Polling Data from MODBUS RTU

Using Serial Protocol

APPLICATION NOTE

AUG-0080-00 2.0 en-US ENGLISH

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Industrial Networks of any inaccuracies or omissions found in this document. HMS Industrial Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Industrial Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Industrial Networks and is subject to change without notice. HMS Industrial Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

Table of Contents

Page

This page intentionally left blank

1 Preface

1.1 About This Document

This document explains in a few steps how the Flexy can poll data registers from a PLC based on MODBUS RTU.

For additional related documentation and file downloads, please visit www.ewon.biz/support.

1.2 Document history

Version	Date	Description
1.0	2015-08-18	First release
2.0	2019-09-03	Major update – reformulation, new document organization

1.3 Related Documents

Document	Author	Document ID
Polling Data from MODBUS TCP Using Ethernet Protocol	HMS	AUG-0059-00
Ewon Flexy Base Units	HMS	IG-0014-00
FLA 3301 - Serial Ports Extension Card	HMS	IG-0016-00
Ewon Flexy 205	HMS	IG-0028-00
Flexy Family Reference Guide	HMS	RG-0008-00

1.4 Trademark Information

Ewon® is a registered trademark of HMS Industrial Networks SA. All other trademarks mentioned in this document are the property of their respective holders.

2 Objectives

The objective of this document is to explain how the Flexy can poll data registers out of one or more PLCs using MODBUS RTU.

Polling PLC data registers implies the following steps :

1. Link the Flexy with the PLC;
2. Configure theFlexy IO server;
3. Create tags in the Flexy;
4. Monitor tags.

3 Requirements

3.1 Hardware

To follow this guide, you need:

- A computer suitable to connect to the Ewon Flexy;
  - → From a computer running a web browser, you will configure the IO server in the Flexy to poll different types of PLC data registers.
  - → You access the Flexy web server either by using (one of) its local LAN port(s) or by another type of access such as VPN IP address.
- A device acting as a MODBUS RTU slave such as a PLC, remote IO, or RTU.
  - → Connection between the Ewon Flexy and the PLC must be done through serial. The serial port could be on the base unit as in the Flexy 202 or on a Flexy extension card: FLA 3301 – 2 serial ports. For polling tags over Modbus TCP with an Ethernet connection, see “Polling Data from MODBUS TCP Using Ethernet Protocol” from *Related Documents, p. 3*.
  - → The device will have its registers read by the tags configured in the IO server of the Flexy.

3.2 Software

3.2.1 eBuddy

The Flexy is configured through its web server. All you need is a standard web browser software such as Google Chrome® or Mozilla Firefox®.

Additionally, we suggest downloading the **eBuddy** utility on our website.

This utility can list all the Ewon Flexy on your network and change the default IP address of a Flexy to match your LAN IP address range. With eBuddy you can also easily upgrade the firmware of your Flexy if required.

3.2.2 Ewon Flexy Firmware

The screenshots of this guide reflect firmware version 13.3s0, but you can expect the basic principles to remain the same in earlier/later versions.

A simple way to upgrade the Flexy firmware is to use eBuddy.

3.3 PLC Protocol Compatibility

The Flexy supports standard Modicon Modbus RTU (Master/Client).

However, Modbus ASCII is not supported.

4 Implementation Steps

4.1 Link the Flexy and the PLC

Modbus RTU can be performed either on an RS232, RS485 or RS422 interface.

Through RS232, the Flexy serial connector acts as a PC serial connector so a null-modem (cross-over) cable is used most of the time. Refer to the cabling schematic of the Flexy and your device to get the right cable.

- Set the serial port dip switch of the Flexy to the right mode according to your Modbus device serial port type: RS232, RS422 or RS485.

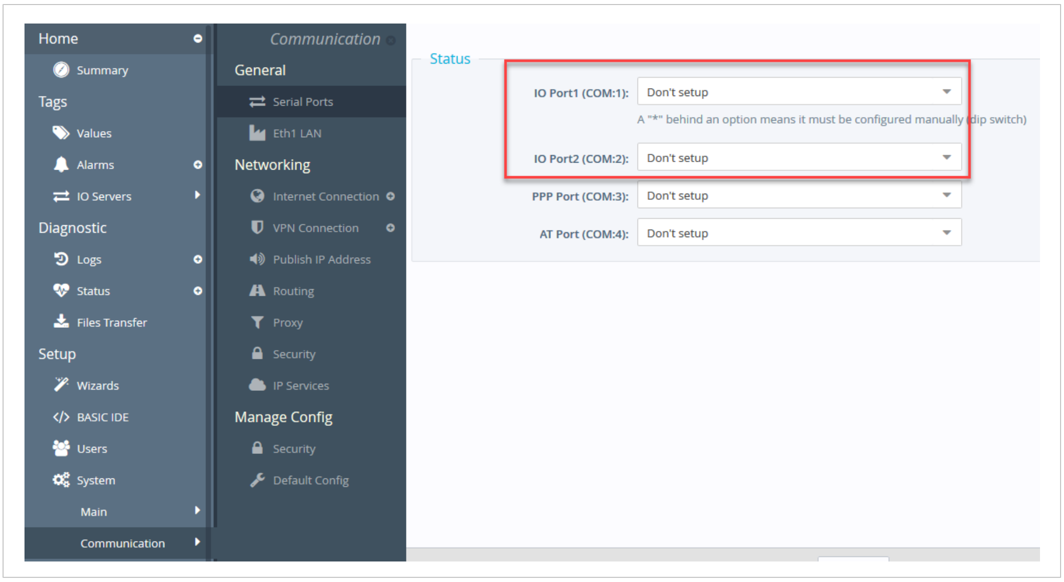
  For the FLA 3301 – 2 Serial Ports Extension card:

  – Port S2 is fixed RS232 mode.

