkInput" blocks) are visible.

In Niagara AX having already accessed your SLATE Device…

Click the “Reports” button.



Click the “Modbus Interface Report” button.



The list below shows the outwardly facing registers in the “Registers” column that are required for testing. Register information is shown in the “Register Name” column. It’s important to keep track of the register data to insure the information transmitted/received during testing is accurate for that register.

Modbus Interface Report

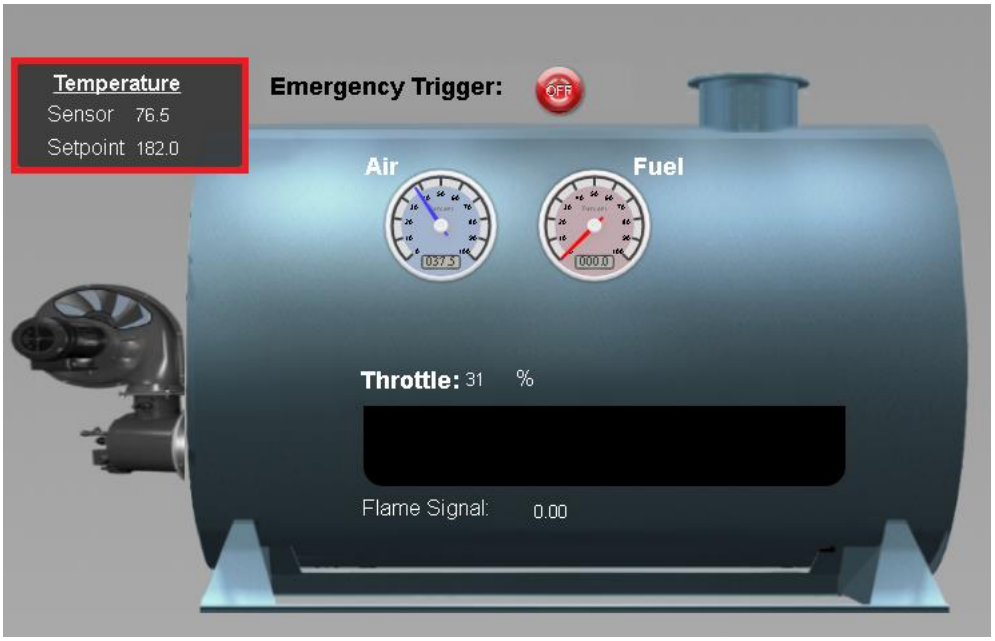
Registers	Register Name	Resource	Data Type	Data Size	Units	Min Value	Max Value	Enumerations
1 - 2	m1ControlProgram_OpSetpoint	m1r1000	F32	4	deg F	-3.402823E+38	3.402823E+38	
3 - 4	m1ControlProgram_InterlockContr	m1r1002	F32	4	float	-3.402823E+38	3.402823E+38	
5 - 6	m1ControlProgram_AutoManualBMSSelect	m1r1003	F32	4	float	-3.402823E+38	3.402823E+38	
7 - 8	m1ControlProgram_BMSThrottle	m1r1004	F32	4	%	-3.402823E+38	3.402823E+38	
9 - 10	m1ControlProgram_Prop	m1r1005	F32	4	deg F	-3.402823E+38	3.402823E+38	
11 - 12	m1ControlProgram_Int	m1r1006	F32	4	float	-3.402823E+38	3.402823E+38	
13 - 14	m1ControlProgram_Der	m1r1007	F32	4	float	-3.402823E+38	3.402823E+38	
15 - 16	m1ControlProgram_LFH_Threshold	m1r1008	F32	4	deg F	-3.402823E+38	3.402823E+38	
17 - 18	m1ControlProgram_LFH_Time	m1r1009	F32	4	s	-3.402823E+38	3.402823E+38	
21 - 22	m1ControlProgram_Trim_Control	m1r1011	F32	4	%	-3.402823E+38	3.402823E+38	
23 - 24	m1ControlProgram_BMS_Demand	m1r1012	F32	4	float	-3.402823E+38	3.402823E+38	
25 - 26	m1ControlProgram_PID_Output	m1r1013	F32	4	%	-3.402823E+38	3.402823E+38	
27 - 28	m1ControlProgram_TempControlDemand	m1r1014	F32	4	float	-3.402823E+38	3.402823E+38	
29 - 30	m1ControlProgram_EStop_Out	m1r1015	F32	4	float	-3.402823E+38	3.402823E+38	
31 - 32	m1ControlProgram_InterlockState	m1r1016	F32	4	deg F	-3.402823E+38	3.402823E+38	
33 - 34	m1ControlProgram_LFH_Sens_Out	m1r1017	F32	4	deg F	-3.402823E+38	3.402823E+38	
35 - 36	m1ControlProgram_Sensor_Output	m1r1018	F32	4	deg F	-3.402823E+38	3.402823E+38	
37 - 38	m1ControlProgram_LimitOutput	m1r1019	F32	4	deg F	-3.402823E+38	3.402823E+38	
39 - 40	m1ControlProgram_LimitThresh1	m1r1020	F32	4	deg F	-3.402823E+38	3.402823E+38	
41 - 42	m1ControlProgram_LimitThresh2	m1r1021	F32	4	deg F	-3.402823E+38	3.402823E+38	
43 - 44	m1ControlProgram_Flame_Signal_Out	m1r1022	F32	4	float	-3.402823E+38	3.402823E+38	
45 - 46	m1ControlProgram_ThrottleOut	m1r1023	F32	4	%	-3.402823E+38	3.402823E+38	
47 - 48	m1ControlProgram_LFH_Active	m1r1024	F32	4	deg F	-3.402823E+38	3.402823E+38	

