Application User Guide

AUG 059 / Rev. 1.0

Polling Data Registers from Modbus Devices

This short guide explains how to poll data registers from a Modbus Device.

support.ewon.biz

Table of Contents

1. Objective

The objective of this document is to explain how the eWON can poll data registers out of one or more Modbus devices.

Polling Modbus data registers can be resumed in four steps :

- Linking the eWON with the Modbus Device
- Configuring the eWON IO Server
- Creating tags in the eWON
- Monitoring tags

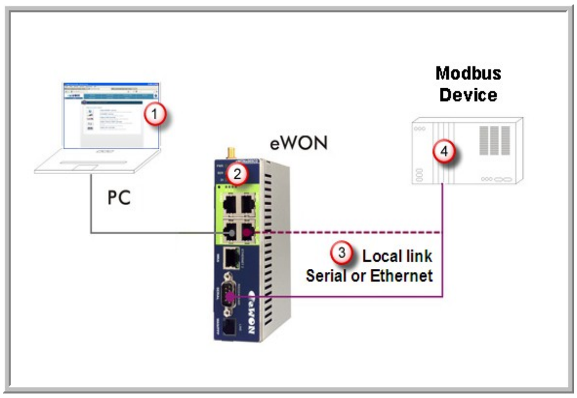
- Note -

🛠 ***Advanced** explanations are indicated by this icon:*

2. Hardware requirements

In order to follow this guide you'll need:

- PC suitable to connect to the eWON
- A Device acting as a Modbus RTU Slave or Modbus TCP Server (PLC, Remote I/Os, RTUs,...)



① From a computer running a web-browser, you will configure the IO Server in the eWON to poll different types of Modbus data registers.

② Access to the eWON web server is done either by using (one of) its local LAN port(s) or by another type of access like VPN IP address, etc.

③ Connection to the Modbus device can be either serial or Ethernet depending on the available interfaces on the device.

④ The connected Modbus device will have its registers read by the tags configured in the IO Server of the eWON.

3. Software requirements

eWON configuration software:

eWON is configured through its web server. All you need is a standard Web Browser software like Internet Explorer or Firefox.

Additionally we suggest you to download the *eBuddy* utility on our website : http://support.ewon.biz/

This utility allows to list all the eWONs on your network and change the default IP address of an eWON to match your LAN IP address range. With eBuddy, you can also easily upgrade the firmware of your eWON (if required).

eWON Firmware Version

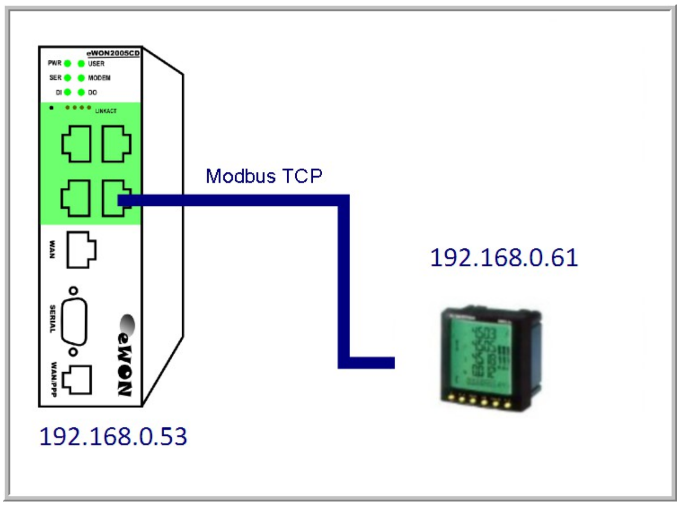
The screen-shots of this guide reflect firmware version 7.0 s2 (2014), but you can expect the basic principles to remain the same in earlier/later versions. *eBuddy* also provides a way to upgrade your firmwares easily.

4. Protocol compatibility

eWON supports standard Modicon Modbus RTU & TCP (Master/Client). So every device that supports standard Modbus protocol can be polled by eWON. By the way, Modbus ASCII is NOT supported

5. Step 1. Linking eWON and Modbus Device

Serial link configuration (Modbus RTU)