  – Port S1 is configurable with dip switches. The proper configuration is printed on the extension card to the left of the switches.

You must also configure the serial interface inside the Flexy:

1. Go to the Flexy web interface.
2. Browse to *Setup* ► *System* ► *Communication* ► *General* ► *Serial ports*.
3. Set the different serial ports you need according to the RS* protocol used by your PLC.



IO Port1 (COM:1): Don't setup

A "*" behind an option means it must be configured manually (dip switch)

IO Port2 (COM:2): Don't setup

PPP Port (COM:3): Don't setup

AT Port (COM:4): Don't setup

Fig. 1 Serial port configuration on Ewon Flexy web interface

The pinout for the serial port on the Flexy is available in the Installation Guides for the Flexy and for the FLA 3301 serial port extension card, depending if you use the serial port of the Flexy base unit or of theFlexy extension card. For more information, refer to *Related Documents, p. 3*.

4.2 Configure the IO Server

1. Go to the Flexy's web interface.
2. Select the ***IO Servers*** menu option.

3. Select the ***IO Server*** corresponding to your PLC type, in this case ***MODBUS***.

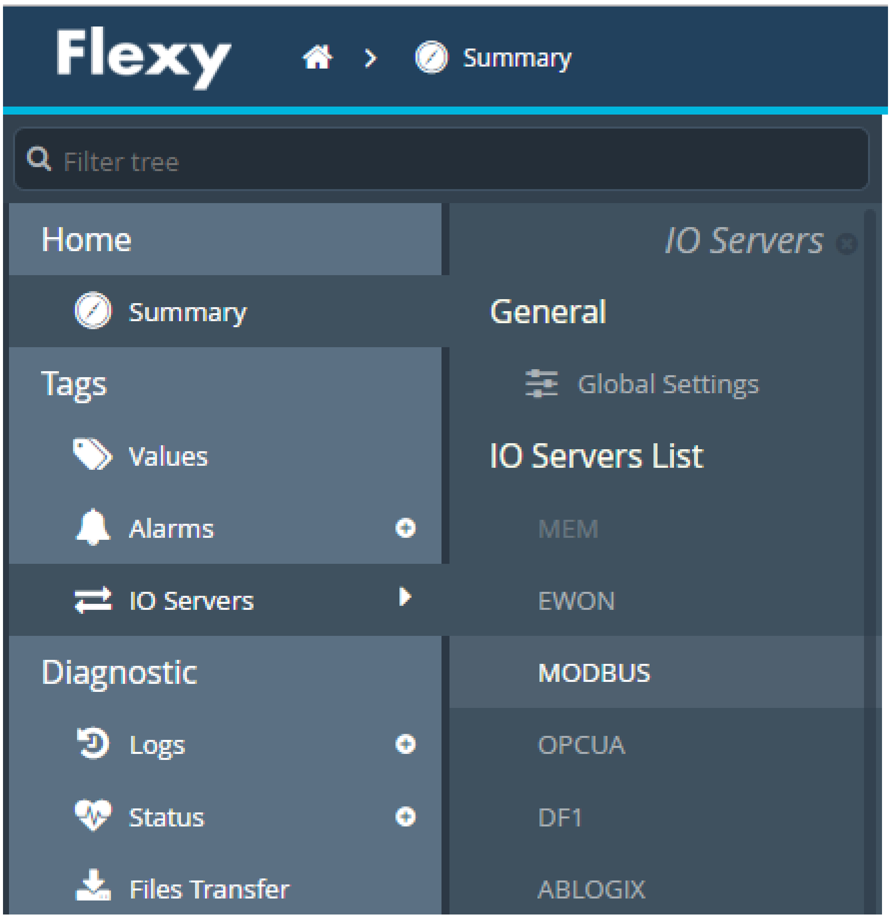


Fig. 2 **MODBUS IO server menu**

4. Set the settings of the IO server with the parameters as defined in the PLC you want to poll:

   a. Select the ***COM port*** to which the PLC is connected.

      The COM port numbering depends on the Flexy model and the number of serial port extension cards installed in the Flexy:

      – For a Flexy 205 with a single FLA 3301 (2 Serial Ports) extension card: port “S1” is *COM 1* and “S2” is *COM 2*.

