

# IO Servers

## REFERENCE GUIDE

RG-0007-01 ENGLISH

## Contents

<b>1. The Ewon IO Servers.....</b>	<b>5</b>
1.1. Introduction.....	5
1.2. IO Servers Setup.....	5
1.3. Assisted Edition .....	7
1.4. Global Settings.....	8
1.5. Standard IO server configuration page .....	8
1.6. Link the Flexy & the PLC viaserial.....	9
<b>2. MODBUS IO Server.....</b>	<b>10</b>
2.1. Introduction.....	10
2.2. Setup .....	10
2.3. Tag name convention .....	14
<b>3. NETMPI IO Server.....</b>	<b>17</b>
3.1. Introduction.....	17
3.2. Setup .....	17
3.3. Tag name convention .....	17
<b>4. UNITE IO Server.....</b>	<b>19</b>
4.1. Introduction.....	19
4.2. Setup .....	19
4.3. Tag name convention .....	21
<b>5. DF1 IO Server .....</b>	<b>23</b>
5.1. Introduction.....	23
5.2. Setup .....	23
5.3. Tag name convention .....	26
<b>6. FINS IO Server.....</b>	<b>30</b>
6.1. Introduction.....	30
6.2. Setup .....	30
6.3. Tag Name Convention .....	33
<b>7. S5-AS511 IO Server.....</b>	<b>35</b>
7.1. Introduction.....	35
7.2. Setup .....	35
7.3. Tag name convention .....	36
<b>8. ABLOGIX IO Server .....</b>	<b>38</b>
8.1. Introduction.....	38
8.2. Setup .....	38
8.3. Tag name convention .....	41
<b>9. Ewon IO Server .....</b>	<b>43</b>
9.1. Introduction.....	43
9.2. Standard Ewon I/O Item Names.....	43

9.3.	Setup .....	48
<b>10.</b>	<b>MEM IO Server .....</b>	<b>50</b>
10.1.	Introduction.....	50
10.2.	Setup .....	50
10.3.	Tag name convention .....	50
<b>11.</b>	<b>S7-300&amp;400 (MPI/Profibus - ISOTCP) .....</b>	<b>51</b>
11.1.	Introduction.....	51
11.2.	Setup .....	51
11.3.	Tag name convention .....	54
<b>12.</b>	<b>S7-200 (PPI - ISOTCP).....</b>	<b>57</b>
12.1.	Introduction.....	57
12.2.	Setup .....	57
12.3.	Tag name convention .....	60
<b>13.</b>	<b>HITACHI IO Server .....</b>	<b>64</b>
13.1.	Introduction.....	64
13.2.	Setup .....	64
13.3.	Tag name convention .....	65
<b>14.</b>	<b>MITSUBISHI FX IO Server .....</b>	<b>69</b>
14.1.	Introduction.....	69
14.2.	Setup .....	69
14.3.	Tag name convention .....	70
<b>15.</b>	<b>SNMP IO Server.....</b>	<b>73</b>
15.1.	Introduction.....	73
15.2.	Setup .....	73
15.3.	Tag name convention .....	74
<b>16.</b>	<b>MITSUBISHI MELSEC IO Server .....</b>	<b>76</b>
16.1.	Introduction.....	76
16.2.	Setup .....	76
16.3.	Tag name convention .....	77
<b>17.</b>	<b>BACNET IO Server .....</b>	<b>80</b>
17.1.	Introduction.....	80
17.2.	Setup .....	80
17.3.	Tag name convention .....	80
<b>18.</b>	<b>OPCUA IO Server.....</b>	<b>82</b>
18.1.	Introduction.....	82
18.2.	Setup .....	82
18.3.	Tag name convention.....	84

---

# 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.

# 1. The Ewon IO Servers

## 1.1. Introduction

An IO Server is the interface between a changing value and the Ewon monitoring engine. It is a kind of driver. Any variable from any *data source* must have a common representation for all IO Servers in order to define common interface in the Ewon.

The *data-source* representation in the Ewon uses 3 fields for the definition of a Tag:

- **The IO Server Name**
- **The Topic name**
- **The Item Name**

A Tag's data-source will be uniquely identified with these 3 parameters:

<b>IO Server name</b>	Is a kind of driver name. For each IO Server there is a specific Topic Name and Item Name syntax. Example: MODBUS, Ewon, MEM
<b>Topic Name</b>	Is used to group items inside an IO Server, for example the memory IO Server uses the blank topic ("") and the retentive topic ("ret"). All Tags of the MEM IO Server defined in the "ret" topic will have their value saved and restored when the Ewon boots. All IO servers do not use a Topic Name. In that case the Topic Name field must be left empty.
<b>Item Name</b>	The item name is a string of characters; its syntax is specific to each IO Server. The Item Name describes the physical variable to monitor that uses the IO Server.

For example, the MODBUS IO Server needs to poll registers or coils from a slave, so it uses an item name representation to define the *register type, register address and slave address*. (Example "40001,5" => Where 4 means "read write register", 0001 is the register number and 5 is the slave Modbus address).



For optimization purpose, the Ewon may disable the polling of "invalid tags"

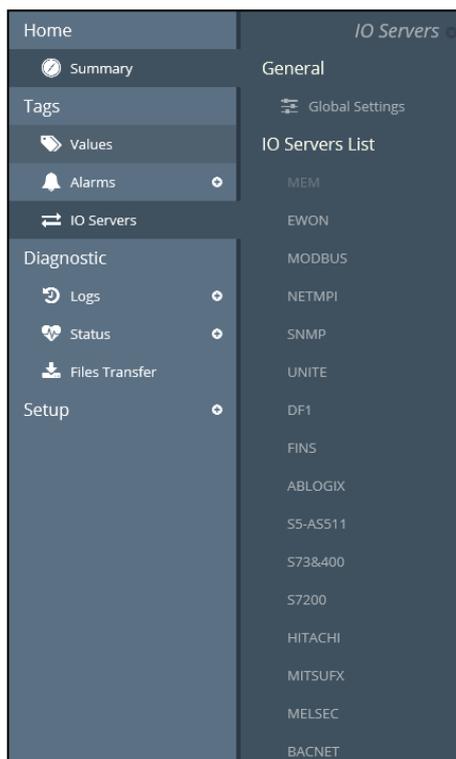
## 1.2. IO Servers Setup

Some of the IO servers are configurable.  
The IO servers window proposes a list of IO servers:

Select an IO server to display its edition window.

There are 3 possible cases regarding the IO server configuration:

- **The IO server is not configurable**
- **The IO server has a dedicated configuration page**
- **The IO server uses the standard IO server configuration page.**



### 1.2.1. IO Server Clear

To avoid the useless consumption of CPU and memory, the unused IO Servers need to be “unloaded” from CPU tasks and the used memory need to be cleared.

For that purpose, the Clear function must be used.

Only set to *disable* the Baud Rate of an IO Server make it idle, but this IO Server is still in memory and use some CPU time (for nothing).

Click on the **Clear** link to reset the config of the displayed IO Server and unload it from memory.



Some IO Servers are not dynamically stoppable and need an Ewon reboot to ensure the unload from memory. You will be warned with the following message.

Config cleared (but IO server is not stoppable, reboot is required to disable IO server).

### 1.2.2. IO Server Init

Click on the **Init** link to initialize an IO Server.

This initialization will reset all IO Server counters and restart the tags validation process.



### 1.3. Assisted Edition

Ewon features an Assisted Edition for the destination device address (the PLC to reach). This allows to encode the correct destination device address for each IOserver.

Topic A Enabled

Global Device Address:  Enter PLC access protocol

Poll Rate:  Default: 2000

[MPI]	MPI
[PROFIBUS]	PROFIBUS
[ISOTCP]	ISOTCP

According to the IOserver you selected, Ewon will display the different possibilities for the PLC device address. For example, for some IO servers you first need to specify the protocol that will be used for the polling.

You can with a mouse click select one of the proposed items, or simply enter the text directly inside the address field.

Topic A Enabled

Global Device Address:  Enter protocol dependent device address

Poll Rate:  Default: 2000

Continue to: ISOTCP address

As long as the device address path is not complete, Ewon will display the address field in black color. This means that some items are still missing inside the address path.

To continue the address path, select or enter one of the new proposed items.

Topic A Enabled

Global Device Address:  Enter MPI/PROFIBUS address: SubnetID,node or node

Poll Rate:  MS Default: 2000

You can now leave the address as it and save the configuration.

Or, if required, you can define supplementary optional address path options. For example, in the picture here above clicking (or typing) the [,] option allows to encode a PLC address that can be reached on a second network behind the first PLC.

An incorrect address syntax will be displayed in red and you'll not be allowed to save the configuration.

Topic A Enabled

Global Device Address:  Enter MPI/PROFIBUS address: SubnetID,node or node

Poll Rate:  MS Default: 2000



Assisted Edition is not applicable for MEM, Ewon or MODBUS servers

### 1.4. Global Settings

eWON servers and gateway IP ports

Modbus TCP Port:	<input type="text" value="502"/>	These changes will only be effective after reboot
EtherNet/IP Port:	<input type="text" value="44818"/>	These changes will only be effective after reboot
ISOTCP Port:	<input type="text" value="102"/>	These changes will only be effective after reboot
FINS Port:	<input type="text" value="9600"/>	These changes will only be effective after reboot

Item	Description
<b>Modbus TCP Port</b>	Encode here the TCP Port number used for the Modbus TCP server protocol. <i>Default is 502</i>
<b>Ethernet/IP Port</b>	Encode here the TCP Port used for the Ethernet/IP protocol (Allen Bradley). <i>Default is 44818</i>
<b>ISOTCP Port</b>	Encode here the TCP Port number used for the ISOTCP protocol (Siemens). <i>Default is 102</i>
<b>FINS Port</b>	Encode here the TCP Port number used for the FINS protocol (Omron). <i>Default is 9600</i>

Advanced Parameters

Disable Tags in Error

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

Item	Description
<b>Disable Tags in Error</b>	Activate the Tags Validation
<b>Default TCP RX/TX Timeout</b>	Value in mSec for IOserver TCP timeout (need reboot to apply)



To disable ports used for transparent gateway, set the corresponding port to 0 and reboot the Ewon.

### 1.5. Standard IO server configuration page

When no dedicated configuration page is defined for configuring an IO server, the standard configuration page is used.

IOserver
Init
Clear

```
MinInterval:10
MaxInterval:268435455
ReverseCount:0
```

As you can see in the above example, the standard configuration screen is a simple text edition area. Each parameter is entered on a separate line, the parameter value is separated from the parameter name by a colon ':'

The generic format of a line is: **PARAM\_NAME : PARAM\_VALUE**



**Extra space must be removed**

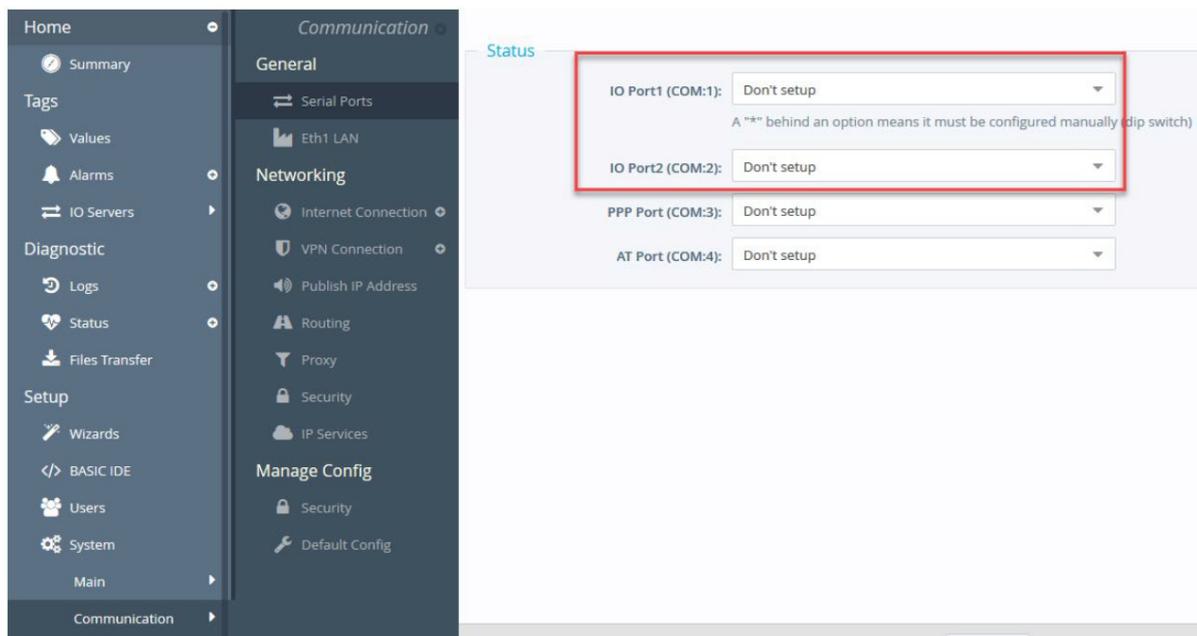
When using this configuration, you must respect the correct syntax of each parameter and the values for each parameter.

The list of valid parameters and their corresponding valid values are listed in the corresponding IO server documentation. Any error that would occur when the Ewon applies the configuration that you have entered would be written to the event file.

## 1.6. Link the Flexy & the PLC via serial

Connection with some PLC can be performed either on an RS232, RS422 or RS485 Interface.  
Set the serial port of the Flexy base unit to the right mode according to your device serial port type: RS232, RS422 or RS485.

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.



For the FLA 3301 – 2 Serial Ports Extension card:  
Set the serial port dip switch to the right mode according to your device serial port type: RS232, RS422 or RS485.  
The proper configuration is printed on the extension card to the left of the switches.

- Port S1 is configurable with dip switches.
- Port S2 is fixed RS232 mode.

## 2. MODBUS IO Server

### 2.1. Introduction

The MODBUS IO Server setup is the standard remote IO communication setup of the Ewon. It is used to configure:

- **The Ewon as a Modbus RTU master.**
- **The Ewon as a Modbus TCP slave and master.**

The first feature (Modbus TCP slave) is specific to the MODBUS IO server; it is actually designed to provide access to Ewon Tag values and, unlike all the other IO servers, for interfacing field values with the Ewon.

The second feature (MODBUS Master) is the actual IO server feature that provides an interface to the field values as a common IO server.

The Ewon MODBUS IO server will give access to values in equipments having a MODBUS interface.

The interface can be:

- **RS485 / RS422 / RS232 – MODBUS RTU protocol will be used**
- **ETHERNET/PPP – MODBUS TCP protocol will be used.**

The Ewon can mix access to MODBUS RTU and MODBUS TCP, depending on the way the Tag address is defined.

### 2.2. Setup

#### 2.2.1. Setup for Ewon Server

MODBUS TCP Server (the eWON publishes data through Modbus TCP)

eWON Server:  Enabled (Otherwise only the gateway is enabled)

Modbus TCP Unit Address:

This page defines the Ewon configuration when used as a Modbus TCP slave.

As described in the Tag configuration paragraph, each Tag can be published to Modbus TCP so that a Modbus TCP can read their values. This setup screen defines the Ewon address, and globally enables or disables the Modbus TCP slave feature.

Ewon Server Properties	Description
<b>Ewon server enable</b>	Globally enables or disables the Modbus TCP Server feature. If disabled, then any request from a Modbus TCP master will be rejected, even if Tags are published.
<b>Modbus TCP Unit address</b>	This feature is used by some gateway but can usually be left to 1 because Modbus TCP appears as a point to point connection.

2.2.2. Setup for Ewon IO server and Gateway - COM Setup

COM Setup

COM Port: IO Port1 (COM:1) ▼

Baud Rate: 19200 ▼

Parity: None ▼

Stop Bit(s): 1 ▼

HW Mode: Half Duplex ▼

Reply Timeout: 1000 MS

8 data bits, RTU mode

If more than one Serial port is available, you must choose on which COM the MODBUS request will be sent. This example of configuration shows an RS485 setup.

Field name	Description
COM Port	Physical COM Port allocated to the Modbus IO Server. Drop down shows alternatives only if the relevant Ewon has multiple serial ports (1).
Baud Rate	Baud rate of the serial link (has to be the same than the target device). Default is Disabled.
Parity	Parity
Stop Bit(s)	Stop bit(s)
HW Mode	HW mode (half Duplex, Full Duplex HW Handshaking, Full Duplex NO Handshaking)
Reply Timeout	Time (in msec) the Ewon waits for an answer from the connected Modbus device. The default value is 1000 msec



The numbering of the serial COM ports starts with the serial port of the Base Unit (if available) and then continues with the upper port of the first Extension Module starting from the left. A maximum of 5 serial ports are supported.



If multiple IO servers are potentially using the same serial port, the unused IO server must be cleared or the baud rate must be set to Disabled.

Example:

If Modbus and UniTE IO servers are configured to the same port, at least one of them must have its baud rate set to Disabled.

If not the case, one of the IO servers will not be able to use the serial line and it will be disabled, with an error written in the event log.

### 2.2.3. Topic configuration

Enabled

Topic A

Slave Address (unit Id):  IP Address (Blank for RTU):

Poll Rate:  M5

Enabled

Topic B

Slave Address (unit Id):  IP Address (Blank for RTU):

Poll Rate:  M5

Three topics can be used for the IO Server.  
 These topics are used to give a common property to a group of MODBUS Tag like:

- **Enable/Disable**
- **Poll rate**
- **Slave address (Modbus RTU)**
- **Unit address and TCP/IP address (Modbus TCP)**

Modbus Server Properties	Description
<b>Topic Enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Slave address</b>	The slave address is a global parameter for all the Tags of the topic. If the slave is connected with Modbus RTU, Slave address must be entered and the IP address must be blank. If the slave is Modbus TCP, its unit address and its IP address must be entered.
<b>Poll rate</b>	This defines the rate to which the Tag will be refreshed. In a complex application, we can imagine that some Tags must be refreshed every second – typically for digital input - and other every minute – typically: temperature-.

**Any slave address that is defined in the Topic configuration overwrites the slave address configured per Tag.**

**If a Tag is defined with Tag Address: 40010,5 and the global address of the topic is 5 and 10.0.0.81, the Tag is entered as Modbus RTU but it is polled as Modbus TCP.**

**So, if you need to address slaves Tag by Tag, let the topic address configuration empty.**

### 2.2.4. Advanced parameters

Advanced parameters

IOSrvData:

**MaxDeltaRegX:** max. number of registers grouped for reading  
**MaxDeltaCoilX:** max. number of coils grouped for reading  
**ErrorRetry:** max. number of retries before entering slow poll mode  
**SlowPollSkip:** number of scans skipped when in slow poll  
**PostErrorDelay:** This delay (MSec) is added each time a device responds with a framing error or a CRC error. Helps the bus to resynchronize  
**PreDelayVal:** Delay in MSec inserted before a request to a slave if previous request was sent to another slave address (not between 2 requests to the same slave).  
**PreDelayFrom:** The PreDelayVal is applied only starting from this address, put this to 256 if pre delay is not used.

Extended parameters have been added to accommodate various special operating conditions.  
 They are entered in the edit box at the bottom of the configuration page, conforming to the syntax below.

Each parameter has a default value,  
 the advanced parameter edit box must only be filled with the parameters for which default values must be changed.

