

# Using ModbusTCP and ModbusRTU and the CommStats Function Block

## Abstract

Why and how you would want to use this functionality of Red Lion Work Bench How you get the CommStats UDFB in your program easily.

## Products

Any product either Red Lion or another company that uses ModbusTCP or ModbusRTU and you are trying to talk to it with one of the RTUs, either the ST-IPm-8460/VT-mIPm-248-D/VT-mIPm-138-D.

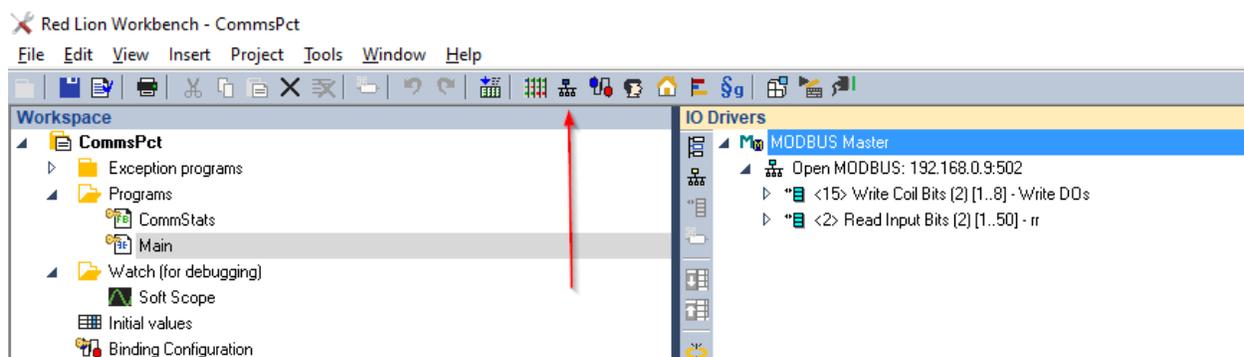
**Use Case: If you are wanting to keep track of your communication performance**

## Required Software

Red Lion Work Bench

## Using Red Lion Work Bench & CommStats Function Block

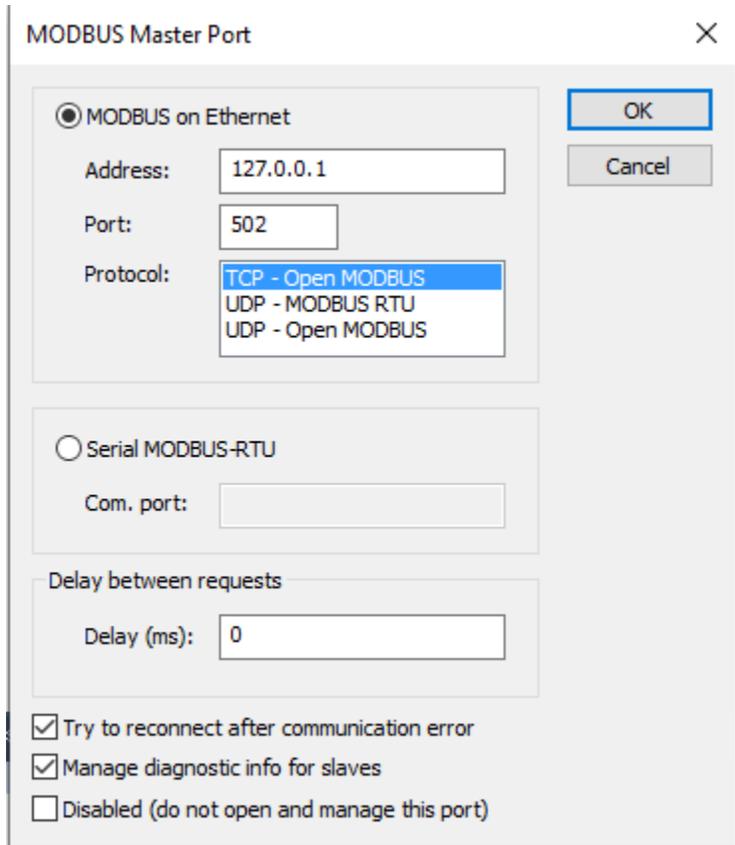
Setting up the Open Fieldbus Configurator



Click on the Icon pointed at below to open up the Open Fieldbus Configurator.