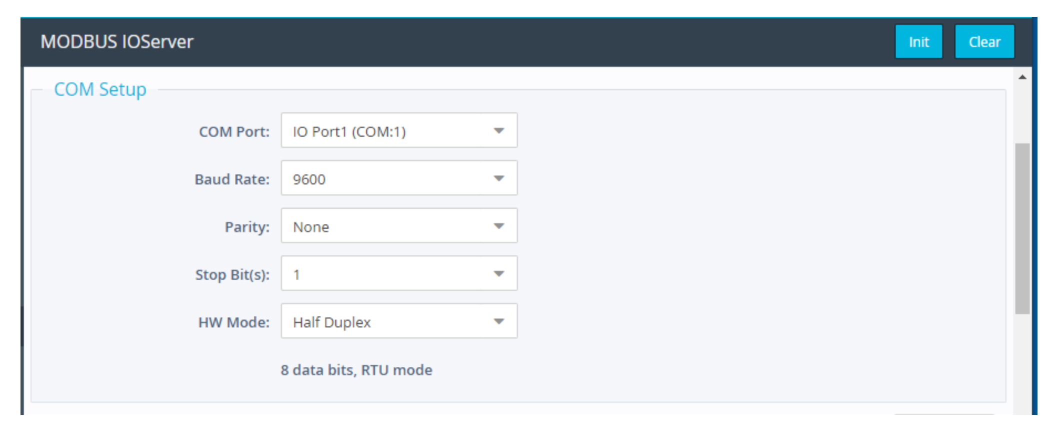
      – For other configurations, see the Installation Guides for your Flexy model and extension cards from the *Related Documents, p. 3*.

   b. Set the ***Baudrate***.

   c. Set the ***Parity***.

   d. Set the ***Stop Bit***.

   e. Set the ***HW Mode***.



COM Port: IO Port1 (COM:1)

Baud Rate: 9600

Parity: None

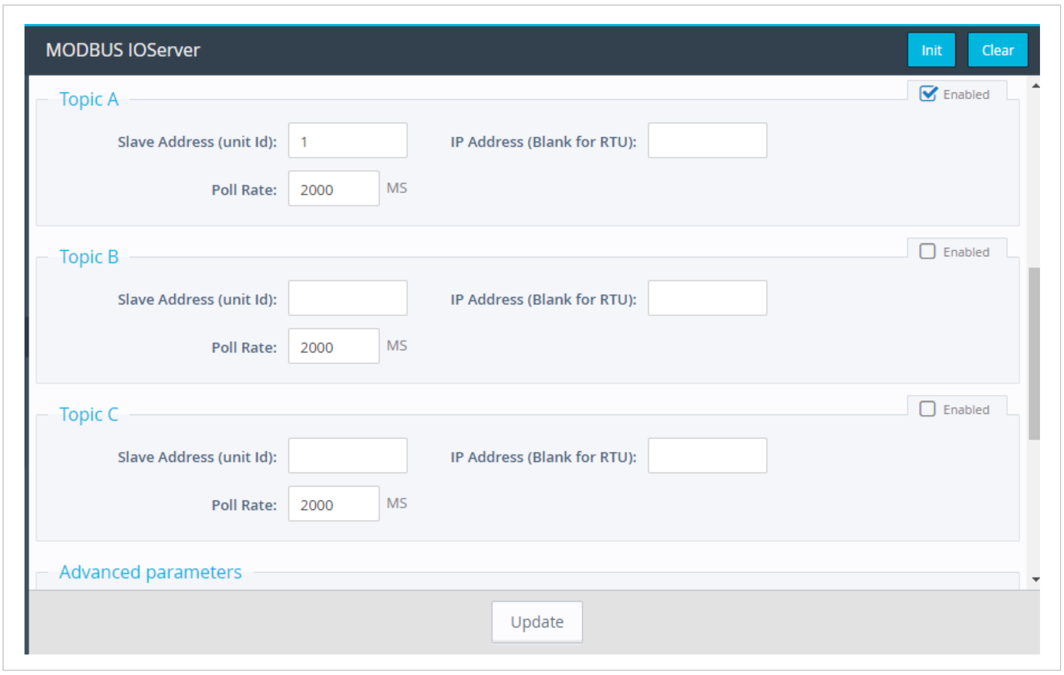
Stop Bit(s): 1

HW Mode: Half Duplex

8 data bits, RTU mode

Fig. 3 MODBUS RTU IO server settings

5. Define at least one ***Topic*** in the IO server configuration page to poll data registers out of your PLC.
   Topics are meant to allocate common properties to a group of tags. Properties include ***Enable/Disable*** polling, ***Poll Rate*** and optionally ***Device Address***.



Topic A ☑ Enabled
Slave Address (unit Id): 1
IP Address (Blank for RTU):
Poll Rate: 2000 MS

Topic B □ Enabled
Slave Address (unit Id):
IP Address (Blank for RTU):
Poll Rate: 2000 MS

Topic C □ Enabled
Slave Address (unit Id):
IP Address (Blank for RTU):
Poll Rate: 2000 MS

Fig. 4 MODBUS IO server settings

6. Enable at least ***Topic A*** by ticking the appropriate box.
7. Enter a valid ***Slave Address*** for the PLC.

   Even though there are only three topics, the Flexy can support polling from more than three PLCs on the same IO server.