Save to .csv File Close

Demo Case program registers; data type = floating.

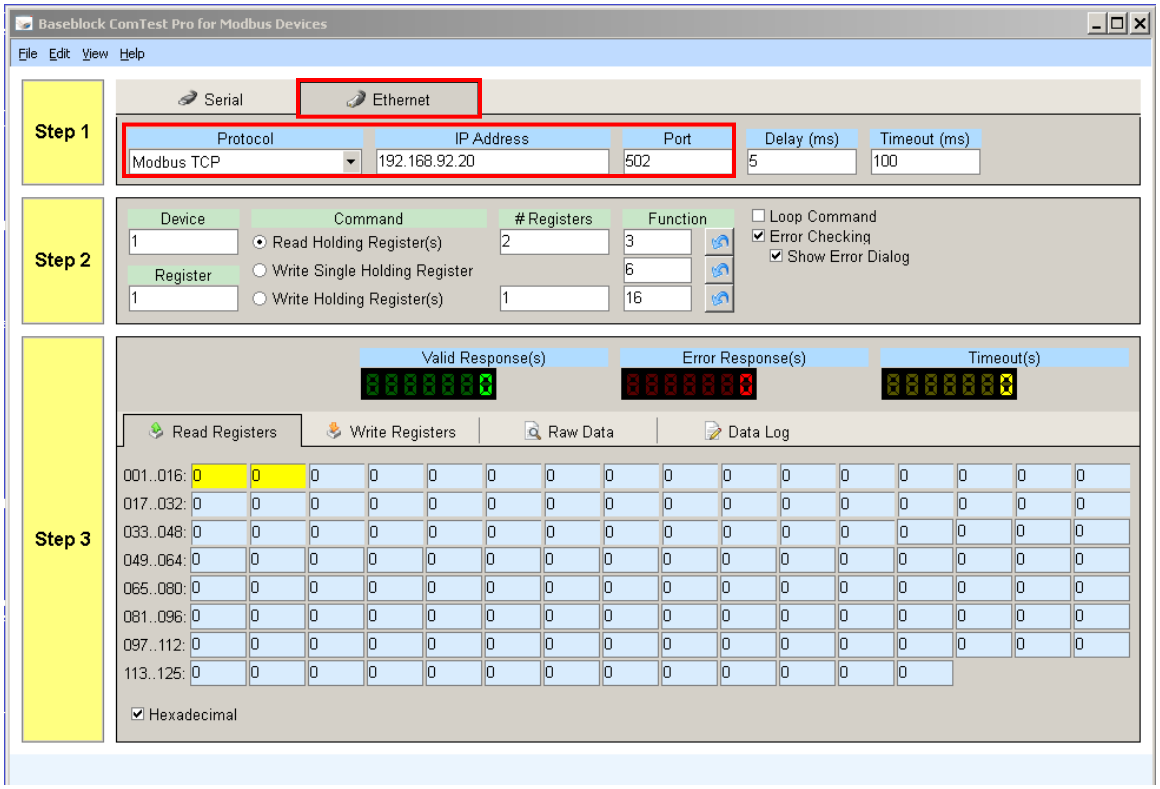
Example:

Register 1: OpSetpoint (writeable)
Register 35: SensorOutput



Open ComTest Pro:

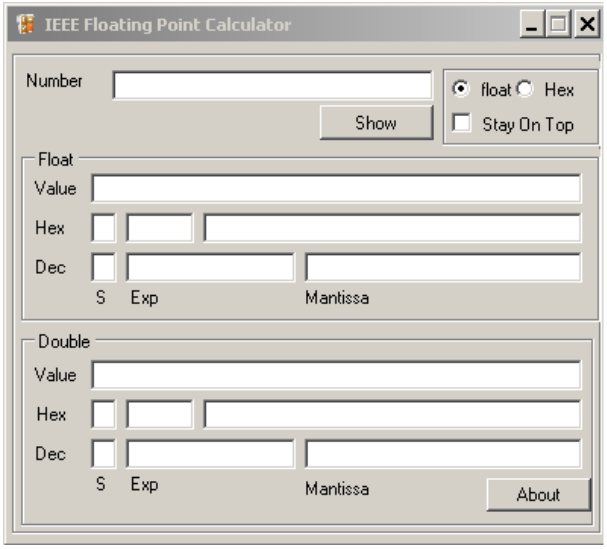
Select the Ethernet tab. Make sure the IP address is set correctly and the port is the same as the SLATE port. The Device value is unused when connecting to SLATE via Ethernet.



In the initial test we will change the burner set-point from its current value to 150F. If the current value is already 150F, select another set-point value to insure Modbus communication is functioning.

From the register list we can see the set-point register is register 1-2 (OpSetpoint). Set the ComTest Pro utility is to display in Hexadecimal. Next we want to convert the OPSetpoint of 150.0 to its floating point hexadecimal representation. For this we can use a conversion calculator. We downloaded IEEE Calc. to convert floats to Hexadecimal.

https://sourceforge.net/projects/ieeecalc/files/latest/download.
This calculator allows us to convert float data types to Hexadecimal.



Convert 150 to Hex…

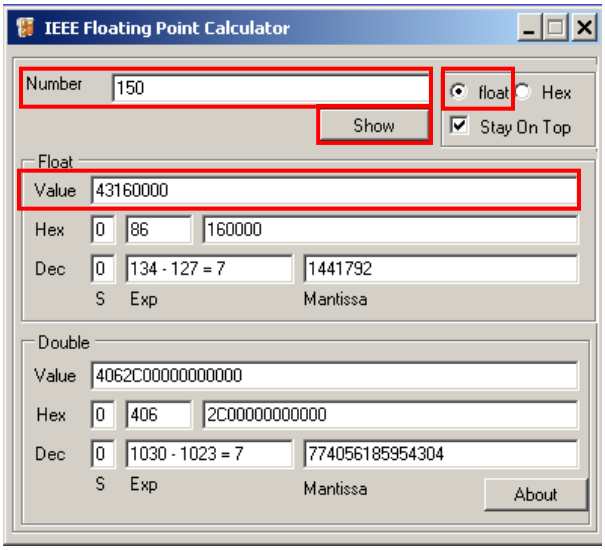
150 is a float that we need converted to its floating point hexadecimal representation.

Type in 150 in the “Number” field.

Insure the type is a float

Click the “Show” button

Result: 150 converted to Hexadecimal = 43160000



Now that we have our floating point Hexadecimal value for 150, we can use ComTest Pro to write the new value to the set-point register.

Select “Write Single Holding Register” or “Write Holding Register(s)”.