This box pops up.

Choose TCP-Open Modbus



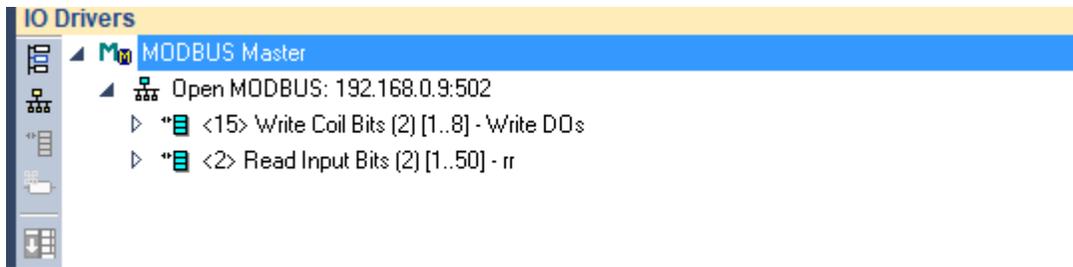
The image shows a dialog box titled "MODBUS Master Port" with a close button (X) in the top right corner. The dialog is divided into three main sections. The first section, "MODBUS on Ethernet", is selected with a radio button. It contains three input fields: "Address" with the value "127.0.0.1", "Port" with the value "502", and "Protocol" with a dropdown menu showing "TCP - Open MODBUS" selected, "UDP - MODBUS RTU", and "UDP - Open MODBUS". The second section, "Serial MODBUS-RTU", is unselected and contains a "Com. port:" input field. The third section, "Delay between requests", contains a "Delay (ms):" input field with the value "0". At the bottom, there are three checkboxes: "Try to reconnect after communication error" (checked), "Manage diagnostic info for slaves" (checked), and "Disabled (do not open and manage this port)" (unchecked). On the right side of the dialog, there are "OK" and "Cancel" buttons.

Put in the IP address of the unit

Put in the port number if different than 502

Put in the delay between requests, this is the actual polling speed for this port