   To poll from more than three PLCs, do not specify the device address in the topic. Instead, include the device address as part of the tag address — when configuring the tag itself — as described later in this document.
8. Define the refresh rate in ms (milliseconds) inside the ***Poll Rate*** which is applicable to all data registers that will be associated with this topic.

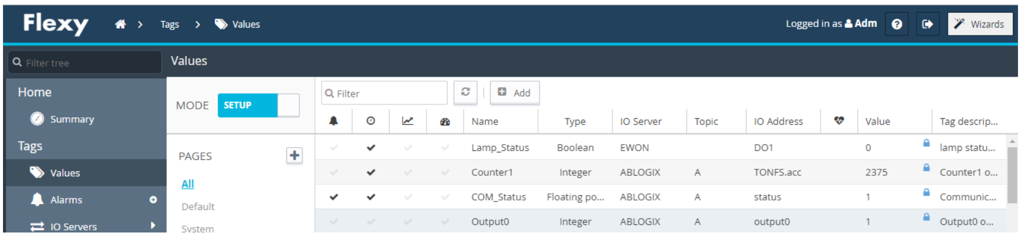
   The default value is 2000 ms (2 seconds).

   The polling rate specified here applies to all tags associated with this topic. If you have tags that need to refresh at different rates, enable and configure multiple topics.
9. Click on ***Update*** to save your settings.

4.3 Create Tags in the Flexy

1. Select the ***Values*** option from the Tags menu.

2. Switch the “Mode” to **Setup**.

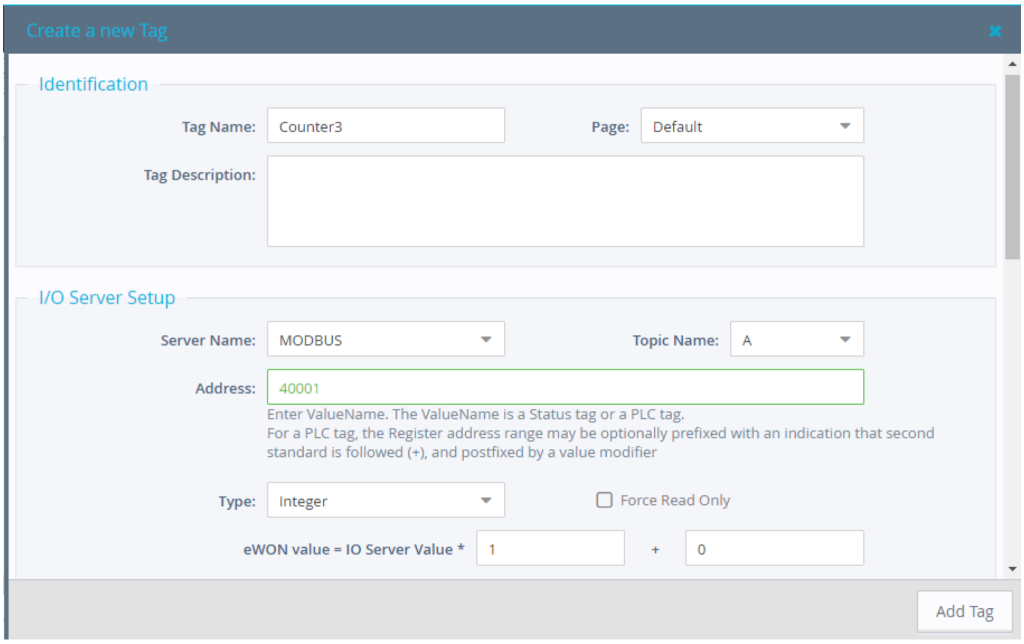


				Name	Type	IO Server	Topic	IO Address		Value	Tag descrip...
	✓			Lamp_Status	Boolean	EWON		DO1		0	lamp statu...
	✓			Counter1	Integer	ABLOGIX	A	TONFS.acc		2375	Counter1 o...
✓	✓			COM_Status	Floating po...	ABLOGIX	A	status		1	Communic...
				Output0	Integer	ABLOGIX	A	output0		1	Output0 o...

Fig. 5 **MODBUS IO server — Tag setup**

3. Click the ***Add*** button to open the tag configuration window.

4. Enter the ***parameters of the tag*** you want to create.



Create a new Tag

Identification

Tag Name: Counter3

Page: Default

Tag Description:

I/O Server Setup

Server Name: MODBUS

Topic Name: A

Address: 40001

Enter ValueName. The ValueName is a Status tag or a PLC tag.
For a PLC tag, the Register address range may be optionally prefixed with an indication that second standard is followed (+), and postfixed by a value modifier