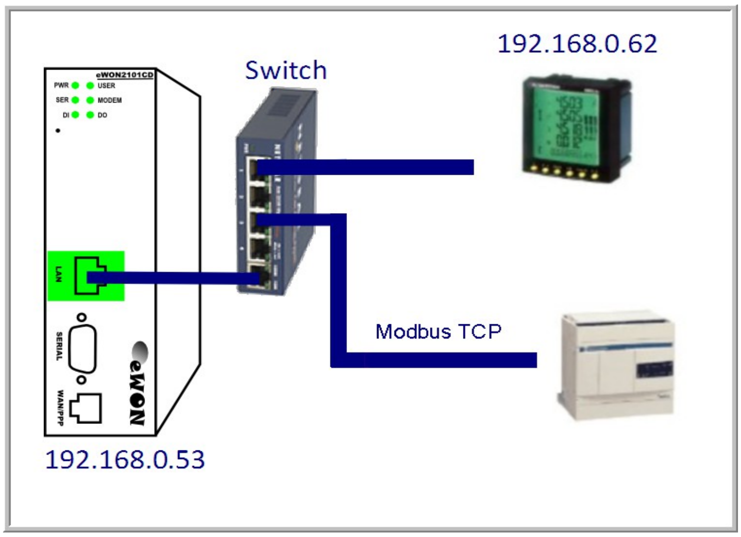
1. Modbus RTU can be performed either on RS232, RS485 or RS422 port. Refer to the cabling schematic of eWON and your device to get the right cable. Through RS232, eWON serial connector acts as a PC serial connector, so a null-modem (cross-over) cable is used most of the time.
2. Set the serial port dip switch of eWON to the right mode according to your Modbus device serial port type : RS232, RS422 or RS485. (Not needed for eWON Flexy backplane serial port or second serial port of eWON Flexy serial card).

Ethernet link configuration (Modbus TCP)

1. Link the LAN interface of the eWON with the Ethernet card of the Modbus device and make sure that the eWON LAN IP address is in the same range as the device IP address. Keep in mind that eBuddy can be used to change the eWON LAN IP address if needed.

- Note -

eWON models supporting 4 LAN ports (2005CD or 4005CD) can be connected to the Ethernet port of the Modbus device directly. eWON models with a single LAN port (2101CD or 4101CD) need[1] to be connected with a crossed cable (single Modbus device) or an external switch (multiple Modbus devices) as the eWON LAN port does not have auto-sense.

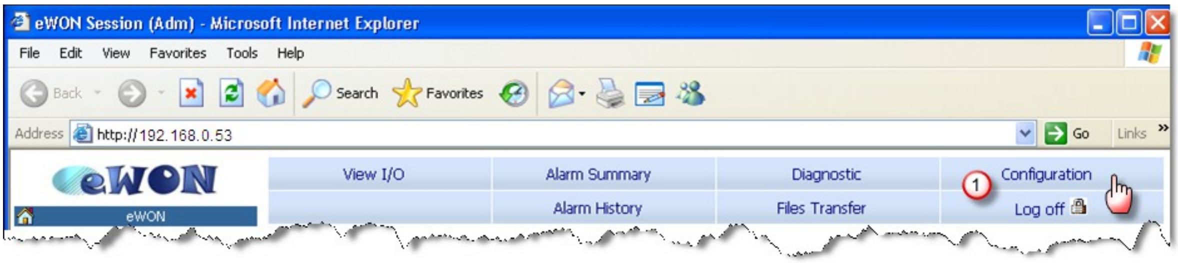


[1] *Actually both straight and crossed cables are acceptable if the Modbus device features an auto-sense Ethernet port.*