## Building Comms to your remote device

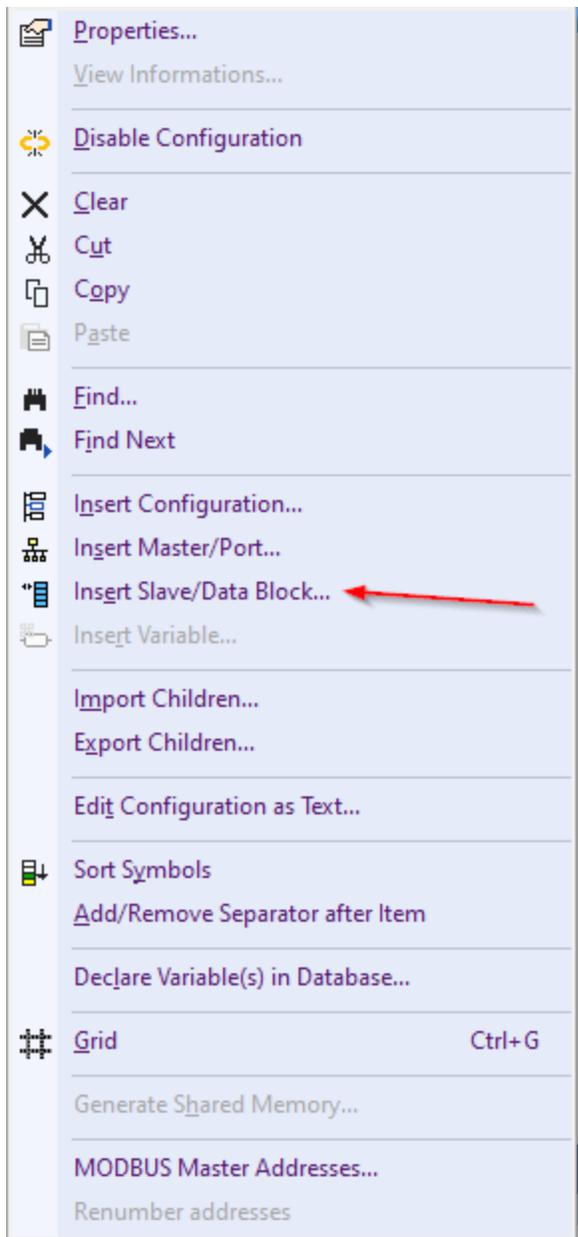


As of right now we have the Modbus Master Driver selected

The IP address and port number of the device set

Now we set up the registers we wish to poll...

After double clicking on the "Open MODBUS" connector like what is shown above, the box below will pop up and, in that drop, down list you will select what the arrow is point too.



MODBUS Master Request ×

Request

Description:

Slave/Unit:  ←

OK

Cancel

MODBUS Request

<1> Read Coil Bits

<2> Read Input Bits

<3> Read Holding Registers

<4> Read Input Registers

Data block

Base address:

Nb items:

Activation

Periodic:  ms  (on error)

On call

On change ←

Misc.

Timeout:  ms Poll Speed

Nb trials:

Declare variables

Prefix:

From:

V1 ... V1

1. Put in a description
2. Select your data type
3. Select if you are using a 0 or 1 as your base address
4. Activation
  - a. Periodic- typical poll/write speed in milliseconds
  - b. On call-using a triggered read
  - c. On change-report by exception
    - i. Do not recommend On change unless you are sure this will work for your application.