Parameter name	Description
<b>PreDelayFrom</b>	Used in conjunction with the next parameter (PreDelayVal), starting at that slave address, the Ewon will insert a delay when switching from one slave address to another. If the PreDelayVal feature is not used, then the value for PreDelayFrom must be set to 256 (default value).
<b>PreDelayVal</b>	Used to define the delay (in Msec) to be inserted before a request to a slave if a request was previously sent to another slave address (not between 2 requests to the same slave). The PreDelayVal is placed only for slaves with an address higher than PreDelayFrom.
<b>ErrorRetry</b>	Defines the number of errors before the device enters in slow poll mode. (Minimum 1)
<b>SlowPollSkip</b>	Defines the number of times the slave is skipped when in slow poll mode. This delay depends on the poll rate.
<b>Gateway ayIPCTimeout</b>	Maximum event lock while waiting for a response to a modbus gateway request from the modbus IO server (router) (in Msec – minimum 1000).
<b>PostErrorDelay</b>	This delay (in Msec) is added each time a device responds with a framing error or a CRC error, in order to help the bus to perform its re synchronization (default value = 50).
<b>MaxDeltaCoilX (X= A,B,C)</b>	Maximum number of Coils that can be grouped in a request (per topic) max=256
<b>MaxDeltaRegX (X= A,B,C)</b>	Maximum number of registers that can be grouped in a request (per topic) max=124
<b>ModbusForceFunction06</b>	Set to 1 to force the use of Modbus function 06 when writing a WORD or INTERGER type into Holding Registers. Not applied on modifiers D,E,L,M,F,H Set to 0: Ewon will use Modbus function 16 when writing into Holding Registers (Default behaviour)

### 2.2.5. Additional advanced parameters

- **Gwdestaddr**

When the Ewon is used as a Modbus gateway, it uses the UnitID from the ModbusTCP request to determine the Modbus RTU destination address.

It is possible to bypass this mechanism and force all the requests received by the Ewon in ModbusTCP to be forwarded to a single ModbusRTU device (EXCEPT for requests with UnitID corresponding to the Ewon's Modbus TCP Unit address (usually 100) when the Ewon Server is enabled - see explanations about this precise point in the above configuration fields descriptions).

Every request is forwarded to the device with Slave address defined by the Modbus IO server advanced parameter called **gwdestaddr**.

If the advanced parameters textfield contains the following entry: **gw destaddr:10**

Then all the requests will be forwarded to the slave with address 10.

REMARK: the Ewon will change the address before sending the request to the slave, then it will change it back when sending the response to the master (Modbus TCP master). So the **gw destaddr** will never appear in any communication between the Master and the Ewon.

- **FastSrv**

FastSrv is a mode which allows a supervisor to read in Modbus TCP more easily the Modbus tags published by the Ewon. This mode completely changes the tag's addressing, and when activated, the Modbus addresses are no more compatible.

You have just to enter "FastSrv:1" in the Advanced Parameters text area to activate it. When done, the Modbus tags can be read as follows:

<b>X</b>	Integer (with scale factors and offset defined)
<b>X+2048</b>	Float (X+2048: 1 <sup>ST</sup> float, X+2048+2: 2 <sup>nd</sup> float, etc.)
<b>X+4096</b>	Alarm status
<b>X+6144</b>	Alarm type



- If the tag is binary read, its value is worth 0 if tag==0, and 1 if tag <>0
- Writing 0 in AlarmStatus acknowledges the alarm (will be logged by Ewon as acknowledged by administrator)
- It is not possible to write a coil in the float area (coil address range: X+2048 to X+4094)
- It is not possible to address more than 1024 registers in float.

Click on the **Update Config** button to validate your inputs or use the **Cancel** button to discard changes

- TCP port**  
 Use the 'TcpPort' parameter to change the default 502 port used when the Ewon is MODBUS TCP CLIENT.  
 If not specified, the port 502 is the default value used. This Port value is used for all the MODBUS TCP client connections.
- UsecReqDelay**  
 Number of uSec the Ewon waits after it receives a message from a slave. (Only used for Modbus RTU)  
 And so before sending another message to the same or to another slave.  
 If this parameter is 0 it will be automatically computed from the baudrate and equals to 5 chars
- MsecInterFrameTO**  
 Time, in millisec, that Ewon waits for detecting the end of the frame at reception over a serial connection.  
 This parameter can be increased in case the frame is not fully received by the Ewon
- # (Bit Selection)**  
 You can use # to read directly a bit (coil or input) on the Modbus device (Example: #1)

**2.3. Tag name convention**

IO Server configuration		
<b>Io Server Name</b>	MODBUS	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	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 used

**2.3.1. ValueName**

The Modbus IO Server Tags can be classified following ranges of values.  
 Two types of ranges are used.  
 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 are used with non-standard equipments that do not allow the reading of some of their values.

In this case, specifying the address in the "write only" ranges informs the Ewon not to 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.

A modifier can be appended after the numerical value representing the register. It will indicate how the register value should be read. The following table describes the different modifiers and their meanings.

Modifier	Description	Automatic Tag type
<b>W</b>	Reads 1 register considered as 16 bits unsigned integer (DEFAULT IF NOT SPECIFIED)	DWord
<b>I</b>	Reads 1 register considered as 16 bits signed integer	Integer
<b>D</b>	Reads 2 regs R1, R2 as a DWORD R1 is Less significant, R2 is most significant (32 bits, unsigned) (*)	DWord
<b>E</b>	Reads 2 regs R1, R2 as a DWORD R2 is Less significant, R1 is most significant (32 bits, unsigned) (*)	DWord
<b>L</b>	Reads 2 regs R1, R2 as a LONG R1 is Less significant, R2 is most significant (32 bits, signed) (*)	Integer
<b>M</b>	Reads 2 regs R1, R2 as a LONG R2 is Less significant, R1 is most significant (32 bits, signed) (*)	Integer
<b>F</b>	Reads 2 regs R1, R2 as a FLOAT R1 is Less significant, R2 is most significant (32 bits, signed)	Float
<b>H</b>	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 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:**

Address	meaning
40001,10	access the <i>Holding Register</i> at address 1 on the UnitID 10
1,11	access the <i>Coil</i> at address 1 on the UnitID 11
+320234,12	access the <i>Input Register</i> at address 20234 on the UnitID 12
40001,100,10.0.0.53	access the <i>Hoding Register</i> at address 1 on the UnitID 100 at IP address 10.0.0.53
40010L,12	access the LONG <i>Holding Register</i> at address 10 (and 11) on the UnitID 12
40008F,15	access the FLOAT <i>Holding Register</i> at address 8 (and 9) on the UnitID 15

• **STATUS TAG:**

The STATUS Tag is a special Tag that returns information about the current state of the communication for a given device. As for other Tags, the status Tag ItemName 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	meaning
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

### 2.3.2. Slave Address

This is the address of the slave device that you want to access.  
It is a number from 0 to 255.

**Example:**

<b>30001,11</b>	Polls a RTU device at address 11.
-----------------	-----------------------------------

### 2.3.3. IP Address

This is the IP address of the device on an Ethernet network.  
It is composed of 4 numbers separated by a dot.

**Example:**

<b>30001,11,10.0.0.50</b>	Polls a device configured with IP address 10.0.0.50 and with Modbus slave address 11.
---------------------------	---

### 2.3.4. Device specific information



**For new users of WAGO modules:**

Keep in mind that coil read and write don't use the same address (offset of 0x200); please consult the Wago™ documentation.

**Example:**

If you use Wago™ systems with two digital inputs and two digital outputs, inputs have addresses 1 and 2, and outputs have the same. The only way to distinguish them it's the read-only access or R/W access.  
Here under, Tags address are related to a device with a node ID 11.

Tag	Modbus address	Comment
MB_DigIn1	10001,11	Digital input module 1 -- read-only
MB_DigIn2	10002,11	Digital input module 2 -- read-only
MB_DigOut1	00001,11	Digital output module 1 for writing - Encode all leading zeroes!
MB_DigOut1Read	10513,11	Digital output module 1 for reading only
MB_DigOut2	00002,11	Digital output module 2 for writing Encode all leading zeroes!
MB_DigOut2Read	10514,11	Digital output module 2 for reading only

In *View I/O page*, you can change the value of MB\_DigOut1 with the update link (set to 1), and if you do that, you view that the value read is always 0.

#### Why?



Because the Ewon reads the value at the WAGO address 1 (thus, DigIn1)!  
If you want to read the state of the DigOut1, you must read it at WAGO address 513!

The same remark is applied for analog Modbus registers.  
It's the documented behavior of Wago™-Modbus modules; keep it in mind.

### 3. NETMPI IO Server

#### 3.1. Introduction

NETMPI IO server is used to communicate with SIEMENS PLC.

The Ewon will connect to the PLC's MPI interface by means of the eLINK adapter.  
 The eLINK adapter is an interface with an MPI interface on one side and an Ethernet TCP/IP interface on the other side.  
 The eLINK interface will be configured with an IP that the Ewon will use to poll data.  
 In addition to the IP address, the Ewon will need the MPI address of the PLC on the MPI bus.

Using that eLINK interface, it is possible to poll different types of item in the PLC. These items types are described below.

#### 3.2. Setup

Topic A

Enabled

Global Device Address:  Enter eLink IP address

Poll Rate:  MS Default: 2000

As for the Modbus and Unite protocols, it is possible to define topics, so that a single configuration can be applied to several Tags.

#### 3.3. Tag name convention

IO Server configuration		
<b>Topic enabled</b>	NETMPI	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName,RemoteStationAddress,eLinkIpAddress	PLC Address is defined Tag by Tag
	ValueName	Topic PLC Address is used

The Item Name can contain the PLC address where the value is polled, or not.  
 If address is also specified at topic level, the address specified at Tag level will be ignored.

### 3.3.1. ValueName

Syntax	Description	Automatic Tag type
<b>DBxBy</b>	Data block x Byte offset y (0-255)	DWord
<b>DBxWy</b>	Data block x Word offset y (0 - 65535)	DWord
<b>DBxFy</b>	Data block x Float offset y (+-3.4e38)	Float
<b>DBxDy</b>	Data block x DWord offset y (-2147483648->2147483647) (*)	Integer
<b>MxB</b>	Memento x as byte	DWord
<b>MxW</b>	Memento x as word	DWord
<b>MxF</b>	Memento x as real	Float
<b>MxD</b>	Memento x as double (*)	Integer
<b>Tx</b>	Timer x (0-65535)	DWord
<b>Cx</b>	Counter x (0-65535)	DWord



(\*) To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

- Bit access modifier:**

In any of the above items, it is possible to access a single bit (except for float items). #x must be appended to the Value Name. (Bit index goes from 0..31)  
The syntax can be used for reading bits and for writing them as well.

**Example:**

DB1W13#3 represents bit 3 of W13 in DB 1

**REM:**

Bit index range depends on a referenced item (for W, bit index maximum value is 15)

- Status register:**

The STATUS Tag is a special Tag that returns information about the current state of the communication for a given device. As for other Tags, the status Tag ValueName is composed of:

**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 analog.

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.

### 3.3.2. Device Address

The device address is either appended to the ValueName in the Item Name definition, or entered in the Topic global address fields. The device address is composed of: RemoteStationAddress,eLinkIpAddress:

- RemoteStationAddress:** is the PLC's MPI address
- eLinkIpAddress:** is the eLINK IP address.



If the PLC address is defined at the Topic level, it can be omitted in the Tag definition. In that case the Tag name will only contain the "ValueName".  
If the PLC address is specified at the Topic level, it will replace any address defined Tag by Tag.

## 4. UNITE IO Server

### 4.1. Introduction

The Ewon RS485 link can be configured as an UNITELWAY SLAVE INTERFACE.

When the Baud Rate in the UNITE IO Server is set to a value different than "Disabled", the Unitelway slave module in the Ewon will be enabled.

This Unitelway slave provides 2 features:

- Poll items in a Unitelway capable device.**  
 The device can be the Unitelway master itself or a device addressable through the Unitelway master on the XWay network.
- Forward XIP requests from TCP/IP XIP to Unitelway bus and thus, act as a gateway between XIP and Unitelway.  
 Using that feature, it is possible to access a PLC connected to the Ewon's Unitelway link by connecting PL7PRO using the XIP driver started with the Ewon IP address as destination.

### 4.2. Setup

#### 4.2.1. Communication Setup

COM-Setup

COM Port:	IO Port1 (COM:1)	
Baud Rate:	19200	Default 9600
Parity:	Odd	Default: ODD
Stop Bit(s):	1	Default: 1
HW Mode:	Half Duplex	Default: Half Duplex
Master response timeout:	<input type="text"/> MS	20..60000, default: 1000
Rx message timeout:	<input type="text"/> MS	1000..60000, default 3000
Tx message timeout:	<input type="text"/> MS	1000..60000, default 3000
	<input checked="" type="checkbox"/> Force Unitelway V2	If unchecked, V1 protocol will be used
	<input type="checkbox"/> Disable 0,254 translation	If not checked, the gateway uses the request's Network & Station destination address
AD0:	<input type="text" value="4"/>	Link address of eWON on unitelway bus (default: 4)
XWAY Network.Station:	<input type="text" value="13"/> . <input type="text" value="14"/>	(0..127),(0..63)

If more than one Serial port is available, you must choose on which COM the Unitelway request will be sent.

Field name	Description
<b>COM Port</b>	Physical COM Port allocated to the Unite IO Server. Drop down shows alternatives only if the relevant Ewon has multiple serial ports (1).
<b>Baud Rate</b>	Baud rate of the serial link (has to be the same than the target device). Default is 9600..
<b>Parity</b>	The parity to apply: none / odd / even. This field is set by default to Odd, as in the main cases in a typical UniTE topology. However, the Ewon allows you to define a different parity type (Even or None), in case this is needful to comply with your industrial network installations.
<b>Stop Bit(s)</b>	Number of stop bits
<b>Master Response Timeout</b>	Maximum time the Ewon will wait for a valid message from the Unitelway master. This value can be critical for a correct operation, depending on the responsiveness of the master. A value of 1000 should be selected to guarantee a correct operation.
<b>Rx message timeout (MSEC)</b>	Maximum time between a request is posted and the response is received
<b>Tx message timeout (MSEC)</b>	Maximum time for a request to be sent

Field name	Description
<b>Force UnitelWay V2</b>	If checked, the Ewon will initiate a communication in V2 with the devices. When used with a TSX PLC, this check box can be left unchecked.
<b>ADO</b>	Link address base. The Ewon will respond to AD0 and AD0+1 on the Unitelway link. The Ewon will act as an Unitelway slave, it will respond to 2 consecutive link addresses AD0 and AD0+1, doing this improves the throughput of data across the Ewon when acting as a gateway.
<b>Xway Network Station</b>	Address of the Ewon on the XWAY network. When acting as an XIP to Unitelway gateway, the Ewon will only respond the XWay network station defined here. Any XIP frame addressed to another network station will be ignored.

(1)For the Ewon Flexy: The numbering of the serial COM ports starts with the serial port of the Base Unit (if available) and then continues with the upper port of the first Extension Module starting from the left. A maximum of 5 serial ports are supported.



When there are multiple IO servers potentially using the serial line, the unused IO servers must be cleared or the baud rate must be set to **Disabled**.

**Example:**

If Modbus and UniTE IO servers are available, at least one of them must have its baud rate configured to Disabled. If not, one of the IO servers will not be able to use the serial line and it will be disabled, with an error written in the event log.

**4.2.2. Topic configuration**

Topic A
 Enabled

Global Device Address:  Enter Gate

Poll Rate:  M5 Default: 2000

Three topics can be used for the IO Server. These topics are used to give a common property to a group of UNITE Tags such as:

- **Enable/Disable**
- **Poll rate**
- **Global Slave address**

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Global Slave Address</b>	This Slave address is a global parameter for all the Tags of the topic. See below for the Device Address syntax. If an address is specified here, it will replace (overload) the address defined Tag by Tag.
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag name must be refreshed every second – typically for digital input - and other every minute – typically: temperature-.

### 4.3. Tag name convention

IO Server configuration		
IO Server Name	UNITE	
Topic Name	A	
	B	
	C	
Item Name	ValueName,Network,Station, Gate,Module,Channel	PLC Address is defined Tag by Tag (and the gate requires a 5 level addressing)
	ValueName,Network,Station, Gate	PLC Address is defined Tag by Tag
	ValueName	Topic PLC Address is used

The Item Name can contain the PLC address where the value is polled, or not.

If the address is also specified at topic level, the address specified at Tag level will be ignored.

If the Topic address is not defined either, then address 0,254,0 will be used.

#### 4.3.1. Value Name

Value name follows the syntax below:

Syntax	Description	Automatic Tag type
<b>MWxW</b>	Internal data word 16 bits (unsigned)	DWord
<b>MWxI</b>	Internal data word 16 bits (signed)	Integer
<b>MWxD</b>	Internal data word 32 bits as DWORD (unsigned) (*)	DWord
<b>MWxF</b>	Internal data word 32 bits as IEEE float	Float
<b>MWxL</b>	Internal data word 32 bits as LONG (signed) (*)	Integer
<b>SWxW</b>	System data word 16 bits (unsigned)	DWord
<b>SWxI</b>	System data word 16 bits (signed)	Integer
<b>SWxD</b>	System data word 32 bits as DWORD (unsigned) (*)	DWord
<b>SWxL</b>	System data word 32 bits as LONG (signed) (*)	Integer
<b>Mx</b>	Internal data bit	DWord
<b>Sx</b>	System data bit	DWord



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

SW type cannot be formatted as float. The Ewon allows you to optimize the requests in case you need to read a lot of Tags that have been created on the UniTelWay device.

Imagine you have 100 Tags to read, the Ewon will group the Tags within a predefined limit in order to make the less as reading operations as possible.

The number of Tags that can be read depends of the types of words or bits that have to be read:

- SW and MW types: by groups of 50
- S and M types: by groups of 200



It is possible to read one bit from a word, the syntax to add is as follows: #0 to #31

That means that if you want to read the fifth bit from internal data word 16 bits unsigned that you address MW0, you have to add "#4" at the end of the address: MW0#4. The type of words for which this syntax can be applied are: MWxW, MWxI, MWxD, MWxL, SWxW, SWxI, SWxD and SWxL (please report to the table above).

Examples:

Address	point to
MW10 or MW10W	the Internal data Word (16bits) at address 10
MW10D	the Internal data Word (as DWord 32bits) at address 10
MW10W#5	the bit 5 (the sixth) from the Word at address 10
SW2 or SW2W	the System data Word (16bits) at address 2

- **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of the communication for a given device. As for other Tags, the status Tag ValueName is composed of:

**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 analog.

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.

#### 4.3.2. The device address syntax

The Device Address is used in the topic definition or in the Tag definition.

If used in the Tag definition, it will be separated from the value name by a coma (',')

- **Network,Station,Gate**

**Example:** 0,254,0 to access the UnitelWay device directly connected on the Ewon

or

- **Network,Station,Gate,Module,Channel**

The second case applies to addresses with 5 levels:

- **Network:** 0..127
- **Station:** 0..63
- **Gate:**
- **Module:**
- **Channel:**

Module and channel can be omitted if not required.



If the address is not specified, 0,254,0 will be used.



If an address is specified in a Topic definition it will replace any address defined Tag by Tag.

## 5. DF1 IO Server

### 5.1. Introduction

The Ewon serial link can be configured as a DF1 INTERFACE. When the Baud Rate in the DF1 IO Server is set to a value different than "Disabled", the DF1 module in the Ewon will be enabled.

This DF1 module provides 3 features:

- Poll items in SLC50x and PLC5 controllers using PCCC requests.
- Forward PCCC requests from EIP (TCP/IP) to DF1 bus and thus, acts as an adapter between EIP and DF1.

Thanks to that feature, it is possible to access a PLC connected to the Ewon's DF1 link by connecting RSLogix 500 using RSLinx TCP driver started with the Ewon IP address as destination.