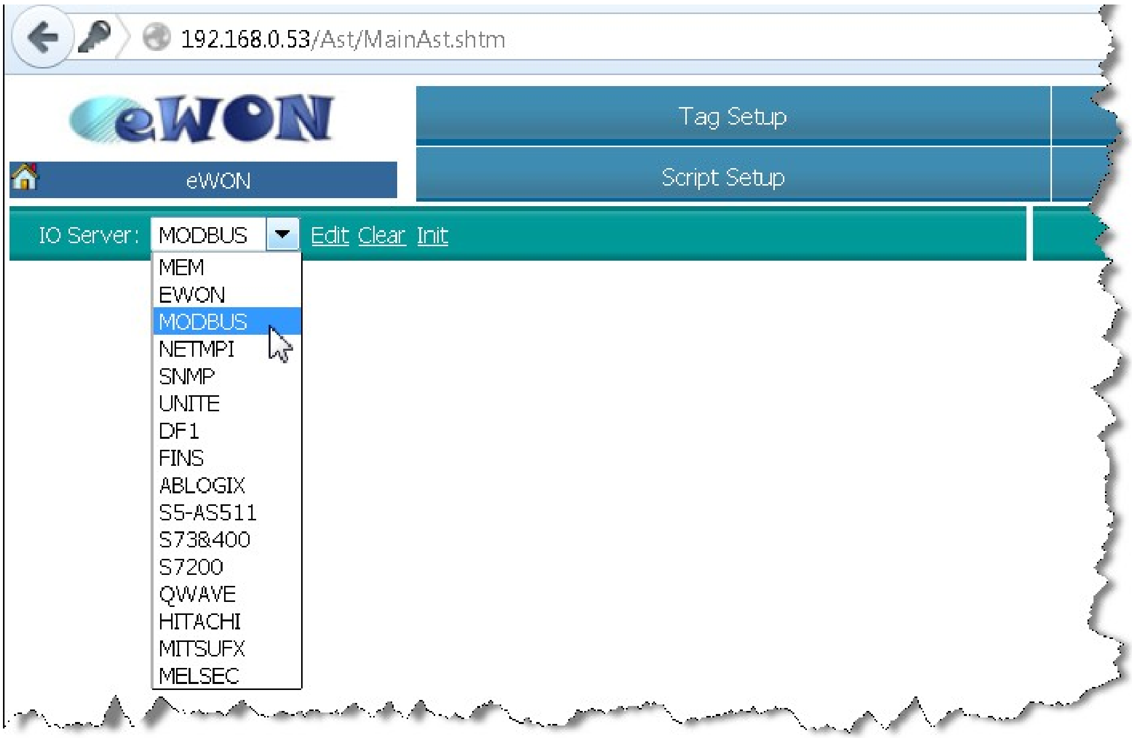
6. Step 2. Configuring the IO Server

Common to serial and Ethernet link

1. Go to the eWON Web page. Open the ***Configuration*** page and go to the ***IO Server Config*** page

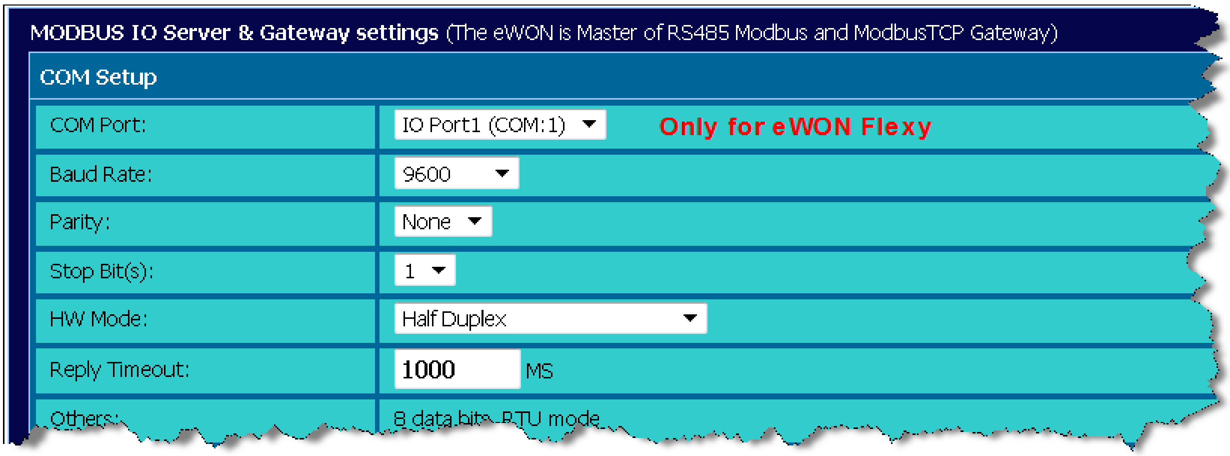


2. Open the ***Configuration*** page and go to the ***IO Server Config*** page
3. In the drop down field select the IO Server "MODBUS"



Specific to serial link (Modbus RTU)

1. Set the ***Port*** *(Only for eWON Flexy)*, ***Baudrate***, ***Parity***, ***Stop Bit*** parameters as defined in the Modbus device you want to connect :



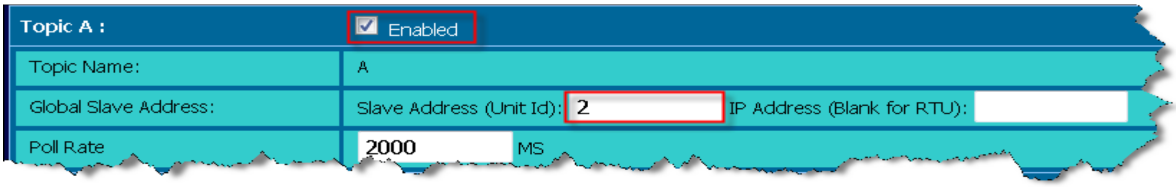
MODBUS IO Server & Gateway settings (The eWON is Master of RS485 Modbus and ModbusTCP Gateway)

COM Setup

COM Port:	IO Port1 (COM:1)	Only for eWON Flexy
Baud Rate:	9600	
Parity:	None	
Stop Bit(s):	1	
HW Mode:	Half Duplex	
Reply Timeout:	1000 MS	
Others:	8 data bits, RTU mode	

The "Reply Timeout" is the time between two retries in case of timeout. 1000 ms is a reasonable value. HW mode option must be set to Half duplex.

2. To be able to poll data registers out of your Modbus Device, you need to define at least one ***Topic*** in the lower part of the IO Server configuration page. Topics are used to allocate common properties to a group of tags (properties include ***Enable/Disable*** polling, ***Poll Rate*** and ***Slave Address***).

