eWON drivers

CP SISTEMI

PRODUCTIVE PROCESS SOFTWARE

www.ewondrivers.com

TABLE OF CONTENTS

TABLE OF CONTENTS 2
1. Preface 3
2. Installation 4
2.1. Remove previous version 4
2.2. Installation 6
2.3. TAG Setup 7
2.4. First run 8
2.5. License Key 8
2.6. Start and Stop commands 9
3. TAG's structure 10
3.1. Serial port or TCP/IP configuration TAG (Type 1) 10
3.2. Device configuration TAG (Type 2) 12
3.3. Variable TAG (Type 3) 13
3.4. TAG hierarchy 15
3.5. State, Commands and Debug 16
3.5.1. Serial port configuration TAG 16
3.5.2. Device configuration TAG 16
3.5.3. Command TAG 16
4. TAG Customization 20
5. Using the MODBUS gateway 20

1. Preface

The **eWon drivers eXtender** provides eWON devices with a flexible solution for extending communication protocols. The software is installed directly into eWON device and can be controlled by the native web interface. For sake of simplicity the configuration is done via web interface, in the same way you setup an eWON TAG.

For each communication driver we provide a sample set of TAG already configured and ready to use.

This software can be evaluated in **demo mode** for 10 minutes from the eWON's startup.
To enable demo mode you need to generate a demo license file for your eWON. To get a license file please visit www.ewondrivers.com

IMPORTANT

If
a) you modify and save any tag's parameter (description, address, etc.) while **eWon drivers eXtender** is running and
b) you are using a driver that is able to write the tag value to field and
c) the global writing flag is enabled

then this will cause the driver to write a "zero" value to the tag's value and so to the field.

In order to avoid this behavior you have two options:
1. Disable the global writing flag during the tag setup phase. Please refer to chapter 3.5.3 Command TAG.
2. Stop **eWon drivers eXtender** during the tag setup phase. Please refer to chapters 2.6 Start and Stop commands and 3.5.3 Command TAG.

Drivers capable of writing tag values to field are available for **eWon drivers eXtender** version 4.0 and above. Please refer to the specific driver documentation to know if it is capable of writing tag values to field.

2. Installation

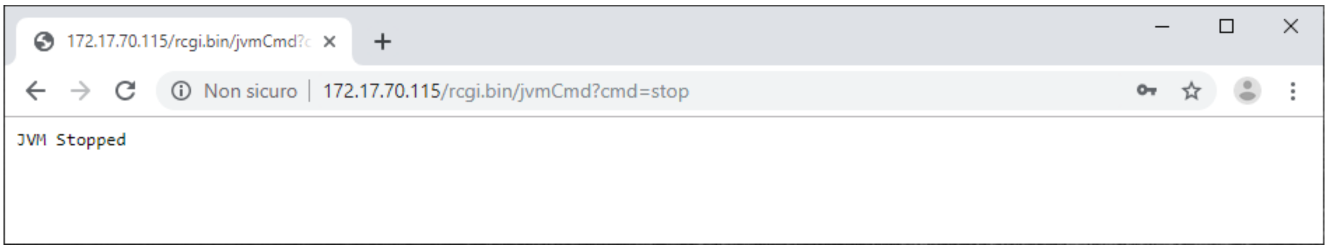
The installation phase consists of five steps:

1. File transfer
2. TAG Setup (driver's specific sample file)
3. First run
4. License Key
5. Start and Stop commands

2.1. Remove previous version

Before installing a new **eWon drivers eXtender** release you need to remove the previous installed version, if any. To do this you need to stop the JVM:

STOP:
http://*IP-eWon*/rcgi.bin/jvmCmd?cmd=stop

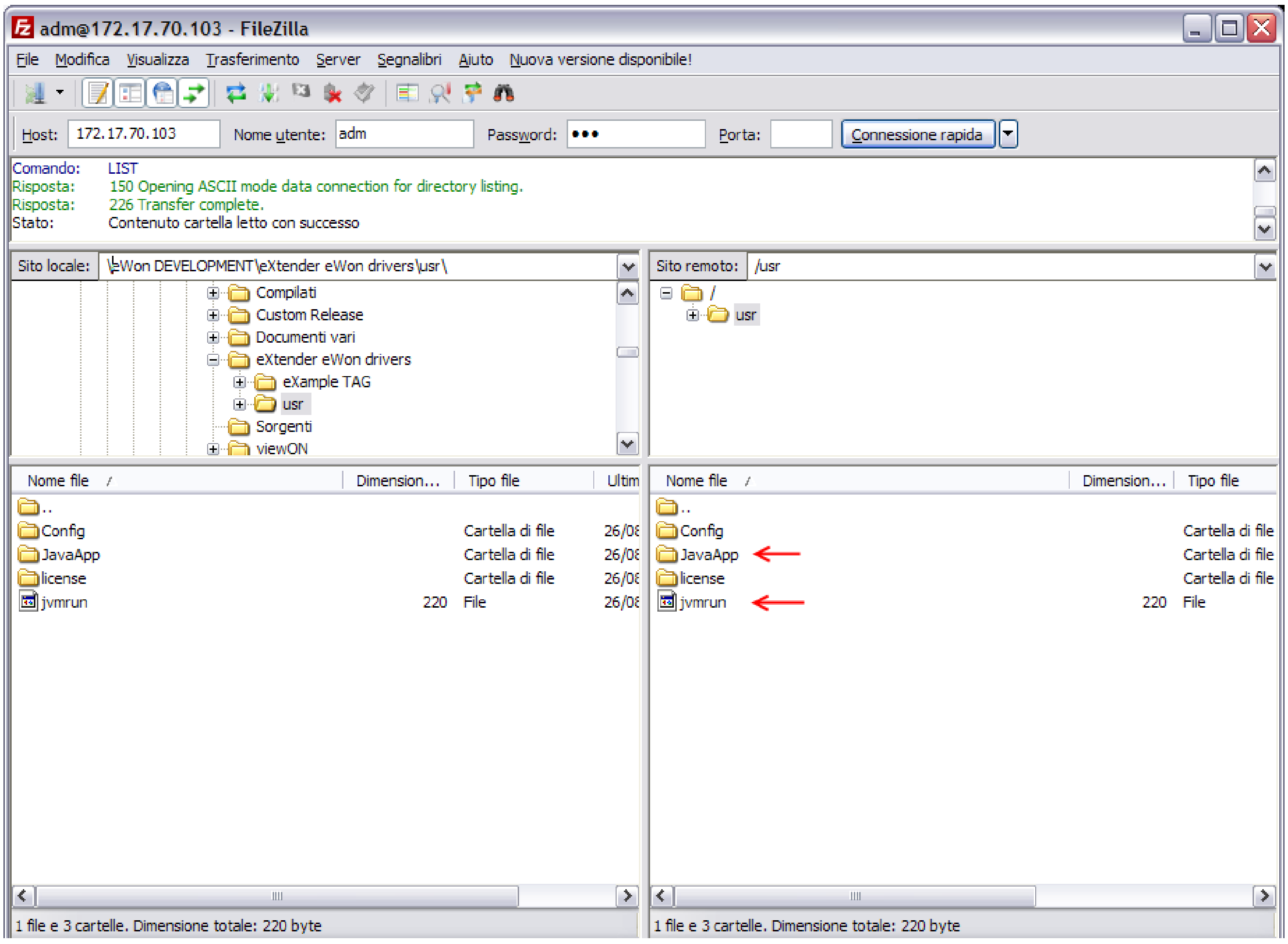


172.17.70.115/rcgi.bin/jvmCmd?cmd=stop

```
JVM Stopped
```

and delete the following contents that you can find under /usr directory using an FTP client like Filezilla:

- JavaApp
  - Directory with **eWon drivers eXtender**.
- Jvmrun
  - If this file exists **eWon drivers eXtender** starts when eWON starts.


adm@172.17.70.103 - FileZilla

Host: 172.17.70.103 Nome utente: adm Password: ••• Porta: Connessione rapida

```
Comando:    LIST
Risposta:   150 Opening ASCII mode data connection for directory listing.
Risposta:   226 Transfer complete.
Stato:      Contenuto cartella letto con successo
```

Sito locale: \eWon DEVELOPMENT\eXtender eWon drivers\usr\

Sito remoto: /usr

Nome file	Dimension...	Tipo file	Ultim
..			
Config		Cartella di file	26/08
JavaApp		Cartella di file	26/08
license		Cartella di file	26/08
jvmrun	220	File	26/08

1 file e 3 cartelle. Dimensione totale: 220 byte

Nome file	Dimension...	Tipo file
..		
Config		Cartella di file
JavaApp		Cartella di file
license		Cartella di file
jvmrun	220	File

1 file e 3 cartelle. Dimensione totale: 220 byte

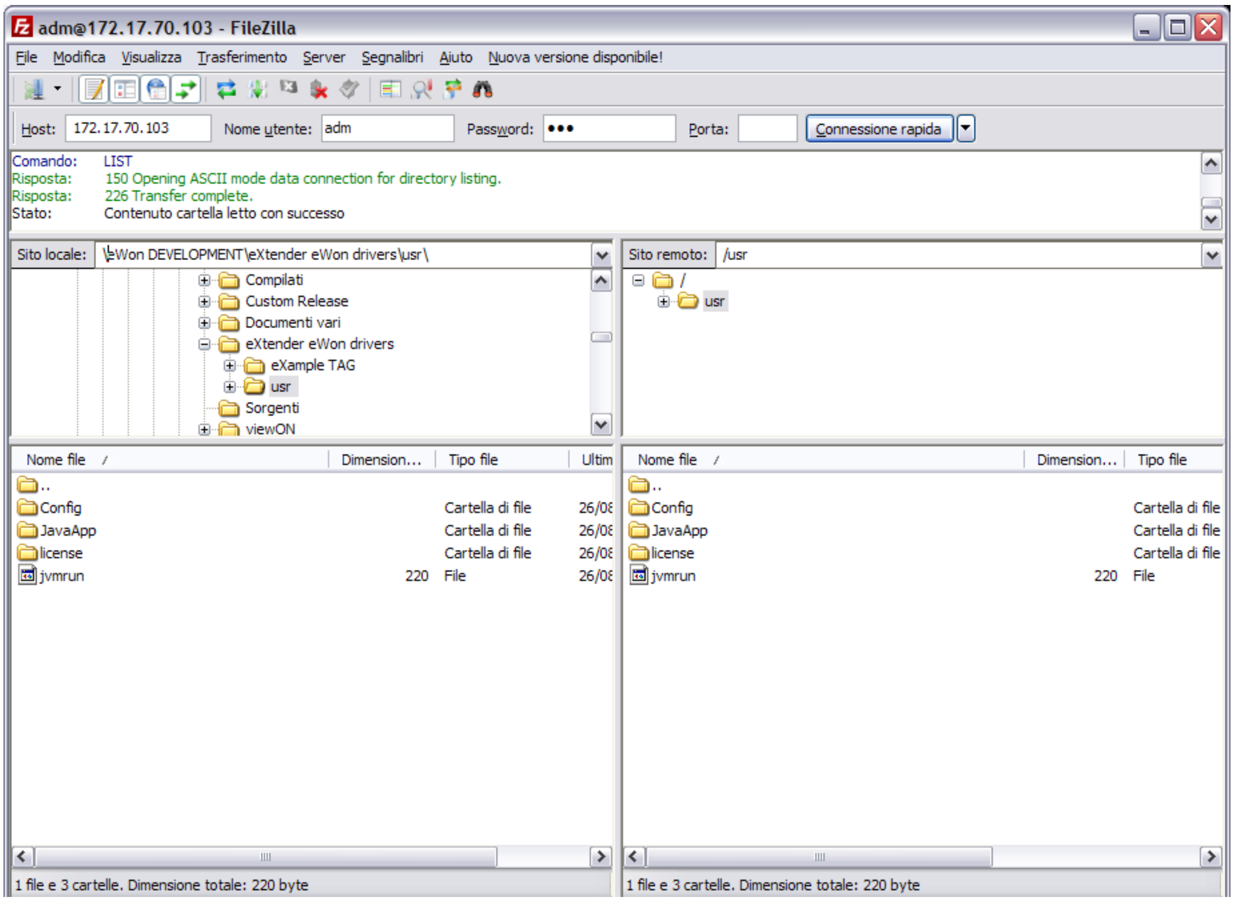
2.2. Installation

Extract the **eWon drivers eXtender** archive and upload the **"usr"** directory content in the eWON's **"/usr"** directory using an FTP client like Filezilla.

<u>Usr's contents:</u>

- Config
  - Directory with CSV file for custom TAG configuration.
- JavaApp
  - Directory with **eWon drivers eXtender**.
- license
  - Directory with license files.
- Jvmrun
  - If this file exists **eWon drivers eXtender** starts when eWON starts.

At the end of the transfer you should be in this state:


adm@172.17.70.103 - FileZilla
File Modifica Visualizza Trasferimento Server Segnalibri Aiuto Nuova versione disponibile!
Connessione rapida

Host: 172.17.70.103 Nome utente: adm Password: ••• Porta:

```
Comando:  LIST
Risposta: 150 Opening ASCII mode data connection for directory listing.
Risposta: 226 Transfer complete.
Stato:    Contenuto cartella letto con successo
```

Sito locale: \eWon DEVELOPMENT\eXtender eWon drivers\usr\

```
⊞ Compilati
⊞ Custom Release
⊞ Documenti vari
⊟ eXtender eWon drivers
    ⊞ eXample TAG
    ⊞ usr
  Sorgenti
⊞ viewON
```

Sito remoto: /usr

```
⊟ /
    ⊞ usr
```

Nome file	Dimension...	Tipo file	Ultim
..			
Config		Cartella di file	26/0ł
JavaApp		Cartella di file	26/0ł
license		Cartella di file	26/0ł
jvmrun	220	File	26/0ł

1 file e 3 cartelle. Dimensione totale: 220 byte

Nome file	Dimension...	Tipo file
..		
Config		Cartella di file
JavaApp		Cartella di file
license		Cartella di file
jvmrun	220	File

