

eWon drivers - Driver Details

Modbus RTU

Summary

1.	Modbus RTU Driver Details	2
1.1.	Port configuration.....	2
1.2.	Device configuration	2
1.3.	TAG list (example).....	3
1.4.	How to create var_lst.csv	4
1.5.	Tested Ethernet/serial bus converters	4

1. Modbus RTU Driver Details

The Modbus RTU driver implements the classic Modbus RTU serial protocol, so it has a similar functionality to the native eWON implementation on serial buses (e.g. RS-232 or RS-485). However, exploiting the flexibility of the **eWon drivers eXtenders**, this driver can be used in conjunction with cheap “dummy” Ethernet/serial bus converters, which are not capable of Modbus/Modnet packet translation.

The Modbus RTU driver can open a TCP connection to the gateway and send/receive serial Modbus RTU packets over-TCP. The gateway will manage TCP-wrapped packets communicating to the eWON and clean Modbus RTU packets communicating to the final Modbus RTU device.

The Modbus RTU driver supports:

- Coil Status flags, Input Status flags, Input Registers, Holding Registers;
- Independent block size configuration for each device;
- Exception responses.

1.1. Port configuration

The Modbus RTU driver, like any other **eWon drivers eXtender**, can use a TCP port or a serial port.

- As described above, configuring a TCP port represents a good opportunity to use a cheap “dummy” Ethernet/serial bus converter (which is not capable of Modbus/Modnet packet translation).
- Configuring a serial port works similarly to the native eWON Modbus RTU protocol on serial buses (RS-232 or RS-485).

TCP port configuration tag description (example):

`ModbusRtuPort**Tcp**192.168.123:2000`

Serial port configuration tag description (example):

`ModbusRtuPort**Serial**comm:com:1;baudrate=9600;bitsperchar=8;stopbits=1;parity=none;blocking=off;autocts=off;autorts=off`

1.2. Device configuration

For each device a tag named *deviceNameN* (e.g. PLC1) must be defined, where *N* is the number of device starting from 1. Description of *deviceNameN* tag is used by the driver to read some parameters, as described below:

*commname**address**timeoutMs**deviceType**unitNumber**blockSize*

Please note that parameters *deviceType* and *unitNumber* must be declared for compatibility reason, in order to specify the last parameter *blockSize*. The default value for parameter *blockSize* is 30, the max value is 200.

Examples:

`COM0**3**2000`

<code>commname = COM0 ;</code>	eWon port
<code>address = 3 ;</code>	Modbus RTU unit number
<code>timeoutMS = 2000 ;</code>	2s timeout for reading

`COM1**0**3000**0**0**10`

<code>commname = COM0 ;</code>	eWon port
<code>address = 0 ;</code>	Modbus unit number
<code>timeoutMS = 3000 ;</code>	3s timeout for reading
<code>deviceType = 0 ;</code>	(not used)
<code>unitNumber = 0 ;</code>	(not used)
<code>blockSize = 10 ;</code>	Modbus RTU block size

1.3. TAG list (example)

The following is an example of partial CSV configuration file:

<i>CmdName</i>	<i>Address</i>	<i>Description</i>
CS0001	CS0001	CS0001
CS0002	CS0002	CS0002
IS0003	IS0003	IS0003
IS0004	IS0004	IS0004
HR0005	HR0005	HR0005
HR0006	HR0006	HR0006
IR0007	IR0007	IR0007
IR0008	IR0008	IR0008

The first two fields (*CmdName*, *Address*) are used by the **eWon drivers eXtender**:

- You can edit the *CmdName* field, then use the new customized prefixes for the eWON tag names.
- You can edit the *Address* field, but note that:
 - Only CS, IS, HR and IR prefixes are supported.

<i>Modbus Area</i>	<i>CmdName prefix</i>
Coil Status	CS
Input Status	IS
Holding Registers	HR
Input Registers	IR

- The address within the area must be an integer between 1 and 9999 (leading zeros are ignored).
- You can edit the *Description* field as you wish.

Please note that defining eWON tags you must respect the following rules:

- Given a Modbus device, the postfix of the eWON tag name must be equal to the *CmdName* field of the CSV configuration file.

To ensure that the **eWon drivers eXtender** reads the values correctly, define the tags in the following way:
deviceNameN_tagPostfix (e.g.: PLC1_ HR0005)

1.4. How to create var_lst.csv

If you want use a var_lst.csv file to add the tag to the eWON, create the file as follows:

Name	Description	ServerName	TopicName	Address	Type	PageId
COM0	ModbusRtuPort**Tcp**192.168.123:2000	MEM		COM0	2	2
PLC1	COM0**1	MEM		PLC1	2	2
PLC1_CS0001	PLC1 - CS0001	MEM		PLC1_CS0001	0	1
PLC1_CS0002	PLC1 - CS0002	MEM		PLC1_CS0002	0	1
PLC1_IS0003	PLC1 - IS0003	MEM		PLC1_IS0003	0	1
PLC1_IS0004	PLC1 - IS0004	MEM		PLC1_IS0004	0	1
PLC1_HR0005	PLC1 - HR0005	MEM		PLC1_HR0005	0	1
PLC1_HR0006	PLC1 - HR0006	MEM		PLC1_HR0006	0	1
PLC1_IR0007	PLC1 - IR0007	MEM		PLC1_IR0007	0	1
PLC1_IR0008	PLC1 - IR0008	MEM		PLC1_IR0008	0	1

- The field *Name* in *var_lst.csv* must be composed of *deviceNameN* (e.g. PLC1) and field *CmdName* (e.g. HR0005).
- The field *Address* in *var_lst.csv* must be equal to field *Name*.
- The field *ServerName* in *var_lst.csv* must be *MEM*.
- The field *Description* in *var_lst.csv*, except for the configuration tags (e.g. COM0, PLC1), has no constraints. It is recommended to fill it with the tag description and other useful informations.
- The field *Type* in *var_lst.csv* is the data type. Use the following mapping to insert a type (it may depend on the eWON firmware version):

Data Format	eWON Format
0	Boolean
1	Float
2	Integer
3	Dword

1.5. Tested Ethernet/serial bus converters

- Anybus Serial Server
- Moxa Device Server NPort Series