MODBUS integration for ONE Series

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

Review: 1

LDA audioTech

Support Handbook

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-458 port, that is physically located on the main board as shown on *Picture 1: RS-485 port location on ONE Series.*

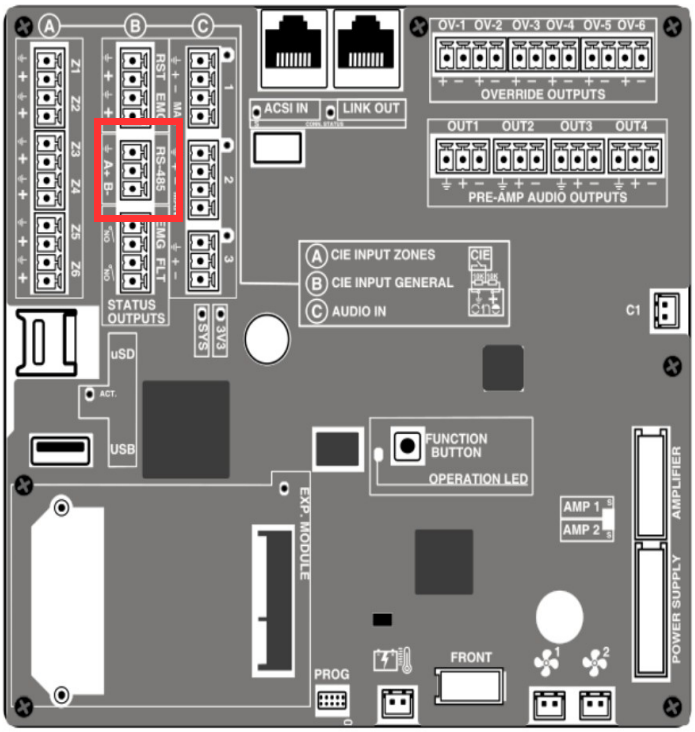
That RS-485 port has its own configuration parameters, that can be reached from the frontal screen interface, using the “Configuration” menu path:

- Configuration → CIE → RS-485 Conf.

NOTE: This action requires a L3 access level at ONE devices. By default, a user will have to introduce the password *0003*.

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.



Picture 1: RS-485 port location on ONE Series

On the other hand, there are also other related MODBUS parameters, summarized at the following table (ID value corresponds to v02.00.01.00 firmware version):

Variable name	Description	ID (Hex)
CIE485_LINK_TIMEOUT	CIE RS-485 bus link timeout for failure detection (in seconds). By default, its value is ZERO, meaning that the supervision is not active. Type: Integer, Read level: 2, Write level: 3 Item size: 1 block, Total size: 1 block Index: n/a	09D3
CIE485_MODBUS_ADDRESS	ONE slave address at the MODBUS bus. By default, its value is 10. Type: Integer, Read level: 2, Write level: 3 Item size: 1 block, Total size: 1 block Index: n/a	09D4
CIE485_MODBUS_ACCESS_LEVEL	Access level given to the MODBUS interface. By default, its value is 2. This value defines which operations the MODBUS remote control is allowed to execute. Type: Integer, Read level: 2, Write level: 3 Item size: 1 block, Total size: 1 block Index: n/a	09D5
DEV_MODBUS_FLAT_REGISTER_ADDRESSING	This mode avoids the register address decrement used in standard MODBUS-RTU communications. By default, its value is “Disabled”. Type: Integer, Read level: 1, Write level: 3 Item size: 1 block, Total size: 1 block Index: n/a	0CC7

Those variables are related to advanced configurations, and they are not frequently used. To modify them, it is needed to access through the frontal interface, using the method described on **ANNEX II: Advanced edition of ONE variables.**

It is recommended to set up this parameters **before using the MODBUS interface**.

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.

2.3 Link supervision

One of the main features of the MODBUS integration is monitoring ONE systems from an external device. Some installations may require a complete supervision of this remote link, and will notify the master in case of failure.