1 file e 3 cartelle. Dimensione totale: 220 byte

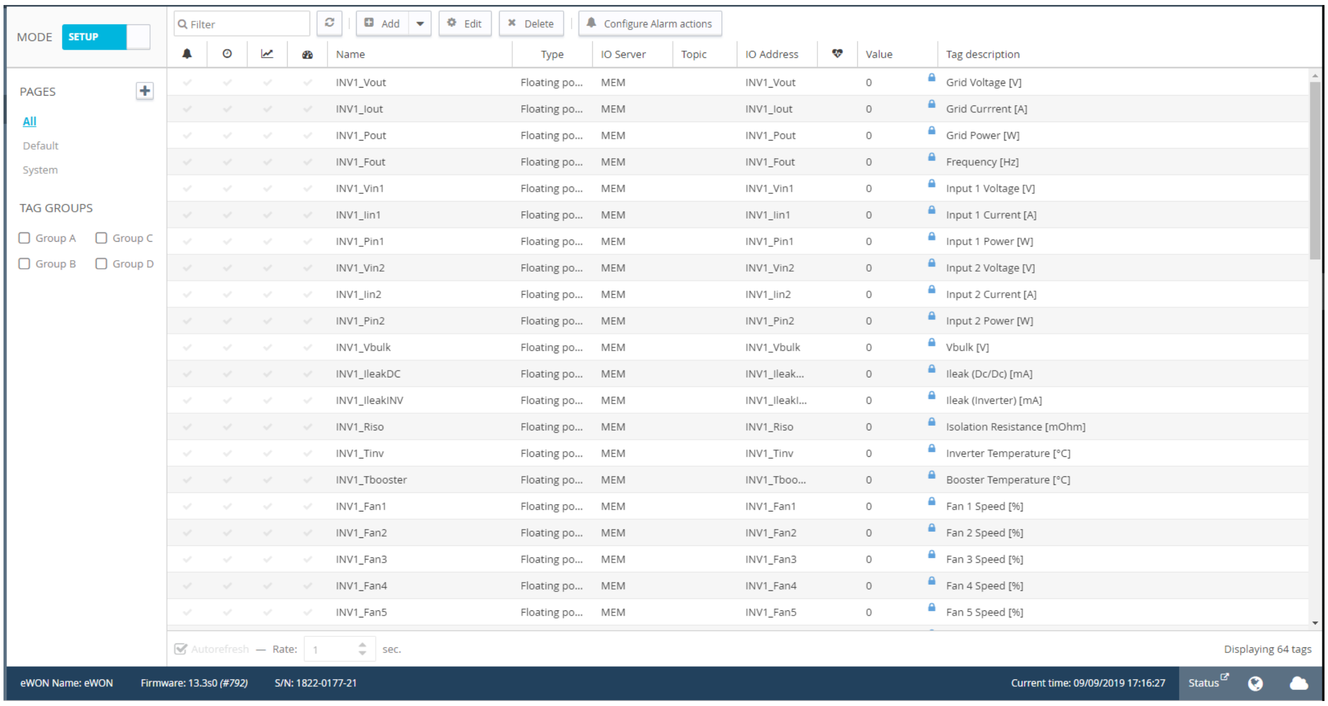
2.3. *TAG Setup*

For a quick configuration, for each type of protocol will be given a file **"var_lst.csv"** for the automatic generation of some sample TAG. The samples in this guide will refer to ***Aurora Power-One*** protocol.

Open the **"Aurora PowerOne"** directory in the **"eXampleTAG"** directory. Here you can find the **"var_lst.csv"** file. Upload **"var_lst.csv"** in the eWON root directory.

At the end of the transfer you got a ready to use TAG List for an **Aurora Power-One** inverter on node address 2. (see chapter 3.2 for more information on node address)

At the end of the transfer you got this TAG list:



				Name	Type	IO Server	Topic	IO Address		Value	Tag description
✓	✓	✓	✓	INV1_Vout	Floating po...	MEM		INV1_Vout		0	Grid Voltage [V]
✓	✓	✓	✓	INV1_Iout	Floating po...	MEM		INV1_Iout		0	Grid Currrent [A]
✓	✓	✓	✓	INV1_Pout	Floating po...	MEM		INV1_Pout		0	Grid Power [W]
✓	✓	✓	✓	INV1_Fout	Floating po...	MEM		INV1_Fout		0	Frequency [Hz]
✓	✓	✓	✓	INV1_Vin1	Floating po...	MEM		INV1_Vin1		0	Input 1 Voltage [V]
✓	✓	✓	✓	INV1_Iin1	Floating po...	MEM		INV1_Iin1		0	Input 1 Current [A]
✓	✓	✓	✓	INV1_Pin1	Floating po...	MEM		INV1_Pin1		0	Input 1 Power [W]
✓	✓	✓	✓	INV1_Vin2	Floating po...	MEM		INV1_Vin2		0	Input 2 Voltage [V]
✓	✓	✓	✓	INV1_Iin2	Floating po...	MEM		INV1_Iin2		0	Input 2 Current [A]
✓	✓	✓	✓	INV1_Pin2	Floating po...	MEM		INV1_Pin2		0	Input 2 Power [W]
✓	✓	✓	✓	INV1_Vbulk	Floating po...	MEM		INV1_Vbulk		0	Vbulk [V]
✓	✓	✓	✓	INV1_IleakDC	Floating po...	MEM		INV1_Ileak...		0	Ileak (Dc/Dc) [mA]
✓	✓	✓	✓	INV1_IleakINV	Floating po...	MEM		INV1_IleakI...		0	Ileak (Inverter) [mA]
✓	✓	✓	✓	INV1_Riso	Floating po...	MEM		INV1_Riso		0	Isolation Resistance [mOhm]
✓	✓	✓	✓	INV1_Tinv	Floating po...	MEM		INV1_Tinv		0	Inverter Temperature [°C]
✓	✓	✓	✓	INV1_Tbooster	Floating po...	MEM		INV1_Tboo...		0	Booster Temperature [°C]
✓	✓	✓	✓	INV1_Fan1	Floating po...	MEM		INV1_Fan1		0	Fan 1 Speed [%]
✓	✓	✓	✓	INV1_Fan2	Floating po...	MEM		INV1_Fan2		0	Fan 2 Speed [%]
✓	✓	✓	✓	INV1_Fan3	Floating po...	MEM		INV1_Fan3		0	Fan 3 Speed [%]
✓	✓	✓	✓	INV1_Fan4	Floating po...	MEM		INV1_Fan4		0	Fan 4 Speed [%]
✓	✓	✓	✓	INV1_Fan5	Floating po...	MEM		INV1_Fan5		0	Fan 5 Speed [%]

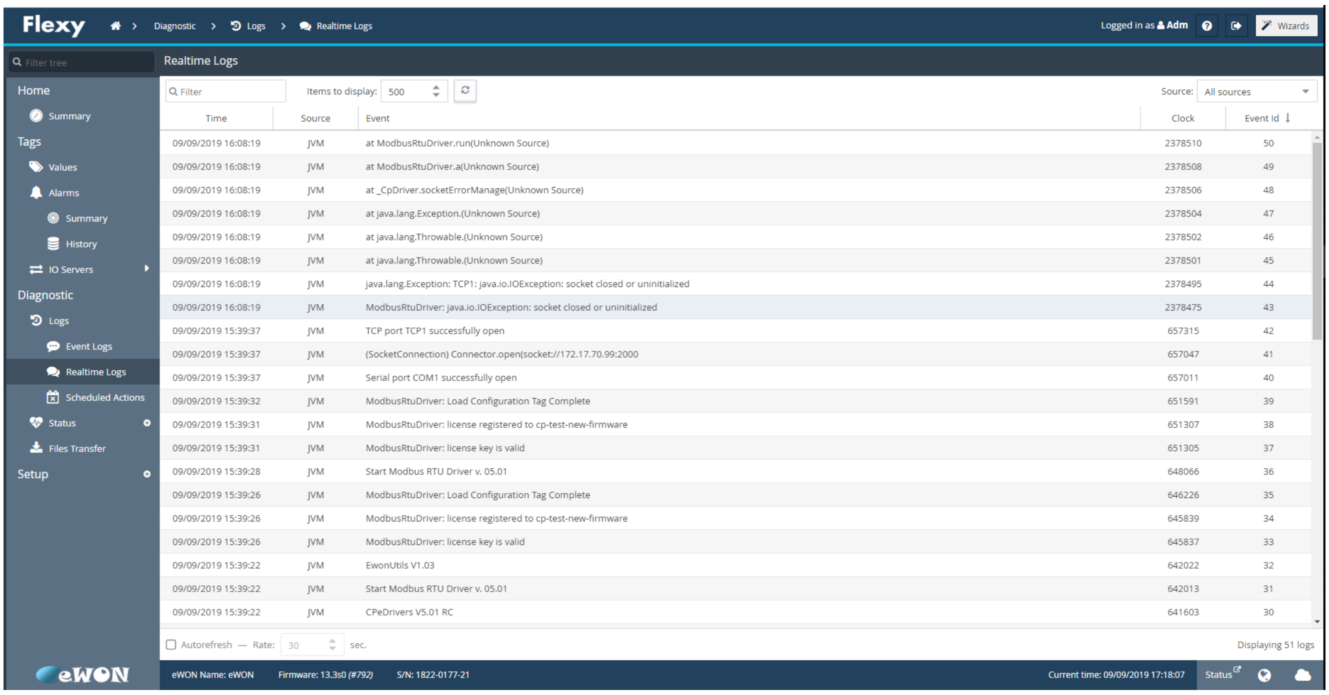
Rate: 1 sec.