- Poll DH+ device using ControlLogix gateway

The DF1 IO Server must be configured in **Full Duplex mode** (Ewon serial link must be configured in RS232 mode, with dipswitch).



Please check in the "Rockwell compatibility matrix" if your device is supported.

## 5.2. Setup

### 5.2.1. Communication Setup

COM Setup

COM Port:	<input type="text" value="IO Port1 (COM:1)"/>	
Baud Rate:	<input type="text" value="9600"/>	Default 9600
Parity:	<input type="text" value="None"/>	Default: NO
Stop Bit(s):	<input type="text" value="1"/>	Default: 1
Frame Error Detection:	<input type="text" value="CRC"/>	Default: CRC
HW Mode:	<input type="text" value="Full Duplex NO Handshaking"/>	Default: Full Duplex
Master response timeout:	<input type="text" value=""/> M5	20..60000, default: 1000
Rx message timeout:	<input type="text" value=""/> M5	1000..60000, default 3000
Tx message timeout:	<input type="text" value=""/> M5	1000..60000, default 3000
DF1 Address:	<input type="text" value=""/>	Device address of eWON on DF1 link (0..254, default: 4)
Destination DF1 Address:	<input type="text" value=""/>	Device address of destination on DF1 link when EIP is used (0..254, default: 1)
	<input checked="" type="checkbox"/> Bridge EIP Connection	Enabled

If more than one Serial port is available, you must choose which COM the DF1 request will be sent to.

The following parameters can be modified:

Field name	Description
COM Port	Physical COM Port allocated to the DF1 IO Server. Note: In the example above, the port selection drop down is not displayed because the Ewon has only one single serial port. For multiple serial ports see remark (1).
Baud Rate	Select the Baud rate from 1200 to 38400 Bauds
Parity	The parity to apply: none / odd / even
Stop Bit(s)	Number of stop bits
Frame Error Detection	Cyclic Redundancy Check (CRC) or Block Check Character (BCC)
HW mode	Full Duplex no handshaking or Half duplex

Field name	Description
<b>Master Response Timeout</b>	Maximum time the Ewon will wait for a valid message from the DF1 master. This value can be critical for a correct operation, depending on the responsiveness of the master. A value of 1000 should be selected to guarantee correct operations
<b>Rx message timeout (MSEC)</b>	Maximum time between a request is posted and the response is received
<b>Tx message timeout (MSEC)</b>	Maximum time for a request to be sent
<b>Ewon DF1 address</b>	The Ewon's device address on the DF1 link When the Ewon acts as a DF1 slave, it will respond to 2 consecutive link addresses; doing this improves the throughput of data across the Ewon when acting as a gateway. (Ewon DF1 address and Ewon DF1 address +1)
<b>Destination DF1 address</b>	Device Address of Destination on DF1 link when EIP is used
<b>Bridge EIP connection</b>	When bridging is configured, EIP connections requests are forwarded to the PLC on the DF1 interface (mandatory for Logix family). This is the default configuration. When not configured, connection requests are canceled by the Ewon (must be disabled for SLC family).

(1)The numbering of the serial COM ports starts with the serial port of the Base Unit (if available) and then continues with the upper port of the first Extension Module starting from the left. A maximum of 5 serial ports are supported.



When there are multiple IO servers potentially using the serial line, the unused IO servers must be cleared or the baud rate must be set to **Disabled**.

**Example:**

If Modbus **and** DF1 servers are available, at least one of them must have its baud rate configured to Disabled. If not, one of the IO servers will not be able to use the serial line and it will be disabled, with an error written in the event log.

**5.2.2. Topic configuration**

✔ Enabled

Topic A

Global Device Address:  Enter Link: nodeID or IP address

Poll Rate:  MS Default: 2000

✔ Enabled

Topic B

Global Device Address:  Enter a sequence of (Port,Link) pairs

Poll Rate:  MS Default: 2000

✔ Enabled

Topic C

Global Device Address:  Enter Link: nodeID or IP address

Poll Rate:  MS Default: 2000

Three topics can be used for the IO Server. These topics are used to give a common property to a group of DF1 Tags like:

- **Enable/Disable**
- **Poll rate**
- **Destination Device Type and Address**

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables the polling of all the Tags in the topic
<b>Destination Device Type and Address</b>	The Destination Device Type and Address is a global parameter for all the Tags of the topic. See below for the Device Address syntax. If an address is specified here, it will replace (overload) the address defined Tag by Tag
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag name must be refreshed every second – typically for digital input - and other every minute – typically: temperature-

### 5.2.2.1. DF1: serial link

The Destination Device Type and address are:

- SLC500-x (where x is the address of your SLC500 or Micrologix Device, range 0..254)
- PLC5-x (where x is the address of your PLC5 Device, range 0..254)

Example :

Topic A Destination = SLC500-1  
Topic B Destination = PLC5-6

For accessing a Micrologix, use the SLC500 syntax  
Topic C Destination = SLC500-3 (accessing the Micrologix at address 3)

### 5.2.2.2. DF1: Ethernet routing

Thanks to the ABLogix IO Server, the Ewon is now able to poll data on SLC500/PLC5 devices by its Ethernet link. Even SLC500 connected behind your ControlLogix Network becomes available for polling.

With this use of DF1 IO Server, the Serial config can be left without configuration (baud rate=disabled), only one Topic enabled is required. You need to use a similar syntax than in ABLogix IO Server.

*[DeviceType-] IP Address [, Port, Link]*

- **DeviceType** = type of PLC ( SLC500 or PLC5)
  - **IP Address** = address on your Ethernet network (i.e.: 10.0.0.50)
  - **Port** = value from 1 to 3 representing: 1 = Backplane, 2 = Channel A, 3 = Channel B
  - **Link** could be:
    - Slot: representing the Slot on the Backplane (0=CPU)
    - Node ID: value from 0 to 99 (for ControlNet)
    - IP address
- To reach a SLC500 with Ethernet address 10.0.0.60:  
Topic A Destination = **10.0.0.60 or SLC500-10.0.0.60**
- To reach a PLC5 with Ethernet address 10.0.0.65:  
Topic A Destination = **PLC5-10.0.0.65**
- To reach a SLC500 connected behind a ControlLogix:  
Topic A Destination = **10.0.0.80,1,3,2,45 or SLC500-10.0.0.80,1,3,2,45**
- 10.0.0.80 = IP address of the ControlLogix
  - 1 = BackPlane
  - 3 = Slot 3 (COM card in ControlNet)
  - 2 = Channel A (of the Card present in Slot 3)
  - 45 = NodeID of the SLC500 (in the ControlNet network)
- To reach a PLC5 connected behind a ControlLogix:  
Topic A Destination = **PLC5-10.0.0.90,1,3,2,44**
- 10.0.0.90 = IP address of the ControlLogix
  - 1 = BackPlane
  - 3 = Slot 3 (COM card)
  - 2 = Channel A (of the Card present in Slot 3)
  - 44 = NodeID of the PLC5 (in the COM network)

### 5.2.2.3. DH+ polling using ControlLogix gateway

This feature allows the Ewon to poll PLC Tags located on a DH+ network, behind a ControlLogix equipped with one or more DHRIO cards. Polling is done using Ewon EIP protocol.

The DF1 IO server is used to access the "SLC500" PLC family. The ABLOGIX IO server is used to access the "LOGIX" PLC family.

The syntax *[Device Type-] IP address [,Port,Link]* is still applicable but in this particular case: the [,Port,Link] part becomes mandatory. The DH+ channel used to reach the "SLC500" PLC must be specified using the "A" or "B" character.

Example:

To reach a SLC500 with DH+ node Id 5, using channel A of a DHRIO card at slot 9 of a ControlLogix at IP address 10.0.30.21.  
Topic A Destination = **SLC500-10.0.30.21,1,9,A,5**

- 10.0.30.21 = IP address of the ControlLogix
- 1 = BackPlane
- 9 = Slot 9 (COM card)
- A = Channel A (of the Card present in Slot 9)
- 5 = NodeID of the SLC500 (in the COM network)

### 5.3. Tag name convention

IO Server configuration		
<b>IO Server Name</b>	DF1	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, DeviceType-DeviceAddress	Device Type and Address are defined Tag by Tag Only device Type SLC500 is supported Device Address is a number between 0..254
	ValueName	Topic PLC Address is used

The Item Name can contain the PLC address where the value is polled, or not.

If the address is also specified at the topic level, the address specified at Tag level will be ignored.

#### 5.3.1. Value Name

##### 5.3.1.1. General Description

The general format of value names for data from SLC-500 controllers matches the naming convention used by the programming software. The format is shown below (The parts of the name shown between square brackets are optional).

**General Value Name Format:**    X[file]:element [.field] [/bit]

X: Identifies the file type. The below table summarizes the valid file types, default file number for each type and the fields allowed.

Symbol	File Type	Default file Number	Fields	Automatic Tag type
O	Output	0		DWord
I	Input	1		DWord
S	Status	2		DWord
B	Binary	3		DWord
T	Timer	4	.PRE, .ACC, .EN, .TT, .DN	DWord
C	Counter	5	.PRE, .ACC, .CU, .CD, .DN, .OV, .UN, .UA	DWord
R	Control	6	.LEN, .POS, .EN, .DN, .ER, .UL, .IN, .FD	DWord
N	Integer	7		DWord
F	Floating	8		Float
A	ASCII	none		DWord
ST	String	none		String
L	Long Word			DWord

- **File:** File number must be 0-255 decimal
- **Element:** Element number within the file
- **Field:** Valid only for Counter, Timer and Control files
- **/bit:** Valid for all types except Floating



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

## 5.3.1.2. Output File Items

Output File Item Format: O[n]:e.s[/b]
---------------------------------------

- "n" represents the file number and is optional. Value is always zero
- "e" indicates the element number in the file (0..30)
- "s" indicates the sub-element number (0..255)
- "b" specifies the bit (0..15) decimal



For the Micrologix Series, the syntax for Output File Items is slightly different: Basically, it drops 1 digit to show O:1 instead of O:0.1

Actually, the Ewon reads consecutive words.

Example for the Micrologix 1400

PLC Addressing	Ewon Addressing
O:0.2	O:2
O:1.0	O:6

This example shows that when stepping to another output card, the incrementation continues in the Ewon (from O:5 to O:6) while in the PLC itself it increments the first digit (from O:0.5 to O:1.0).

## 5.3.1.3. Input File Items

Input File Item Format: I[n]:e.s[/b]
--------------------------------------

- "n" represents the file number and is optional. Value is always one
- "e" indicates the element number in the file (0..30)
- "s" indicates the sub-element number (0..255)
- "b" specifies the bit (0..15) decimal



For the Micrologix Series, the syntax for Output File Items is slightly different: Basically, it drops 1 digit to show O:1 instead of O:0.1

Actually, the Ewon reads consecutive words.

Example for the Micrologix 1400

PLC Addressing	Ewon Addressing
I:0.7	I:7
I:1.0	I:8

This example shows that when stepping to another input card, the incrementation continues in the Ewon (from I:7 to I:8) while in the PLC itself it increments the first digit (from I:0.7 to I:1.0).

## 5.3.1.4. Status File Items

Status File Item Format: S[n]:e[/b]
-------------------------------------

- "n" represents the file number and is optional. If not specified, it is assumed to be 2
- "e" indicates the element number in the file (0..255)
- "b" specifies the bit (0..15) decimal

## 5.3.1.5. Binary File Items

Binary File Item Format: B[n]:e/b
-----------------------------------

- "n" represents the file number & is optional. If not specified, it is assumed to be 3, otherwise must be between 3 and 255 decimal
- "e" indicates the element number in the file (0..255)
- "b" specifies the bit (0..15) decimal



The format B[n]/b is not supported.

## 5.3.1.6. Timer File Items

Timer File Item Format: T[n]: e [.f] [/b]
---

- "n" represents the file number and is optional.  
If not specified, it is assumed to be 4, otherwise must be between 4 and 255 decimal
- "e" indicates the element number (3 words per element) in the file (0..255)
- "f" identifies one of the valid values for timer fields specified in the table above.  
If omitted it is assumed to be the word containing the status bits
- "b" specifies the bit (0..15) decimal

## 5.3.1.7. Counter File Items

Counter File Item Format: C[n]: e [.f] [/b]
---

- "n" represents the file number and is optional.  
If not specified, it is assumed to be 5, otherwise must be between 5 and 255 decimal
- "e" indicates the element number (3 words per element) in the file (0..255)
- "f" identifies one of the valid values for counter fields specified in the table above.  
If omitted it is assumed to be the word containing the status bits
- "b" specifies the bit (0..15) decimal

## 5.3.1.8. Control File Items

Counter File Item Format: C[n]: e [.f] [/b]
---

- "n" represents the file number and is optional. If not specified, it is assumed to be 6, otherwise must be between 6 and 255 decimal
- "e" indicates the element number (3 words per element) in the file (0..255)
- "f" identifies one of the valid values for counter fields specified in the table above.  
If omitted it is assumed to be the word containing the status bits
- "b" specifies the bit (0..15) decimal

## 5.3.1.9. Integer File Items

Integer File Item Format: N[n]:e[/b]
--------------------------------------

- "n" represents the file number and is optional. If not specified, it is assumed to be 7, otherwise must be between 7 and 255 decimal
- "e" indicates the element number in the file (0..255)
- "b" specifies the bit (0..15) decimal

## 5.3.1.10. Floating File Items

Floating File Item Format: F[n]:e
-----------------------------------

- "n" represents the file number and is optional. If not specified, it is assumed to be 8, otherwise must be between 8 and 255 decimal
- "e" indicates the element number in the file (0..255)

## 5.3.1.11. ASCII File Items

ASCII File Item Format: An:e[/b]
----------------------------------

- "n" represents the file number and is not optional
- "e" indicates the element number in the file (0..255)
- "b" specifies the bit (0..15) decimal

## 5.3.1.12. String File Items

String File Item Format: ST[n]:e
----------------------------------

- "n" represents the file number and is optional.
- "e" indicates the element number in the file (0..255)

**Note:** The size of a string Tag is limited to 1024 bytes. If you read a string greater than 1024 bytes, the value will be truncated at 1021 bytes + "..." (3 dots) to show that the string was truncated.

**5.3.1.13. Status register**

The STATUS Tag is a special Tag that returns information about the current state of the communication for a given device. As for other Tags, the status Tag ValueName is composed of:

**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 analog.

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.

**5.3.2. Destination Device Type and Address**

The Device Address is used in the topic definition or in the Tag definition. If used in the Tag definition, it will be separated from the value name by a coma (',').

**Format: DeviceType-DeviceAddress**

The Destination Device Type and address are:

- *SLC500-x* (where x is the address of your SLC500 or Micrologix Device, range 0..254)
- *PLC5-x* (where x is the address of your PLC5 Device, range 0..254)

**Example:**

`Topic A Destination = SLC500-1`  
`Topic B Destination = PLC5-6`

`For accessing a Micrologix, use the SLC500 syntax`

`Topic C Destination = SLC500-3 (accessing the Micrologix at address 3)`

**Format: [DeviceType-] IP address [,Port,Link]**

**Example:**

10.0.0.80 to access a SLC500 (or a Micrologix)

## 6. FINS IO Server

### 6.1. Introduction

The FINS IO Server includes the configuration of:

**The Ewon as a FINS Hostlink client (master)  
to give access on values in CSCJ series OMRON equipments reachable using Ewon serial port(s).**

**The Ewon as a FINS TCP/UDP client (master)  
to give access on values in CSCJ OMRON equipments reachable using Ewon Ethernet interface.**

**The Ewon as a FINS TCP/UDP server acting as a gateway ay between the Ethernet/PPP interface and the serial interface  
(used to connect remotely programming/monitoring software to OMRON FINS supporting equipment reachable using Ewon serial ports).**

The FINS IO Server is designed to provide simultaneous access to OMRON equipments on its serial interface and Ethernet interface. The correct protocol will depend on the topic that the Tag belongs to. UDP and TCP protocols can be mixed as well on the Ethernet interface.

When the Baud Rate in the FINS IO Server is set to a value different than "Disabled", the serial Hostlink Client will be enabled. The

FINS IO Server can be configured in 3 modes:

- **Full Duplex mode (Ewon serial link must be configured in RS232 mode) without HW handshaking**
- **Full Duplex mode with HW handshaking**
- **Half Duplex slave mode (RS485 mode)**

### 6.2. Setup

#### 6.2.1. Communication Setup

COM Setup

COM Port:	<input type="text" value="IO Port1 (COM:1)"/>	
Baud Rate:	<input type="text" value="9600"/>	Default 9600
Parity:	<input type="text" value="Even"/>	Default: EVEN
Databits:	<input type="text" value="7"/>	Default: 7
Stop Bit(s):	<input type="text" value="2"/>	Default: 2
HW Mode:	<input type="text" value="Full Duplex NO Handshaking"/>	Default: Full duplex no Handshaking
Master response timeout:	<input type="text" value="50.50000"/> MS	50..50000, default: 3000
Ethernet FINS network:	<input type="text"/>	0..127, default: 0
Ethernet FINS node:	<input type="text"/>	0..254, default: 0
Serial FINS network:	<input type="text"/>	0..127, default: 0
Serial FINS node:	<input type="text"/>	0..254, default: 0

The following parameters can be modified:

Field name	Description
COM Port	Physical COM Port allocated to the FINS IO Server. Note: The port selection drop down is not displayed if the Ewon has only one single serial port. For multiple serial ports see remark (1).
Baud Rate	Disabled, 1200, 2400, 4800, 9600, 19200, 38400, 57600
Parity	None, Odd, Even
Data Bits	7, 8
Stop Bit(s)	1,2
HW mode	Full Duplex no HW handshaking, Full Duplex HW handshaking, Half Duplex
ReplyTimeout	The maximum time the Ewon will wait for a valid FINS message response (applicable for Ethernet and serial interface).
Ethernet FINS network	Source Network Address (SNA) filled in a FINS request message originating from the Ewon and sent out on the Ethernet interface.
Ethernet FINS node	Source Node Address (SA1) filled in a FINS request message originating from the Ewon and sent out on the Ethernet interface. It uniquely identifies the Ewon on the Ethernet network.
Serial FINS network	Source Network Address (SNA) filled in a FINS request message originating from the Ewon and sent out on the serial interface.
Serial FINS node	Source Node Address (SA1) filled in a FINS request message originating from the Ewon and sent out on the serial interface. It uniquely identifies the Ewon on the serial network.

(1)The numbering of the serial COM ports starts with the serial port of the Base Unit (if available) and then continues with the upper port of the first Extension Module starting from the left. A maximum of 5 serial ports are supported.



When there are multiple IO servers potentially using the serial line, the unused IO servers must be cleared or the baud rate must be set to **Disabled**.

**Example:** if Modbus and FINS servers are available, at least one of them must have its baud rate configured to Disabled. If not, one of the IO servers will not be able to use the serial line and it will be disabled, with an error written in the event log.

