Modbus TCP and RTU on the RAMOS Ultra

Although this manual was prepared and checked with the best care, CONTEG, spol. s r.o. cannot accept any liability for omissions or errors in this publication. Due to the continuous development and progress, CONTEG, spol. s r.o. also reserves the right to change details and technical specifications of the products described in this manual. Such changes along with eventual errors or printing errata shall not constitute grounds for compensation.

Content

1. Introduction to Modbus

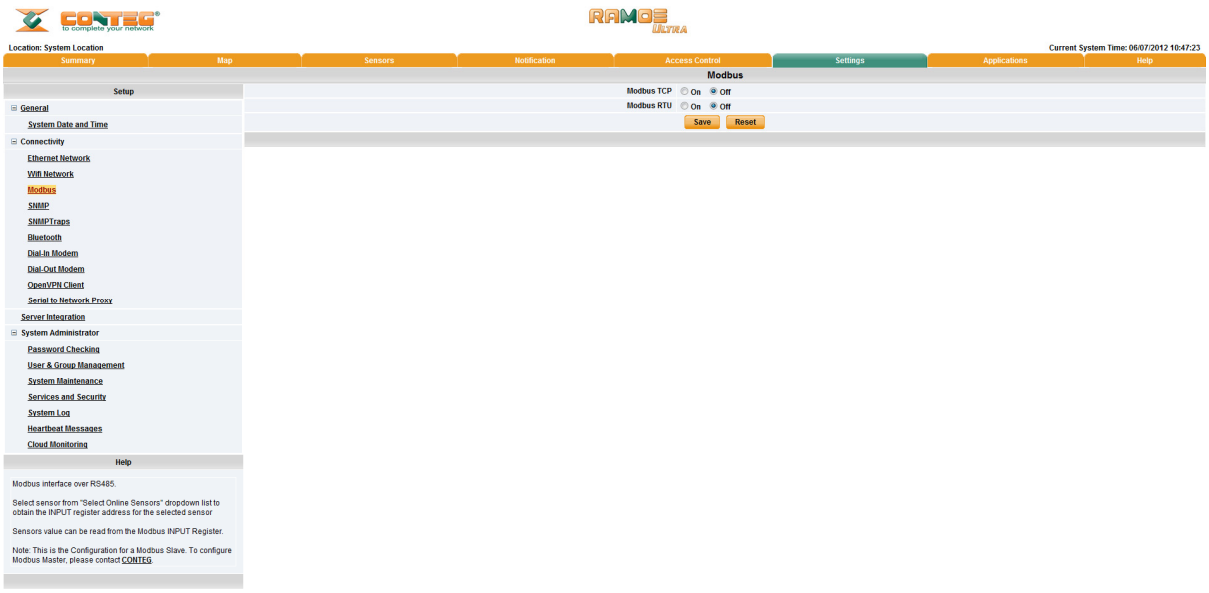
MODBUS RTU is a non-proprietary serial communications protocol that is widely used in the process control industry actuation. The hardware over which MODBUS RTU communications are performed is not defined by the protocol. MODBUS RTU is supported on RS-232, RS-422, RS-485, Ethernet and other electrical standards. The RAMOS Ultra supports RS-485 and Ethernet. The Modbus/RTU protocol defines how a “master” device polls one or more “slave” devices to read and write data in real time over the RS-485 serial data communication. Modbus/TCP, an extension of Modbus/RTU, defines how Modbus/RTU and Modbus/ASCII messages are encoded within and transported over the TCP/IP-based networks. Modbus/TCP is just as simple to implement and flexible to apply as the original Modbus/RTU. The RAMOS Ultra can represent both “master” and “slave” devices and supports both Modbus RTU and Modbus TCP protocols.

Transactions on Modbus Networks

Controllers communicate using a master–slave technique, in which only one device (the master) can initiate transactions (called ‘queries’). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. Typical master devices include host processors and programming panels. Typical slaves include programmable controllers.
The master can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a message (called a ‘response’) to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