Displaying 64 tags

eWON Name: eWON Firmware: 13.3s0 (#792) S/N: 1822-0177-21 Current time: 09/09/2019 17:16:27

2.4. First run

If you correctly uploaded **“jvmrun”** on the **“/usr”** eWON’s directory just reboot and wait for loading. Log messages are visible in the ***Real Time Log*** page of the eWON:



Time	Source	Event	Clock	Event Id
09/09/2019 16:08:19	JVM	at ModbusRtuDriver.run(Unknown Source)	2378510	50
09/09/2019 16:08:19	JVM	at ModbusRtuDriver.a(Unknown Source)	2378508	49
09/09/2019 16:08:19	JVM	at _CpDriver.socketErrorManage(Unknown Source)	2378506	48
09/09/2019 16:08:19	JVM	at java.lang.Exception.(Unknown Source)	2378504	47
09/09/2019 16:08:19	JVM	at java.lang.Throwable.(Unknown Source)	2378502	46
09/09/2019 16:08:19	JVM	at java.lang.Throwable.(Unknown Source)	2378501	45
09/09/2019 16:08:19	JVM	java.lang.Exception: TCP1: java.io.IOException: socket closed or uninitialized	2378495	44
09/09/2019 16:08:19	JVM	ModbusRtuDriver: java.io.IOException: socket closed or uninitialized	2378475	43
09/09/2019 15:39:37	JVM	TCP port TCP1 successfully open	657315	42
09/09/2019 15:39:37	JVM	(SocketConnection) Connector.open(socket://172.17.70.99:2000	657047	41
09/09/2019 15:39:37	JVM	Serial port COM1 successfully open	657011	40
09/09/2019 15:39:32	JVM	ModbusRtuDriver: Load Configuration Tag Complete	651591	39
09/09/2019 15:39:31	JVM	ModbusRtuDriver: license registered to cp-test-new-firmware	651307	38
09/09/2019 15:39:31	JVM	ModbusRtuDriver: license key is valid	651305	37
09/09/2019 15:39:28	JVM	Start Modbus RTU Driver v. 05.01	648066	36
09/09/2019 15:39:26	JVM	ModbusRtuDriver: Load Configuration Tag Complete	646226	35
09/09/2019 15:39:26	JVM	ModbusRtuDriver: license registered to cp-test-new-firmware	645839	34
09/09/2019 15:39:26	JVM	ModbusRtuDriver: license key is valid	645837	33
09/09/2019 15:39:22	JVM	EwonUtils V1.03	642022	32
09/09/2019 15:39:22	JVM	Start Modbus RTU Driver v. 05.01	642013	31
09/09/2019 15:39:22	JVM	CPeDrivers V5.01 RC	641603	30

eWON Name: eWON Firmware: 13.3s0 *(#792)* S/N: 1822-0177-21 Current time: 09/09/2019 17:18:07

2.5. License Key

The license file has the extension **“.eKey”** and must be uploaded in the **“/usr/license”** directory of the eWON. Every license file is used to enable the protocol to which it refers. Licenses are hardware bounded and cannot be transferred from one eWON to another.

There are two kind of license:

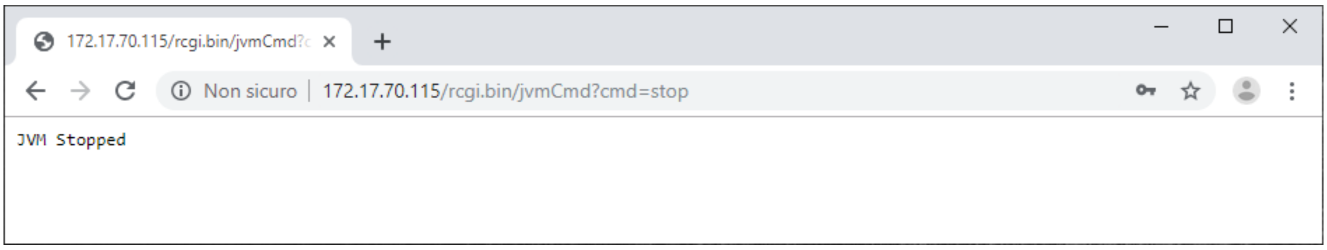
- demo license: all feature and drivers evaluable for 10 minute*
- full license: unlimited time license

*You can restart the eWON device to restart the evaluation period.

2.6. *Start and Stop commands*

If you need to Stop and Start the *Java Virtual Machine* and thus the **eWon drivers eXtender** open the following link address with your preferred web browser:

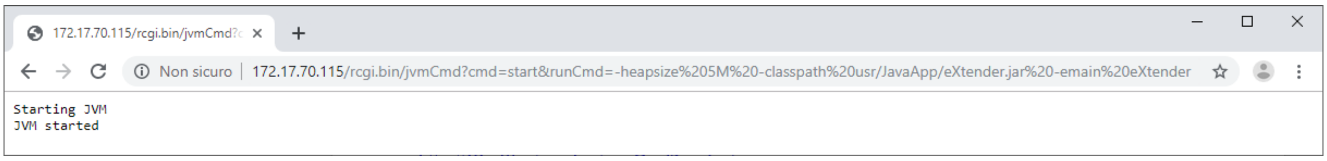
STOP:
http://*IP-eWon*/rcgi.bin/jvmCmd?cmd=stop



```
172.17.70.115/rcgi.bin/jvmCmd?cmd=stop
```

```
JVM Stopped
```

START:
http://*IP-eWon*/rcgi.bin/jvmCmd?cmd=start&runCmd=-heapsize 5M -classpath usr/JavaApp/eXtender.jar -emain eXtender



```
172.17.70.115/rcgi.bin/jvmCmd?cmd=start&runCmd=-heapsize%205M%20-classpath%20usr/JavaApp/eXtender.jar%20-emain%20eXtender
```

```
Starting JVM
JVM started
```

3. TAG's structure

TAGs used by **eWon drivers eXtender** can be divided into three types:

Type 1: Serial ports configuration TAG
Type 2: Devices configuration TAG
Type 3: Variables TAG

For types 1 & 2 the **Description** field is used to store configuration parameters delimited by two star character (**). Every TAG must be defined on the **MEM** I/O server.
For types 1 & 2 the **Type** field must be set to **Integer**.
For types 3 the **Type** field depends on the type of data.

3.1. *Serial port or TCP/IP configuration TAG (Type 1)*

This TAG contains the configuration parameters of serial port or Tcp/Ip socket.

Below are the steps for proper configuration:

1. Create a new TAG with a significant name, for example: **COM0**
2. Insert in the **Description** field the configuration parameter of the protocol:
   - Parameter 1: Name of the protocol. (ex. ***AuroraPort*** for Aurora Power-One)
   - Separator: "**"
   - Parameter 2: Type of port: ***Serial*** for serial port, ***Tcp*** for Tcp/Ip socket.
   - Parameter 3: A comma separated list of serial port parameters or Tcp/Ip port parameters.