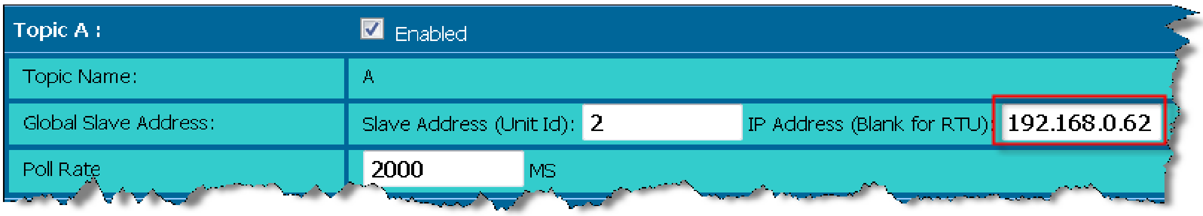


Topic A :	☑ Enabled	
Topic Name:	A	
Global Slave Address:	Slave Address (Unit Id): 2	IP Address (Blank for RTU):
Poll Rate	2000 MS	

3. ***Enable*** at least Topic A by ticking the appropriate box.
4. In ***Poll Rate***, you can define the refresh rate in ms (milliseconds) applicable to all data registers that will be included in this topic. If you have tags that need to be refreshed at different rates, enable and configure multiple topics.
5. Save your settings by clicking on ***Update Config.***

Specific to Ethernet link

1. If you communicate only with Ethernet-connected device(s) and do not use the serial link, then set the ***Baudrate*** to ***Disabled*** and leave all other parameters in the ***COM Setup*** part as they are. If you use both type of communication simultaneously, keep your serial settings as described above.
2. To be able to poll data registers out of your device, you need to define at least one ***Topic*** in the lower part of the IO Server configuration page. Topics are meant to allocate common properties to a group of tags (properties include ***Enable/Disable***, ***Poll Rate, Slave Address and IP address***).



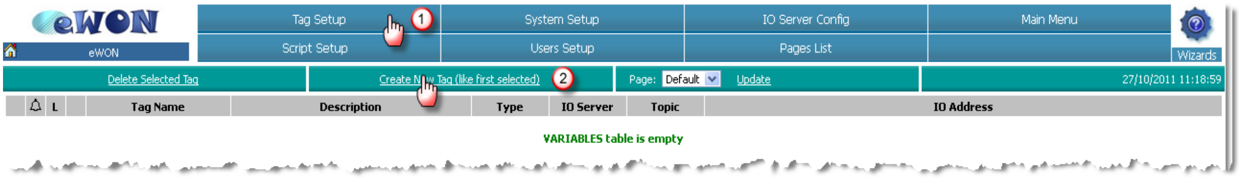
Topic A :	☑ Enabled			
Topic Name:	A			
Global Slave Address:	Slave Address (Unit Id):	2	IP Address (Blank for RTU):	192.168.0.62
Poll Rate	2000	MS		

3. Enable at least Topic A by ticking the appropriate box.
4. Enter the ***Slave Address*** and the ***IP address*** of your Modbus device.
5. In ***Poll Rate***, you can define the refresh rate in ms (milliseconds) applicable to all data registers that will be included in this this topic. If you leave this field blank, the default value applied will be 2000 ms (2 seconds).
6. The polling rate specified here is applied to all tags associated with this topic. If you have tags that need to be refreshed at different rates, enable and configure multiple topics.
7. Save your settings by clicking on ***Update Config***.

7. Step 3. Creating tags in the eWON

Common to all families

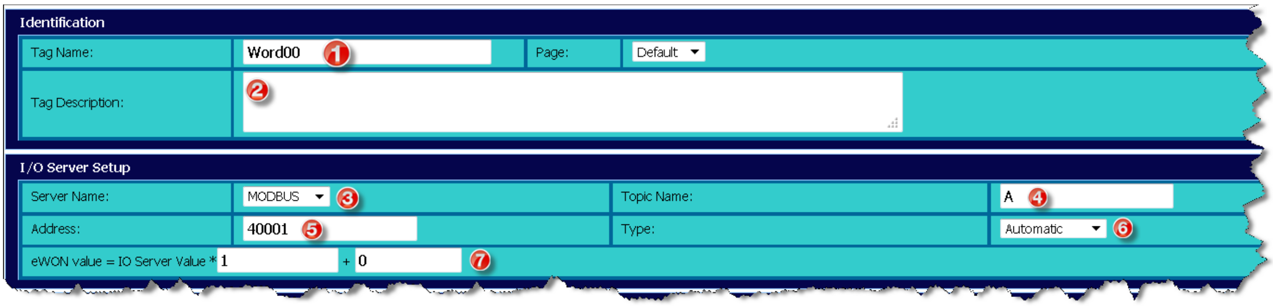
1. Go back to the configuration menu and select ***Tag Setup***. If you are starting from scratch, the list of tags should be empty.



L	Tag Name	Description	Type	IO Server	Topic	IO Address

VARIABLES table is empty

2. Select Create New to open the tag configuration window and enter the parameters of the tag you want to create.



Identification

Tag Name: Word00
Page: Default
Tag Description:

I/O Server Setup

Server Name: MODBUS
Topic Name: A
Address: 40001
Type: Automatic
eWON value = IO Server Value * 1 + 0

① Enter a ***Tag Name*** – free text, no spaces, no symbols -, =, %, $, @, # etc.

