

# MODBUS integration for ONE Series

Instructions and examples  
for a remote control of ONE  
devices using MODBUS protocol

Review: 3



## 1 INTRODUCTION

This handbook details how to **manage ONE devices using MODBUS protocol**. The following sections explain the requirements that must be adapted in order to configure the units.

Chapter 3 APPLICATION EXAMPLE: CONNECTION AND SETUP OF USR-N510 CONVERTER shows how to control an ONE device remotely using an external MODBUS-TCP to MODBUS-RTU converter. In this case, we will use a PC as MODBUS protocol's master.

### 1.1 MODBUS

MODBUS is a communication protocol between devices based on queries from the main device, defined as “Master”; to other devices that will receive those queries. Those receivers will be called “Slaves”.

On ANNEX I: Definitions these concepts will be explained in detail.

### 1.2 MODBUS integration for ONE

ONE integrates MODBUS protocol in its version MODBUS-RTU, using the serial bus RS-485 as a physical channel for the communications.

ONE 500 devices have a **slave** role in the bus, executing the master's requirements. These functionalities are available at ONE devices with firmware version v01.00.03.01 or newer.

## 2 MODBUS INTERFACE CONFIGURATION IN ONE

### 2.1 ONE local setup

The integration of MODBUS protocol with ONE devices is made through a RS-485 port, that is physically located on the main board as shown on *Picture 1: RS-485 port location on ONE Series*.

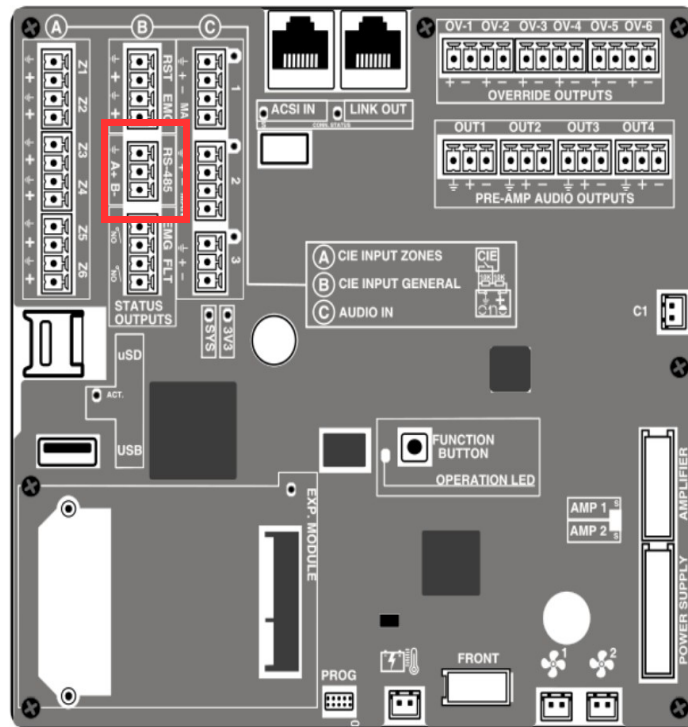
That RS-485 port has its own configuration parameters, that can be reached from the frontal screen interface, using the “Configuration” menu path:

- Configuration → CIE → RS-485 Conf.

The following parameters are the typical ones for a serial bus connection:

- Baudrate: Transmission speed of the serial bus, measured on baud. By default, its value is 115200 bauds.
- Parity: Serial communications parity. By default, given value is “Even”.
- Stop bits: Bits for the serial communication stop. By default, its value is 1.

## MODBUS integration for ONE Series



Picture 1: RS-485 port location on ONE Series

### 2.2 Access level and connection security

ONE devices have different user level accesses for each configuration level.

User access level varies depending on the used interface. That means that **the access level of the frontal screen of ONE will be different to the RS-485 interface.**

To change the access level for RS-485 interface, the CIE485\_MODBUS\_ACCESS\_LEVEL variable, which has been explained in the previous section, must be used.

Communications between the master and the slave use a defined address. By default, ONE 500 has the MODBUS address 10. This value could be modified by editing the parameter CIE485\_MODBUS\_ADDRESS, as explained in the previous section.