Parameters for **Serial** port:
(parameters must be inserted with the following format: Pa1=Value1;Par2=Value2;…)

Parameter	**Default**	**Description**
comm	platform dependent	The logical name of the port.
baudrate	platform dependent	The speed of the port.
bitsperchar	8	The number bits per character(7 or 8).
stopbits	1	The number of stop bits per char(1 or 2)
parity	none	The parity can be odd, even, or none.
blocking	on	If on, wait for a full buffer when reading.
autocts	on	If on, wait for the CTS line to be on before writing. IMPLIES autorts=on (even if not set - or set differently) This option is not used in half duplex mode
autorts	on	If on, turn on the RTS line when the input buffer is not full. If off, the RTS line is always on. IMPLIES autocts=on (even if not set - or set differently) This option is not used in half duplex mode
halfduplex	off	If on, turn on the half duplex mode for the serial line This is only valid if the dip switch is set to RS485/422 mode on the device. In this configuration, the device will switch the emitter on only when transmiting data. The transmiter is off for the rest of time and the device is listening to the line. autocts and autorts are not used if halfduplex=on

reuseport	off	If on, the driver will try to reuse an open port. If it fails, the it will open the port. In any case, the settings passed are applied.
keepopened	off	If on, then when the connection object is closed, the driver will remain opened. WARNING: in any case the cleanup function will close the driver. Even if keepopened is specified. The user must cleanly close the connection if he wants to keep it opened.

Parameters for **Tcp/Ip** socket: (parameter must be inserted with the following format: *host:port*)

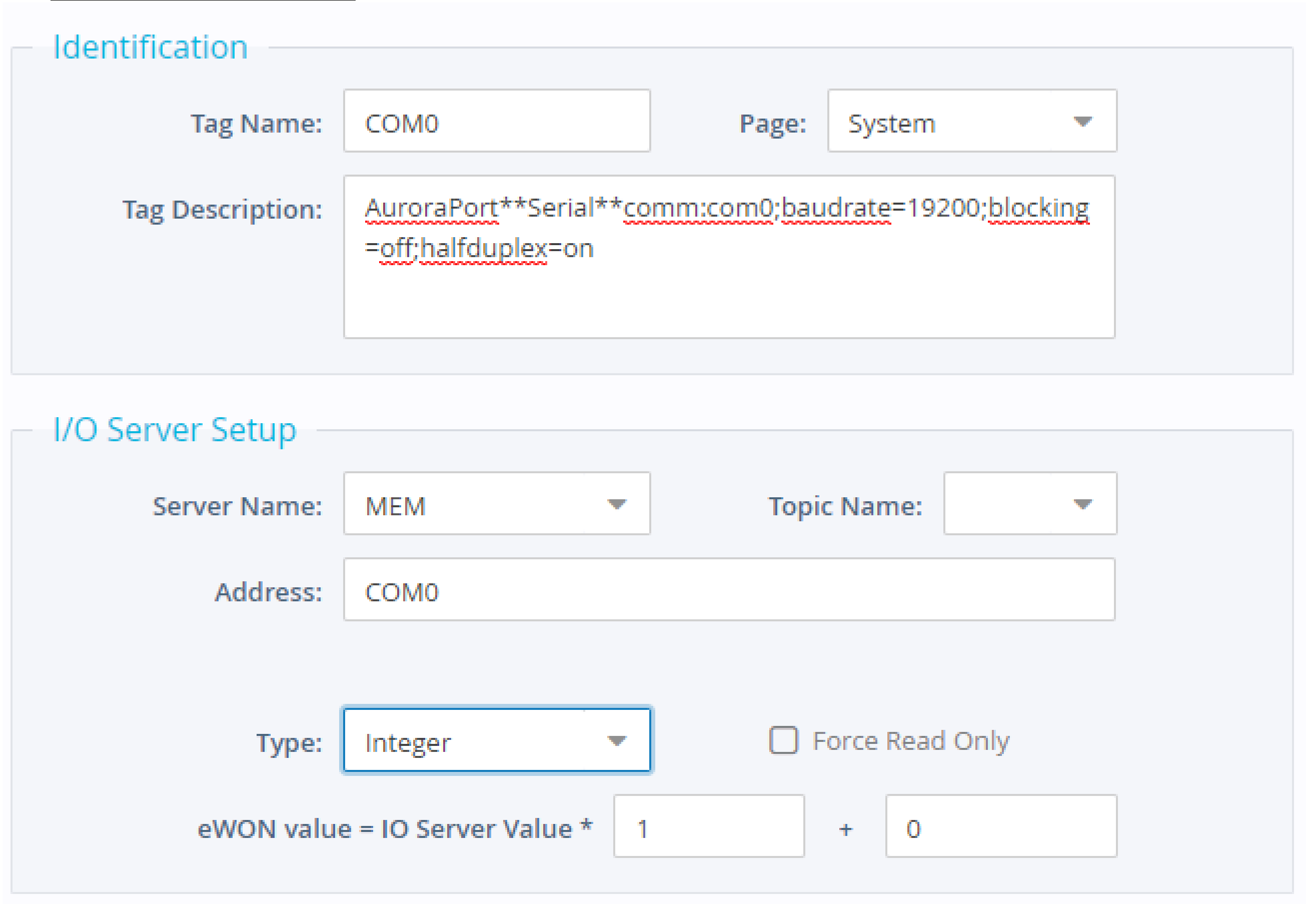
Parameter	**Default Value**	**Description**
host	–	host name or IP address
port	–	TCP/IP port nuumber

In the following example we use COM0 to communicate to Aurora Power-One inverter using the Aurora driver with baudrate 19200, 8 data bits, no parity, no blocking and half-duplex enabled.

eWon TAG Name: COM0

eWon TAG Description:
`AuroraPort**Serial**comm:com0;baudrate=19200;blocking=off;halfduplex=on`

Configuration example:



Identification

Tag Name: COM0
Page: System
Tag Description: AuroraPort**Serial**comm:com0;baudrate=19200;blocking=off;halfduplex=on

I/O Server Setup

Server Name: MEM
Topic Name:
Address: COM0
Type: Integer
☐ Force Read Only
eWON value = IO Server Value * 1 + 0

In the following example we use TcpIp to communicate to Aurora Power-One inverter using the Aurora driver connected with host 172.17.70.111 port 2001.

eWon TAG Name: `COM0`

eWon TAG Description: `AuroraPort**Tcp**172.17.70.111:2001`

3.2. Device configuration TAG (Type 2)

This kind of TAG contains the configuration parameters of the device.

Below are the steps for proper configuration:

3. Create a new TAG with a significant name, for example: **INV1**
4. Insert in the **Description** field the configuration parameter of the protocol:

Parameter1:	Name of the "Serial port configuration TAG"
Separator:	"**"
Parameter 2:	Device's node address
Separator:	"**"
Parameter 3:	Time-Out (default: 10000 ms)
Separator:	"**"
Parameter 4:	Device type (required for some protocols)
Separator:	"**"
Parameter 5:	Device's unit address (required for some protocols)

