

Polling Data from MODBUS TCP

Using Ethernet Protocol

APPLICATION NOTE

AUG-0059-00 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

1	Preface	3
1.1	About This Document	3
1.2	Document history	3
1.3	Related Documents	3
1.4	Trademark Information	3
2	Objectives	4
3	Requirements	5
3.1	Hardware	5
3.2	Software.....	5
3.3	PLC Protocol Compatibility.....	5
4	Implementation Steps.....	6
4.1	Link the Flexy and the PLC	6
4.2	Configure the IO Server.....	6
4.3	Create Tags in the Flexy	7
4.4	Monitor Tags.....	10
5	Troubleshoot Tags in Error	12
5.1	False Positive	12
A	MODBUS Tag Address Syntax.....	15
A.1	ValueName	15
A.2	Slave Address	16
A.3	IP Address	17

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 TCP.

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	2019-07-23	Major update – reformulation, new document organization

1.3 Related Documents

Document	Author	Document ID
Polling Data from MODBUS RTU Using Serial Protocol	HMS	AUG-0080-00
Ewon Flexy Base Units	HMS	IG-0014-00
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 TCP.

Polling PLC data registers implies the following steps :

1. Link the Flexy with the PLC;
2. Configure the Flexy 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 TCP server such as a PLC, remote IO, or RTU.
 - Connection between the Ewon Flexy and the PLC must be done through Ethernet protocol. For polling tags over Modbus RTU with a serial connection, see “Polling Data from MODBUS RTU Using Serial 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, among others, standard Modicon Modbus TCP (Master/Client).

However, Modbus ASCII is not supported.

4 Implementation Steps

4.1 Link the Flexy and the PLC

1. Link the LAN interface of the Flexy with the Ethernet card of the PLC.
2. Make sure that the Flexy's LAN IP address is in the same range as the PLC IP address.

You can use eBuddy to change the Flexy's LAN IP address if it is not within the same range as the PLC.

For Ewon Flexy with configurable LAN / WAN ports such as the Flexy 205, a LAN port is indicated by a green LED and a WAN port is indicated by a red LED.

For a Flexy with permanent LAN ports such as the Flexy 201 or 101, the LAN ports are the Ethernet ports on the base unit itself.

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. 1 MODBUS IO server menu

- 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**.

Fig. 2 MODBUS IO server settings

- Enable at least **Topic A** by ticking the appropriate box.
- Enter a valid **Slave Address** and an **IP 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.

- 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.

- Click on **Update** to save your settings.

4.3 Create Tags in the Flexy

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

2. Switch the “Mode” to **Setup**.

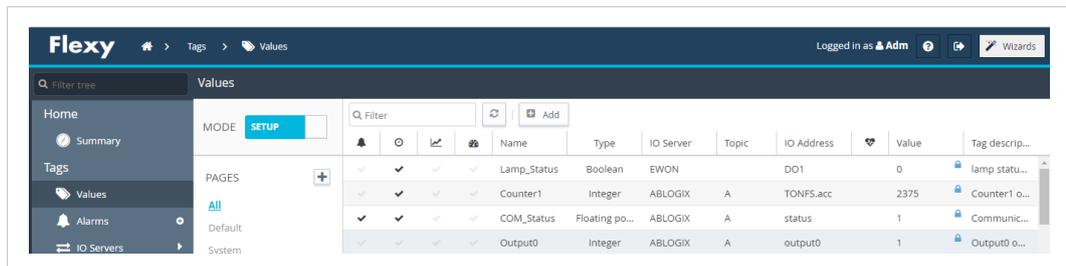


Fig. 3 MODBUS IO server — Tag setup

3. Click the **Add** button to open the tag configuration window.
4. Enter the **parameters of the tag** you want to create.

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

Add Tag

Fig. 4 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.