### 3 APPLICATION EXAMPLE: CONNECTION AND SETUP OF USR-N510 CONVERTER

To manage ONE devices through the RS-485 port via MODBUS it is required to use an TCP/IP to RS-485 converter between ONE and the controller device. The device must work using the Modbus TCP option in the device webserver:

Socket A	Parameters
Work Mode:	TCP Server <input type="button" value="v"/> ModbusTCP <input type="button" value="v"/>
TCP Server MAX Sockets:	8 <input type="button" value="v"/> Up to MAX KICK <input type="button" value="v"/>
Local/Remote Port Number:	502 <input type="text"/> 8899 <input type="text"/> (1~65535)
PRINT:	<input type="checkbox"/>
ModbusTCP Poll:	<input checked="" type="checkbox"/> Poll Timeout : 1000 <input type="text"/> (200~9999) ms
Enable Net Heartbeat Packet:	<input type="checkbox"/>
Registry Type:	None <input type="button" value="v"/> Location <input type="button" value="Connect With"/> <input type="button" value="v"/>

In this section the **USR-N510 bidirectional converter from TCP to RS-485** and its configuration system will be used as example to explain how to set the system. The converter will be used to emulate a communication with ONE through an external software called *Modbus Poll*. Nevertheless, general information about that setup can also be used on any other converter device or similar systems.

A **TCP-RS-485 bidirectional converter** will be used to **control a ONE device via MODBUS protocol** through the RS-485 port.

This system can be used to **modify and manage the ONE system variables parameters** using an external application. The application will need a system configuration similar to the one shown at *Picture 2: Communication performance between MODBUS RTU and ONE*.

## MODBUS integration for ONE Series

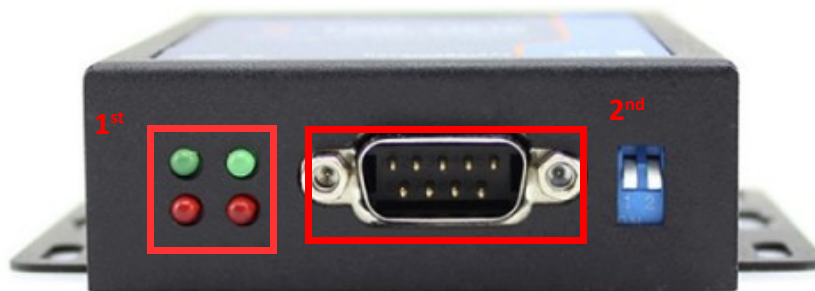


Picture 2: Communication performance between MODBUS RTU and ONE

### 3.1 USR-N510 connection

Using the USR-N510 converter module as example of external device, some steps should be followed to a proper installation.

The device will get started after the user connects its power supply, and the “PWR” red LED will light on. This LED is located on the connector side of the equipment. At *Picture 3: Status LEDs disposition* the LED cluster is marked as 1<sup>st</sup>. The LED that indicates the power connection is the lower left LED, located at the red marked square. The other LEDs refer to the current work status of the device performance.



Picture 3: Status LEDs disposition

The USR-N510 should be connected to the network to manage ONE devices. The connection should be done via its Ethernet port, located at the upper side of the device.

Connection to ONE devices must be done using the RS-485 port. To use the RS-485 port of the USR-N510, we must connect the RS-485 to RS-232 adapter as shown on *Picture 3: Status LEDs disposition*, (port marked as 2<sup>nd</sup>).

After connecting the adapter, the wiring from RS-485 must be done to ONE poles T+(A), T-(B) and GND. See Table 1: RS-485 equivalences between ONE and USR-N510 for further information:

## MODBUS integration for ONE Series

USR-N510	ONE
T+(A)	A+
T+(B)	B-
GND	GND

Table 1: RS-485 equivalences between ONE and USR-N510

To make the connection to RS-485 on ONE Series, it is recommended to use a 3 ways (3,81 mm) euroblock. Please verify the pin connection before finishing the installation.

**NOTE:** At **ONE extended systems** (multiple ONE devices), it is important to know which **device has the direct connection to the Modbus master device**, because this ONE will execute the commands. It is important to notice that the master ONE device has particular features that affect the extended system.