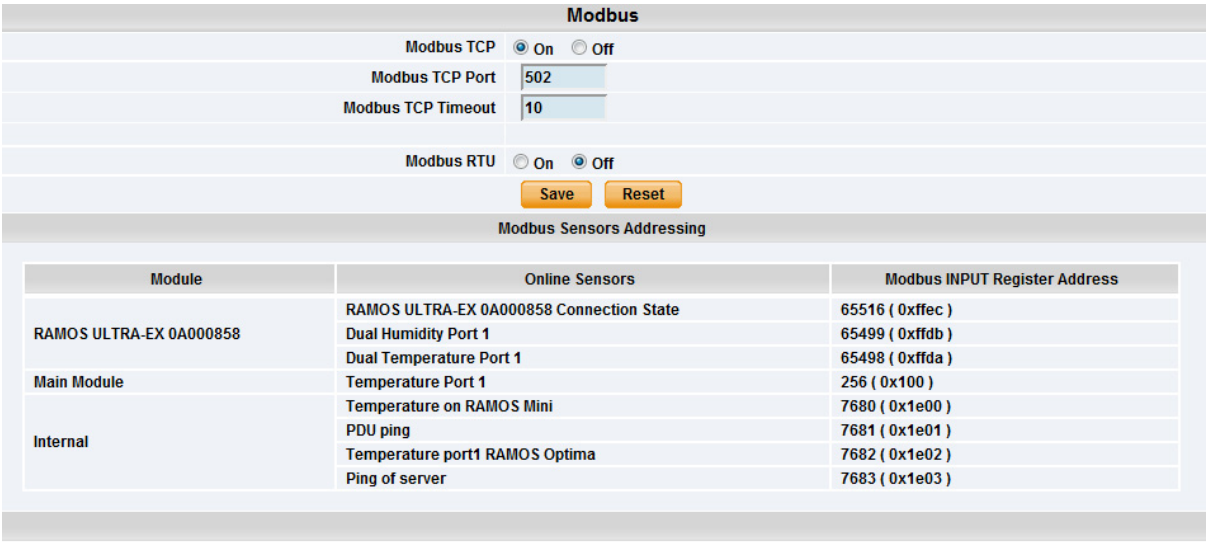
The Modbus protocol establishes the format for the master’s query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error–checking field. The slave’s response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error–checking field. If an error occurrs in receipt of the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send it as its response.

2. Setting up MODBUS on the RAMOS Ultra

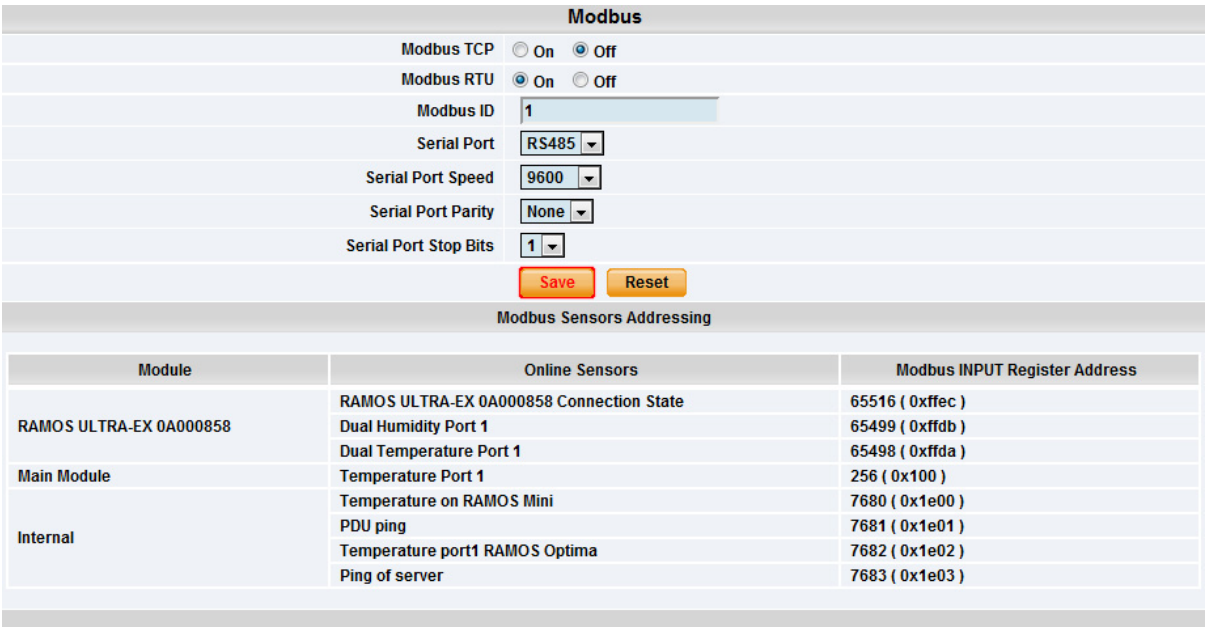


2.1. From the "Settings" tab in the RAMOS Ultras web interface, expand "Connectivity, then click on "Modbus" in the left hand column. Set the Modbus TCP and Modbus RTU to "off" for the Master device (this is the Reading unit). See screen shot above.