Type: Integer

☐ Force Read Only

eWON value = IO Server Value * 1 + 0

Fig. 6 **MODBUS IO server — Tag setup 2**

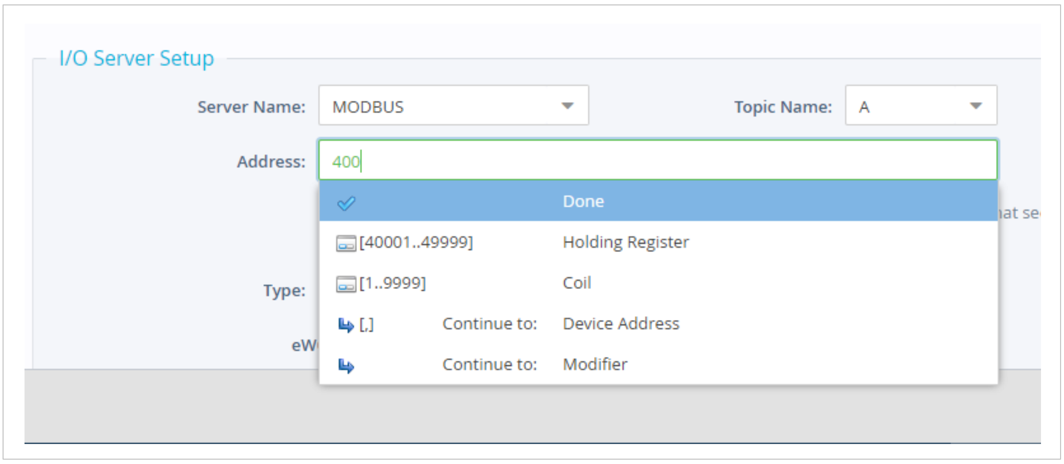
5. Enter a ***Tag Name***.
Free text, no spaces, no symbols -, =, %, $, @, # etc.

6. Enter a ***Description***.
Free text.

7. Select ***MODBUS*** as IO server.

8. Enter the ***PLC register*** in the “Address” field which will be polled from the PLC.

   As the address is entered, a tag helper appears to help properly format the tag address.



I/O Server Setup

Server Name: MODBUS

Topic Name: A

Address: 400

Done
[40001..49999] Holding Register
[1..9999] Coil
[,] Continue to: Device Address
Continue to: Modifier
Type:
eW
at se

Fig. 7 MODBUS IO server — Tag setup 3

Example of tag address syntax

Address	Description
40001	Reading a “Holding Register” using the format “Word” at address 1.
+320500F	Reading an “Analog Input” using the format “Float” at address 20500.
1	Reading a “Coil” at address 1.

The register configured in the Flexy must exist in the PLC. If you enter a wrong address, the tag creation will be rejected and an error message will be displayed.

For more information on data register ID syntax, see *MODBUS Tag Address Syntax, p. 17*.

9. Enter a ***Topic Name***: A, B or C.

   The topic must have been configured in the IO server page (see *Configure the IO Server, p. 6*).

10. ***Type***: The data type of the tag such as *Floating Point* or *Boolean*.

    The ***Automatic*** option lets the Flexy decide the format depending on the IO server register/modifier type.

11. [Optional] The remaining fields are mostly left with their default value:

    – ***Force Read Only***: Unchecked is the default.

      When it is checked, users will not be able to change a value in ***View*** mode on the ***Values*** page.

The tag remains read/write for commands written in the embedded BASIC script program or on custom webpages.

    – ***Ewon value***: Defaults are **1+0*.

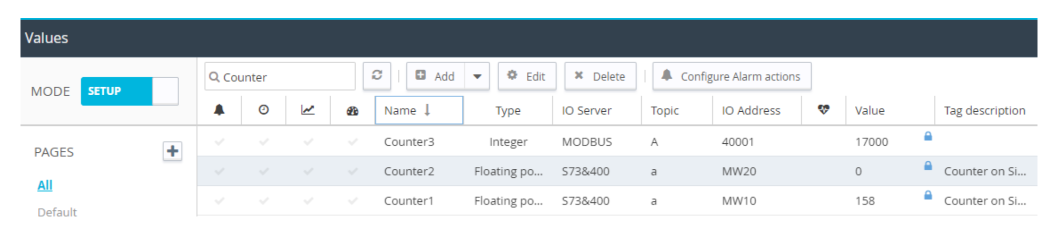
      This field applies a ***scale factor*** and an ***offset*** to the raw value coming from the IO server.

      The scale factor and offset are float values. Negative values are accepted.

      TAGval = IOSERVERval * scale factor + offset.

12. Click the ***Add*** button when your tag configuration is complete.

If everything is OK the new tag appears in the tag list.



Name	Type	IO Server	Topic	IO Address	Value	Tag description
Counter3	Integer	MODBUS	A	40001	17000	
Counter2	Floating po...	S73&400	a	MW20	0	Counter on Si...
Counter1	Floating po...	S73&400	a	MW10	158	Counter on Si...

Fig. 8 MODBUS IO server — Tags list

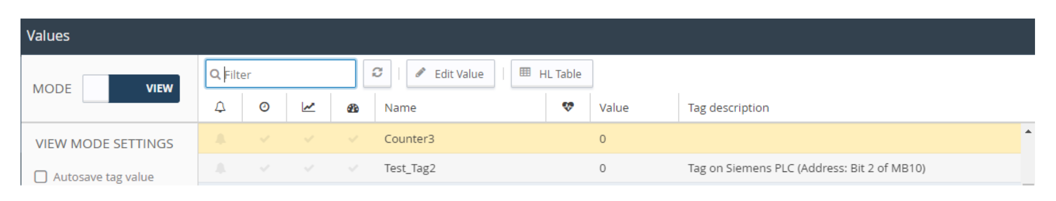
You can repeat the same sequence for any other tags. If you need to create new tags that have almost the same properties as an existing tag in the list, select the source tag and click the ***Add as selected*** option.