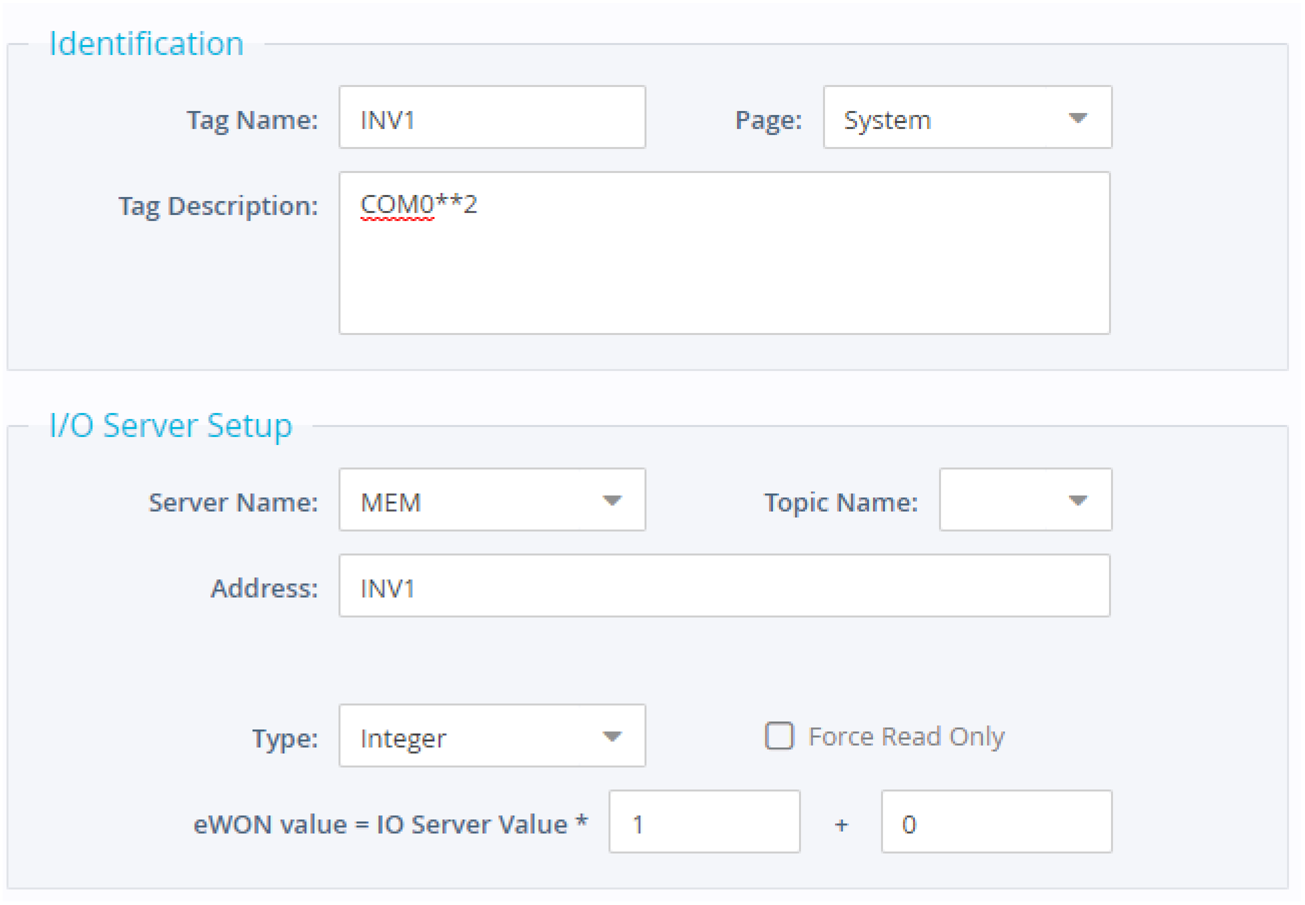
Note: Parameters may be driver-dependent.

In the following example we use COM0 to communicate to two Aurora Power-One inverters at node address 2 and 3:

eWon TAG Name: `INV1`
eWon TAG Description: `COM0**2`

eWon TAG Name: `INV2`
eWon TAG Description: `COM0**3`

Configuration example:



Tag Name: INV1
Page: System
Tag Description: COM0**2

Server Name: MEM
Topic Name:
Address: INV1
Type: Integer
☐ Force Read Only
eWON value = IO Server Value * 1 + 0

3.3. *Variable TAG (Type 3)*

This kind of TAG contains the values read from the devices. The name of the TAG implies the measure to read and is composed of two references:

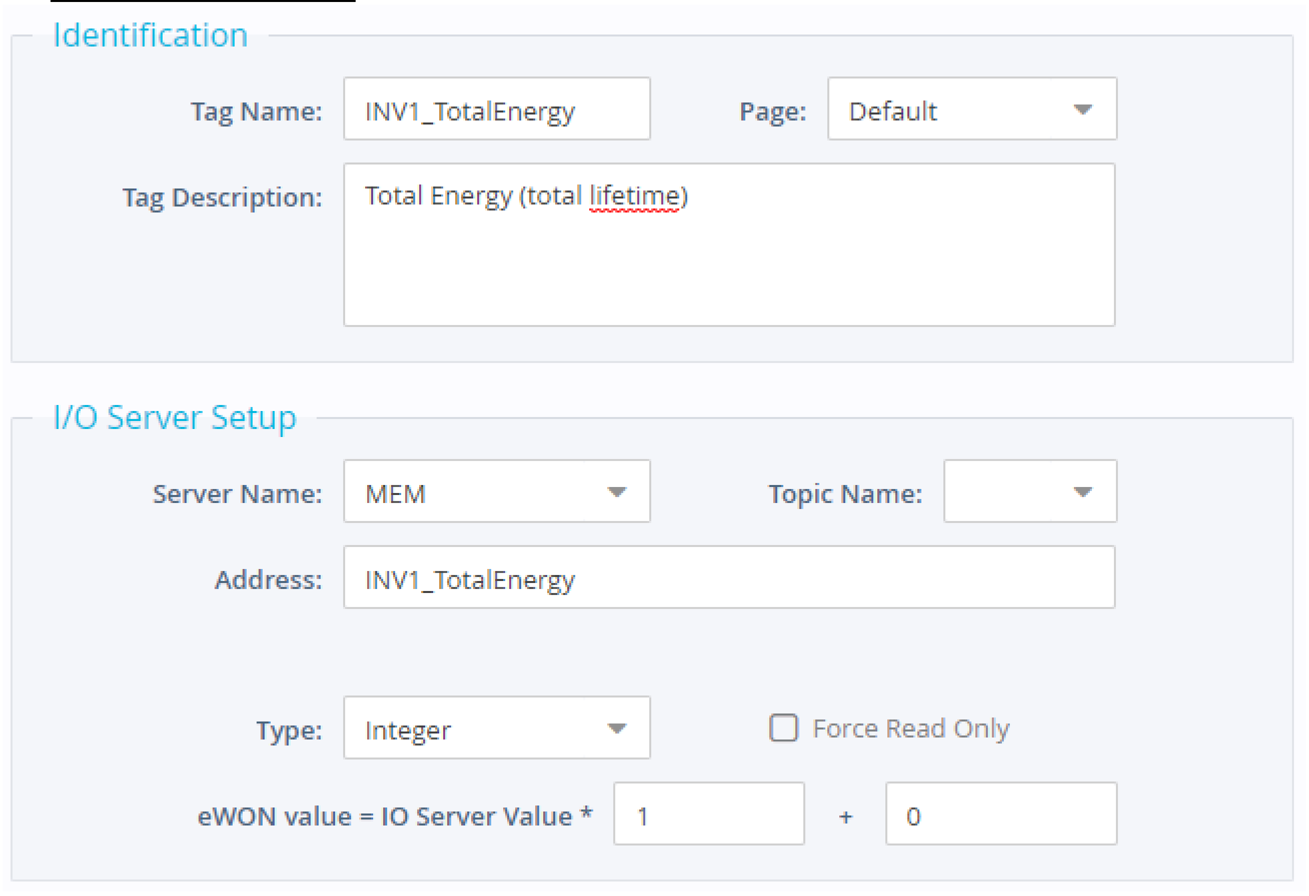
- Prefix: Name of the "Device configuration TAG"
- Separator: "_"
- Suffix: Name of the measure to read.

In the following example we define a DWord TAG used to read the TotalEnergy measure from inverter defined in the device configuration TAG INV1:

eWon TAG Name: `INV1_TotalEnergy`

Nota: The suffix is driver-dependent and can be customized.