### 3.2 USR-N510 local setup

In order to configure the entire system, serial bus parameters must be the same as the ones configured in ONE.

To enable the RS-485 port on USR-N510, the switches on the back side of the device must be turned up, as shown on *Picture 4: Switch disposition to RS-485 performance*.



Picture 4: Switch disposition to RS-485 performance

The device has its own dedicated software ([USR-TCP232-M4K3](#)). It is also possible to make all settings from the application hosted on the device's IP address.

To access to it, the device IP address once connected to a network, by default it is ([192.168.0.7](#)). Default user name and password can be found at *Table 2: Default credentials to USR-N510 configuration software*.

User name:	admin
Password:	admin

Table 2: Default credentials to USR-N510 configuration software

## MODBUS integration for ONE Series

**NOTE:** Network access to USR-510 device is mandatory. That means it is necessary to reach the devices through a local network.

Once the setup application is started, *Current Status* window shows the general configuration parameters. Only *PORT 1* section needs to be edited, and so, it is possible to leave the rest of the dependencies as configured by default.

The screenshot displays the 'parameter' configuration window for the USR-N510 device. The 'PORT 1' section is highlighted with a red box. The configuration parameters are as follows:

Parameter	Value	Unit/Range
Baud Rate	115200	bps(600~230400)bps
Data Size	8	bit
Parity	Even	
Stop Bits	1	bit
Flow Control	None	
UART Packet Time	0	(0~255)ms
UART Packet Length	0	(0~1460)chars
Sync Baudrate(RF2217 Similar)	<input checked="" type="checkbox"/>	
Enable Uart Heartbeat Packet	<input type="checkbox"/>	

Below the 'PORT 1' section, the 'Socket A Parameters' are visible:

Parameter	Value	Unit/Range
Work Mode	TCP Server	None
Local/Remote Port Number	8899	8899 (1~65535)
PRINT	<input type="checkbox"/>	
ModbusTCP Poll	<input type="checkbox"/>	Poll Timeout : 200 (200~9999) ms
Enable Net Heartbeat Packet	<input type="checkbox"/>	
Registry Type	None	Location Connect With

The 'Socket B Parameters' section shows:

Parameter	Value
Work Mode	NONE

At the bottom of the window, there are 'Save' and 'Cancel' buttons.

Picture 5: USR-N510 setup (PORT 1 section)

By default, the main window view is *Current Status*. On this guide only *PORT 1* will be configured. Required configuration of *PORT 1* is shown at *Picture 5: USR-N510 setup (PORT 1 section)*. Values must coincide with linked ONE device's values. In case of any modification, that modification should be done on both devices.