② Enter a ***Description*** – free text

③ Select ***Modbus*** as IO server

④ In the ***Address*** field, enter the register address that has to be polled from the Modbus device.

The address syntax is "**PAAAAF**" where :

P	The Modicon prefix that defines the type of data to read and intrinsically the Modbus function used in the reading request. There exists 4 prefixes : • 0 : Data is a "Coil" and Modbus function used is 01. • 1 : Data is a "Digital Input" (read-only) and Modbus function used is 02. • 3 : Data is a "Analog Input"(read-only) and Modbus function used is 04. • 4 : Data is a "Holding Register" and Modbus function used is 03. **-Note -** *Check in your Modbus Device manual what are the supported functions. Some devices only support some functions.*
AAAA	The address of the variable to read from the Slave Device. By default, 4 digits are accepted. If the register address to access is higher than 9999 (5 digits), you need to add a "+" in front of the prefix (+4AAAAAF). **-Note -** *eWON uses Modicon standard. So, it considers that addresses starts at 1. If your Modbus Slavedevice uses JBus standard, it considers that Modbus addresses start at 0. In this case, you just need to add 1 to the register addresses you want to poll.*
F	The Format of the analog value [Optional]. By default, an analog value is interpreted like a Word. If the format is different (Dword,Float, Inverted Dword,...), it must be specified as follows :

Format	Description
W	16 bits unsigned (Word → [0...65535]) (Default).
I	16 bits signed (Integer → [-32768 ... +32767])
D	32 bits unsigned format (DWord → [0 ... +4294967296]). Two successive addresses are read from Slave device starting at definied address (= R1). R1 is the less significant register and R2 is the most significant register.
E	Inverted 32 bits unsigned format (DWord → [0 ...+4294967296]). Two successive addresses are read from Slave device starting at address defined (= R1). R2 is the less significant register and R1 is the most significant register (**E** = SwapRegisters (**D**))
L	32 bits signed format (Long → [-2147483648...+2147483647]). Two successive addresses are read from Slave device starting at address defined (= R1). R1 is the less significant register and R2 is the most significant register.
M	Inverted 32 bits signed format (Long → [-2147483648...+2147483647]). Two successive addresses are read from Slave device starting at address defined (= R1). R2 is the less significant register and R1 is the most significant register. (**L** = SwapRegisters (**M**))
F	Float format (IEEE Single precision format). Two successive addresses are read from Slave device starting at address defined (= R1). R1 is the less significant register and R2 is the most significant register.
H	Inverted Float format (IEEE Single precision format). Two successive addresses are read from Slave device starting at address defined (= R1). R2 is the less significant register and R1 is the most significant

	register. (F = Swap Registers(H))

Note -

> *For Boolean values (Coil and Digital Input), no format must be defined.*

Example

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

⑤ Enter a ***Topic Name*** A, B or C. The topic must have been configured in the IO server page (see § 6 Step 2. Configuring the IO Server).

The remaining fields are mostly left with their default value:

⑥ ***Type***: DataType of the tag (Automatic, Floating Point, Boolean, Integer, Dword). All tags are stored on 4 bytes. The default ***Automatic*** option lets the eWON decide the format depending on the IOServer register/modifier type.

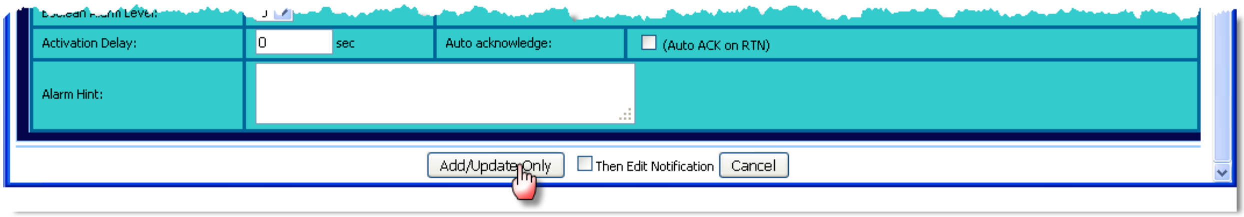
⑦ ***Force Read Only***: Unchecked is the default. When it is checked, users will not be able to write a value by using the ***Update*** command in the ***View IO page***.

The tag remains however read/write for commands written in the embedded BASIC program.

⑧ ***eWON value***: Defaults are *1+0. 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.