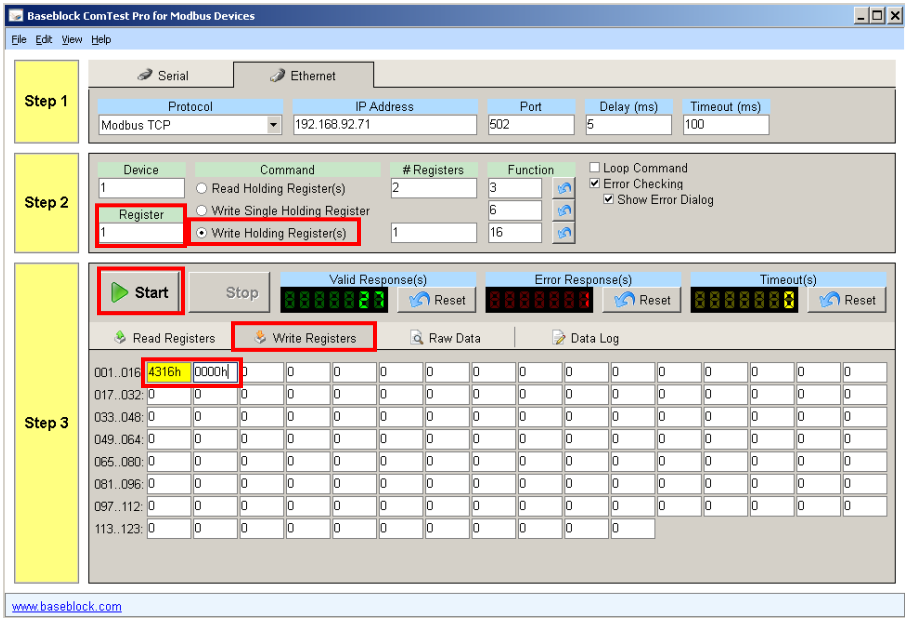
Select the “Write Registers” tab.

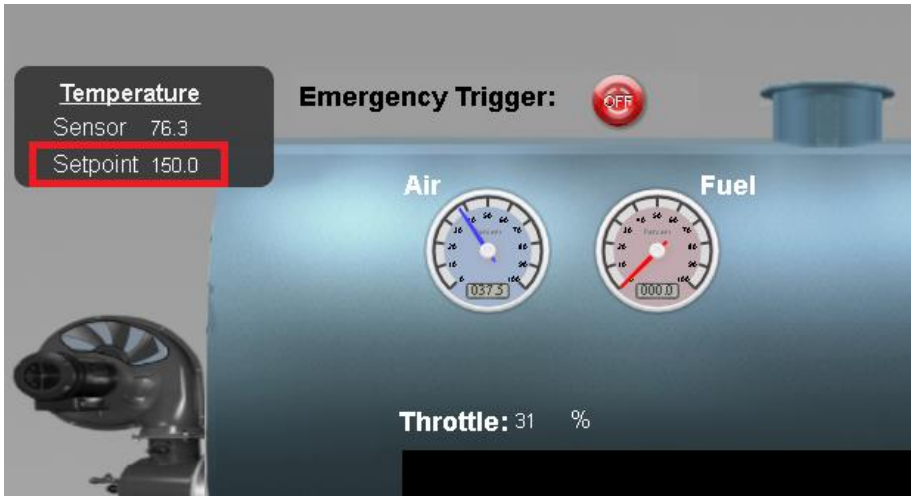
Look up the appropriate register showing the “Setpoint” value. From the Modbus report we see the register with the “Setpoint” value is register 1-2 (OpSetpoint).

Type in the Hexadecimal value of the new set-point in the first register field followed by “h” representing Hexadecimal.

Click the “Start” button.

Verify that the ”Setpoint” on the Main page now shows 150.





Next, we will read the sensor value from the Main page of the demo.

To read data from a register using ComTest Pro:

Select “Read Holding Register(s)”

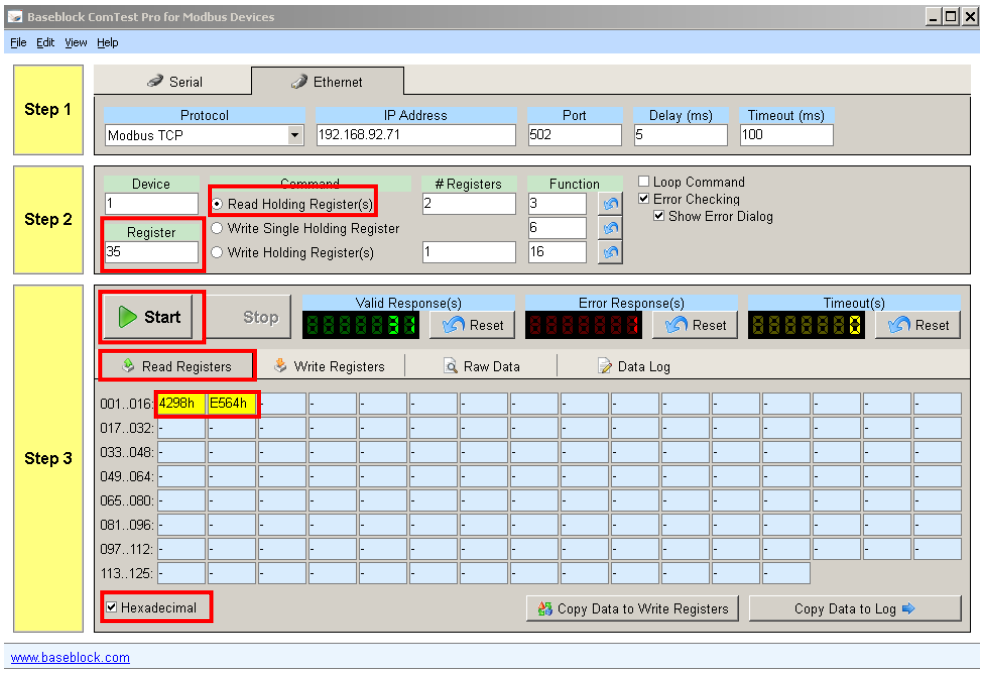
Look up the appropriate register showing the “Sensor” value. From the Modbus report we see the register with the “Sensor” value is register 35 (Sensor_Output).