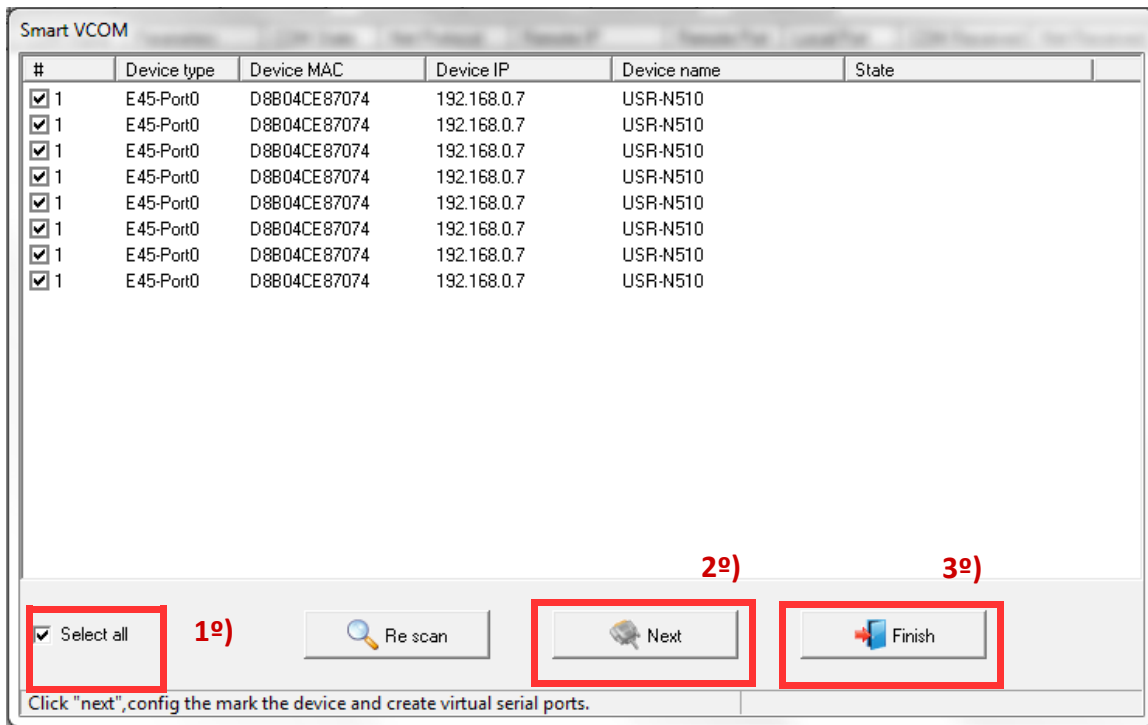
### 3.3 COM ports setup from PC

After the the device is configured, a COM port must be opened between the USR-N510 and the server. It is recommended by the manufacturer to use the dedicated software [USR-VCOM](#).

When the USR-N510 device is connected to the network, the user will have to select the Smart VCOM label to search the device available COM ports. By selecting those devices

(Select All option), the virtual ports will be created.

Once the ports are opened, the software shows these ports on a main window, either if they are network connected or not (*Connected* or *Connection Failed*) at the *Next State* labelled column. Opened COM ports will be listed at *Device Name* column. *Picture 6: Smart V-COM section* shows these software windows, with the steps overwritten in red.



Picture 6: Smart V-COM section

### 3.4 Communication with ONE via Modbus Poll

Once the COM port communication is created, the transmission between the *master* device (that can be wherever on the network) and the *slave* (ONE, in this case) is set. To use that communication it is possible to work with an external application. In this Handbook, [Modbus Poll](#)<sup>1</sup> is used to perform a simulation of a master controller.

On the software main window, select *Connection* → *Connect*.

In the *Serial Port* connection type the requested parameters must be set using the same values as the ones used on the ONE device. A COM port must be opened and configured according to 3.3 COM ports setup from PC section. This port must be available, no other software at the server should be using the same port.

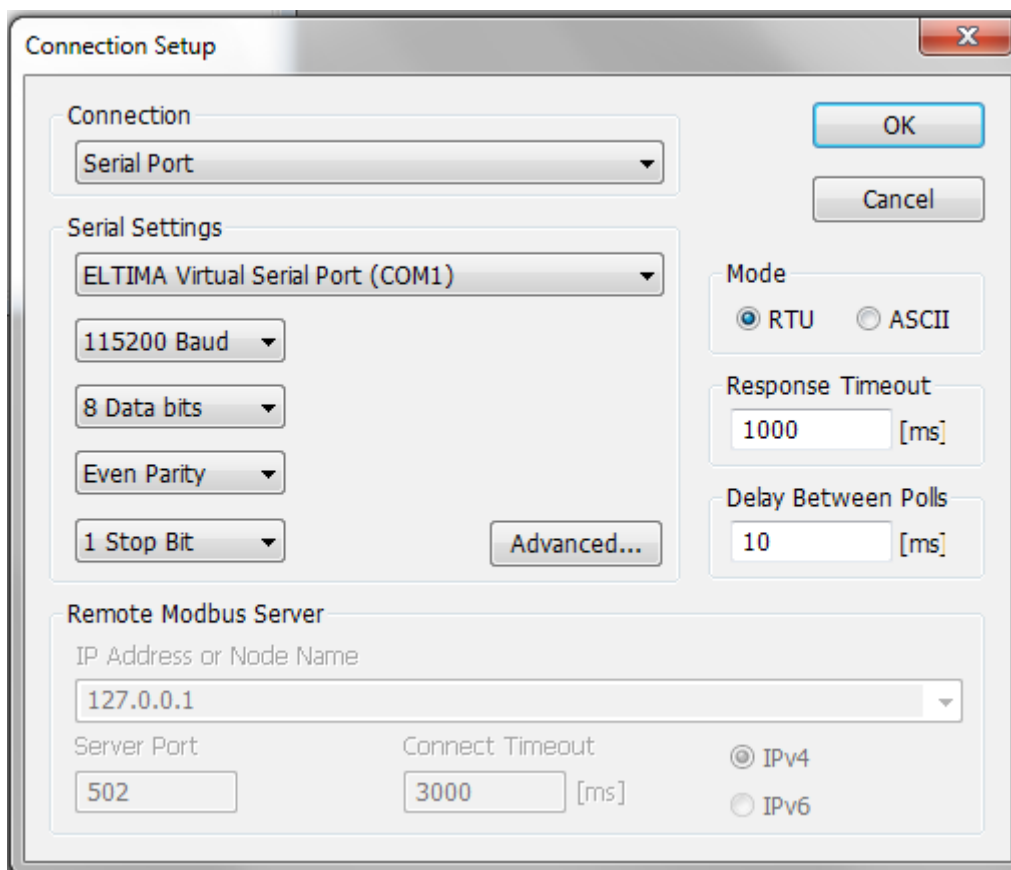
It is mandatory to test that the *Net State* is *Connected* on USR-VCOM software before using it.