2.2. Now set the Modbus TCP or Modbus RTU (this depends on your connection type) setting to "on" for your Client device



2.2.1. For setting up the Modbus TCP, first set the TCP port and TCP timeout (default TCP port is 502 and TCP timeout is 10 seconds). See screen shot above.



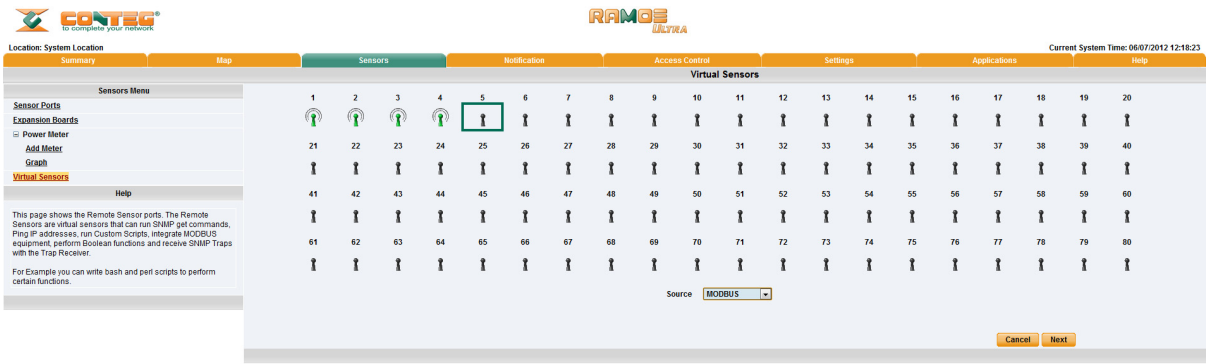
Module	Online Sensors	Modbus INPUT Register Address
RAMOS ULTRA-EX 0A000858	RAMOS ULTRA-EX 0A000858 Connection State	65516 (0xffec)
	Dual Humidity Port 1	65499 (0xffdb)
	Dual Temperature Port 1	65498 (0xffda)
Main Module	Temperature Port 1	256 (0x100)
Internal	Temperature on RAMOS Mini	7680 (0x1e00)
	PDU ping	7681 (0x1e01)
	Temperature port1 RAMOS Optima	7682 (0x1e02)
	Ping of server	7683 (0x1e03)

2.2.2. To set up the Modbus RTU

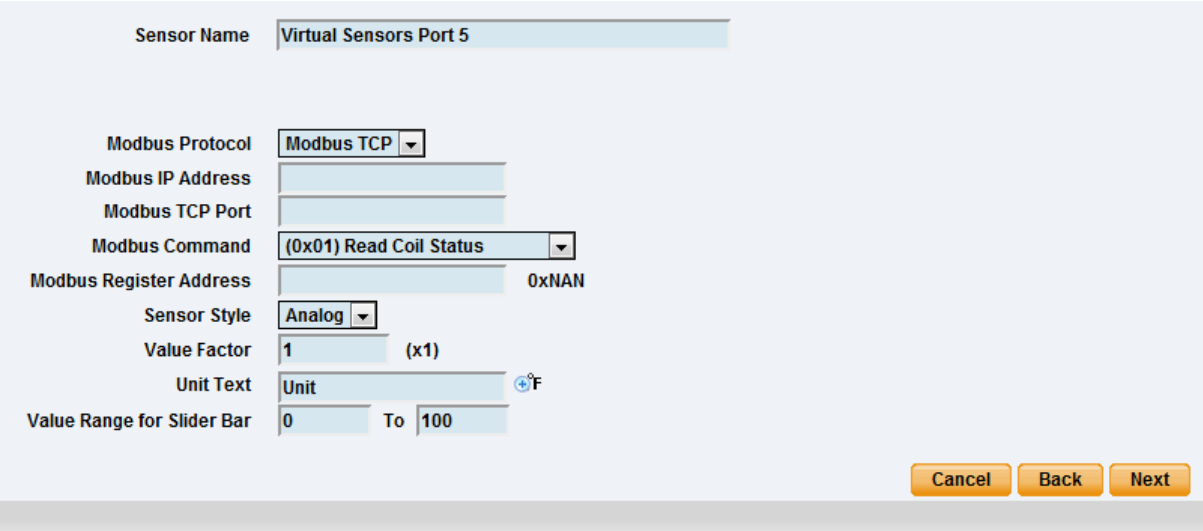
- First set the Modbus Address,
- Serial Port, Serial Port Speed,
- Serial Port Parity
- Serial Port Stop bits