### 6.2.2. Topic Configuration

Enabled

Protocol: Fins TCP      Default: SERIAL

Global Device Address: 0,0,192.168.120.148      Enter PLC address: FINS network, Fins Node, IP address (on ethernet link) or HostLinkId (on seriallink)

Poll Rate: 2000 MS      Default: 2000

Enabled

Protocol: Fins Serial      Default: SERIAL

Global Device Address: 0,0,2      Enter protocol dependent device address

Poll Rate: 2000 MS      Default: 2000

Three (3) topics can be used for the IO Server.  
 These topics are used to give a common property to a group of FINS Tags like:

- **Enable/Disable**
- **Protocol**
- **Global Device Address**
- **Polling Rate**

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables the polling of all the Tags in the topic.
<b>Protocol</b>	Protocol used for the tags belonging to this topic: FINS Serial, FINS UDP, FINS TCP.
<b>Global Device Address</b>	See below for the Device Address Syntax. If an address is specified here, it will replace (overload) the address-defined Tag by Tag.
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

### 6.2.3. Gateway Configuration

The following parameters can be modified:

Gateway Configuration

FINS TCP server node:  0..254, default: 1

**Routing Entry**

FINS Destination Network: <input type="text"/>	Relay Node: <input type="text"/> (0..127, 0..254)
FINS Destination Network: <input type="text"/>	Relay Node: <input type="text"/> (0..127, 0..254)
FINS Destination Network: <input type="text"/>	Relay Node: <input type="text"/> (0..127, 0..254)
FINS Destination Network: <input type="text"/>	Relay Node: <input type="text"/> (0..127, 0..254)

<b>FINS TCP Server Node</b>	The Ewon FINS server node address used during FINS TCP session establishment (exchange of the FINS node address messages, and allocation of a FINS TCP Client node if necessary).
<b>Routing Entry 1..4</b>	For each defined destination network, gives the matching relay destination node. This is used to fill in the host link unit ID in the hostlink frame which encapsulates the FINS message sent out on the serial interface.

### 6.3. Tag Name Convention

IO Server configuration		
<b>IO Server Name</b>	FINS	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, FINS Network, FINS Node, Hostlink or Ip Address	If FINS Serial has been chosen at topic level, Hostlink value has to be defined. If FINS UDP or FINS TCP has been chosen at topic level, IP address has to be defined.
	ValueName	Topic PLC Address is used.

The Item Name can contain the PLC address where the value is polled, or not.

If address is also specified at the topic level, the address specified at the Tag level will be ignored.

#### 6.3.1. Value Name

##### 6.3.1.1. General Description

The format of value names for data from CS/CJ OMRON controllers is shown below. Its is based on the naming convention used by the CX Programmer programming software. The format is shown below (The parts of the name shown in square brackets are optional).

<b>General Value Name Format:</b> <b>X[bank number:][modifier]word address[#bit address] *</b>
--

- **X identifies the Memory area acronym.**
- **[bank number:] is only supported by the E memory area. Values 0 to max values for memory area.**
- **[modifier] allow you to interpret the data in a specific type (Word if omitted).**
- **[#bit address] is only supported by A, D, CIO, H and W memory areas. Values 0 to 15.**

*\*items between brackets "[]" are optional (the brackets should not be used!)*

##### Supported memory areas:

<b>X</b>	Memory area
<b>A</b>	Auxiliary area
<b>C</b>	Counter area
<b>CIO</b>	Core I/O area
<b>D</b>	Data Memory area
<b>E</b>	Extended Data Memory area
<b>H</b>	Holding area
<b>T</b>	Timer Area
<b>W</b>	Work Area

Modifier:

Symbol	Modifier	value range	Automatic Tag type
<b>W</b>	Word (default)	0 .. 65535	DWord
<b>S</b>	signed Word	-32768 .. 32767	Integer
<b>D</b>	DWord	0 .. 4294967296 (*)	DWord
<b>L</b>	signed DWord	-2147483648 .. 2147483647 (*)	Integer
<b>F</b>	Float	+/- 3.4e38	Float



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

Examples:

Address	point to
<b>A5</b>	the Word at address 5 in the Auxiliary Area (equivalent to AW5)
<b>AS5</b>	the Signed Word at address 5 in the Auxiliary Area
<b>A2#5</b>	the bit 5 from the Word at address 2 in the Auxiliary Area
<b>E0:S3</b>	the signed Word at address 3 from the Extended Data Memory 0
<b>C5</b>	the Counter at address 5
<b>D4</b>	the Data Memory at address 4

• **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for the other Tags, the status Tag ValueName is composed of:

**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 analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

**6.3.2. Global Device Address**

The global device address is used in the topic definition or in the Tag definition. If it is used in the Tag definition, it will be separated from the value name by a coma.

**Format: Network,Node,HostLin or Network,Node,IP**

**Topic A**  Enabled

Protocol: Fins TCP Default: SERIAL

Global Device Address: 0,0,192.168.120.148 Enter PLC address: FINS network, Fins Node, IP address (on ethernet link) or HostLinkId (on seriallink)

Poll Rate: 2000 MS Default: 2000

**Topic B**  Enabled

Protocol: Fins Serial Default: SERIAL

Global Device Address: 0,0,2 Enter protocol dependent device address

Poll Rate: 2000 MS Default: 2000

## 7. S5-AS511 IO Server

### 7.1. Introduction

This IO server is intended for use with Siemens S5 PLCs communicating via the front programming port using AS511 protocol. The AS511 protocol is specific for each Siemens device.

This IO Server has been designed to operate with a set range for Siemens equipment. Use of the IO Server on devices other than those listed is not recommended and not supported.

The Siemens S5 PLC family has a unique memory structure. Data within the PLC are not located at fixed locations within the PLC's memory space. This memory space is continuously updated and revised as you create and modify your PLC logic. When these revisions occur, the location of key data elements such as flags, timers, counters, I/O and data blocks can move around in the PLC's memory. The Siemens S5 IO Server has been designed to read the location of these memory elements when the driver first begins operation upon detecting a communication error or after a request (which is not a read or a write request) has been transmitted to the PLC. If you change your PLC configuration you must restart the Siemens S5 IO Server or simply unplug/re-plug the connection cable. Both actions will cause the Siemens S5 IO Server to reacquire the location of all PLC memory elements.

### 7.2. Setup

Topic A
Enabled

Global Device Address:

Poll Rate:  MS

Serial port

Default: 2000

The numbering of the serial COM ports starts with the serial port of the Base Unit (if available) and then continues with the upper port of the first Extension Module starting from the left. A maximum of 5 serial ports are supported.



When there are multiple IO servers potentially using the serial line, the unused IO servers must be cleared or the baud rate must be set to **Disabled**.

#### Example:

If Modbus **and** S5-AS511 servers are available, at least one of them must have its baud rate configured to Disabled. If not, one of the IO servers will not be able to use the serial line and it will be disabled, with an error written in the event log.

#### 7.2.1. Communication setup

The AS511 link uses a RS-232 Current Loop functioning mode. A specific cable (with electronic) is therefore required (Siemens S5 Cable: ACT'L ref EW40910)

The communication parameters are fixed to the following values:

- 9600 Baud (Fixed)
- Even Parity (Fixed)
- 8 Data Bits (Fixed)
- 1 Stop Bit (Fixed)
- Full duplex no handshaking (Fixed)

## 7.2.1.1. Supported Devices

- Siemens S5 - 90U
- Siemens S5 - 95U
- Siemens S5 - 100U - 100
- Siemens S5 - 100U - 101
- Siemens S5 - 100U - 103
- Siemens S5 - 101U
- Siemens S5 - 115U - 941
- Siemens S5 - 115U - 942
- Siemens S5 - 115U - 943
- Siemens S5 - 115U - 944
- Siemens S5 - 115U - 945

## 7.3. Tag name convention

<b>IO server Name</b>	S5-AS511	
<b>Topic name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, ComPortNum	COM port is defined Tag by Tag
	ValueName	Topic COM port is used (or default)

**ComPortNum:** The port number used to access the PLC; if not specified, the default COM port is used.

- If a port address is specified in the topic, it is used and overloads per Tagaddress.
- If no address is specified, neither in the topic, nor at the Tag level, then the default port is used (Default port is the ECIA Port - COM:1).

## 7.3.1. ValueName

Syntax	Description	Automatic Tag type
<b>DBxLy</b>	Data block x Word offset y, left byte of word (0 - 255)	DWord
<b>DBxYy</b>	Data block x Word offset y, left char of word (-128 - 127)	Integer
<b>DBxRy</b>	Data block x Word offset y, right byte of word (0 - 255)	DWord
<b>DBxZy</b>	Data block x Word offset y, right char of word (-128 - 127)	Integer
<b>DBxWy</b>	Data block x Word offset y, full word (0 - 65535)	DWord
<b>DBxSy</b>	Data block x Word offset y, integer (-32768 - 32767)	Integer
<b>DBxDy</b>	Data block x DWord offset y (0 - 4294967296 ) (*)	DWord
<b>Tx or Kx</b>	Timer x (0-65535)	DWord
<b>Cx or Zx</b>	Counter x (0-65535)	DWord
<b>MxB or FxB</b>	Memento as Byte	DWord
<b>MxW or FxW</b>	Memento as Word	DWord
<b>MxD or FxD</b>	Memento as DWord (*)	DWord
<b>Ix or Ex</b>	Input	DWord
<b>Qx or Ax</b>	Output	DWord



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

- **Bit access modifier:**

In any of the above items, it is possible to access a single bit.  
 #x must be appended to the Value Name (Bit index goes from 0..31).  
 The syntax can be used for reading bits and for writing them as well.

**Example:**

`DB1W13#3` represents bit 3 of W13 in DB 1 `Q32B#0` represents bit 0 of Q32.



You must define the type byte (B) to successfully extracts a bit (Q32#0 will not work).

**REM:** Bit index range depends on item referenced (for W, bit index maximum value is 15)



- For DB the smallest element is a WORD and we count addresses in Word. This means that DB1W0 and DB1W1 WILL NOT overlap.
- Timers and Counters are always Words and addresses are counted in WORD as well. This means that T0 and T1 WILL NOT overlap.
- For M, I, Q addresses are counted in BYTES. This means that M0W and M1W WILL overlap.
- Both syntax (German and English) can be used for I, Q, C which in German gives: E, A, Z.

- **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for other Tags, the status Tag ValueName is composed of:

**Status, ComPortNum**

- You can define a status Tag for each COM port used.
- If you use the status address, the Tag must be configured as analog.

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.

## 8. ABLOGIX IO Server

### 8.1. Introduction

ABLogix IO server is used to communicate with Allen Bradley Logix Series PLCs on the Ethernet link or on the Serial link (in DF1 protocol).

- It allows the Ewon to poll ABLOGIX PLCs in order to read data.
- It allows the Ewon to act as EIP-DF1 gateway for remote maintenance of the ABLOGIX PLCs.
- It allows the Ewon to poll DH+ device using ControlLogix gateway.

The Ewon only allows DF1 communication with devices directly connected to the Ewon Serial link (the ABLOGIX PLC may not be located behind another PLC connected on the DF1 serial link acting itself as gateway).



Please check in the "Rockwell compatibility matrix" if your device is supported.

### 8.2. Setup

COM Setup (Duplicate of DF1 IO server)

COM Port:	<input type="text" value="IO Port1 (COM:1)"/>	
Baud Rate:	<input type="text" value="9600"/>	Default 9600
Parity:	<input type="text" value="None"/>	Default: NO
Stop Bit(s):	<input type="text" value="1"/>	Default: 1
Frame Error Detection:	<input type="text" value="CRC"/>	Default: CRC
HW Mode:	<input type="text" value="Full Duplex NO Handshaking"/>	Default: Full Duplex
Master response timeout:	<input type="text"/> MS	20..60000, default: 1000
Rx message timeout:	<input type="text"/> MS	1000..60000, default 3000
Tx message timeout:	<input type="text"/> MS	1000..60000, default 3000
DF1 Address:	<input type="text"/>	Device address of on DF1 link (0..254, default: 4)
Destination DF1 Address:	<input type="text"/>	Device address of destination on DF1 link when EIP is used (0..254, default: 1)
<input checked="" type="checkbox"/> Bridge EIP Connection		Enabled

### 8.2.1. Communication Setup

In the current implementation, the serial link configuration may be done using the DF1 IO Server configuration page or the AB LOGIX IO server page.

The same settings apply for both IO servers (DF1 and ABLogix).



In the config.txt file the serial link parameters are defined in the DF1 IO server entry.

Field name	Description
<b>COM Port</b>	Physical COM Port allocated to the ABLOGIX IO Server. Note: The port selection drop down is not displayed if the Ewon has only one single serial port. For multiple serial ports see remark (1).
<b>Baud Rate</b>	Select the Baud rate from 1200 to 38400 Bauds
<b>Parity</b>	The parity to apply: none / odd / even
<b>Stop Bit(s)</b>	Number of stop bits
<b>Frame Error Detection</b>	Cyclic Redundancy Check (CRC) or Block Check Character (BCC)
<b>HW mode</b>	Full Duplex no handshaking or Half duplex
<b>Master Response Timeout</b>	Maximum time the Ewon will wait for a valid message from the DF1 master. This value can be critical for a correct operation, depending on the responsiveness of the master. A value of 1000 should be selected to guarantee correct operation
<b>Rx message timeout (MSEC)</b>	Maximum time between a request is posted and the response is received
<b>Tx message timeout (MSEC)</b>	Maximum time for a request to be sent
<b>Ewon DF1 address</b>	Device Address of Ewon on DF1 link When the Ewon acts as a DF1 slave, it will respond to 2 consecutive link addresses; doing this improves the throughput of data across the Ewon when acting as a gateway. (Ewon DF1 address and Ewon DF1 address +1)
<b>Destination DF1 address</b>	Device Address of Destination on DF1 link when EIP is used
<b>Bridge EIP connection</b>	When bridging is configured, EIP connections requests are forwarded to the PLC on the DF1 interface (mandatory for Logix family). This is the default configuration. When not configured, connection requests are canceled by the Ewon (must be disabled for SLC family).

(1) The numbering of the serial COM ports starts with the serial port of the Base Unit (if available) and then continues with the upper port of the first Extension Module starting from the left. A maximum of 5 serial ports are supported.



When there are multiple IO servers potentially using the serial line, the unused IO servers must be cleared or the baud rate must be set to **Disabled**.

#### Example:

If Modbus and ABLOGIX servers are available, at least one of them must have its baud rate configured to Disabled.  
If not, one of the IO servers will not be able to use the serial line and it will be disabled, with an error written in the event log.

### 8.2.2. Topic configuration

Three topics can be used for the IO Server. These topics are used to give a common property, to a group of Tags, like:

- **Enable/Disable**
- **Global Device Address**
- **Poll rate**

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables the polling of all the Tags in the topic
<b>Global Device Address</b>	The Global Device Address is a global parameter for all the Tags of the topic. See below for the Device Address syntax. If an address is specified here, it will replace (overload) the address defined Tag by Tag
<b>Poll rate</b>	Defines the refresh rate (in mSec) of the Tag name. In a complex application, we can imagine that some Tag name must be refreshed every second – typically for digital input - and other every minute – typically: temperature-

#### 8.2.2.1. ABLogix : DF1 serial link

The Global Device Address has the following syntax: **Address**

- **Address** = DF1 Node ID of the device (i.e.: 5)

#### 8.2.2.2. ABLogix : EIP Ethernet link

The Global Device Address has the following syntax : **IP Address,Port,Link**

- **IP Address** = address on your Ethernet network (i.e.: 10.0.30.7)
- **Port** = value from 1 to 3 representing: 1 = Backplane, 2 = Channel A, 3 = Channel B
- **Link** could be:
  - *Slot*: representing the Slot on the Backplane (for example: 0=CPU)
  - *Node ID*: value from 0 to 99 (for ControlNet)
  - *IP address*

**REM: In the case of using a ControlLogix as Gateway, the Device Address could be like following:**

**IP Address, Port, Link[, Port,Link][, Port,Link]...**

#### 8.2.2.3. DH+ polling using ControlLogix gateway

This feature allows the Ewon to poll PLC tags located on a DH+ network, behind a ControlLogix equipped with one or more DHRIO cards. Polling is done using the Ewon EIP protocol.

The ABLOGIX IO server is used to access the "LOGIX" PLC family.

The DF1 IO server is used to access the "SLC500" PLC family

The general syntax *IP address,Port,Link* is still applicable.

There is no specific character to indicate the DH+ channel ("2" or "3" should be used).

#### Example:

To reach a SLC500 (with CPU in slot 0) with DH+ node Id 3 using channel A of a DHRIO card at slot 9 of a ControlLogix at IP address 10.0.30.21.

Topic A Destination = **SLC500-10.0.30.21,1,9,2,3,1,0**

- **10.0.30.21** = IP address of the ControlLogix
- **1** = BackPlane
- **9** = Slot 9 (COM card)
- **=** Channel A (of the Card present in Slot9)
- **=** NodeID of the PLC5 (in the COM network)
- **1** = BackPlane
- **0** = slot (of the CPU in our example)

### 8.3. Tag name convention

The Item Name can contain the PLC address where the value is polled, or not.

If the address is also specified at topic level, the address specified at Tag level will be ignored.

IO Server Configuration		
IO Server Name	ABLOGIX	
TOPIC Name	A	
	B	
	C	
Item Name	ValueName	Topic PLC address is used
	ValueName, IP address[, Port, Link]	PLC address is defined Tag by Tag
	ValueName, NodeID	PLC address is defined Tag by Tag

#### 8.3.1. ValueName

ValueName follows the below syntax:

```
[PROGRAM: ProgName . ] SymbolicTagName
```

- **ProgName is the name of the program where the Tag is.**

If no ProgName, the Tag is in the global scope (tag is controller type)

- **SymbolicTagName:**

- Symbol of the tag.  
Only following atomic type are supported: BOOL, SINT, INT, FLOAT, DINT (\*), BIT ARRAY, STRING
- Bit selection with <SymbolicTagName>/bit where bit is the bit number (from 0 to 31)  
ex: controlbit/4 read the bit 4 of the controlbit register
- TIMER, CONTROL, COUNTER predefined types with <SymbolicTagName>.acc (or ctl or pre)  
ex: MyVar.acc read the counter of MyVar
- Element of a table <Symbol\_1>[idx\_1].<Symbol\_2>[idx\_2].<Symbol\_3>[idx\_3].<Symbol> With maximum 3 index.  
ex: table1[2].subtable[6].element read data named 'element' on the index 6 of the 'sub table' from the index 2 of the 'table1'
- Element of a structure <Symbol\_1>.<Symbol\_2>  
ex: CounterObj.init read the 'init' part of the structure 'CounterObj'
- The size of a string Tag is limited to 1024 bytes  
If you read a string greater than 1024 bytes, the value will be truncated at 1021 bytes + "... " (3 dots) to show that the string was truncated.



**(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag. If you choose "Automatic" Tag type, Ewon will use Floating Point as storage type.**

**Examples :**

<b>Address</b>	<b>point to</b>
<b>myvar</b>	the data named "myvar" in the scope <i>global</i> (controller) of the PLC
<b>PROGRAM:myprog.myvar</b>	the data named "myvar" in the scope of the program named "myprog"
<b>myvar/4</b>	the bit 4 (the fifth) from the data "myvar"
<b>mystruct.mydata</b>	the data named "mydata" from the structure named "mystruct"
<b>mytable[6]</b>	the element 6 (the seventh) of the table "mytable"
<b>mytablestruct[4].mydata[12]</b>	the element 12 of the table "mydata" from the structure at index 4 of the table "mytablestruct"

- **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for other Tags, the status Tag ValueName is composed of:

**Status, GlobalDeviceAddress**

- You can define a status Tag for each COM port used.
- If you use the status address, the Tag must be configured as analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

## 9. Ewon IO Server

### 9.1. Introduction

The Ewon IO server is used to interface the Ewon INPUTS and OUTPUTS. Depending on your Ewon model, you have:

Ewon types	Flexy 10x - 20x (Base Units)	Flexy FLX 3401/3402 (Extension Card)
Digital Inputs	2	8
Digital Outputs	1	2
Analog Inputs	0	4

Additionally, there are a number of Tags that can be addressed with this IO server and which are computed by the Ewon IO Server. These additional Tags are used for energy management. In Energy management, the following requirements are taken into accounts:

- Using digital inputs as counter inputs
- Count for a given interval and latch computed result (also save it in historical)
- Reject the measurement interval if too long or too short
- Adjust the Ewon's Real Time Clock based on a digital input

These Tags will be computed if the energy module is enabled.  
There is no topic name to define for the Ewon IO server.