*Picture 7: Modbus Poll network configuration* shows the parameters, according to ONE default values. Also, the configuration must always be the same as the previous menu configuration plus selection of RTU performance (Mode → RTU).

1 **NOTE:** For extra information about Modbus Poll software, kindly consult the [manufacturer user manual](#).



## MODBUS integration for ONE Series



Picture 7: Modbus Poll network configuration

After these configurations are made, it is possible to manage the ONE Series parameters from the master controller software. It can be done just by selecting any cell at 00000 column at the *poll configuration* table.

Once a cell is selected, a new window gets opened. In this window, the command that would be sent to the ONE device can be written. From here it can be sent to other ONE devices connected to the network. The required values are:

- *Slave ID*: ONE device identifier according to Modbus (value by default: 10). See 2.2 Access level and connection security .
- *Address*: Variable or register address to be modified. By default, it requires the decimal address value.<sup>2</sup>
- *Value*: Desired new value for that variable.

Option 16: *Write multiple registers* should be selected on “*Use function*” section.

Those inputs are marked at *Picture 8: TEST function for ONE Series from Modbus Poll*.

---

<sup>2</sup> **NOTE:** Under the exceptions discussed onto Error: No se encuentra la fuente de referencia Error: No se encuentra la fuente de referencia.

## MODBUS integration for ONE Series

The screenshot shows a 'Write Single Register' dialog box with the following fields and options:

- Slave ID: 10
- Address: 4098
- Value: 1
- Result: N/A
- Close dialog on "Response ok"
- Use Function:
  - 06: Write single register
  - 16: Write multiple registers
- Request:
  - RTU: 0A 10 10 02 00 01 02 00 01 04 83
  - ASCII: 3A 30 41 31 30 31 30 30 32 30 30 30 31 30 32 30 30 30 31 44 30 0D 0A

Picture 8: TEST function for ONE Series from Modbus Poll

**EXAMPLE:** Picture 8: TEST function for ONE Series from Modbus Poll shows an example of writing of the variable 0x1003 (4099 on decimal, 4098 used in this software), that represents the TEST function on ONE devices interface

Clicking on *Send* automatically activates this command to the ONE device.

In the *Request* → *RTU* section on this window, the Modbus frame that modifies the device parameters is generated. In order to test this, the application function *Web to serial* can be used. It is located at the IP address of the connected USR-N510 converter, (by default [192.168.0.7](http://192.168.0.7)).

Picture 9: TEST function activation command on ONE device sent via USR-N510 shows the copied Modbus frame, on the second box. By clicking on button *send hex data* the frame will be transmitted and the ONE Series device will execute it. The frame line must have no spaces.