(The default Modbus Address is 1, Serial Port is RS485, Serial Port Speed is 9660, Serial Port Parity is none and Serial Port Stop bit is 1)

2.3. The Modbus Sensors Addressing is now showing the Modbus INPUT Register Address of each sensor that is online on client device. Example, the Modbus INPUT Register Address of the Temperature on port 1 is 256 (0x0100)

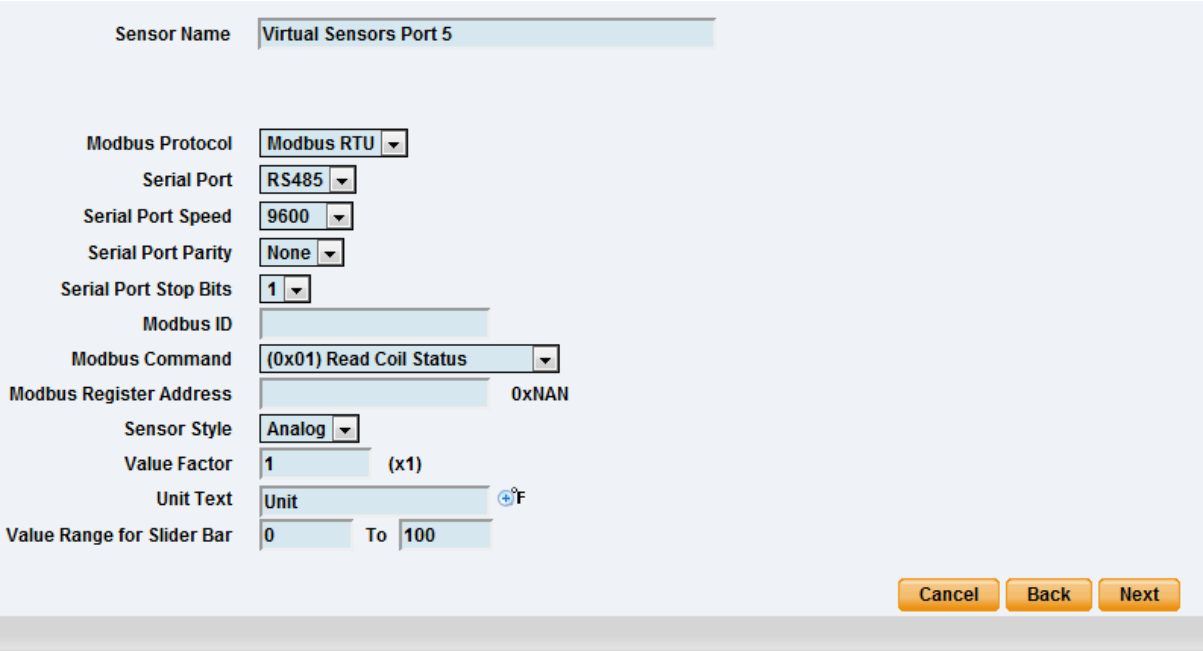


2.4. From the “Sensors” tab (on Master device) select virtual sensors, then set the “Source” to Modbus and then click “Next”. See screen shot above



2.4.1. For the Modbus TCP

- Set the “Modbus Protocol” to Modbus TCP
- Enter the Client device IP address in “Modbus IP Address”
- Enter the TCP port in “Modbus TCP Port”
- Set the “Modbus command: to get the sensor value or get the sensor status, (0x03 Read Holding Register for getting the sensor status and 0x04 Read Input Register for getting the sensor value).
- Now set the Modbus INPUT Register Address value to Modbus Register Address (see #3 above for the Modbus INPUT Register Address Value) Example; Set 258 for the read value of the Temperature on port 3 is 258). See screen shot above