<u>Configuration example:</u>



Identification

Tag Name:	INV1_TotalEnergy	Page:	Default
Tag Description:	Total Energy (total lifetime)		

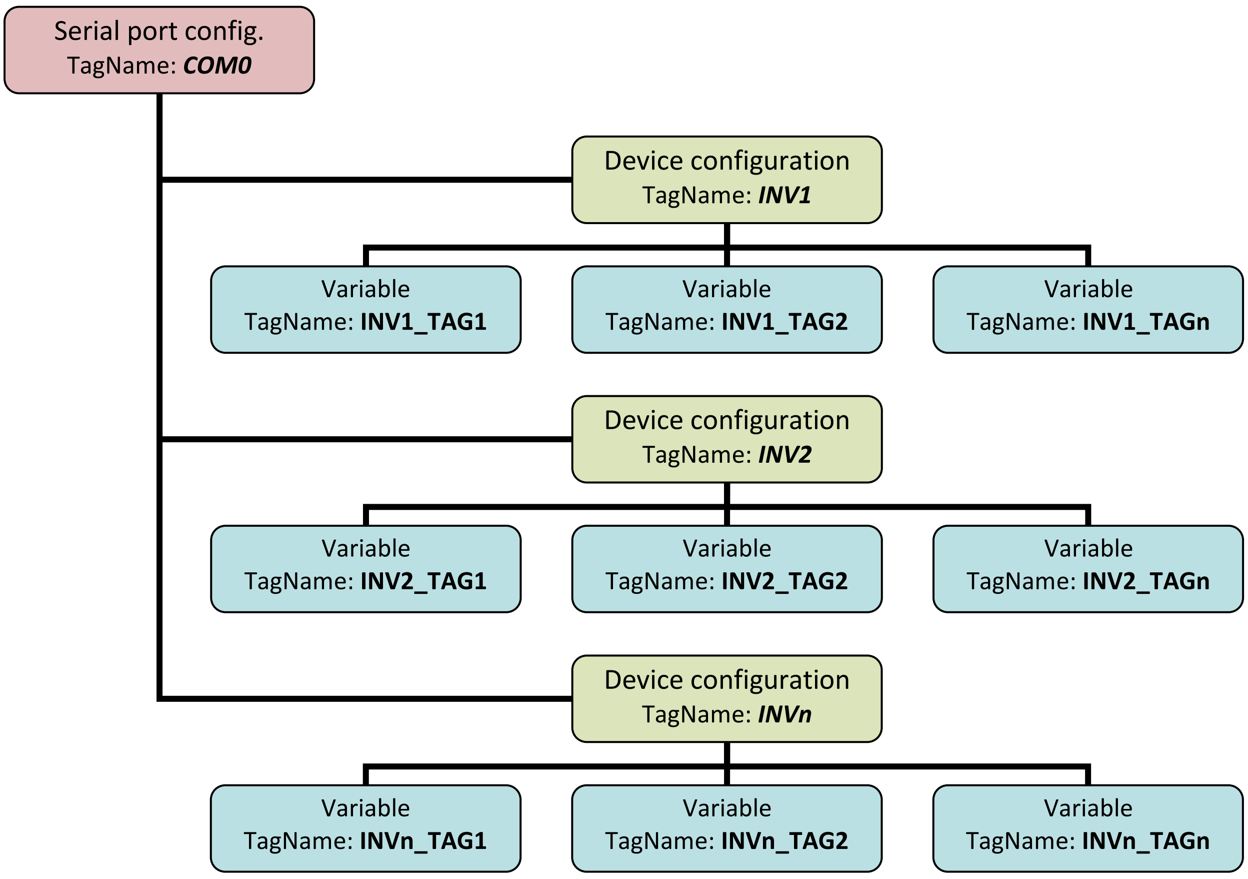
I/O Server Setup

Server Name:	MEM	Topic Name:	
Address:	INV1_TotalEnergy		
Type:	Integer	☐ Force Read Only	

eWON value = IO Server Value * 1 + 0

3.4. TAG hierarchy

Each TAG used by **eWon drivers eXtender** are related to each other according to a tree structure. The root node is a “Serial port configuration TAG”. A “Serial port configuration TAG” can refer to one or mode “Device configuration TAG”. A “Device configuration TAG” can refer to one or mode “Variable TAG”.



eWon drivers eXtender detect as “Serial port configuration TAG” each TAG with a proper configured Description field and with a valid driver name as first parameter (es: AuroraSerialPort).

eWon drivers eXtender detect as “Device configuration TAG” each TAG with a proper configured Description field and with a valid “Serial port configuration TAG” name as first parameter (es: COM0).

eWon drivers eXtender detect as “Variable TAG” each TAG that starts with the name of one “Device configuration TAG” (es: INV1_TotalEnergy).

3.5. State, Commands and Debug

eWon drivers eXtender allow the user to send commands to the driver, read the state of the devices and read some diagnostic information.

3.5.1. Serial port configuration TAG

The value of a “Serial port configuration TAG” reports the status of the Serial port:

Status:

0	Port free
1	Port in use by **eWon drivers eXtender**

There are two diagnostic TAG for each “Serial port configuration TAG” automatically created by **eWon drivers eXtender** to analyze the number of packets sent and received.

If the “Serial port configuration TAG” name is COM0, the two diagnostic TAG will be:

- **COM0_SendPacket**: number of packets sent by **eWon drivers eXtender**
- **COM0_RecPacket**: number of valid packets received by **eWon drivers eXtender**

3.5.2. Device configuration TAG

The value of a “Device configuration TAG” reports the status of the Device:

Status:

0	Device offline
1	Device online

3.5.3. Command TAG

CPeDriverStatus: From this TAG you can read the driver’s status or write a command to modify it.

Status:

0	Driver is loading
1	Driver is running
2	Driver is stopped (the serial port is free to use)

Command:

-1	Reload the configuration and start the driver
0	Start the driver without reloading the configuration
2	Stop the driver

CPdebug: This TAG defines the log level of **eWon drivers eXtender**’s debug messages. Log messages are visible in the eWON’s ***Real Time Log*** web page.

Status:

0	Debug log disabled
1	Debug log enabled
2	Debug log enabled and serial communication log enabled

<u>**Command:**</u>

0	Disable debug log
1	Enable debug log
2	Enable debug log and serial communication log

CPeWritingEnable: This command TAG is managed only for **eWon drivers eXtender** version 4.0 and above. From this TAG you can read the global writing flag status or write a command to modify it.

<u>**Status:**</u>

0	Global writing flag is disabled (no driver will write tag values to field, even if capable)
1	Global writing flag is enabled (all capable drivers will write tag values to field)