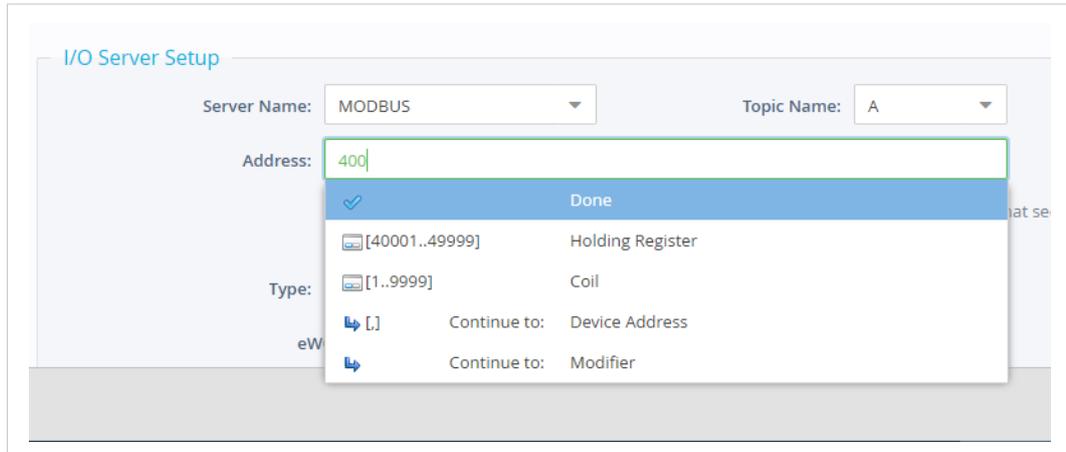


Fig. 5 MODBUS IO server — Tag setup 3

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. 15](#).

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

- **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.
- **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.
 $TAGval = IOSERVERval * scale\ factor + offset.$

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

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

Values		Q Counter	↻	+	Add	⚙	Edit	✖	Delete	🔔	Configure Alarm actions		
MODE		SETUP											
PAGES		+											
All													
Default													
		🔔	🕒	📄	🔍	Name ↓	Type	IO Server	Topic	IO Address	🔒	Value	Tag description
		✓	✓	✓	✓	Counter3	Integer	MODBUS	A	40001		17000	🔒
		✓	✓	✓	✓	Counter2	Floating po...	573&400	a	MW20		0	🔒 Counter on Si...
		✓	✓	✓	✓	Counter1	Floating po...	573&400	a	MW10		158	🔒 Counter on Si...

Fig. 6 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.

Values		Q Filter	↻	✎	Edit Value	📄	HL Table		
MODE		VIEW							
VIEW MODE SETTINGS		🔔	🕒	📄	🔍	Name	🔒	Value	Tag description
<input type="checkbox"/>	Autosave tag value	🔔	✓	✓	✓	Counter3		0	
		🔔	✓	✓	✓	Test_Tag2		0	Tag on Siemens PLC (Address: Bit 2 of MB10)

Fig. 7 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.

The screenshot shows a web interface titled "Values" for a MODBUS IO server. It features a table of tags with columns for Name, Value, and Tag description. The "Test_Tag1" row is highlighted in yellow, indicating it is the current focus. The interface includes a search filter, an "Apply" button, and various settings and page options.

MODE	VIEW	Filter	Edit Value	Apply	HL Table	Name	Value	Tag description
VIEW MODE SETTINGS						Lamp_Status	1	Light controlled by eWON digital output (1=ON - 0=OFF)
<input type="checkbox"/> Autosave tag value						COM_Status	1	Communication status with S7-300 MPI (1=OK, 2= NOT OK)
<input type="checkbox"/> Auto edit the next tag						Test_Tag1	0 → 5	Tag on Siemens PLC (Address: MB10)
PAGES						Output_1	1	Output on PLC (at Q0.1)
All						Output_0	0	Output on PLC (at Q0.0)
Default						Counter1	106 → 68	Counter on Siemens PLC (Address: MW10)
						Counter2	0	Counter on Siemens PLC (Address: MW20)

Fig. 8 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*.