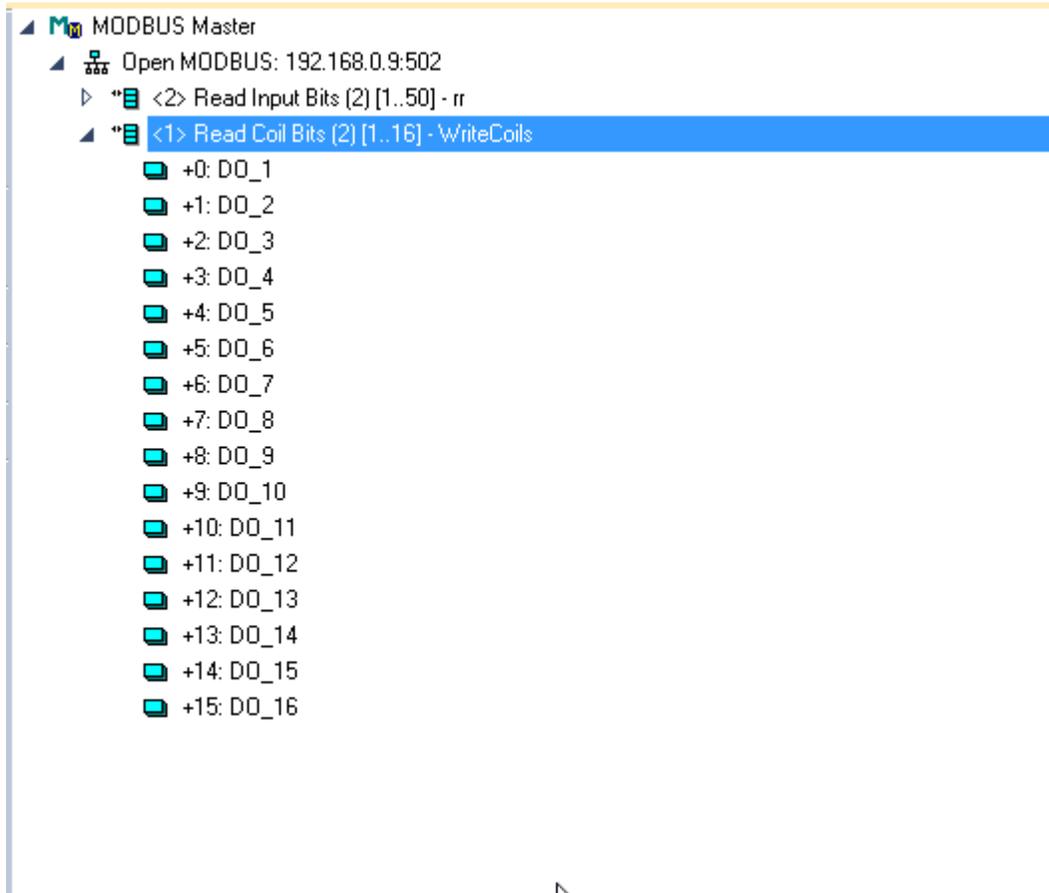
**This is a typical setup screen.**

The screenshot shows a dialog box titled "MODBUS Master Request" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Request:** Contains a "Description:" field with the text "WriteCoils" and a "Slave/Unit:" field with the value "2". To the right of this section are "OK" and "Cancel" buttons.
- MODBUS Request:** A list box containing four options: "<1> Read Coil Bits" (highlighted in blue), "<2> Read Input Bits", "<3> Read Holding Registers", and "<4> Read Input Registers".
- Data block:** Contains a "Base address:" field with the value "1" and a "Nb items:" field with the value "1".
- Activation:** Contains three radio button options: "Periodic:" (selected), "On call", and "On change". The "Periodic:" option has a "100" field followed by "ms" and a "0" field followed by "(on error)".
- Misc.:** Contains a "Timeout:" field with the value "3000" followed by "ms" and a "Nb trials:" field with the value "1".
- Declare variables:** A checked checkbox labeled "Declare variables". Below it is a "Prefix:" field with the value "DO\_%" and a dropdown menu showing "BOOL". Below that is a "From:" field with the value "1". At the bottom, the text "DO\_1 ... DO\_1" is displayed.

**You can see right here that by using the % mark the software will automatically create the IO point number for you. You have to supply the nomenclature before the % mark for this to work.**

**This I show it shows up once things are filled out.**



**Now comes the part where we put in the communication statistics tags.**

Right click on the communications poll/write you wish to monitor and this box will appear.

The screenshot shows the 'IO Drivers' software interface. On the left, a tree view under 'MODBUS Master' shows a connection to 'Open MODBUS: 192.168.0.9:502'. Underneath, there are two communication tasks: '<2> Read Input Bits (2) [1..50] - rr' and '<1> Read Coil Bits (2) [1..16] - WriteCoils'. The 'WriteCoils' task is expanded to show 16 individual coil addresses from '+0: DO\_1' to '+15: DO\_16'. On the right, a table lists the configuration parameters for the selected task.

Name	Value
Request	<1>
Slave/Unit	2
Address	1
Nb Item	16
Activation	Peri
Period (ms)	100
Period on error	0
Timeout (ms)	300
Number of trials	1
Description	Writ

In the foreground, a 'MODBUS Variable' dialog box is open. It has a 'Symbol' field containing '???' and 'OK'/'Cancel' buttons. Below are two radio buttons: 'Status / Control' (unselected) and 'Data exchange' (selected). Under 'Data exchange', there are fields for 'Offset' (0), 'Mask' (FFFF), and 'Storage' (Default). There are also 'Range' fields for 'Min' and 'Max', and 'Signal' fields for 'Min' and 'Max'.