For this purpose, ONE uses a timer. That timer will be reset any time that ONE receives a command. If the CIE485_LINK_TIMEOUT value countdown finishes without any valid MODBUS command reception, the device detects it as a communication loss and activates the CIE link error.

If this error is active, it will automatically disappear in case the communications are recovered again (or after a manual error status restart).

When any new command is received, the timer will reset itself. The timer is configured by setting the CIE485_LINK_TIMEOUT parameter. By default, CIE_LINK_TIMEOUT value is 0, meaning that supervision is disabled.

2.4 Special register addressing (flat-mode)

MODBUS protocol standards include a decrease of 1 unit at the value of the command address before sending it. Under that performance, the slave recovers the correct value (by applying the increment) and reads it properly.

EXAMPLE: To access at register 100, when the command is created; the data string transmitted is 99. The slave that receives that command then applies a post-increment, recovering the 100 original addressing.

However, it is possible to find devices that translate the addresses directly to the MODBUS command string. In this case, you should use the flat-mode enabling the DEV_MODBUS_FLAT_REGISTER_ADDRESSING. (Possible values: “Enabled” or “Disabled”)

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.

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*.

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 1st. 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 2nd).

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:

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.

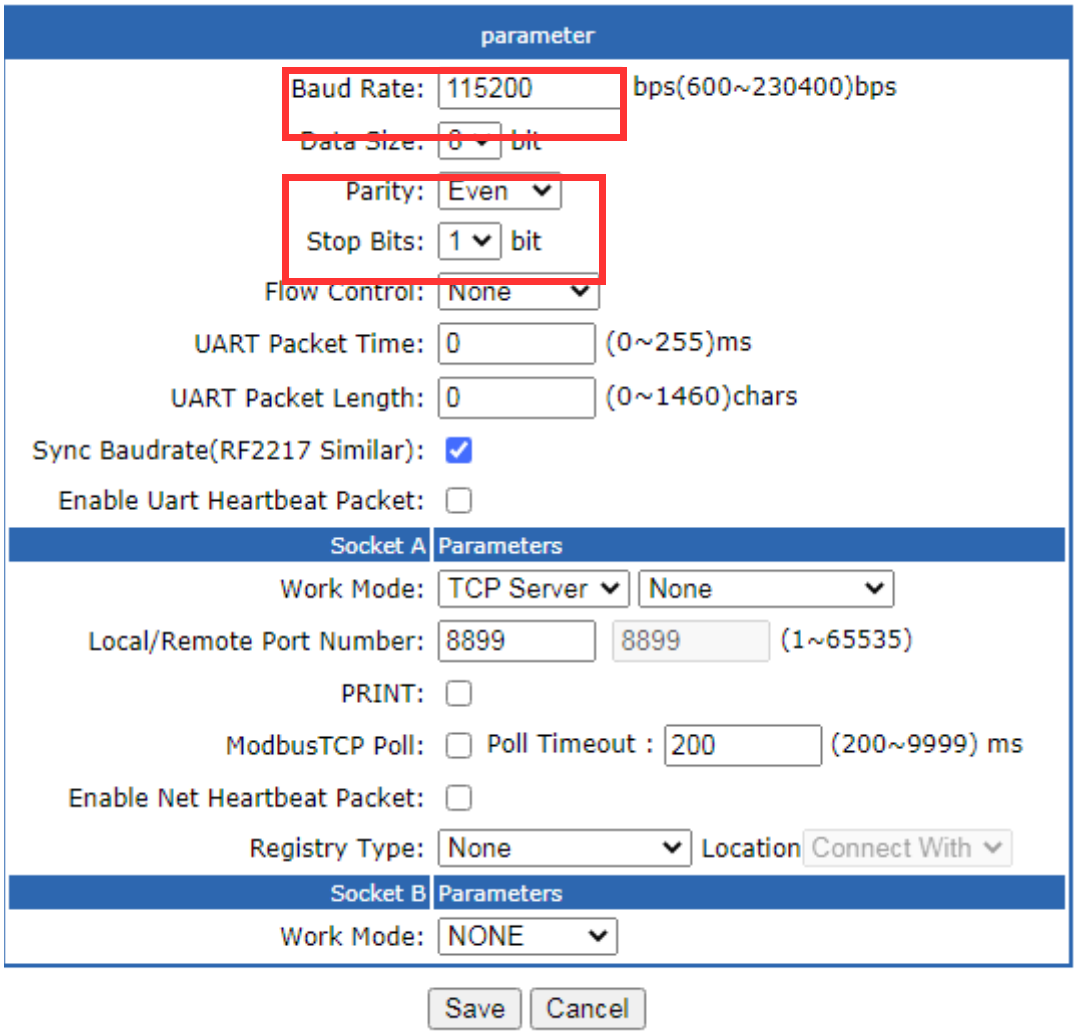
To access to it, the device IP address once connected to a network, by default it is (192.168.0.7). Default user name and password can be found at *Table 2: Default credentials to USR-N510 configuration software*.