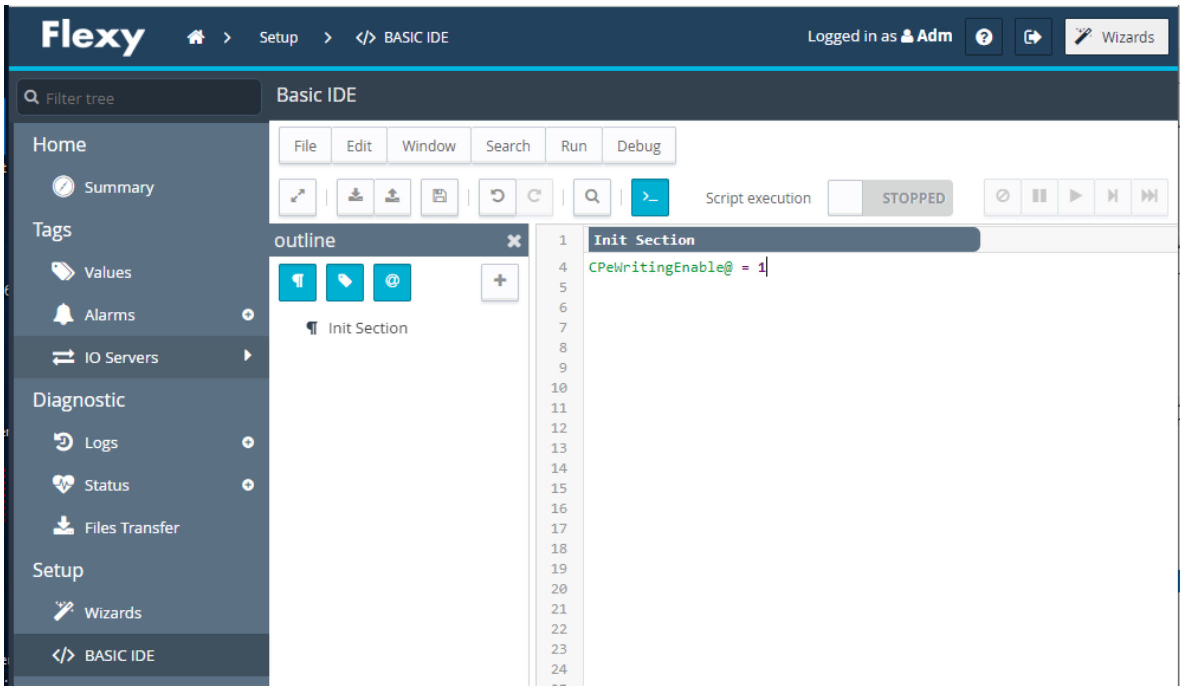
<u>**Command:**</u>

0	Disable global writing flag (no driver will write tag values to field, even if capable)
1	Enable global writing flag (all capable drivers will write tag values to field)

If you want to enable the global writing flag at eWON startup, you can add the following BASIC instruction to the Init Section in the Script Setup:

```
CPeWritingEnable@ = 1
```

Init Section example:



```
Init Section
CPeWritingEnable@ = 1
```

IMPORTANT

If

d) you modify and save any tag's parameter (description, address, etc.) while **eWon drivers eXtender** is running and
e) you are using a driver that is able to write the tag value to field and
f) the global writing flag is enabled

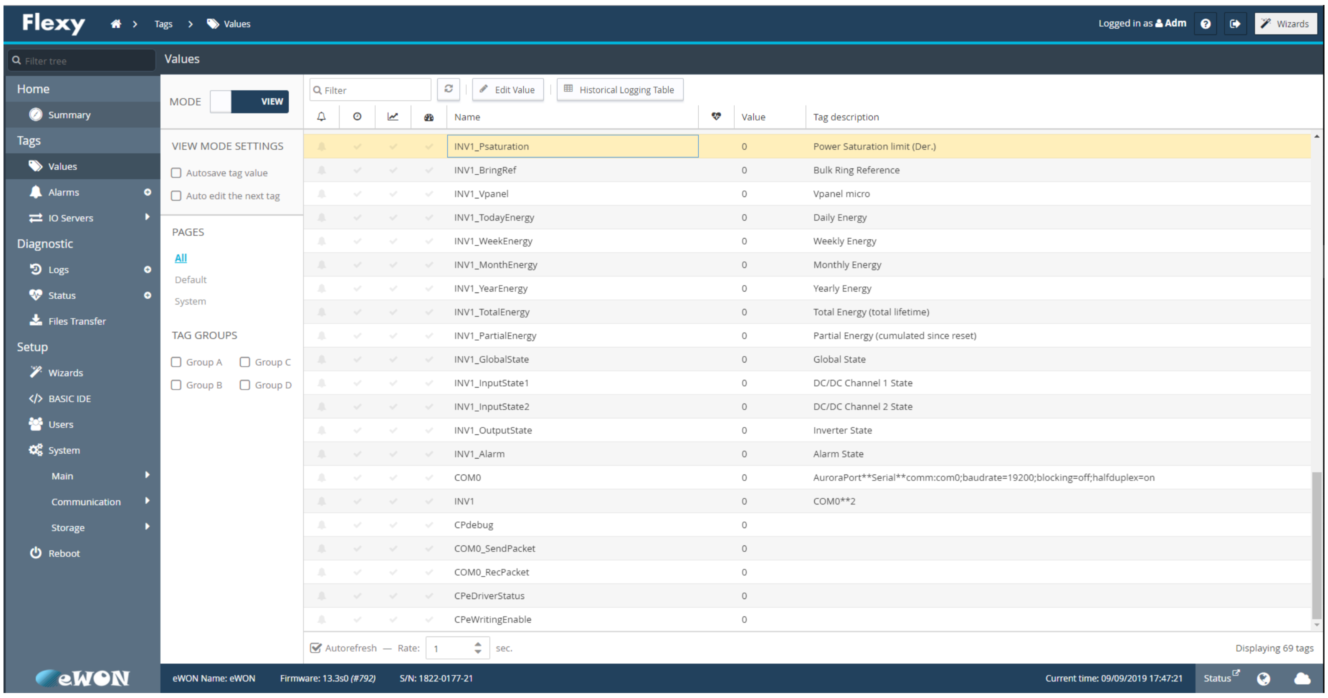
then this will cause the driver to write a "zero" value to the tag's value and so to the field.

In order to avoid this behavior you have two options:

3. Disable the global writing flag during the tag setup phase. Please refer to chapter 3.5.3 Command TAG.
4. Stop **eWon drivers eXtender** during the tag setup phase. Please refer to chapters 2.6 Start and Stop commands and 3.5.3 Command TAG.

Drivers capable of writing tag values to field are available for **eWon drivers eXtender** version 4.0 and above. Please refer to the specific driver documentation to know if it is capable of writing tag values to field.

Command tags example:



Name	Value	Tag description
INV1_Psaturation	0	Power Saturation limit (Der.)
INV1_BringRef	0	Bulk Ring Reference
INV1_Vpanel	0	Vpanel micro
INV1_TodayEnergy	0	Daily Energy
INV1_WeekEnergy	0	Weekly Energy
INV1_MonthEnergy	0	Monthly Energy
INV1_YearEnergy	0	Yearly Energy
INV1_TotalEnergy	0	Total Energy (total lifetime)
INV1_PartialEnergy	0	Partial Energy (cumulated since reset)
INV1_GlobalState	0	Global State
INV1_InputState1	0	DC/DC Channel 1 State
INV1_InputState2	0	DC/DC Channel 2 State
INV1_OutputState	0	Inverter State
INV1_Alarm	0	Alarm State
COM0	0	AuroraPort**Serial**comm:com0;baudrate=19200;blocking=off;halfduplex=on
INV1	0	COM0**2
CPdebug	0	
COM0_SendPacket	0	
COM0_RecPacket	0	
CPeDriverStatus	0	
CPeWritingEnable	0	

Autorefresh — Rate: 1 sec.   Displaying 69 tags

eWON Name: eWON   Firmware: 13.3s0 (#792)   S/N: 1822-0177-21   Current time: 09/09/2019 17:47:21   Status

4. TAG Customization

eWon drivers eXtender uses a CVS configuration file to know what data put to which tag. This file is contained in the eWON's directory: "***/usr/Config***". You can modify the file to use custom "Variable TAG" suffix.

Example:
If we need to change TAG's name from **INV1_TotalEnergy** to **INV1_Energy**:

0 Open the driver's configuration file **"/usr/Config/PowerOne.csv"**
1 Find on the first column the suffix **TotalEnegy** and rename it to **Energy**
2 Save the file
3 Rename, using the eWON web interface, the name of the **INV1_TotalEnergy** TAG to **INV1_Energy**
4 Reload the configuration or reboot the eWON

5. Using the MODBUS gateway

All Tags read with eWon Drivers eXtender are compatible with eWON's native MODBUS gateway.