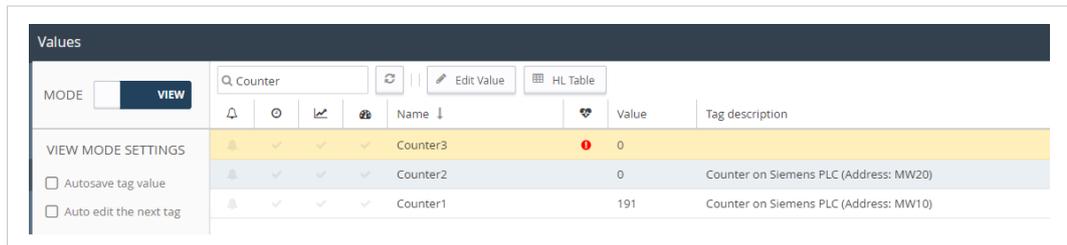


Fig. 9 MODBUS IO server — Tag error

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.

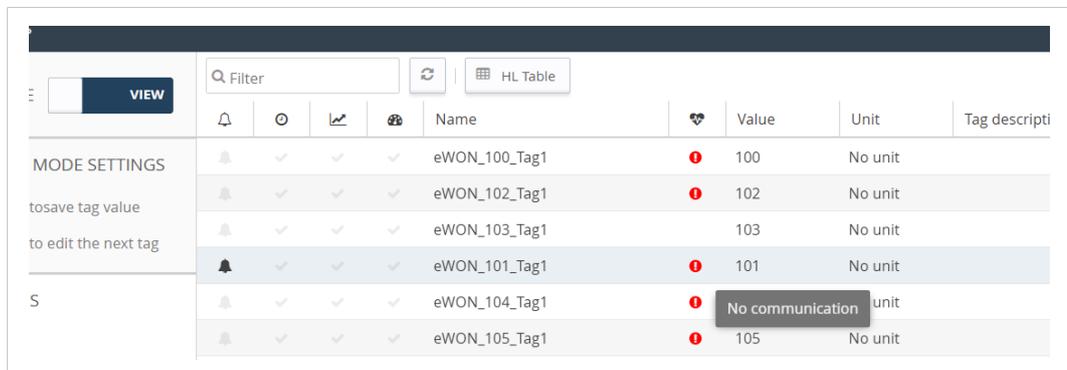


Fig. 10 MODBUS IO server — Tag Error

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*.



Fig. 11 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.



Fig. 12 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



To avoid loss of precision of “D”, “E”, “L” or “M” due to integer to float conversion, choose the right storage Data Type 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 <i>Holding Register</i> on address 1 from the UnitID 10
1,11	access the <i>Coil</i> on address 1 from the UnitID 11
+320234,12	access the <i>Input Register</i> on address 20234 from the UnitID 12
40001,100,10.0.0.53	access the <i>Holding Register</i> on address 1 from the UnitID 100 which IP address is 10.0.0.53
40010L,12	access the <i>LONG Holding Register</i> on address 10 (and 11) from the UnitID 12
40008F,15	access the <i>FLOAT Holding Register</i> on address 8 (and 9) from the UnitID 15

A.1.1 Status Tag

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

As for other tags, the status tag of an address is composed of: *Status,Address*.

If the address is omitted, the topic address will be used, e.g.: “status,11” points to the status of the slave 11.

You can define a status tag for each address used. If you use the status MODBUS address, the tag must be configured as analog:

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

A.2 Slave Address

This is the address of the slave device that you want to access.

It is a number from 0 to 255, e.g.: 30001,11 will poll an RTU device on address 11.

A.3 IP Address

This is the IP address of the device on an Ethernet network.

It is composed of 4 numbers separated by a dot, e.g.: 30001,11,10.0.0.50 will poll a device configured with an IP address 10.0.0.50 and with Modbus slave address 11.