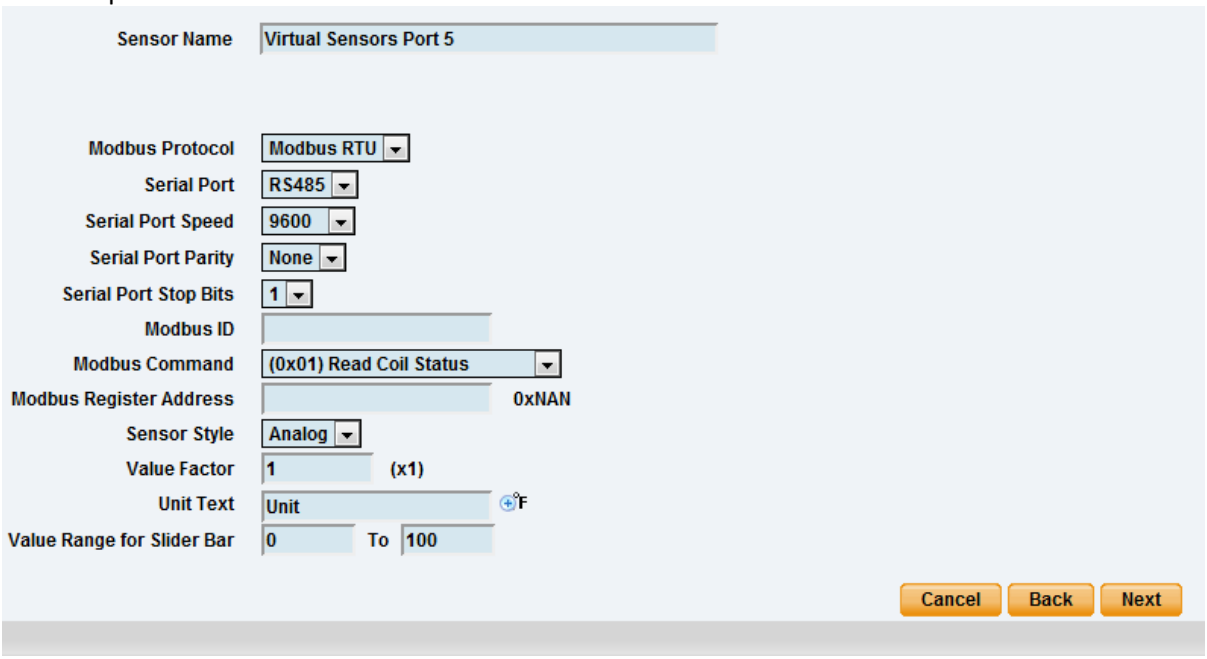


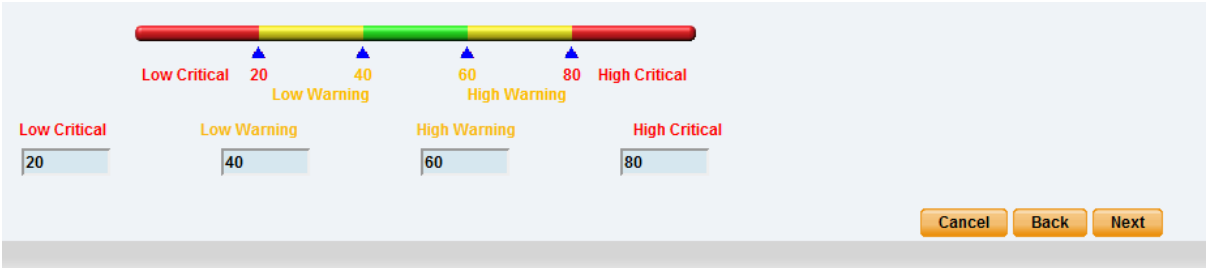
2.4.1.1. For Modbus RTU

- Set “Modbus Protocol” to Modbus RTU
- Set “Serial Port” to RS485 or other (this setting depends on the connection you are using)
- Set “Serial Port Speed” (this setting should be the same as shown in number 2.2 above)
- Set “Serial Port Parity” (this setting should be the same as shown in number 2.2 above)
- Set “Serial Port Stop Bits” (this setting should be the same as shown in number 2.2 above)
- Input the same “Modbus Address” value from step number 2.2 into “Modbus Slave ID”
- Select the Modbus command for getting the sensor value or the sensor status (0x03 Read Holding Register for getting the sensor status and 0x04 Read Input Register for getting the sensor value).
- Input the Modbus INPUT Register Address value into “Modbus Register Address”. An example of the Modbus INPUT Register Address Value can be found in step number 3 (Example input of 258 for the read value of the Temperature on port 3 is 258)