### 9.2. Standard Ewon I/O Item Names

#### 9.2.1. Tag address convention

The following Tags addresses are available for standard Ewon Inputs & Outputs access:

IO Server configuration		Comment		
IO Server Name	Ewon			
Topic Name	Empty			
Item Address	DI#	Digital Input (1)		Boolean
	CI#	Counter Input (1)	0 to 2.147.400.000(2)	Analog
	FI#	Counter Input	0 to 255	Analog
	LI#	Latched Counter Input (1)		Analog
	DO#	Digital Output (1)		Boolean
	AI#	Analog Input (1)		Analog
	BI#	Button Input (1)		Boolean



The button input (BI1) can be used during normal Ewon operations, if it is pressed for more than 4 seconds while the Ewon is booting, the flash file system will be erased.

(1)The number of items depend of the Ewon type. For the Ewon Flexy range, the numbering of the IOs starts with the embedded IOs of the Base Unit and continues with the first IO Extension Card starting from the left (if available).

IO Server configuration		Comment	
<b>IO Server Name</b>	<b>Ewon</b>		
<b>Topic Name</b>	<b>SYS</b>		
	<b>SN_LO</b>	Serial number, low part (see example of use below)	DWord
	<b>SN_HI</b>	Serial number, high part (see example of use below)	DWord
	<b>SYS_UP</b>	Number of seconds since Power Up	DWord
	<b>GSM_REG</b>	GSM Status: 1: Home network 2: Searching registration 3: Registration denied 4: unknown registration 5: Roaming 100: Not applicable 101: Registration in progress  Usually:1 or 5: means registered Other: not registered	DWord
	<b>GSM_LEV</b>	GSM level (antenna reception)	DWord

Example of use of SN\_LO and SN\_HI items in a Basic program:

```

a% = Int(SN_HI@ * 65536) + Int(SN_LO@)
Rem Product code
b% = a% Mod 256
Rem Sequential number
c% = Int(a% / 256) Mod 1024
Rem Week number
d% = Int(a% / 262144) Mod 64
Rem Year number
e% = Int(a% / 16777216) Mod 128
Print "SN: ";e%;" ";d%;"-" ";c%;"-"b%
    
```

For instance, you can define a Tag for the digital input 1 as follows:

*Server name: Ewon - Topic: empty - Address: DI1*

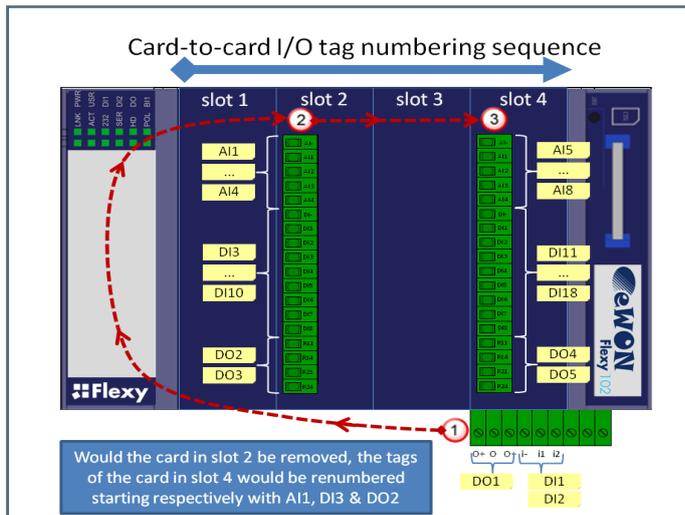
	Syntax	Description	Automatic Tag type
<b>Ewon Flexy 10x/20x</b> Base Units only. See note below.	<b>DI1... DI2</b>	Status of input 1 and 2 on the Base Unit main connector	Boolean
	<b>CI1</b>	Counter on input 1 and 2 on the Base Unit main connector	DWord
	<b>LI1</b>	Latched counter on input 1 and 2 on the Base Unit main connector	DWord
	<b>DO1</b>	Digital command of relay 1 on the Base Unit main connector	
	<b>BI1</b>	Status of button (reset) on the front face	

**9.2.2. Basic Principle of the Ewon Flexy I/O Tag Addresses**

The internal I/O tag addresses of the Ewon Flexy always start with the Inputs/Outputs of the Base Unit. Remember that all Base Units feature 1 Digital Output and 2 Digital Inputs, those are the first ones that have to be considered when creating tags in the Ewon.

The example on the right helps you to understand the syntax of the I/O tag addresses in the case of 2 I/O Extension Cards.

In the case of multiple I/O cards, the tag numbering sequence is following the left-to-right order of slots. First the I/O Extension Card plugged in the most left slot, then the next on its right and so on. Removing an I/O card other than the utmost right one will result in an internal reallocation of tag addresses that may result in a mismatch between physical I/Os and their software configuration. The software tag addresses can be “frozen” by adding the slot number in the tag definition. This prevents unwanted I/O mismatch (see below).



Slot number append to prevent tag address mismatch:

To prevent any risk of mismatching tag addresses you can append the slot number after the I/O address. The syntax to use is as follows:

**ABx,Ey**

Where AB is the type of I/O (DI, AI, DO) x is the order number  
 E is a constant prefix to the slot number  
 y is the slot number of the card (0 = main board, 1 = slot #1, 2 = Slot #2, etc..)

Let's imagine an Ewon Flexy featuring 2 I/O cards FLX 3401 in slots #1 and #2.

Name	Type	IO Server	Topic	IO Address	Value	Tag description
DO1	Floating po...	EWON		DO1,E0	1	Digital Output 1
DO2	Floating po...	EWON		DO2	0	Digital Output 2
DO3	Floating po...	EWON		DO3,E2	0	Digital Output 3 on slot 1
DO4	Floating po...	EWON		DO4,E1	0	Digital Output 4 on slot 2

The main I/O syntax and order numbering remains unchanged. In the example above, the I/O server tag addresses are as follows:

Tag Syntax	Explanation
DO1,E0	Digital Output 1, main board (though E0 is not necessary in this case)
DO2	Digital Output 2, no position specified = second detected DO = first DO on first extension card detected (in slot #2 most left).
DO3,E2	Digital Output 3, board in slot #1 = second DO on extension card in slot #2
DO4,E3	Digital Output 4, board in slot #2 = first DO on extension card in slot #3

**Behavior if the card in slot #1 is removed:**

In the **Tag View** page,

- DO2 will appear normally as it was not frozen by a slot number append. But it can no longer reflect the status of the first DO of the board in slot #1 that was removed. DO2 now reflects the status of the first DO of the board in slot #2.
- The 2 other DOs with specified slot number E2 and E3 will all appear in error (red cross, value 0), because:
- The address of DO3,E1 of the card in slot #1 could no longer be found.
- The software address of DO4,E2 of the card in slot #2 do no longer match with the detected order of physical addresses.
- The error messages in the **Trace** log file are "Invalid IO Tag name (DOx, addr. DOx,Ex)".

To have the DOs of the card in slot #2 responding, edit the tags and change their software addresses as follows: DO4,E2 to DO2,E2

### 9.2.3. Energy configuration with the Llx Tags

The following parameters can be added to the Ewon IO server in order to activate the energy support (enter *Energy:1*, validate and the other parameters will be automatically added):

Parameter	Default Value	Description	
<i>Energy</i>	<b>0 or 1</b>	Enabled or not	Once Energy is defined, the Ewon will automatically add the other parameters with default value. When 0, no energy computation at all are performed, saving Ewon CPU resources.
<i>Debug</i>	<b>0 or 1</b>	Not automatically added	When :1 all synchronizations are logged in the Real time event log.
<i>RefTime</i>		This is the reference time to compute interval. Letting this empty will start at 1/1/1970 00:00. You may enter a date with hour in the form 08/07/2002 11:15:00 to set a new reference (usually not required).	
<i>IntTime</i>	<b>15</b>	This is the integration time in MINUTES.	
<i>IntToIS</i>	<b>5</b>	This is a tolerance on the integration time in SECONDS. If the measurement interval is shorter or longer than this number of seconds, the period is considered as valid. For example IntTime=15 IntToIS =5 means "interval is valid if between 11.14:55 and 11.15:05".	
<i>SyncIO</i>	<b>5</b>	This is the IO number for clock synchronization. 0 means no synchro available, 1..8 is the IO number. Synchronization is based on the counter input associated with the digital input, each time the IO changes the Ewon will try to perform a synchronization (see also <i>SyncToIS</i> )	
<i>SyncToIS</i>	<b>5</b>	This is the tolerance for accepting the synchronization pulse in SECONDS. If the synchronization pulse arrives outside the interval, the pulse is rejected. <b>ERROR REPORT:</b> the first time the pulse is rejected, an error is logged in the event log. Next errors are not logged (except in Real time log if Debug is enabled); When sync are accepted again, a trace is logged in the event file for the first accepted synchronization.  REM: Because the internal clock has a precision of 1 second, the time is only updated if the absolute value of the offset is greater than 1 (2 or more). Otherwise the offset jitters between -1 and +1 all the time.	

#### Example:

```

MinInterval:10
MaxInterval:268435455
ReverseCount:0
Energy:1
RefTime:
IntTime:15
SyncIO:0
SyncToIS:5
IntToIS:5

```

**9.2.3.1. ENERGY TAGS**

The following tags are available for energy management:

<b>LI1..LI8</b>	<b>Latched input</b>	<b>When the Integration Period expires, the number of pulses counted during the integration period are logged in the corresponding LIx.</b>
<b>ST1..ST4</b>	<b>Status variables:</b>	
<b>ST1</b>	This is the absolute time when counter were latched. Expressed in seconds since 1/1/1970 minus 0x30000000 HEX (or 805306368 decimal). This huge value is subtracted to maintain precision in the Float storage. (*)	
<b>Latch time - 0x30000000</b>		
<b>ST2</b>	0 means period is valid (within tolerance), 1 means period is invalid.	
<b>Period status</b>		
<b>ST3</b>	Length of the period in second.	
<b>Period length sec</b>		
<b>ST4</b>	This 32 bit counter counts is increased by 1 after each integration period. It can be used with ONCHANGE to perform operations when period expires.	
<b>Period Num</b>		

**REM1:** Energy tags are logged after each new period IF Logging is enabled, even if the value has not changed since previous period.

**REM2:** Even if Deadband is -1 and Interval is 0. If not, you will have additional points and even maybe duplicate points.



**(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.**

**9.2.3.2. 32 BITS COUNTER TAGS:**

The Ewon IO Server provides 8 32 bits (\*) counter tags named: C11..C18.  
These counters are writable but writing in these registers affects the LIx inputs of the Energy IO server module.



**(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.**

### 9.2.4. Analog Input Value Ranges

On the Ewon Flexy Extension Card FLX 3401/3402, the Analog Inputs (AI1 to AI4) are processed by a 16 bit ADC. The converted values range from 0 to 65536.

Ewon Flexy Extensions Card FLX 3401/3402 analog inputs AI1 to AI4

AI#	Volt
0	0
65535	10



If you need to read the Tag value converted in Volt, you can set a factor of 0.0001526 = 10/65535 with an offset of 0.

Create a new Tag

**Identification**

Tag Name:  Page:

Tag Description:

**I/O Server Setup**

Server Name:  Topic Name:

Address:   
Enter AI index and optional Extended Syntax address

Type:   Force Read Only

**EWON value = IO Server Value \***  +

### 9.3. Setup

The Energy IO Server is based on the Ewon IO Server.

The configuration consists in advanced parameters from the latter IO Server. The following values are the standard empty Ewon IO Server configuration page.

```
MinInterval:10
MaxInterval:268435455
ReverseCount:0
```

### 9.3.1. Configuration of the counter pulse length

The pulse length for counter can be configured on all the Ewon types.  
The configuration must be entered in the Ewon IO server edit area.  
The following parameters are used:

<b>MinInterval:</b>	default=10, min=10
<b>MaxInterval:</b>	default=268435455, max=268435455
<b>MinInterval &amp; MaxInterval are entered in milliseconds</b>	
<b>ReverseCount:</b>	default=0

The pulse length must be between MinInterval and MaxInterval to be accepted,  
the measurement resolution is 5 msec (the precision of the pulse length measured is 5 msec)

ReverseCount can be used if pulses entered in the Ewon are reversed,  
reversed means that signal is normally high on the Ewon input and it goes low when the pulse occurs.

#### Example of configuration:

```
MinInterval:40
MaxInterval:1000
ReverseCount:0
```



Counts pulse with length longer than 40 msec and shorter than 1000 msec, other pulses are ignored.

## 10. MEM IO Server

### 10.1. Introduction

The MEMORY IO Server is not a real IO server because values do not come from a peripheral. Memory Tags (Tags defined with the MEM IO server) are rather sorts of variables that can be modified by a user input or by a BASIC application.

These Tags are very useful for combining different Tags and consider the result as an actual Tag i.e. having data logging capabilities and alarm management capabilities like for all other Tags.

### 10.2. Setup

There is no setup for the MEM Ewon IO server.

### 10.3. Tag name convention

IO Server configuration		
IO Server Name	MEM	
Topic Name		If no topic is specified, the Tag is a standard memory Tag. Its value is set to 0 when the Ewon boots and the Tag is read/write, it can be updated through user actions with script or web pages
	RET	If topic is set to RET, the Tag is retentive, each change is saved to flash and when the Ewon boots, the last value of the Tag is restored. This feature can be convenient to use Tags as parameters.
Item Name	DON'T CARE	The MEM "Item Name" is insignificant, the TagName is automatically copied in the MEM address by the Ewon, the user can change it, but the value will immediately be overwritten again by the TagName. Data type "Automatic" is equivalent to Floating Point.
Protocol Type		Automatic, Floating Point, Boolean, Integer, Dword, String



RET values are saved in the FLASH memory, this is not an immediate operation and the FLASH memory can only be written and erased a limited (even if huge 100.000) number of times.

This means that you should not use a BASIC script for changing retentive values at very high rate. 1 Tag/Sec. is a maximum AVERAGE rate (occasional higher rates are not prohibited at all).

## 11. S7-300&400 (MPI/Profibus - ISOTCP)

### 11.1. Introduction

S73&400 IO Server is dedicated to direct communication with SIEMENS PLC (S7-300 & S7-400).  
The Ewon will connect directly to the PLC's MPI interface or Profibus DP interface.



**The MPI/Profibus interface requires a special serial port hardware!  
ONLY the Ewons MPI/Profibus are able to communicate in MPI/Profibus**

**Others Serial IO Server (i.e.: modbus RTU, DF1, UniteWay)  
are not available on Ewon MPI/Profibus.**

**The ISOTCP (Ethernet) is available on all Ewon types.**

**Multi Session:** the ISOTCP-MPI gateway allows multiple clients to access the same PLC at the same time.  
**PPI MultiMaster (for S7-200):** it is supported by the MPI port of the Ewon

### 11.2. Setup

#### 11.2.1. COM configuration

**MPI/PROFIBUS/PPI MultiMaster Setup**

Protocol Type:	<input type="text" value="MPI"/>	Default MPI, PPI Multimaster may only be used with the S7200 IO server
Baud Rate:	<input type="text" value="187500"/>	Default 187500
Reply Timeout:	<input type="text" value=""/> MS	50..50000, default: 3000
MPI/PROFIBUS/PPI MultiMaster Address:	<input type="text" value="10"/>	Device address of eWON on link (0..126, default: 0)
MPI/PROFIBUS/PPI MultiMaster Highest Station Address:	<input type="text" value="31"/>	Default: 31

Parameter	Description
<b>Gateway MPI/Profibus Destination Node</b>	Identifies the PLC MPI/Profibus Destination address when ISOTCP is used
<b>Protocol Type</b>	Select protocol MPI or PROFIBUS. As the MPI port configuration is shared by the S73&400 and the S7-200 IOServers, this combo box shows also the PPI MULTIMASTER protocol  <i>PPI MultiMaster and MPI/Profibus are mutually exclusive. It is not allowed to use the S7-200 and the S73&amp;400 IOServers on the MPI port at the same time. The string type (ASCII) is included in the S73&amp;400 I/O server.</i>
<b>Baud Rate</b>	Speed of the MPI port. Available speeds are:  Disable 9.6 k Bauds 19.2 k Bauds 45.45 k Bauds 93.75 k Bauds 187.5 k Bauds 500 K Bauds 1.5 M Bauds 3 M Bauds 6 M Bauds 12M Bauds
<b>Reply Timeout</b>	The maximum time the Ewon will wait for a valid MPI message response
<b>MPI Address</b>	The device address of the Ewon on the MPI link (0..126, default is 0)
<b>MPI Highest Station Address</b>	The highest station address polled by Ewon. Select between 15, 31, 63 or 127.
<b>Maximum length</b>	The length is 222 bytes (2 size bytes + 220 characters). It means that if S7 IOServer has a longer string, it will be truncated to 220 characters when read or written

### 11.2.2. Advanced Routing Setup

Advanced Routing Setup allows to access devices that make part of another MPI/Profibus network connected indirectly to the Ewon through another (or various other) PLC(s).

Advanced Routing Setup items are visible when the **Enable** checkbox is ticked.



Parameters that would have been set in this section remain in memory but are visually hidden AND not taken into account by the firmware when the **Enable** checkbox is unticked.

Enabled

Gateway Destination S7 Subnet ID:  (xxxx-xxxx, default: 0000-0000)

ISOTCP S7-Subnet ID:  (xxxx-xxxx, default: 0000-0000)

MPI/PROFIBUS S7-Subnet ID:  (xxxx-xxxx, default: 0000-0000)

Default Next MPI/PROFIBUS Node:  (0..126, default: 2)

---

Gateway Entry 1

Destination S7-Subnet ID:  default:0000-0000

Next MPI/PROFIBUS Node:  default:2

Parameter	Description	Default value
<b>Advanced Routing Setup</b>	Enable/disable Advanced Routing Setup feature	0 (disabled)
<b>Gateway/Destination S7-Subnet ID:</b>	Used when no subnet ID is specified in the ISOTCP message to relay on the MPI bus	0000-0000
<b>ISOTCP S7-Subnet ID:</b>	Ewon subnet ID at the ISOTCP interface side	0000-0000 (invalid subnet ID)
<b>MPI/PROFIBUS S7-Subnet ID:</b>	Ewon subnet ID at the MPI interface side	0000-0000
<b>Default Next MPI/PROFIBUS Node:</b>	MPI node of the PLC towards which the MPI messages will be sent out if the destination subnet ID specified is not the one of the Ewon (see previous field), nor one of the ones specified in the gateway entries (see next fields)	2
<b>Gateway Entry 1..3 Destination S7-Subnet ID</b>	MPI messages with destination subnet ID equal to this destination ID will be forwarded to the next MPI/Profibus Node (see next field)	0000-0000
<b>Gateway Entry 1..3 Next MPI/Profibus Node ID</b>	See previous row	2

11.2.3. Topic configuration

Topic A

Enabled

Global Device Address:  Enter MPI/POFIBUS address: SubnetID,node or node

Poll Rate:  MS Default: 2000

Topic B

Enabled

Global Device Address:  Enter MPI/POFIBUS address: SubnetID,node or node

Poll Rate:  MS Default: 2000

Topic C

Enabled

Global Device Address:  Enter MPI/PROFIBUS address: SubnetID,node or node

Poll Rate:  MS Default: 2000

Three (3) topics can be used for the IO Server.