All properties of the existing tag will be copied in the new tag creation wizard. Copied properties include the tag name. Since the tag name must be unique, make sure you change the name of the new tag.

It is the first selected tag that will be copied if more than one single tag is selected in the list.

4.4 Monitor Tags

You can change the mode to ***View*** to monitor tags values and their status.



Name	Value	Tag description
Counter3	0	
Test_Tag2	0	Tag on Siemens PLC (Address: Bit 2 of MB10)

Fig. 9 MODBUS IO server — Monitor tags

This page shows the tags and their last polled values as well as alarm information and logging information for tags with alarming and logging enabled.

The page refreshes automatically at the rate set on the bottom of the page.

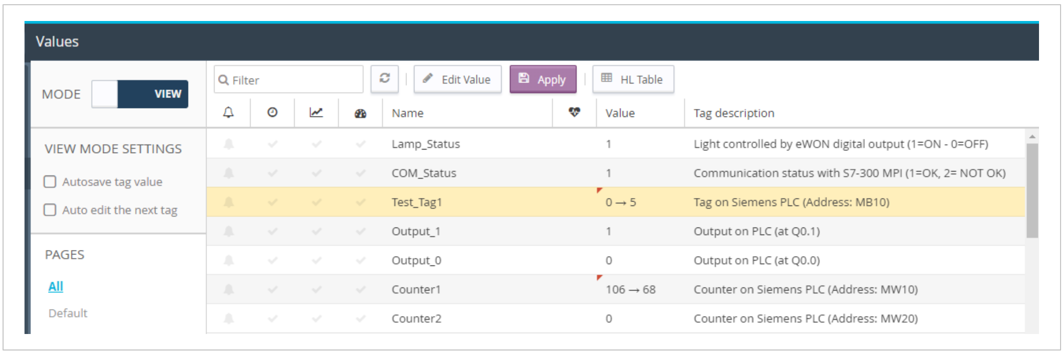
On this page, you can sort, filter, and search for tags to easily find specific tags.

Information about configuring additional tag features is available in the *Flexy Family Reference Guide*, see *Related Documents, p. 3*.

You can change the value of tags that are configured as read/write (unless the box *Force Read Only* was ticked in the tag creation wizard).

To change the value of a tag:

1. Select the tag and press the ***Edit Value*** button, or double click its current value.
2. Enter the new value.
3. Click the ***Apply*** button to update all of the edited tags.



Values

MODE VIEW

VIEW MODE SETTINGS

□ Autosave tag value

□ Auto edit the next tag

PAGES

All

Default

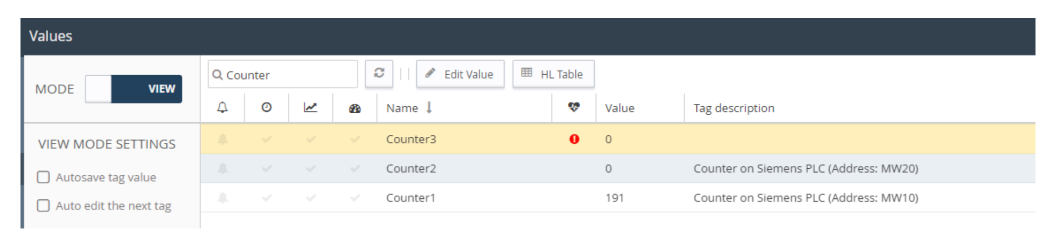
🔔	🕒	📈		Name	♥	Value	Tag description
🔔	✓	✓	✓	Lamp_Status		1	Light controlled by eWON digital output (1=ON - 0=OFF)
🔔	✓	✓	✓	COM_Status		1	Communication status with S7-300 MPI (1=OK, 2= NOT OK)
🔔	✓	✓	✓	Test_Tag1		0 → 5	Tag on Siemens PLC (Address: MB10)
🔔	✓	✓	✓	Output_1		1	Output on PLC (at Q0.1)
🔔	✓	✓	✓	Output_0		0	Output on PLC (at Q0.0)
🔔	✓	✓	✓	Counter1		106 → 68	Counter on Siemens PLC (Address: MW10)
🔔	✓	✓	✓	Counter2		0	Counter on Siemens PLC (Address: MW20)

Fig. 10 MODBUS IO server — Edit a tag

By clicking the ***Apply*** button, the Flexy sends the new values to the registers of the PLC. Since the value displayed in the value column is the last value read from the PLC, the new value will not appear until the next poll takes place.

5 Troubleshoot Tags in Error

A tag value displayed with a red icon in the *Quality/Status* column indicates that the quality for this value is considered as *bad*.