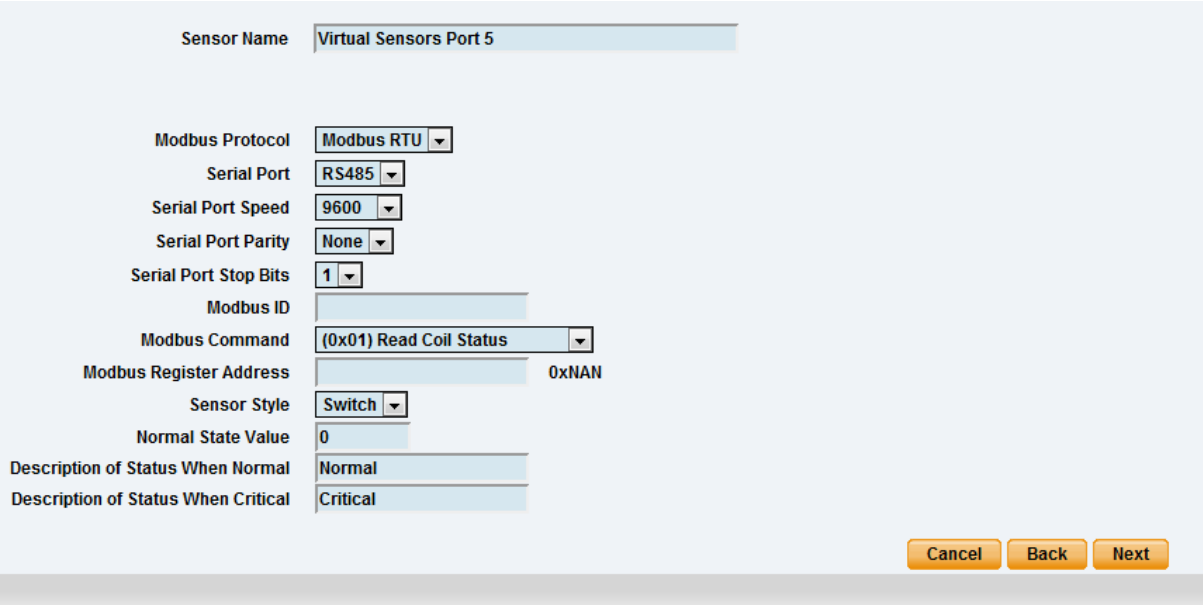
2.4.2. Sensor Style. Set the Sensor Style to Analog

2.4.2.1. Analog Style “Value Factor”, set “Value Factor” for multiple with normal value (Example; if raw value is 1234 and needs to show a value to 12.34, then this should be set to x0.01. Default is x1). Input the Unit of sensor value into the “Unit Text” field, or click to input a special character





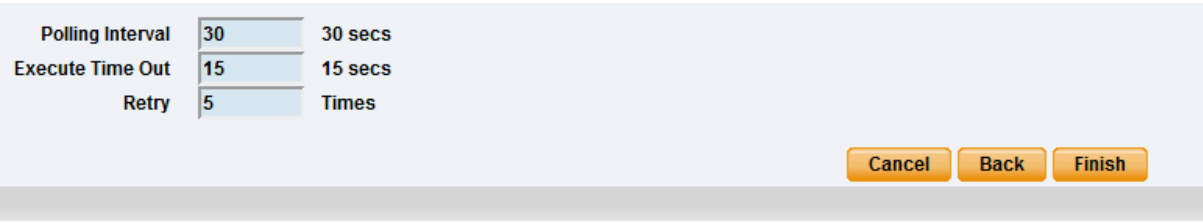
Input the Minimum and Maximum of the sensor values into the “Value Range for Slider Bar” field then click “Next”.
Set the Low Critical, Low Warning, High Warning and High Critical thresholds as shown in the screen shot above, then click “Next” again.



2.4.3.2. Switch Style

- Set the “Sensor Style” to Switch
- Input the Normal value of the sensor into “Normal State Value”.
- Input the Description of Normal and Critical status into “Description of Status When Normal” and “Description of Status When Critical”

(See screen shot on previous page)



2.5. Set Time to Polling

- Input the time to polling into “Polling Interval” (Default is 15 secs)
- Input “Execute Time Out” (Default is 10 secs) and “Retry” (Default is 3 times)
- Click the “Finish” button

3. Return value from Holding Register

The following is a list of what each of the return values from the Holding Register represent

1 No Status (Sensor has not been initialized)
2 Normal
3 High Warning
4 High Critical
5 Low Warning

6 Log Critical
7 Sensor Error
8 Switch Low (Output)
9 Switch High (Output)

For the INPUT Register
If the sensor is OFFLINE, the INPUT register will return 32767
If the sensor is No Status or Sensor Error, the INPUT register is -32768