These topics are used to give a common property to a group of MPI/ISOTCP Tags like:

- Enable/Disable
- Global Device Address
- Polling Rate

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Global Device Address</b>	See below for the Device Address Syntax. If an address is specified here, it will replace (overload) the address-defined Tag by Tag.
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

### 11.3. Tag name convention

<b>IO Server Name</b>	S73&400	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, Global Device Address	PLC address is defined Tag by Tag
	ValueName	Topic PLC Address is used.

The Item Name can contain the PLC address where the value is polled, or not.  
If the address is also specified at the topic level, the address specified at Tag level will be ignored.

#### 11.3.1. Value Name

The syntax is the following:

<Memory Type><Modifier><address>

Symbol	Memory Type	Modifier	Address
<b>DBX</b>	<b>Data Block number x</b>	B, C, W, S, D, L, F,T	Byte offset
<b>M</b>	<b>Internal Memory</b>	B, C, W, S, D, L, F,T	Byte offset
<b>C</b>	<b>Counter</b>	W, S	Object number
<b>T</b>	<b>Timer</b>	W, S	Object number
<b>I</b>	<b>Discrete Inputs</b>	B, C, W, S, D, L, F,T	Byte offset
<b>Q</b>	<b>Discrete Outputs</b>	B, C, W, S, D, L, F,T	Byte offset
<b>PI</b>	<b>Peripheral Inputs</b>	B, C, W, S, D, L, F,T	Byte offset
<b>PQ</b>	<b>Peripheral Outputs</b>	B, C, W, S, D, L, F,T	Byte offset



All the addresses are always in BYTES (except for Counter and Timer that are objects)



The Modifier can be omitted, the modifier in bold will be used (DB type must have a modifier).

Symbol	Modifier	value range	Automatic Tag type
<b>B</b>	<b>Byte</b>	0 .. 255	DWord
<b>C</b>	<b>signed Byte</b>	-128 .. 127	Integer
<b>W</b>	<b>Word</b>	0 .. 65535	DWord
<b>S</b>	<b>signed Word</b>	-32768 .. 32767	Integer
<b>D</b>	<b>DWord</b>	0 .. 4294967296 (*)	DWord
<b>L</b>	<b>signed DWord</b>	-2147483648 .. 2147483647 (*)	Integer
<b>F</b>	<b>Float</b>	+/- 3.4e38	Float
<b>T</b>	<b>String</b>	2-222 bytes (Text)	String



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

**Examples**

<b>Address</b>	<b>point to</b>
<b>MW4</b>	the Word at address 4 (in bytes) in the Internal Memory
<b>CS1</b>	the Counter number 1, read it as Signed Word
<b>DB2L5</b>	the DWord at address 5 (in bytes) in the Data Block 2
<b>IB3</b>	the Byte at address 3 (in bytes) in the Discrete Inputs zone
<b>I5#2</b>	the bit 2 from the Byte (read 'Bit access modifier' note below) at address 5 in the Discrete Inputs zone
<b>DB5T20</b>	The string starting at address 20 (in bytes) in the Data Block 5

The maximum length of a S7&400 string tag in the Flexy is 222 bytes (2 size bytes + 220 characters). This means that if you have a longer string in your S7 PLC, it will be truncated to 220 characters when read or written.

- **Bit access modifier:**

In any Memory Type (excluding Counter and Timer), it is possible to access a single bit. A #x must be appended to the Value name.

**As the address is always in byte, the Bit index goes only from 0 to 7, no modifier are allowed** (except B to separate DB block from the address).

The syntax can be used for reading bits and for writing them as well. If Tag type "Automatic" is used, the type Boolean will be chosen.

**Example:**

```
DB1B13#3 represents bit 3 of Byte 13 in DB 1
invalid: IW5#2 wrong because there is a Modifier
         I5#10 wrong because bit number greater than 7
```

- **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of the communication for a given device. As for other Tags, the status Tag ValueName is composed of:

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 analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

### 11.3.2. Global Device Address

The device address is either appended to the ValueName in the Item Name definition or entered in the Topic global address fields. The device address is composed like following:

<b>MPI</b>	
<b>MPI, MPI node address</b>	<b>Node address: MPI node address of the PLC</b> <b>Global Device Address: MPI, 2</b> <b>Reach by MPI the PLC with the node address 2</b>
<b>PROFIBUS, Profibus node addr</b>	<b>Node address: PROFIBUS node address of the PLC</b> <b>Global Device Address: PROFIBUS, 2</b> <b>Reach by Profibus the PLC with the node address 9</b>

<b>MPI</b>	
<b>MPI, Subnet ID, node address</b>	<p>When the Advanced Routing Setup is <i>Enabled</i>, the following syntax can be applied to reach a device on the MPI network indirectly connected to the Ewon (behind one or several other PLCs)</p> <p><b>Global Device Address: MPI, 4859-4565, 4</b></p> <p><i>Subnet ID</i>: ID of the destination S7-Subnet ID  <i>node address</i>: MPI/Profibus node address of the PLC to reach            Reach by MPI/Profibus the PLC with the node address 4 on the subnet with subnet ID 4859-4565</p>
<b>ISOTCP</b>	
<b>ISOTCP, IP address, CalledTSAP</b>	<p><i>IP address</i> : IP address of the PLC (ex: 10.0.120.204)  <i>Called TSAP</i> : Transport Service Access Point (ex: 03.02)</p> <p><b>Global Device Address: ISOTCP, 10.0.120.204, 03.02</b></p> <p>Reach by ISOTCP the PLC at IP 10.0.120.204 with the TSAP 03.02</p>
<b>ISOTCP, IP address, CalledTSAP, Subnet ID, MPI node address</b>	<p><i>IP address</i> : IP address of the PLC (ex: 10.0.120.204) <i>Called TSAP</i> : Transport Service Access Point (ex: 03.02) <i>Subnet ID</i> : ID of the MPI subnet  <i>MPI node address</i> : address of the PLC</p> <p><b>Global Device Address:            ISOTCP, 10.0.120.204, 03.02, 0020-0001, 2</b></p>



If the PLC address is defined at the Topic level, it can be omitted in the Tag definition.

In that case the Tag name will only contain the "ValueName".

If the PLC address is specified at the Topic level, it will replace any address defined Tag by Tag.



For an S7-300, it's generally always 03.02 .

For an S7-400, you have to take into account the slot number of the CPU.

For example if the CPU is on slot 4 rack 0 use TSAP 03.04, if the CPU is on slot 5 rack 1 use TSAP 03,25.

## 12. S7-200 (PPI - ISOTCP)

### 12.1. Introduction

The S7200 IO Server is dedicated to direct communication with SIEMENS PLC of family S7-200.

All Ewon types, with serial port or with MPI port, are able to communicate in both PPI modes (MonoMaster and MultiMaster).

To communicate with a S7-200 device you could use either an Ewon with a serial port or an Ewon with an MPI port.

Usage	Ewon with SERIAL	Ewon with MPI port
<b>MonoMaster</b> Ewon is the ONLY Master on the PPI network.	Yes	Yes
<b>MultiMaster</b> Ewon is not the only Master on the PPI network. Other masters (HMI or PPI multimaster programming cables) may be present at the same time on the PPI bus.	Yes	Yes

Usually, the Ewon will be connected directly to the PLC's PPI interface.

Ewons with an MPI port are able to communicate on a PPI MultiMaster bus.  
 Ewons with a serial port are able to communicate on a PPI MultiMaster bus.

The Ewon acts always as MASTER.

The standard serial port of Ewon must be configure in RS-485 to communicate in PPI.  
 The ISOTCP (Ethernet) is available on all the Ewon types.

## 12.2. Setup

### 12.2.1. Gateway configuration

Gateway Configuration

Destination PPI Address:  0..126,default: 2

Parameter	Description
Destination PPI Address	Identifies the PLC PPI Destination address when ISOTCP is used

12.2.2. COM configuration (serial port)

COM Setup

COM Port:	<input type="text" value="IO Port1 (COM:1)"/>	
Baud Rate:	<input type="text" value="9600"/>	Default: 9600
Parity:	<input type="text" value="Even"/>	Default: EVEN
Databits:	<input type="text" value="8"/>	Default: 8
Stop Bit(s):	<input type="text" value="1"/>	Default: 1
HW Mode:	<input type="text" value="Half Duplex"/>	Default: half duplex no Handshaking
Reply Timeout:	<input type="text" value=""/> MS	50..50000, default: 3000
PPI Address:	<input type="text" value=""/>	Device address of eWON on PPI link (0..126, default: 0)

Parameter	Description
<b>COM Port</b>	Choose your COM port (only on Ewon4002)
<b>Baud Rate</b>	Speed of the PPI port. Available speeds are 9600, 19200 Bauds or disable
<b>Parity</b>	The parity to apply (None / Even /Odd)
<b>Databits</b>	Number of bits in data (7 or 8)
<b>Stop bit(s)</b>	Number of Stop bits (1 or 2)
<b>HW mode</b>	Choose the Hardware mode of communication: Half-Duplex by default or Full Duplex with/without hardware handshaking.
<b>Reply Timeout</b>	Maximum time the Ewon will wait for a valid PPI message response
<b>PPI Address</b>	Device address of Ewon on PPI link (0.. 126, default is 0)



By default, Ewon with Serial port will act as MonoMaster on the PPI link.

12.2.2.1. Serial PPI MultiMaster configuration

To configure the serial port as PPI MultiMaster you need to configure the following parameters.

PPI Address:	<input type="text" value=""/>	Device address of eWON on PPI link (0..126, default: 0)
HSA:	<input type="text" value=""/>	Highest Station Address (MultiMaster Advanced Parameter, default: 31)
StatusInterleave:	<input type="text" value=""/>	status request interleave (MultiMaster Advanced Parameter, default: 0)
Poll Retry:	<input type="text" value=""/>	number of poll request retries (MultiMaster Advanced Parameter, default: 50)
Gw Retry:	<input type="text" value=""/>	number of gateway request retries (MultiMaster Advanced Parameter, default: 0)

Serial Multimaster Parameter	Description
<b>StatusInterleave</b>	Number of token messages sent by the Ewon before one status request message is sent out. Status messages are used to get the configuration of the network. In the case interleave number is high bandwidth is used more efficiently for data exchange but network discovery and recovery is slow. Value range: 0..20 - default value: 0 (Ewon is mono master no exchange of status or token pass messages)
<b>NbrPollRetry</b>	Number of retrials for one user request (poll = ask for the response after reception of a positive ACK from the PLC) before aborting the request. Useful in case of high PLC load. Value range: 0..100 (0 means only one poll for one request) - default value: 15
<b>NbrGwRetry</b>	Number of retrials for one gateway request (poll = ask for the response after reception of a positive ACK from the PLC) before aborting the request. Useful in case of bus conflict due to high PLC load (reception of lot of NACK). Value range: 0..100 - default value: 15
<b>HSA</b>	Highest station active value range: 15..127 - default value: 31

12.2.3. COM configuration (MPI port)

MPI/PROFIBUS/PPI MultiMaster Setup

Protocol Type:	<input type="text" value="PPI MULTIMASTER"/>	Default MPI. PPI Multimaster may only be used with the S7200 IO server
Baud Rate:	<input type="text" value="187500"/>	Default 187500
Reply Timeout:	<input type="text"/> MS	50..50000, default: 3000
MPI/PROFIBUS/PPI MultiMaster Address:	<input type="text" value="10"/>	Device address of eWON on link (0..126, default: 0)
MPI/PROFIBUS/PPI MultiMaster Highest Station Address:	<input type="text" value="31"/>	Default: 31

Parameter	Description
<b>Protocol Type</b>	Select protocol PPI MULTIMASTER. As the MPI port configuration is shared by the S7-200 and the S73&400  IO Servers, this combo box shows also the MPI & Profibus protocols. <i>PPI MultiMaster and MPI/Profibus are mutually exclusive. It is not allowed to use at the same time the S7-200 and the S73&amp;400 IO Servers on the MPI port. The string type (ASCII) is included in the S73&amp;400 I/O server.</i>
<b>Baud Rate</b>	Speed of the MPI port. Available speeds are: Disable 9.6 k Bauds 19.2 k Bauds 45.45 k Bauds 93.75 k Bauds 187.5 k Bauds 500 k Bauds 1.5 M Bauds 3 M Bauds 6 M Bauds 12 M Bauds
<b>Reply Timeout</b>	The maximum time the Ewon will wait for a valid MPI message response
<b>PPI MultiMaster Address</b>	The device address of Ewon on PPI link (0..126, default is 0)
<b>PPI Highest Station Address</b>	The highest station address polled by Ewon. Select between 15, 31, 63 or 127.



When using the PPI MultiMaster, the status counters to be checked are the MPI counters and not the PPI counters.

### 12.2.4. Topic configuration

Topic A

Enabled

Global Device Address:  Enter PPI node

Poll Rate:  M5 Default: 2000

Three (3) topics can be used for the IO Server.

These topics are used to give a common property to a group of PPI/ISOTCP Tags like:

- Enable/Disable
- Global Device Address
- Polling Rate

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Global Device Address</b>	See below for the Device Address Syntax. If an address is specified here, it will replace (overload) the address-defined Tag by Tag.
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

### 12.3. Tag name convention

<b>IO Server Name</b>	S7200	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, Global Device Address	PLC address is defined Tag by Tag
	ValueName	Topic PLC Address is used.

The Item Name can contain the PLC address where the value is polled, or not.

If the address is also specified at topic level, the address specified at Tag level will be ignored.

12.3.1. Value Name

The syntax is the following:

<Memory Type><Modifier><address>

Symbol	Memory Type	Modifier allowed	Address
<b>M</b>	Internal Memory	B, C, W, S, D, L, F	Byte offset
<b>SM</b>	Special Memory	B, C, W, S, D, L, F	Byte offset
<b>V</b>	Variable Memory	B, C, W, S, D, L, F	Byte offset
<b>C</b>	Counter	W, S	Object number
<b>HC</b>	High-Speed Counter	D, L	Object number
<b>T</b>	Timer	D, L	Object number
<b>AI</b>	Analog Inputs	W, S	Byte offset
<b>AQ</b>	Analog Outputs	W, S	Byte offset
<b>I</b>	Discrete Inputs	B, C, W, S, D, L, F	Byte offset
<b>Q</b>	Discrete Outputs	B, C, W, S, D, L, F	Byte offset



All addresses are always in BYTES (except for Counters and Timer that are objects).



The Modifier can be omitted, the modifier in bold will be used.

Symbol	Modifier	value range	Automatic Tag type
<b>B</b>	Byte	0 .. 255	DWord
<b>C</b>	<b>signed Byte</b>	-128 .. 127	Integer
<b>W</b>	Word	0 .. 65535	DWord
<b>S</b>	<b>signed Word</b>	-32768 .. 32767	Integer
<b>D</b>	DWord	0 .. 4294967296 (*)	DWord
<b>L</b>	<b>signed DWord</b>	-2147483648 .. 2147483647 (*)	Integer
<b>F</b>	Float	+/- 3.4e38	Float
<b>T</b>	String	Text	String



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

Examples:

Address	point to
<b>MW4</b>	the Word at address 4 (in bytes) in the Internal Memory
<b>HCL1</b>	the HighSpeed Counter number 1 and read it in Signed DWord
<b>AQW5</b>	the Word at address 5 (in bytes) in the Analog Output zone
<b>IB3</b>	the Byte at address 3 (in bytes) in the Discrete Inputs zone
<b>I5#2</b>	the bit 2 from the Byte (read the 'Bit access modifier' note below) at address 5 (in bytes) in the Discrete Inputszone

- **Bit access modifier:**

For Memory Type **M, SM, V, I and Q**, it is possible to access a single bit.

A **#x** must be appended to the Value name.

**As the address is always in byte, the Bit index goes only from 0 to 7, and no modifiers are allowed.**

The syntax can be used for reading bits and for writing them as well.

If Tag type "Automatic" is used, the type Boolean will be chosen.

**Example:**

**AQ10#5 represents bit 5 of Byte 10 in Analog Output zone.**

**Invalid: IW5#2 wrong because there is a Modifier**

**I5#10 wrong because bit number greater than 7**

- **Status register:**

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

As for the other Tags, the status Tag ValueName is composed of:

**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 analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

### 12.3.2. Global Device Address

The device address is either appended to the ValueName in the Item Name definition or entered in the Topic global address fields. The device address is composed like the following:

<b>PPI</b>	
<b>PPI, PPI node address</b>	<i>Node address:</i> PPI node address of the PLC <b>Global Device Address: PPI,4</b> Reach by PPI the PLC with the node address 4
<b>ISOTCP</b>	
<b>ISOTCP, IP address, CalledTSAP</b>	<i>IP address :</i> IP address of the PLC (ex: 10.0.120.204) <i>Called TSAP :</i> Transport Service Access Point (ex: 4D.57) <b>Global Device Address: ISOTCP,10.0.120.204,4D.57</b> Reach by ISOTCP the PLC at IP 10.0.120.204 with the TSAP 4D.57

#### 12.3.2.1. ISOTCP polling:

You can always use the TSAP 4D.57. This is the default TSAP for a Step 7-Micro/win connection. If you specify your own connection in the S7-200, be sure to:

- Use the same TSAP for the Server and the Client (ex: Local TSAP:12.00, Remote TSAP:12.00)
- Check the "Enable the Keep Alive function for this connection" feature for this connection otherwise the Ewon will not be able to poll the device after a connection lost due to the Ewon reboot.



**If the PLC address is defined at the Topic level, it can be omitted in the Tag definition.  
 In that case the Tag name will only contain the "ValueName".  
 If the PLC address is specified at the Topic level, it will replace any address defined Tag by Tag.**

## 13. HITACHI IO Server

### 13.1. Introduction

The Hitachi IOServer allows the Ewon to poll tags belonging to a HITACHI EH series PLC using its serial link or Ethernet interface. Ewon uses standard H protocol (for serial RS232), station number H protocol (for serial RS485), and Ethernet H-protocol (for Ethernet).

No remote maintenance functionality is provided by this IO server.

### 13.2. Setup

#### 13.2.1. COM configuration

COM Setup

COM Port:	IO Port1 (COM:1)	
Baud Rate:	9600	Default 9600
Parity:	Even	Default: EVEN
Databits:	7	Default: 7
Stop Bit(s):	1	Default: 1
HW Mode:	Full Duplex NO Handshaking	Default: Full Duplex
Master response timeout:	<input type="text"/> MS	50..50000, default: 3000

Parameter	Description
<b>Baud Rate</b>	Speed of the Serial port. Available speeds are 1200, 2400, 4800, 9600, 19200 Bauds or disabled
<b>Parity</b>	The parity to apply (None / Even / Odd)
<b>Databits</b>	The number of bits in data (7 or 8)
<b>Stop bit(s)</b>	The number of Stop bits (1 or 2)
<b>HW mode</b>	Choose the Hardware mode of communication: <ul style="list-style-type: none"> <li>• Full Duplex HW handshaking (by default)</li> <li>• Full Duplex NO handshaking</li> <li>• Half duplex</li> </ul>
<b>Reply Timeout</b>	Maximum time the Ewon will wait for a valid message response

The Ewon implements Transmission Control Procedure 1 (One-way activation Control Procedure).

When half-duplex mode is used (RS485), the Ewon acts as master.

So, no other master may be on the bus (eg: control panel or host).