For advanced explanations for these fields, please refer to the General Reference Guide RG-001 available here: Wiki Support

3. Click on the ***Add/Update*** Only button when your tag configuration is complete

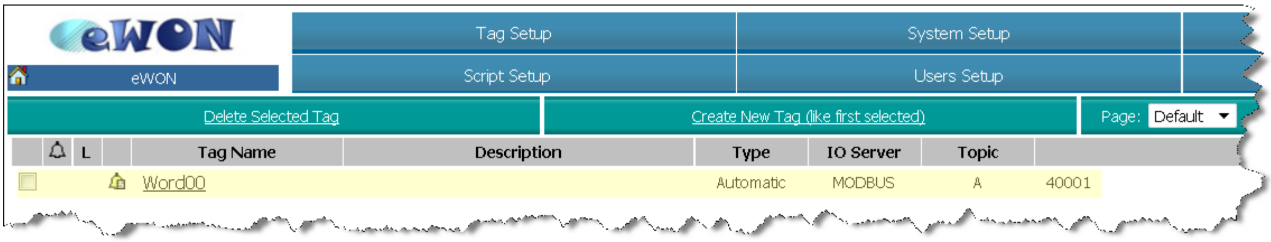


Activation Delay: 0 sec

Auto acknowledge: □ (Auto ACK on RTN)

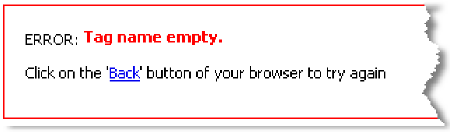
□ Then Edit Notification

4. If everything is OK, our new tag appears in the tag list:



	L	Tag Name	Description	Type	IO Server	Topic	
□		Word00		Automatic	MODBUS	A	40001

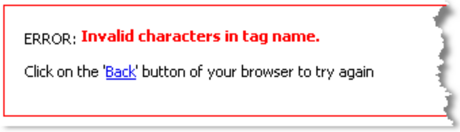
If not, here are a couple of examples of error messages that can appear:



ERROR: **Tag name empty.**

Click on the 'Back' button of your browser to try again

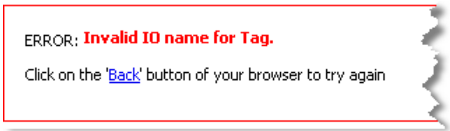
Tag name empty: check tag name field and type name in it.



ERROR: **Invalid characters in tag name.**

Click on the 'Back' button of your browser to try again

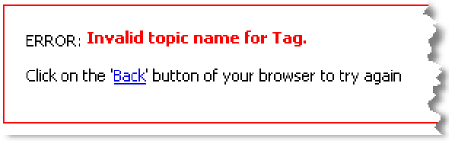
Invalid character in tag name: check name for spaces and invalid characters.



ERROR: **Invalid IO name for Tag.**

Click on the 'Back' button of your browser to try again

Invalid IO name for tag: check tag address syntax.

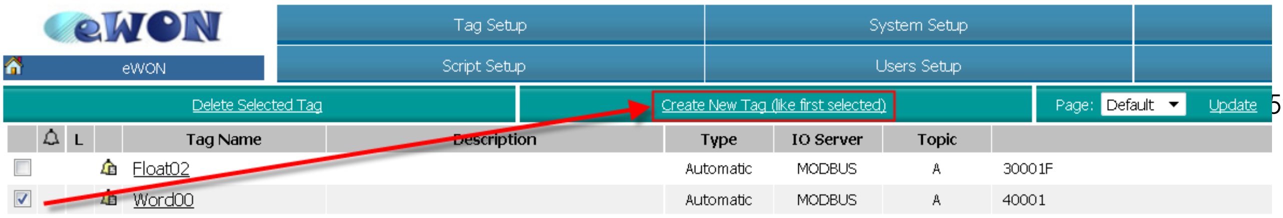


ERROR: **Invalid topic name for Tag.**

Click on the 'Back' button of your browser to try again

***Invalid topic name for tag**: check if topic was enabled in IO server, check topic field and name.*

5. Redo the same sequence from point 1 for the other tags you need to create.
6. If you need to create new tags that have almost the same properties as an existing tag in the list, then check the box next to the source tag before clicking on the Create New Tag (like first selected) link.
7. All (*) properties of the existing tag will be copied in the new tag creation



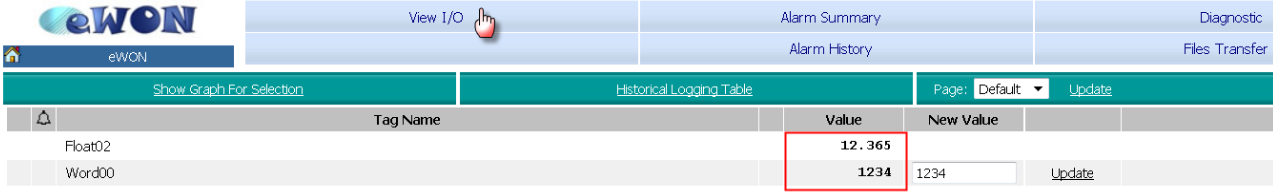
	L	Tag Name	Description	Type	IO Server	Topic	
□		Float02		Automatic	MODBUS	A	30001F
☑		Word00		Automatic	MODBUS	A	40001

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

(*) Copied properties include the Tag Name. Since the Tag Name must be unique, make sure you change the name of the new tag.