☐ Autosave tag value

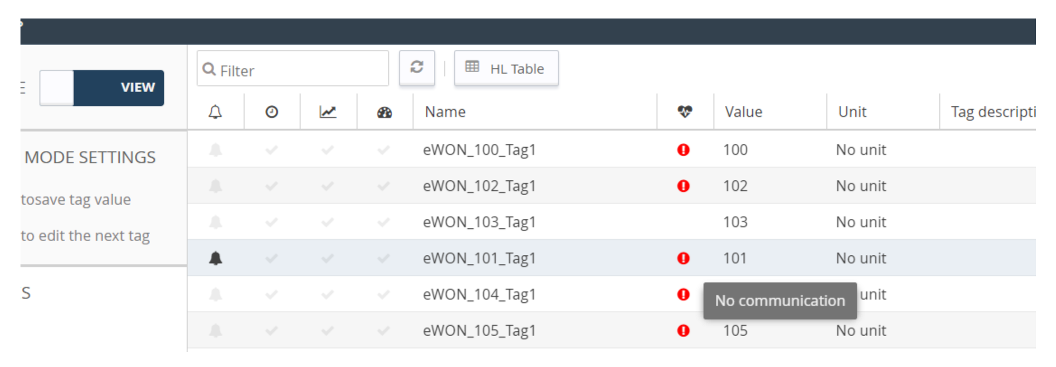
☐ Auto edit the next tag

Name		Value	Tag description
Counter3	❗	0	
Counter2		0	Counter on Siemens PLC (Address: MW20)
Counter1		191	Counter on Siemens PLC (Address: MW10)

Fig. 11 MODBUS IO server — Tag error

As long as the quality of the displayed value is *good*, no icon appears in this column.

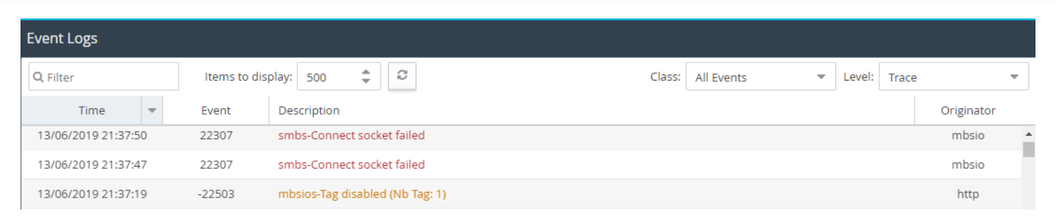
More information about the nature of the problem can be obtained by placing the mouse cursor on the icon.



Name		Value	Unit	Tag descripti
eWON_100_Tag1	❗	100	No unit	
eWON_102_Tag1	❗	102	No unit	
eWON_103_Tag1		103	No unit	
eWON_101_Tag1	❗	101	No unit	
eWON_104_Tag1	❗		unit	
eWON_105_Tag1	❗	105	No unit	

Fig. 12 MODBUS IO server — Tag Error

To get more information about the nature of the error and the sequence of events before and after the error occurred, you can check the events appearing in the *Event Log*.



Time	Event	Description	Originator
13/06/2019 21:37:50	22307	smbs-Connect socket failed	mbsio
13/06/2019 21:37:47	22307	smbs-Connect socket failed	mbsio
13/06/2019 21:37:19	-22503	mbsios-Tag disabled (Nb Tag: 1)	http

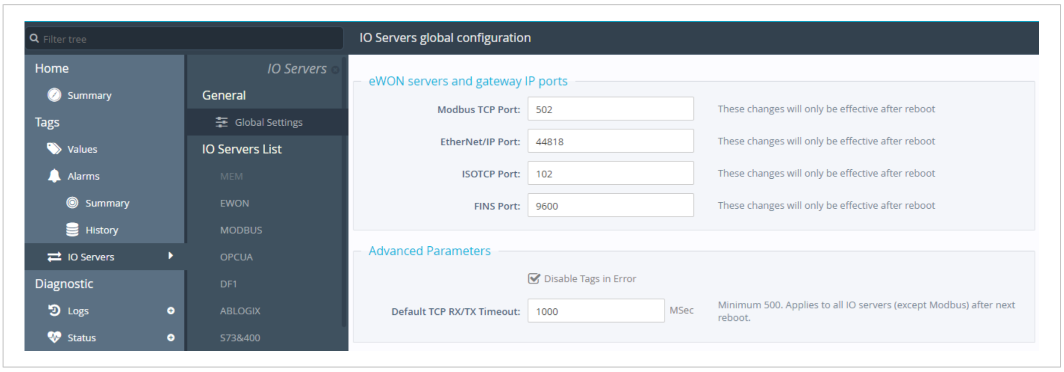
Fig. 13 MODBUS IO server — Error logs

5.1 False Positive

A single tag in error (*truly bad*) can cause a number of other (*good*) tags to appear in error as well because tag requests and responses are grouped in one single *envelope* for communication optimization purposes. The whole group is then affected with the same error status.

During commissioning or maintenance, you may want to isolate the *truly bad* tag from the others. Therefore, you have to disable the polling of tags in error.