### 13.2.2. Topic configuration

Enabled

Destination Device Type and Address:  Enter network configuration dependent PLC address.  
On a serial link: EH or EH,station or EH,LUMP or EH, station, LUMP.  
 On ethernet link: EH,IP address, Protocol,and optionally LUMP

Poll Rate:  MS Default: 2000

Three (3) topics can be used for the IO Server.

These topics are used to give a common property to a group of Hitachi Tags like:

- Enable/Disable
- Global Device Address
- Polling Rate

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Global Device Address</b>	See below for the Device Address Syntax. If an address is specified here, it will replace (overload) the address-defined Tag by Tag.
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

### 13.3. Tag name convention

<b>IO Server Name</b>	HITACHI	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, Global Device Address	PLC address is defined Tag by Tag
	ValueName	Topic PLC Address is used.

The Item Name can contain the PLC address where the value is polled or not.

If an address is also specified at topic level, the address specified at Tag level will be ignored.

## 13.3.1. Value Name

The syntax is the following:

<Memory Type> [<Modifier>] <address>

Symbol	Memory type	Modifier allowed (optional)	Address
<b>X</b>	<b>External Input Bit</b>		Hrusbb H stands for hexadecimal r: remote number (1 digit hexadecimal) u: unit number (1digit hexadecimal) s: slot number (1 digit hexadecimal) bb: bit offset (00 ..FF 2 digits hexadecimal)
<b>WX</b>	<b>External Input Word</b>	<b>W, S</b>	Hrusw H stands for hexadecimal r: remote number (1 digit hexadecimal) u: unit number (1digit hexadecimal) s: slot number (1 digit hexadecimal) w: word offset (1digit hexadecimal)
<b>DX</b>	<b>External Input Double</b>	<b>D, L, F</b>	Hrusw H stands for hexadecimal r: remote number (1 digit hexadecimal) u: unit number (1digit hexadecimal) s: slot number (1 digit hexadecimal) w: word offset (1digit hexadecimal)
<b>Y</b>	<b>External Output Bit</b>		Hrusbb H stands for hexadecimal r: remote number (1 digit hexadecimal) u: unit number (1 digit hexadecimal) s: slot number (1 digit hexadecimal) bb: bit offset (00..FF: 2 digits hexadecimal)
<b>WY</b>	<b>External Output Word</b>	<b>W, S</b>	Hrusw H stands for hexadecimal r: remote number (1 digit hexadecimal) u: unit number (1digit hexadecimal) s: slot number (1 digit hexadecimal) w: word offset (1digit hexadecimal)
<b>DY</b>	<b>External Output Double</b>	<b>D, L, F</b>	Hrusw H stands for hexadecimal r: remote number (1 digit hexadecimal) u: unit number (1digit hexadecimal) s: slot number (1 digit hexadecimal) w: word offset (1digit hexadecimal)
<b>R</b>	<b>Internal Output Bit dedicated</b>		Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating bit offset
<b>WR</b>	<b>Internal Output Word dedicated</b>	<b>W, S</b>	Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating word offset
<b>DR</b>	<b>Internal Output Word dedicated Double</b>	<b>D, L, F</b>	Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating word offset
<b>M</b>	<b>Internal Bit Output Common</b>		Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating bit offset
<b>WM</b>	<b>Internal Word Output Common</b>	<b>W, S</b>	Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating word offset
<b>DM</b>	<b>Internal Word Output Common Double</b>	<b>D, L, F</b>	Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating word offset

<b>L</b>	<b>CPU Link Bit</b>		Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating bit offset
<b>WL</b>	<b>CPU Link Word</b>	<b>W, S</b>	Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating word offset
<b>DL</b>	<b>CPU Link Double</b>	<b>D, L, F</b>	Hxxxxxx H stands for hexadecimal x: 1 to 6 digits hexadecimal indicating word offset



The Modifier can be omitted, the modifier in bold will be used.

Symbol	Modifier	value range	Automatic Tag type
<b>W</b>	<b>Word</b>	0 .. 65535	DWord
<b>S</b>	<b>signed Word</b>	-32768 .. 32767	Integer
<b>D</b>	<b>DWord</b>	0 .. 4294967296 (*)	DWord
<b>L</b>	<b>signed DWord</b>	-2147483648 .. 2147483647 (*)	Integer
<b>F</b>	<b>Float</b>	+/- 3.4e38	Float



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

**Examples**

Address	point to
<b>XH00103</b>	selection of bit3 of word 0 of the input device located at remote:0 unit:0 slot: 1
<b>WYH0011</b>	selection of word 0 of the output device located at at remote:0 unit:0 slot 1
<b>RH105</b>	bit 105 hex (261 decimal) of the Internal Output Bit Dedicated area
<b>WRSH10</b>	short located at word 10 hex of the WR area
<b>WMH10</b>	word 10 hex of the Internal Word Output Common area
<b>DMH10</b>	double word 10 + 11 hex of the Internal Word Output Common area
<b>WMH000010</b>	word 10 hex of the Internal Word Output Common area (WMH10 is more optimal)

• **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for the other Tags, the status Tag ValueName is composed of:

**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 analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

**13.3.2. Global Device Address**

The device address is either appended to the ValueName in the Item Name definition, or entered in the Topic global address fields. The device address syntax is the following:

**EH [, [ [ipaddress , TCP|UDP , port ] | [station] ] [ , LUMP ] ]**

**EH Serial**

<b>EH,station</b>	station : address of the PLC on the network EH,4 the Ewon will access PLC number 4.
<b>EH</b>	By configuration, an HITACHI PLC could have an undefined station number. If this PLC is alone on a serial link, you don't need to specify a station number.
<b>EH,station,LUMP</b>	station : address of the PLC on the network LUMP : Link Unit Port Module EH,4,FFFF0000 the Ewon will access PLC number 4 with the LUMP FFFF0000.
<b>EH,LUMP</b>	By configuration, an HITACHI PLC could have an undefined station number. If this PLC is alone on a serial link, you don't need to specify a station number. LUMP : Link Unit Port Module EH,FFFF0000 the Ewon will access PLC with the LUMP FFFF0000.

**EH ethernet**

<b>EH,ipaddress,TCP UDP,port</b>	ipaddress : IP address of the PLC TCP UDP : select the protocol used port : IP port used <i>EH,10.0.120.203,TCP,3004 the Ewon will access the PLC at 10.0.120.203 with protocol TCP and port 3004</i>
<b>EH,ipaddress,TCP UDP,port,LUMP</b>	ipaddress : IP address of the PLC TCP UDP : select the protocol used port : IP port used LUMP : Link Unit Port Module <i>EH,10.0.120.203,TCP,3004,FFFF0000 the Ewon will access the PLC at 10.0.120.203, protocol TCP, port 3004 and LUMP FFFF0000</i>



**If the PLC address is defined at the Topic level, it can be omitted in the Tag definition. In that case the Tag name will only contain the "ValueName".**  
**If the PLC address is specified at the Topic level, it will replace any address defined Tag by Tag.**

## 14. MITSUBISHI FX IO Server

### 14.1. Introduction

This MITSUFX IOServer allows the Ewon to poll tags belonging to a MITSUBISHI FX series PLC using its RS-422 base unit programming connector. No remote maintenance functionality is provided by this IO server.

The VCOM feature of the Ewon has to be used to cover a remote maintenance. MITSUFX IOServer is available in Ewon with Firmware 5.5 and above.

### 14.2. Setup

#### 14.2.1. COM configuration

COM Setup

COM Port:	<input type="text" value="IO Port1 (COM:1)"/>	
Baud Rate:	<input type="text" value="9600"/>	Default 9600. Higher baud rates are used only with FX3U PLC series
Parity:	<input type="text" value="Even"/>	Default: EVEN
Databits:	<input type="text" value="7"/>	Default: 7
Stop Bit(s):	<input type="text" value="1"/>	Default: 1
HW Mode:	<input type="text" value="Full Duplex NO Handshaking"/>	Default: Full Duplex
Master response timeout:	<input type="text" value=""/>	MS 50..50000, default: 3000

Parameter	Description
<b>Baud Rate</b>	Speed of the Serial port. Available speeds are from 9600 until 115200 Bps or disabled
<b>Parity</b>	The parity to apply (None / Even / Odd)
<b>Databits</b>	The number of bits in data (7 or 8)
<b>Stop bit(s)</b>	The number of Stop bits (1 or 2)
<b>HW mode</b>	Choose the Hardware mode of communication: Full Duplex HW handshaking Full Duplex NO handshaking (by default) Half duplex
<b>Reply Timeout</b>	The maximum time the Ewon will wait for a valid message response



The switches of the Ewon must be configured in RS485/RS422 mode.

### 14.2.2. Topic configuration

Topic A
✔ Enabled

Destination Device Type and Address:  Enter PLC address

Poll Rate:  MS Default: 2000

Topic B
✔ Enabled

Destination Device Type and Address:  Enter PLC address

Poll Rate:  MS Default: 2000

Three (3) topics can be used for the IO Server.

These topics are used to give a common property to a group of Mitsubishi Tags like:

- Enable/Disable
- Global Device Address
- Polling Rate

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Global Device Address</b>	Select the type of Mitsubishi PLC to link. Available FX type are: FX, FX0, FX0N, FX2N or FX3U  There is no address (number) to add because the Ewon is connected directly to one PLC by its programming port. If a device is specified here, it will replace (overload) the device-defined Tag by Tag.
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

### 14.3. Tag name convention

<b>IO Server Name</b>	MITSUFX	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, Global Device Address	PLC address is defined Tag by Tag
	ValueName	Topic PLC Address is used.

The Item Name can contain the PLC address where the value is polled, or not.

If the address is also specified at topic level, the address specified at Tag level will be ignored.

14.3.1. Value Name

The syntax is the following:

<Memory Type Symbol>[<Modifier>]<address>

Symbol	Memory type	Modifier allowed (optional)	Address
<b>X</b>	<b>Input Bit (Boolean)</b>		1 to 3 OCTAL digits
<b>Y</b>	<b>Output Bits (Boolean)</b>		1 to 3 OCTAL digits
<b>M</b>	<b>Auxiliary relays (Boolean)</b>		1 to 4 decimal digits (max 7999)
<b>M</b>	<b>Special Auxiliary relays (Boolean)</b>		1 to 4 decimal digits (min 8000)
<b>S</b>	<b>States (Boolean)</b>		1 to 4 decimal digits
<b>TC</b>	<b>Timer Contacts (Boolean)</b>		1 to 3 decimal digits
<b>CC</b>	<b>Counter Contacts (Boolean)</b>		1 to 3 decimal digits
<b>TR</b>	<b>Timer Reset (Boolean)</b>		1 to 3 decimal digits
<b>CR</b>	<b>Counter Reset (Boolean)</b>		1 to 3 decimal digits
<b>T</b>	<b>Timer Value</b>	W, S	1 to 3 decimal digits
<b>C</b>	<b>Counter Value</b>	W, S	1 to 3 decimal digits (max 199)
<b>C</b>	<b>High Speed Counter Value</b>	L, D	1 to 3 decimal digits (min 200)
<b>D</b>	<b>Data Registers</b>	S, W, L, D, F	1 to 4 decimal digits (max 7999)
<b>D</b>	<b>Special Data Registers</b>	S, W, L, D, F	1 to 4 decimal digits (min 8000)



The Modifier can be omitted, the modifier in bold will be used.

Symbol	Modifier	value range	Automatic Tag type
<b>W</b>	<b>Word</b>	0 .. 65535	DWord
<b>S</b>	<b>signed Word</b>	-32768 .. 32767	Integer
<b>D</b>	<b>DWord</b>	0 .. 4294967296 (*)	DWord
<b>L</b>	<b>signed DWord</b>	-2147483648 .. 2147483647 (*)	Integer
<b>F</b>	<b>Float</b>	+/- 3.4e38	Float



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

## Examples

Address	point to
<b>X14</b>	input bit at octal address 14
<b>D3</b>	data register at address 3 (read as Signed Word)
<b>DD3</b>	data register at address 3 (read as DWord)
<b>DF3</b>	data register at address 3 (read as Float)
<b>D8010</b>	special data register at address 8010 (read as Signed Word)
<b>C199</b>	counter value at address 199 (read as Word)
<b>C200</b>	high speed counter at address 200 (read as DWord)

- **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for the other Tags, the status Tag ValueName is composed of:

**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 analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

## 15. SNMP IO Server

### 15.1. Introduction

The SNMP IO server (Simple Network Management Protocol) is used to communicate with any SNMPv1 device on the network. It allows both read and write operations on 32-bit objects. It supports access to several different devices at the same time.

The following object types are supported: INTEGER, 32bits COUNTER, GAUGE, UNSIGNED, TIMETICK and extracted FLOAT. There is no handling of MIB files.

### 15.2. Setup

Topic A
 Enabled

Destination Device Type and Address:  Enter IP address

Poll Rate:  MS Default: 2000

Three (3) topics can be used for the IO Server. These topics are used to give a common property to a group of SNMP Tags like:

- Enable/Disable
- Global Device Address
- Polling Rate

Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Global Device Address</b>	<p>Select the target SNMP Server device. Syntax:                      IP_Addr,Read_Community,Write_community IP_Addr: IP address of the SNMP Server                      Read_Community: Community used forReading Write_Community: Community used for Writing</p> <p>IP_Addr,Community                      IP_Addr: IP address of the SNMP Server                      Community: Community used for Reading and Writing</p> <p>If an address is specified here, it will replace (overload) the address-defined Tag by Tag.</p>
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

### 15.3. Tag name convention

<b>IO Server Name</b>	SNMP	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, Global Device Address	PLC address is defined Tag by Tag
	ValueName	Topic PLC Address is used.

The Item Name can contain the Device address where the value is polled, or not.  
If the address is also specified at topic level, the address specified at Tag level will be ignored.

#### 15.3.1. Value Name

As the SNMP use an Abstract Syntax Notation (ASN), encode the full OID (Object ID) of the data you want to Read/Write.  
You cannot import MIB files into the Ewon.

INTEGER, 32bits COUNTER, GAUGE, UNSIGNED, TIMETICK and extracted FLOAT (see Note below) are supported

**For all integer objects (not for the extracted FLOAT), use the Integer type to save them.  
Choosing the Automatic type will use the Integer type.  
Otherwise, they could be truncated when Ewon will store them as Float).**



*the SNMP IOServer can read STRING-type OID and converts them in FLOAT. If, for example, a tag contains the string [AnyText 6.78], the extracted result will be [6.78] while Ewon converts it by keeping only the number part of the string. As FLOAT does not exist in SNMP, this is a work around to this limitation. This works in read only, not in write.*

#### Examples

<b>Address</b>	<b>point to</b>
<b>.1.3.6.1.2.1.4.3.0</b>	read the .iso.org.dod.internet.mgmt.mib-2.ip.ipInReceives object
<b>.1.3.6.1.2.1.4.3.0,10.0.0.55,public,private</b>	read the .iso.org.dod.internet.mgmt.mib-2.ip.ipInReceives object at the IP 10.0.0.55 using community "public" for reading and the community "private" for writing.



The address of SCALAR objects ends with a 0 while the address of array items ends with their index in the array (starting from 1).

For example, to read *.iso.org.dod.internet.mgmt.mib-2.ip.ipInReceives* that have the address ".1.3.6.1.2.1.4.3", you must add a ".0" at the end to obtain ".1.3.6.1.2.1.4.3.0".

And to read *.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu* (".1.3.6.1.2.1.2.2.1.4") of the second interface,

because this object is a table, you must add ".2" at the end to obtain ".1.3.6.1.2.1.2.2.1.4.2".

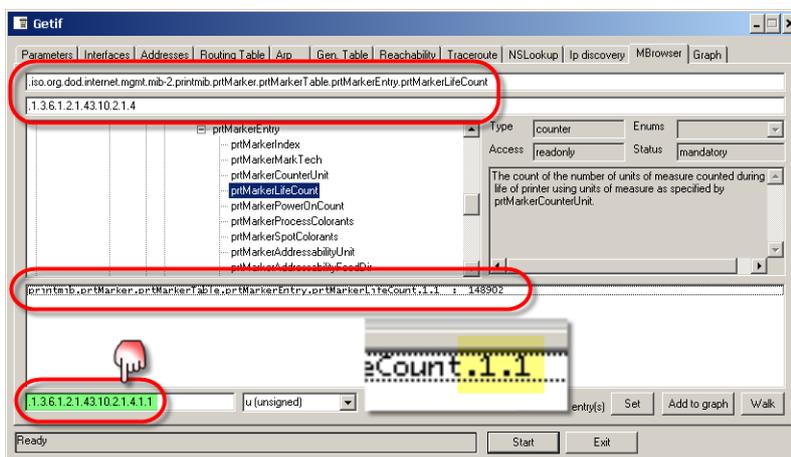


We recommend you to check the SNMP address with an SNMP Client to find the right address. (like Getif <http://www.wtcs.org/snmp4tpc/getif.htm>)

**Example:**

Searching in the MIB file of a network printer to find the address of the "TotalNumberOfPagePrinted", we found *.iso.org.dod.internet.mgmt.mib-2.printmib.prtMarker.prtMarkerTable.prtMarkerEntry.prtMarkerLifeCount* at address ".1.3.6.1.2.1.43.10.2.1.4"

Using the MIB file with Getif to retrieve the whole address of this Counter, we see that the Counter is in a 2 dimensions table and we need to add ".1.1" to get its value.



• **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for the other Tags, the status Tag ValueName is composed of:

**Status, Global Device Address**

- You can define a status Tag for each Device used.
- If you use the status address, the Tag must be configured as analog.

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.

## 16. MITSUBISHI MELSEC IO Server

### 16.1. Introduction

This MELSEC IO Server allows the Ewon to poll tags belonging to the MITSUBISHI Q-Series PLC using the protocol on the Ethernet interface. To cover remote maintenance for the Mitsubishi PLCs, the Ethernet routing feature or VCOM feature (serial) of the Ewon have to be used.

The MELSEC IO Server is available from firmware version 6.3 onwards.

Ewon also integrates the E1 frames of the MC protocol to allow polling the FX3U and A-Series PLC's through their Ethernet interface

### 16.2. Setup

#### 16.2.1. Topic configuration

Enabled

Destination Device Type and Address:  Enter PC Number

Poll Rate:  MS Default: 2000

Three (3) topics can be used for the IO Server.

These topics are used to give a common property to a group of Mitsubishi Tags like:

- Enable/Disable
- Global Device Address
- Polling Rate

Topic configuration item	Description
<b>Topic enabled</b>	<b>Enables or disables polling of all the Tags in the topic.</b>
<b>Global Device Address Q-Series QCPU,QnACPU &amp; QOCPU</b>	Syntax: Model,Protocol,IP Address,Port Number,Network Number,PC Number [,ReqDestModuleIONo,ReqDestModuleStationNo], where:  Model: QCPU, QnACPU or QOCPU Protocol: TCP, UDP IP Address: is the IP address of the PLC Port Number: is the Port of the PLC on which it is listening for MC protocol Network Number: network number of the target (1..239) (0) PC Number: Target Station Number (1..64) (FF) Optional ReqDestModuleIONo default FF 03 Optional ReqDestModuleStationNo default 0 Example: QCPU,UDP,198.162.120.155,5000,1,1

Topic configuration item	Description
<b>Global Device Address A-Series</b>  <b>ACPU</b>	Syntax: Model,Protocol,IP Address,Port Number,PC Number where:  Model: ACPUCPU Protocol: TCP, UDP IP Address: is the IP address of the PLC Port Number: is the Port of the PLC on which it is listening for MC protocol Network Number: network number of the target (1..239) (0) Example: ACPUCPU,UDP,198.162.120.155,5000,1
<b>Global Device Address FX-Series</b>  <b>FX3UCPU</b>	Syntax: Model,Protocol,IP Address,Port Number where:  Model: FX3UCPU Protocol: TCP, UDP IP Address: is the IP address of the PLC Port Number: is the Port of the PLC on which it is listening for MC protocol Example: FX3UCPU,UDP,198.162.120.155,5000
<b>Poll rate</b>	Defines the refresh rate of the Tag name. In a complex application, we can imagine that some Tag names must be refreshed every second - typically for digital input - and other every minute - typically: temperature-.