Select the “Read Registers” tab.

Fill in the Register field with appropriate register (35).

Make sure the Hexadecimal box is checked.

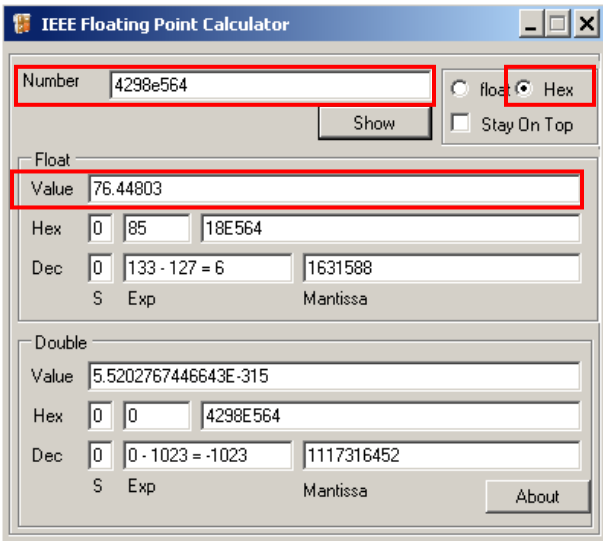
Click the “Start” button

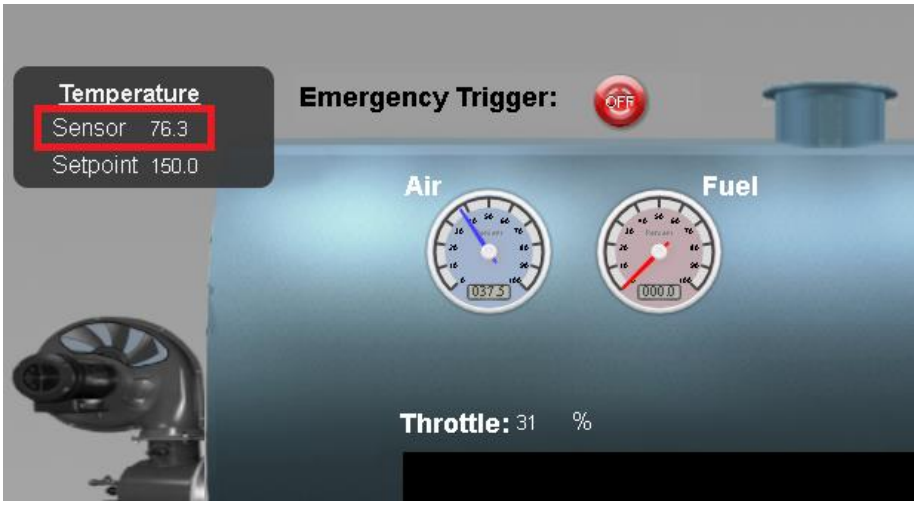


Record the register readings and plug them in the IEEE calculator. As this value is a Hexadecimal value, select the “Hex” option radial button in the calculator.

Click the “Show” button.

Under Float, the decimal conversion is shown which should match the reading on the Main Demo page.





Repeat these steps to receive and transmit other values if desired.

NOTE: Integer based registers such as Booleans or Enums (see Data Type column of the Modbus Interface Report) require simple Hex to Decimal conversion, not the IEEE calculator used for converting float data types.