This can be done in the ***IO Server*** ► ***Global Settings*** parameters.



IO Servers global configuration

eWON servers and gateway IP ports

Modbus TCP Port: 502 These changes will only be effective after reboot

EtherNet/IP Port: 44818 These changes will only be effective after reboot

ISOTCP Port: 102 These changes will only be effective after reboot

FINS Port: 9600 These changes will only be effective after reboot

Advanced Parameters

☑ Disable Tags in Error

Default TCP RX/TX Timeout: 1000 MSec Minimum 500. Applies to all IO servers (except Modbus) after next reboot.

Fig. 14 MODBUS IO server — Disable tags in error

This page intentionally left blank

A MODBUS Tag Address Syntax

The following convention for the address syntax is in place:

IO server configuration		
IO server name	MODBUS	
Topic name	A	
	B	
	C	
Address	ValueName,SlaveAddress	The PLC address is defined tag by tag on serial link (RTU Master).
	ValueName,SlaveAddress,IPAddress	The PLC address is defined tag by tag on TCP link.
	ValueName	The topic PLC address is in use.

The values of the address field are explained in the following sub-sections.

A.1 ValueName

The tags of the *MODBUS* IO server can be classified following 2 ranges of values.

The two following tables describe the different ranges of value, for each of the two standards:

First Standard			
Modbus Type	**IO Type**	**Access**	**Register address**
Coil	Digital Output	R/W	1 ~ 9999
Contact	Digital Input	R	10001 ~ 19999
Input Register	Analog Input	R	30001 ~ 39999
Holding Register	Analog Output	R/W	40001 ~ 49999
Output Coil	Digital Output	W	50001 ~ 59999
Output Registers	Analog Output	W	60001 ~ 69999

Second Standard			
Modbus Type	**IO Type**	**Access**	**Register address**
Coil	Digital Output	R/W	+1 ~ +65535
Contact	Digital Input	R	+100001 ~ +165535
Input Register	Analog Input	R	+300001 ~ +365535
Holding Register	Analog Output	R/W	+400001 ~ +465535
Output Coil	Digital Output	W	+500001 ~ +565535
Output Registers	Analog Output	W	+600001 ~ +665535

The second standard allows more than 9999 values in each range. Notice the “+” sign before the register value.

The two last ranges “Output Coil” & “Output Registers” are used with non-standard equipment that do not allow the reading of (some of) their values.

In this case, specifying the address in the “write only” ranges informs the Flexy that it should not read the values after setting them, which is normally done in the other cases. If those registers are read, the returned value will always be 0.

After the numerical value, the characters “F”, “L”, “I”, “D” or “W” can be used to specify how to read the value.

The following table describes the different character meaning:

Character	Description	Automatic tag type
W	Reads 1 register considered as 16 bits unsigned integer (default is not specified).	DWord
I	Reads 1 register considered as 16 bits signed integer.	Integer
D	Reads 2 regs R1, R2 as a DWORD R1 is Less significant, R2 is most significant (32 bits, unsigned) (*)	DWord
E	Reads 2 regs R1, R2 as a DWORD R2 is Less significant, R1 is most significant (32 bits, unsigned) (*)	DWord
L	Reads 2 regs R1, R2 as a LONG R1 is Less significant, R2 is most significant (32 bits, signed) (*)	Integer
M	Reads 2 regs R1, R2 as a LONG R2 is Less significant, R1 is most significant (32 bits, signed) (*)	Integer
F	Reads 2 regs R1, R2 as a FLOAT R1 is Less significant, R2 is most significant (32 bits, signed)	Float
H	Reads 2 regs R1, R2 as a FLOAT R2 is Less significant, R1 is most significant (32 bits, signed)	Float

ℹ *To avoid loss of precision of “D”, “E”, “L” or “M” due to integer to float conversion, choose the right storage DataType for your tag.*

When reading a 32 bits value, two consecutive registers or coils are read and combined.

E.g.: “40001L,11” to access in Long representation the reg 1 on the slave 11.

Examples of Modbus addresses

Address	Description
40001,10	access the *Holding Register* on address 1 from the UnitID 10
1,11	access the *Coil* on address 1 from the UnitID 11
+320234,12	access the *Input Register* on address 20234 from the UnitID 12
40001,100,10.0.0.53	access the *Hoding Register* on address 1 from the UnitID 100 which IP address is 10.0.0.53
40010L,12	access the LONG *Holding Register* on address 10 (and 11) from the UnitID 12
40008F,15	access the FLOAT *Holding Register* on address 8 (and 9) from the UnitID 15

A.1.1 Status Register

The status tag is a special tag that returns information about the current state of the communication for a given device.

The status tag address syntax is as follows:

```
Status[Global Device Address]
```

You can define a status tag for each PLC used.

If you use the status address, the tag must be configured as an analog data type such as floating point or integer.

Status value	Description
0	Communication not initialized: Status UNKNOWN. If no tag is polled on that device address, the communication status is unknown.
1	Communication OK
2	Communication NOT OK

This page intentionally left blank

© 2019 HMS Industrial Networks
Box 4126
300 04 Halmstad, Sweden

info@hms.se