Temporary hidden table

#### Examples of Global Device Address syntax

CPU	Protocol	IP Address	Port #	Network ID of PLC	Network ID of PC	IO Server settings
Q	TCP	192.168.140.10	5000	1	5	QCPU, TCP,192.168.140.10, 5000,1,5
Q	UDP	192.168.140.10	5000	1	5	QCPU, UDP,192.168.140.10, 5000,1,5
QNA	TCP	192.168.140.10	5000	1	5	QNACPU, TCP,192.168.140.10, 5000,1,5
QNA	UDP	192.168.140.10	5000	1	5	QNACPU, UDP,192.168.140.10, 5000,1,5
QO	TCP	192.168.140.10	5000	1	5	QOCPU, TCP,192.168.140.10, 5000,1,5
QO	UDP	192.168.140.10	5000	1	5	QOCPU, UDP,192.168.140.10, 5000,1,5
A	TCP	192.168.1.70	1284	None	255	ACPU, TCP,192.168.1.70, 1284, 255
A	UDP	192.168.1.70	1284	None	255	ACPU, UDP,192.168.1.70, 1284, 255
FX3U	TCP	192.168.140.10	5001	None	None	FX3UCPU, TCP,192.168.140.10, 5001,1,5
FX3U	UDP	192.168.140.10	5002	None	None	FX3UCPU, UDP,192.168.140.10, 5002,1,5

### 16.3. Tag name convention

<b>IO Server Name</b>	MELSEC	
<b>Topic Name</b>	A	
	B	
	C	
<b>Item Name</b>	ValueName, Global Device Address	PLC address is defined Tag by Tag
	ValueName	Topic PLC Address is used.

The Item Name can contain the PLC address where the value is polled, or not.

If the address is also specified at topic level, the address specified at Tag level will be ignored.

### 16.3.1. Value Name

The syntax is the following:

<Memory Type Symbol>[<Modifier>]<address>

Symbol	Memory type	Modifier allowed (optional)	Address
X	Input		1 to 6 hexadecimal bits
DX	Direct Input		1 to 6 hexadecimal bits
Y	Output		1 to 6 hexadecimal bits
DY	Direct Output		1 to 6 hexadecimal bits
B	Link Relay		1 to 6 hexadecimal bits
SB	Special Link Relay		1 to 6 hexadecimal bits
M	Internal Relay		1 to 6 decimal digits
SM	Special Internal Relay		1 to 6 decimal digits
L	Latch Relay		1 to 6 decimal digits
F	Annunciator Relay		1 to 6 decimal digits
V	Edge Relay		1 to 6 decimal digits
S	Step relay		1 to 6 decimal digits
TS	Timer Contact		1 to 6 decimal digits
TC	Timer Coil		1 to 6 decimal digits
TN	Timer Value	<b>W, S</b>	1 to 6 decimal digits
SS	Integrating Timer Contact		1 to 6 decimal digits
SC	Integrating Timer Coil		1 to 6 decimal digits
SN	IntegratingTimer Value	<b>W, S</b>	1 to 6 decimal digits
CS	Counter Contact		1 to 6 decimal digits
CC	Counter Coil		1 to 6 decimal digits
CN	Counter Value	<b>W, S</b>	1 to 6 decimal digits
D	Data Register	<b>S, W, L, D, F</b>	1 to 6 decimal digits
SD	Special Data Register	<b>S, W, L, D, F</b>	1 to 6 decimal digits
W	Link Register	<b>S, W, L, D, F</b>	1 to 6 decimal digits
SW	Special Link Register	<b>S, W, L, D, F</b>	1 to 6 decimal digits
R	File Register	<b>S, W, L, D, F</b>	1 to 6 decimal digits
Z	Index Register	<b>S, W, L, D, F</b>	1 to 6 decimal digits



The Modifier can be omitted, the modifier in bold will be used.

Symbol	Modifier	value range	Automatic Tag type
<b>W</b>	<b>Word</b>	0 .. 65535	DWord
<b>S</b>	<b>signed Word</b>	-32768 .. 32767	Integer
<b>D</b>	<b>DWord</b>	0 .. 4294967296 (*)	DWord
<b>L</b>	<b>signed DWord</b>	-2147483648 .. 2147483647 (*)	Integer
<b>F</b>	<b>Float</b>	+/- 3.4e38	Float



(\*)To avoid loss of precision due to Integer to float conversion, choose the right storage DataType for your Tag.

**Examples**

Address	point to
X14	input bit at octal address 14
D3	data register at address 3 (read as Signed Word)
DD3	data register at address 3 (read as DWord)
DF3	data register at address 3 (read as Float)
D8010	special data register at address 8010 (read as Signed Word)

- **Status register:**

The STATUS Tag is a special Tag that returns information about the current state of communication for a given device. As for the other Tags, the status Tag ValueName is composed of:

**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 analog.

<b>0</b>	Communication not initialized. Status UNKNOWN. If no Tag is polled on that device address, the communication status is unknown.
<b>1</b>	Communication OK.
<b>2</b>	Communication NOT OK.

## 17. BACNET IO Server

### 17.1. Introduction

This IO Server uses the BACNET/IP protocol (based on UDP/IP) to collect the data from field devices acting as BACNET/IP Servers.

### 17.2. Setup

#### 17.2.1. General Setup

**General Setup**

eWON Bacnet Device ID:  0..4194303, leave empty to get a value assigned automatically

eWON Bacnet Device Port:  1024..65535, default is 47808

Parameter	Description
<b>Ewon BACNET Device ID</b>	Set the BACNET device ID of the Flexy. This number must be unique in your BACNET network. Leave it blank to let the Flexy choose a device ID
<b>Ewon BACTNET Port</b>	Select the UDP port used for the BACNET IP communication

#### 17.2.2. Topic configuration

The BACNET IO Server provides also three Topics used to define different polling rates.

The topic settings (except the pollrate) are optional and can be defined in the Tag address as well. This allows you to poll more than 3 BACNET devices.

**Topic A** Enabled

Destination Device Type and Address:  Enter Bacnet Device ID

Poll Rate:  MS Default: 2000

Write Priority:  Value between 1 (highest) and 16 (lowest) or 0 (no priority specified)

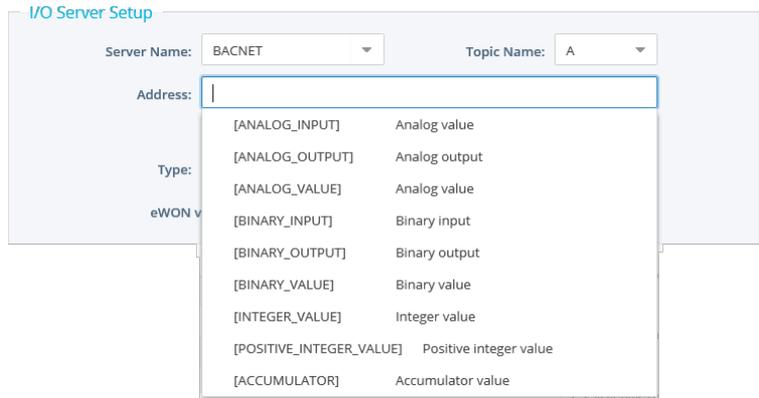
Topic configuration item	Description
<b>Topic enabled</b>	Enables or disables polling of all the Tags in the topic.
<b>Destination Device Type and Address</b>	Device ID of the device to poll. There is no IP address to define for the device as the BACNET IP protocol can detect the Bacnet devices on the network using an UDP Broadcast packet.  the IP address of the Bacnet device must still be in the same range as the Ewon IP address
<b>Poll rate</b>	Polling rate for this topic (in milliseconds). Default: 2000 ms
<b>Write priority</b>	0 or blank : no priority defined, 1 : highest priority, 16 : lowest priority

### 17.3. Tag name convention

The address of the Tag is composed of four parameters :

1. Object Type (mandatory)
2. Instance Number (mandatory)
3. Property of the object (optional. If not defined, the property "Present Value" is used)
4. Device ID (optional if defined in the topic settings)

**17.3.1. Object Type**



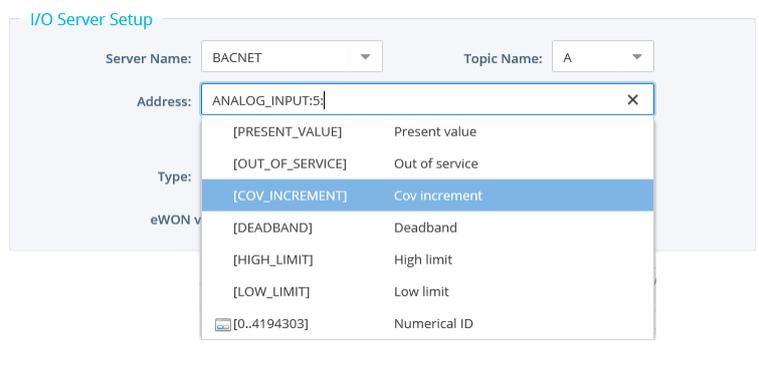
This is the list of the supported object types:

- ACCUMULATOR (23)
- ANALOG\_INPUT (0)
- ANALOG\_OUTPUT (1)
- ANALOG\_VALUE (2)
- BINARY\_INPUT (3)
- BINARY\_OUTPUT (4)
- BINARY\_VALUE (5)
- INTEGER\_VALUE (45)
- POSITIVE\_INTEGER\_VALUE (48)
- PULSE\_CONVERTER (24)

**17.3.2. Instance Number**

The instance number defines the ID of the object you want to access to

**17.3.3. Property of the object**



An object is composed of different properties. These properties can be accessed through different Tags.

You have therefore to define which property you want to access for your Tag. If no property is defined, the property "PRESENT\_VALUE" is used.

Here is the list of the property you have access to:

- PRESENT\_VALUE (85)
- OUT\_OF\_SERVICE (81)
- COV\_INCREMENT (22)
- DEADBAND (25)
- HIGH\_LIMIT (45)
- LOW\_LIMIT (5)
- Any numerical ID, if not listed (Example: STATUS\_FLAGS value by using the numerical ID 111)

## 18. OPCUA IO Server

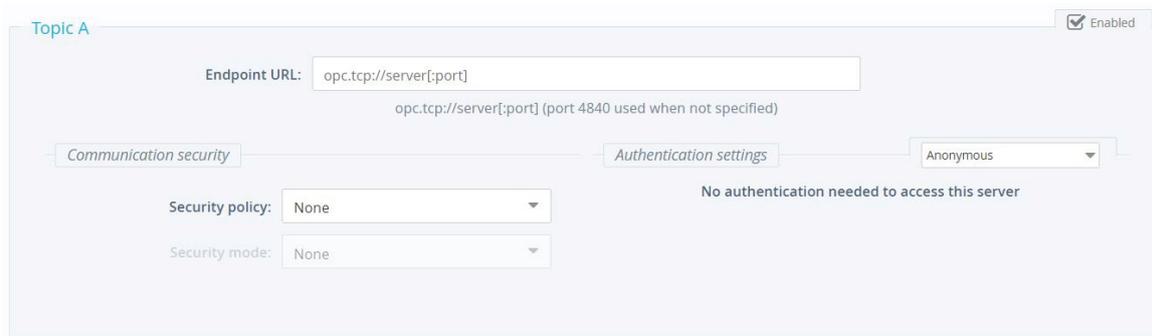
### 18.1. Introduction

This info applies to Ewon Flexy devices. Since firmware 13.0s0, the Flexy integrates a new IO Server called OPCUA. This IO Server allows Ewon to act as an OPCUA client in order to retrieve tag values from other OPCUA servers, like PLCs, HMIs, etc. The OPCUA IO-server allows Ewon to connect to a maximum of 3 different OPCUA devices.

### 18.2. Setup

#### 18.2.1. Topic configuration

In this IO Server, you can configure 3 different topics with 3 different OPCUA Servers (different endpoints and authentication settings).



Topic configuration item	Description
<b>Topic enabled</b>	By checking the box, you enable the topic to be published.
<b>Endpoint URL</b>	The address of the OPC UA device you want to retrieve tags from, which must follow the syntax: <code>opc.tcp://[server]:[port]</code> . This address must be in the same range than the IP address of your Flexy.
<b>Security policy</b>	The level of security policy.
<b>Security mode</b>	The security method applied to the communication: None: no security Sign: will allow you to sign the communication to ensure both sides (client and server) know each other. Sign and encrypt: will do the same as Sign and encrypt your communication in a secure way
<b>Authentication settings</b>	The authentication to be able to publish to the Ewon OPC UA server can be anonymous (server widely open), protected by a combination of user/password or by a certificate. In the case of a user/password combination, the OPC UA client will have to log in with credentials accepted by the OPC UA server. In case of certificates, you will need to select a certificate (.der file) and a private key (. pem file). These files will need to be uploaded in the PKI frame first.

Encode the OPCUA endpoint URL of the device you want to connect with.

As for example: **`opc.tcp://192.168.0.54:4840`**  
where 192.168.0.54 is the IP address of the OPCUA device and 4840 the port used for OPCUA on this device.

Once configured, you need to enable the topic.

**Note:** The IP address of the OPCUA device needs to be compatible with your Ewon network setup.

### 18.2.2. Certificate Management

This allows the encryption and the authentication with certificates for the OPC UA server of the Ewon.

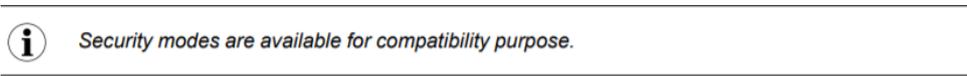
The Ewon is capable of handling different security policy & security mode.

Security policy:

- None: no encryption
- Basic256
- Basic256Sha256

Security mode:

- None: no encryption
- Sign: the access control, which allows to trust or reject the connection on both sides
- Sign and encrypt: the access control and encryption



The recommended setup would be:

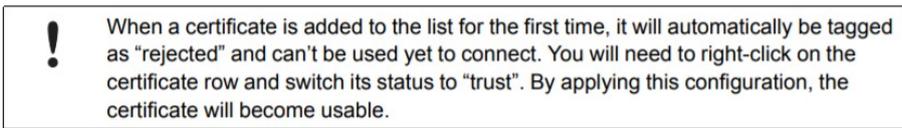
- Basic256Sha256
- Sign and encrypt

Certificates can be manually added by clicking the “+” icon which makes it possible to upload new certificates/private key for the authentication part.

These certificates are divided in 3 categories:

- **Own**: the client certificate, used for signing/encryption.
- **Auth**: the authentication certificate, used to log in with a certificate.
- **Trusted/Rejected**: this is used to control the access (signing procedure) and ensure to know who can and cannot connect from/to.

Only certificates tagged as “own”, “auth” and “trusted” will be used. Certificates tagged as “rejected” will be listed but ignored.



If the certificates can be dealt via the GUI, it can also be managed through FTP. All the operations (the PKI) can be done by switching certificates/keys from one folder to another one.

Each PKI is stored in the /usr dir of your Flexy. The default folder for the OPC UA client is /usr/ pki\_cli.

In the PKI directory, there are 4 subdirectories:

- Auth**: the storage of the certificates and private keys used for authentication.
- Own**: the storage of its own certificates (identity) for signing purpose.
- Rejected**: the storage of the rejected certificates (default folder).
- Trusted**: The storage of the accepted/trusted certificates.

### 18.2.3. OPCUA certificate check option

The System config parameter “OpcuaIOSrvIgnCertErr” allows to configure the Opcua IOserver to bypass some certificate check errors.

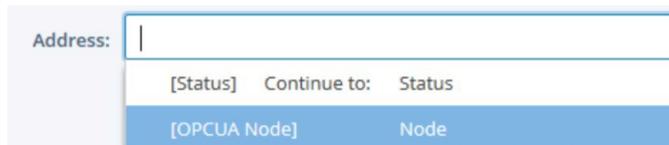
OpcuaIOSrvIgnCertErr	Description
0	Default value: No Ignore of errors, Certificates must be valid
1	Ignore “untrusted certificate error”
2	Ignore “hostname invalid error”
4	Ignore “use not allowed error”
8	Ignore “URI invalid error”

### 18.3. Tag name convention

When configuring an Ewon Tag on the OPCUA IOserver, you have the possibility to browse the OPCUA server to which you want to connect. Create a TAG by specifying a Tag Name and select for the Server Name "OPCUA".

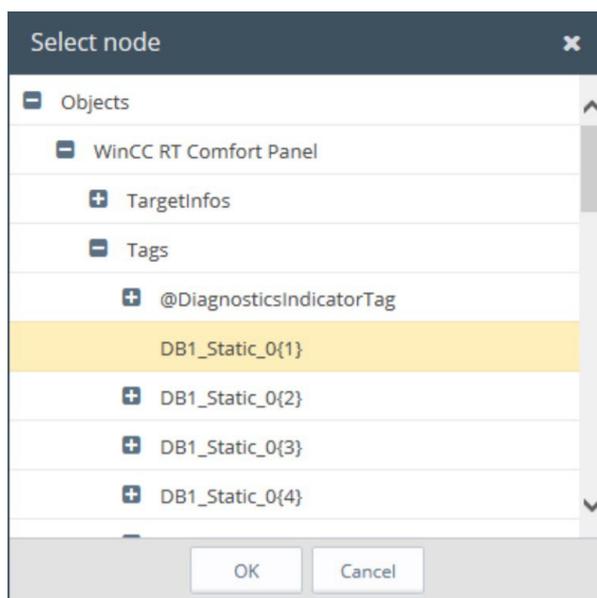
For the Topic Name, select the one you just configured before (A,B or C). This will link the Tag to the OPCUA device you want to reach.

To specify the address, click on the field and the Tag Helper will propose you 2 choices:



- Status: indicates the current status of the tag.
- OPCUA Node: will link the Flexy tag to one of the OPC UA device value

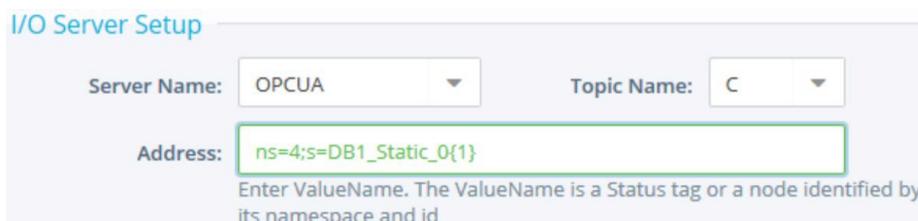
To link the Tag to the device you want to connect to, select [OPCUA Node] The OPCUA browser window will open:



The browsing is dynamic and refreshes each time you click on "+".

**Note:** You never have a complete view of the node architecture because it can be huge and it's not really a tree It's a graph and can contain cycles.

Once you found your OPCUA node on your OPCUA server, click on it and then click on "OK" to validate the IO address. The Tag Address field will now contain the correct syntax for the OPCUA node.



The Tag type field (Boolean, Integer, etc.) is automatically updated with the corresponding type. Fill in the other Tag parameters if required (alarm setting, historical settings, etc.) and click on "Add Tag" to create the Tag.

**!** Strings data type are currently in ASCII format and it is supported by OPCUA I/O server. UTF-8 is not supported yet.