8. Step 4. Monitoring tags

1. Go to the ***View I/O*** page from the ***Main Menu*** to check tag values and status



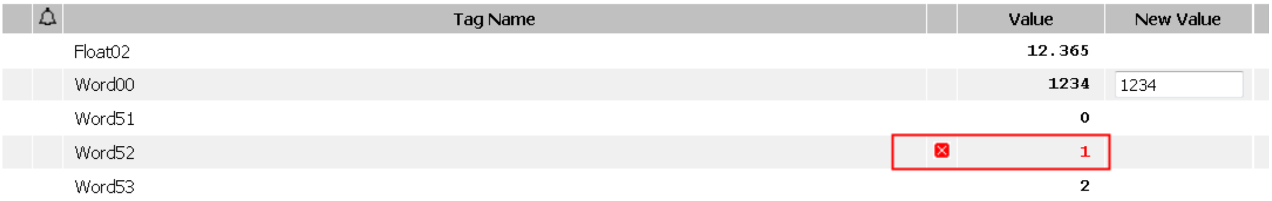
🔔	Tag Name	Value	New Value	
	Float02	12.365		
	Word00	1234	1234	Update

2. 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, edit the ***New Value*** field and click on the ***Update*** ① link as shown below.

- Note -

Clicking this **Update** ① link sends the new value to the register of the Modbus Device (Using other Modbus Functions → FCT 5 for Digital value, FCT 6 and 16 for analog value) . The value will actually be returned with the next poll. In addition, you need to instruct your browser to refresh the value shown on screen. To update the value on the screen, click the page **Update** ② link.

3. Tags in error appear with a red icon as shown below:



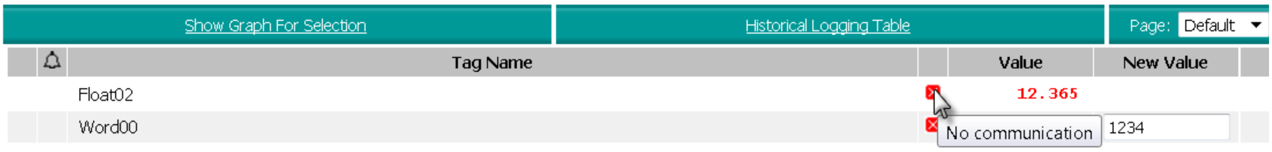
🔔	Tag Name		Value	New Value
	Float02		12.365	
	Word00		1234	1234
	Word51		0	
	Word52	☒	1	
	Word53		2	

Please refer to § 9 Troubleshooting tags in error if this appears.

9. Troubleshooting tags in error

A tag value displayed in red in the View IO page indicates that the quality for this value is considered as bad. When the quality of the tag is bad, a red icon is displayed in the column between the tag name and the tag value.

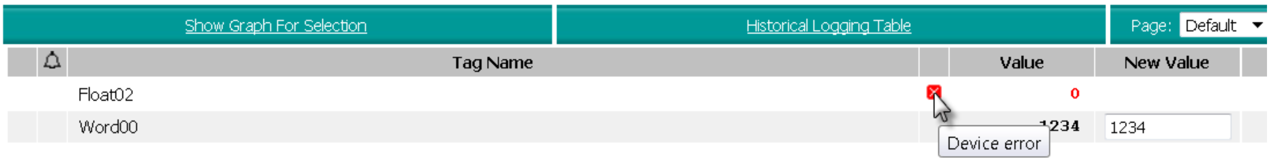
As long as the quality of the displayed value is good, no icon appears in this column. A short destcription about the origin of the problem can be obtained by placing the mouse cursor on the icon as shown below:



	Tag Name		Value	New Value
	Float02	☒	12.365	
	Word00	☒		1234

No communication

or

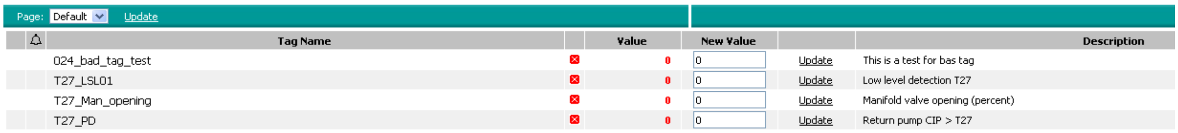


	Tag Name		Value	New Value
	Float02	☒	0	
	Word00	☒	1234	1234

Device error

- Note-

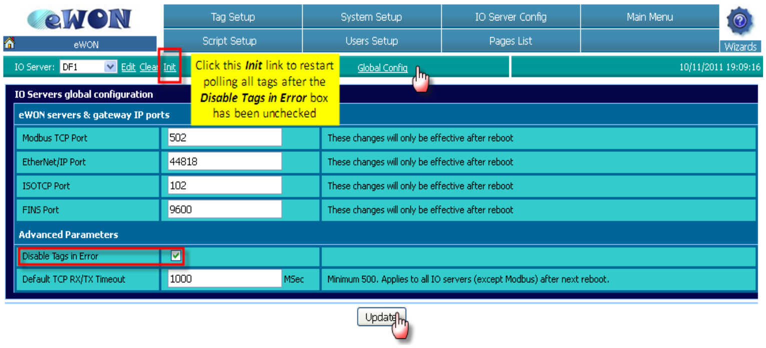
A single tag in error (truly bad) can cause communication errors for other Tags if they are grouped in a single read request (especially when several tags are configured with successive Modbus addresses). Indeed, the Modbus device will simply respond that the whole request is invalid.