Timeout (ms)	3000
Number of trials	1
Description	WriteC

MODBUS Variable

Symbol  
DOWrite\_Successful\_Writes

OK  
Cancel

Tag Name

Status / Control  
Success counter

Data exchange  
Offset: 0  
Mask: FFFF  
Storage: Default

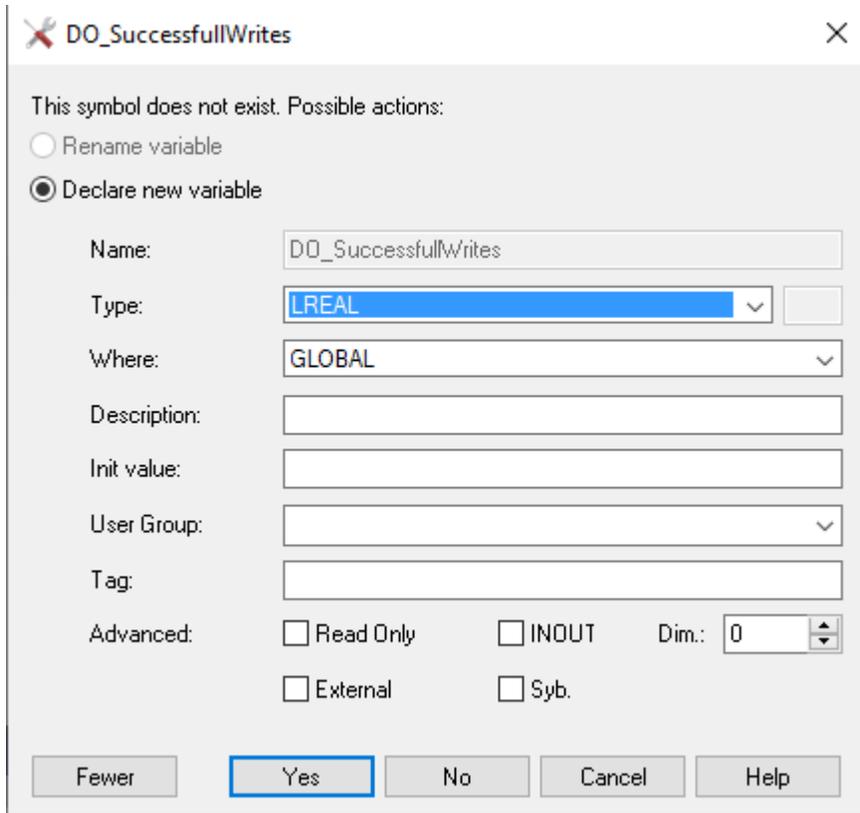
Range  
Min:  
Max:

Signal  
Min:  
Max:

Select this from drop down list.

0 Drivers CommStats

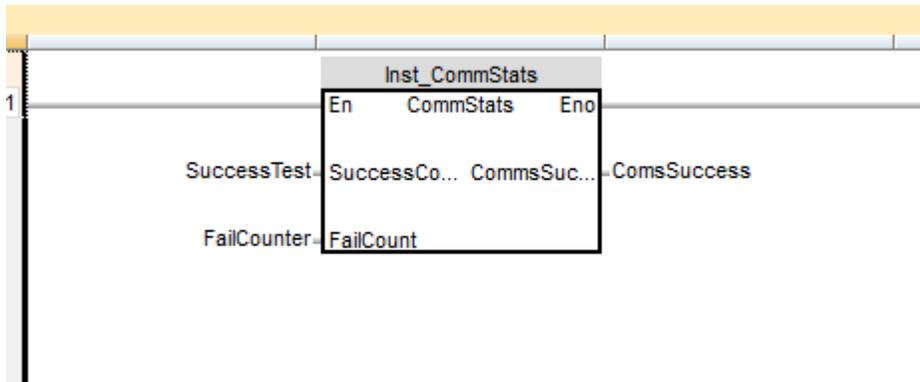
**You have just created your successful writes monitoring tag, be sure to make this a float so it can work with our CommStats block as there is internal floating-point math involved.**



**Make this tag global and an LReal or Long Real/64bit real also known as a “double precision float”.**

**Now do the same for the Write Fails**

Now go out and download off of our website the User Defined Function Block (UDFB) CommStats. Attach these two tags to it and one more for the output. Now you have the comms stats reporting back in a percentage basis...



## Why would you use this block?

- You want to monitor the comms to all remote devices from the processor
- You want to generate alarms based on the success percentages
- You are having issues with your overall connections and want to know which remote devices might be causing it.

## Disclaimer

It is the customer's responsibility to review the advice provided herein and its applicability to the system. Red Lion makes no representation about specific knowledge of the customer's system or the specific performance of the system. Red Lion is not responsible for any damage to equipment or connected systems. The use of this document is at your own risk. Red Lion standard product warranty applies.

## Red Lion Technical Support

If you have any questions or trouble contact Red Lion Technical Support by clicking [here](#) or calling 1-877-432-9908.

For more information: <http://www.redlion.net/support/policies-statements/warranty-statement>