# MODBUS integration for ONE Series

firmware revision: v3031 中文

**USR IOT**  
-IOT Experts- *Be Honest, Do Best!*

Current Status  
Local IP Config  
PORT1  
**Web to Serial**  
Misc Config  
Reboot

parameter

Websocket connection: 0

Receive HEX: 0xa 0x10 0x10 0x2 0x0 0x1 0xa5 0xb2

0A10100200010200010483

send ascii data **send hex data** clear

help

- **web to serial**  
this page use websocket to transmit data between webpage and uart

Copyright © 2009 - 2015 · JiNan Usr IOT Technology Limited website: [www.usriot.com](http://www.usriot.com)

Picture 9: TEST function activation command on ONE device sent via USR-N510

## ANNEX I: Definitions

- **Master:** Controller device that starts a communication request.
- **Slave:** Receiver device, passive, that answers the master queries.
- **RS-485:** Communications standard that implements a multipoint serial bus based on differential transmission.
- **Modbus:** Communication protocol based on master-slave relationships.
- **Modbus RTU:** Modbus transfer mode designed to be used by devices under the electrical standards RS-232, RS-485 and RS-422.
- **Modbus TCP:** Modbus transfer mode designed to work at TCP/IP networks.
- **Modbus TCP to Modbus RTU converter:** Device used to transform from Modbus RTU to Modbus TCP protocol.

On this guide, an **USR-N510 converter** model is used as example, connected to the local network.

- **Modbus master simulator:** Software used to simulate the master controller. It is an application that manages the ONE parameter values. This will be in charge of developing the RTU data packages to ONE.

## ANNEX II: Advanced edition of ONE variables

Some advanced configurations are based on ONE devices internal variables. Each of these variables has a unique identifier, labelled as ID. With that identifier, users may check the variable status and also modify its value (in case this user has authorisation).

To access to an advanced configuration variable, the OK and BCK buttons on the front interface must be pressed simultaneously for 2 seconds. After that, a special screen will be shown. In this screen, the user will enter the ID. If necessary, the user will be asked to register with the required access level by the variable. If he does not have enough access level, it will be denied.

## ANNEX III: Modbus Commands Overview

Modbus command frames are determined by the Modbus protocol and its specification. These frames are usually called ADU (Application Data Unit) inside Modbus context. For a complete information it is recommended to visit their official website <https://modbus.org>.

For more information on the MODBUS integration variables for ONE please consult your LDA sales partner.

However, below there are a some key points in order to understand how Modbus works and how a valid Modbus command can be built:

- **Data encoding:** Modbus uses a “big-endian” representation for addresses and data items.
- **Data models:** Modbus allows designers to define 1-bit or 16-bits data operations. In this case, ONE implements a 16-bits register data model.

## MODBUS integration for ONE Series

- Function Codes: Modbus protocol defines different operations that Modbus devices can perform between them. The function code to use will depend on what data model and what kind of access is implemented on target device. In our case, ONE uses 16-bits register data model with this available operations:
  - Read Holding Registers: function code **03** ("03" in hex)
  - Write Multiple Registers: function code **16** ("10" in hex)
- Command frame contents:

Address (1 byte)	Function Code (1 byte)	Data (N bytes, depends on selected function)	CRC (2 bytes)
---------------------	---------------------------	---	------------------

At this point we are able to analyze Modbus commands at a very low level and see what's inside them. Here we will review the first one:

- Performed operation: set "Master PA volume" to 0 dB value.
- ONE's variable address: "1500" in hexadecimal.
- Modbus command: "0A 10 14 FF 00 01 02 00 00 95 6E".
- Command decoding:
  - Address (1 byte): hex value "**0A**". Matches address 10 in decimal, which is ONE's Modbus default address.
  - Function Code (1 byte): hex value "**10**" ("Write Multiple Registers" operation, with decimal value of 16).
  - Data (these are the parameters for "Write Multiple Registers" operation):
    - Starting Address (2 bytes): hex value "**14FF**". This means register "1500"
    - Quantity of Registers (2 bytes): hex value "**0001**". This variable has only 1 block length, so we just need to write one register.
    - Byte Count (1 byte): hex value "**02**". Each register has two bytes length, and that's the size we need to write.
    - Registers Value (N x 2 bytes): hex value "**0000**". Here is where we set the desired value to be written. In this case we are sending zero to set a volume of 0 dB.
  - CRC (2 bytes): hex value "**956E**". This value comes from the calculation of the check value specified by Modbus protocol.