	Tag Name		Value	New Value		Description
	024_bad_tag_test	☒	0	0	Update	This is a test for bas tag
	T27_LSL01	☒	0	0	Update	Low level detection T27
	T27_Man_opening	☒	0	0	Update	Manifold valve opening (percent)
	T27_PD	☒	0	0	Update	Return pump CIP > T27

*To force the eWON to read one tag at a time and hence know which tag makes the problem, you can check the option "**Disable tag in error**" in **IO Server → Global Config**.*

*Do not forget to deactivate the option (and click "**Init**" link in Modbus IO Server menu) once the error have been resolved.*



Click this ***Init*** link to restart polling all tags after the ***Disable Tags in Error*** box has been unchecked

IO Servers global configuration

eWON servers & gateway IP ports

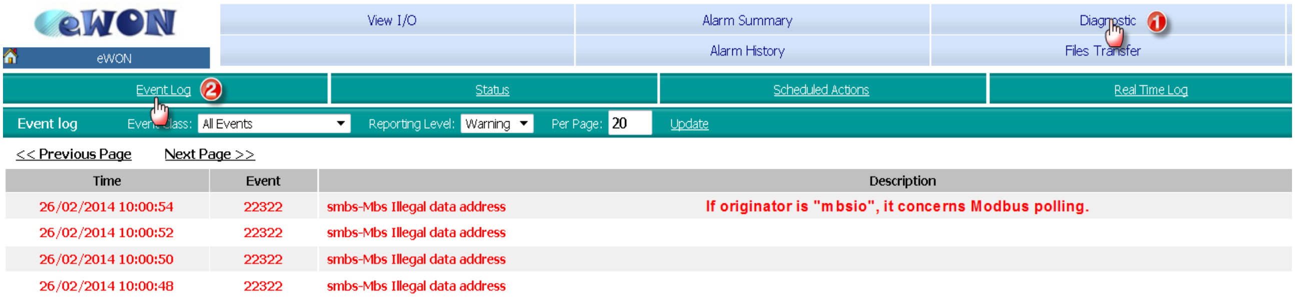
Modbus TCP Port	502	These changes will only be effective after reboot
EtherNet/IP Port	44818	These changes will only be effective after reboot
ISOTCP Port	102	These changes will only be effective after reboot
FINS Port	9600	These changes will only be effective after reboot

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.

To get more details about the origin of the error and the sequence of events before and after the error occurred, you can check the events appearing in the ***Event Log***. The ***Event Log*** ② can be accessed through the ***Main Menu, Diagnostic*** ① .

The log originator "**mbsio**" indicates that the log is related to the Modbus IO server. However, a connection issue can generate logs with a different originator.



Event log Event Class: All Events Reporting Level: Warning Per Page: 20

<< Previous Page Next Page >>

Time	Event	Description
26/02/2014 10:00:54	22322	smbs-Mbs Illegal data address
26/02/2014 10:00:52	22322	smbs-Mbs Illegal data address
26/02/2014 10:00:50	22322	smbs-Mbs Illegal data address
26/02/2014 10:00:48	22322	smbs-Mbs Illegal data address

If originator is "mbsio", it concerns Modbus polling.

In case the error displayed in the tooltip indicates "No communication", the error is probably related to the communication link. You should then verify your communication parameters :

- Serial communication (Modbus RTU) :
  - Serial settings such as Baudrate, Parity,... (Must match between eWON and Modbus device).
  - Dip switch of eWON Serial port
  - Serial cable schematic.
- IP communication (Modbus TCP) :
  - IP address and TCP port (Is the Modbus device IP address in the same range as eWON LAN IP ?).
  - Ethernet cable (Do you need a straight or cross-over cable?).

In case the error displayed in the tool-tip indicates "Device Error", the error is then probably related to a wrong tag configuration. The event log can be helpful in this case.

A typical Modbus exception is "Illegal Data Address". This means that the Modbus address that eWON tries to read does not exist in the target device. You should therefore check the address of the tags in error.

Revision

Revision History

Revision Level	Date	Description
1.0	18/08/2015	Original Copie

Document build number: 10

Note concerning the warranty and the rights of ownership:

The information contained in this document is subject to modification without notice. Check https://www.ewon.biz/support for the latest documents releases.

The vendor and the authors of this manual are not liable for the errors it may contain, nor for their eventual consequences.

No liability or warranty, explicit or implicit, is made concerning the quality, the accuracy and the correctness of the information contained in this document. In no case the manufacturer's responsibility could be called for direct, indirect, accidental or other damage occurring from any defect of the product of errors coming from this document.

The product names are mentioned in this manual for information purposes only. The trade marks and the product names or marks contained in this document are the property of their respective owners.

HMS Industrial Networks