User name:	admin
Password:	admin

Table 2: Default credentials to USR-N510 configuration software

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.



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

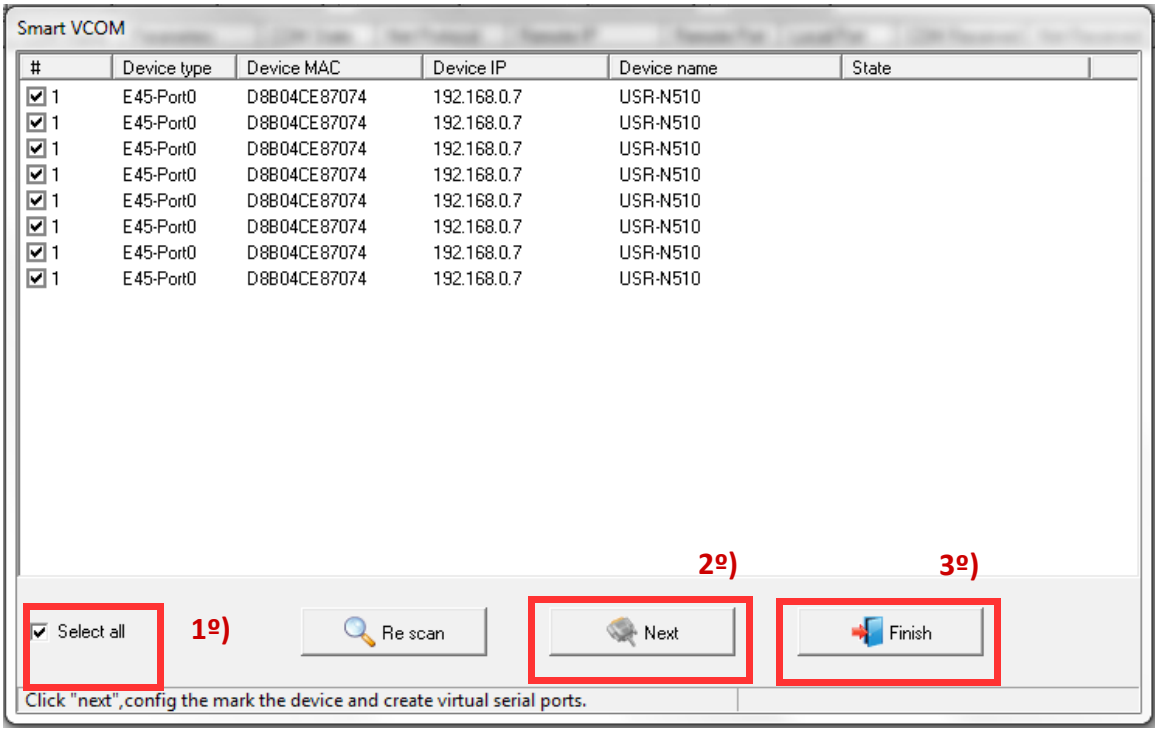
By default, the main window view is *Current Status.* On this guide only *PORT 1* will be configured. Required configuration of *PORT 1* is shown at *Picture 5: USR-N510 setup (PORT 1 section).* Values must coincide with linked ONE device's values. In case of any modification, that modification should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

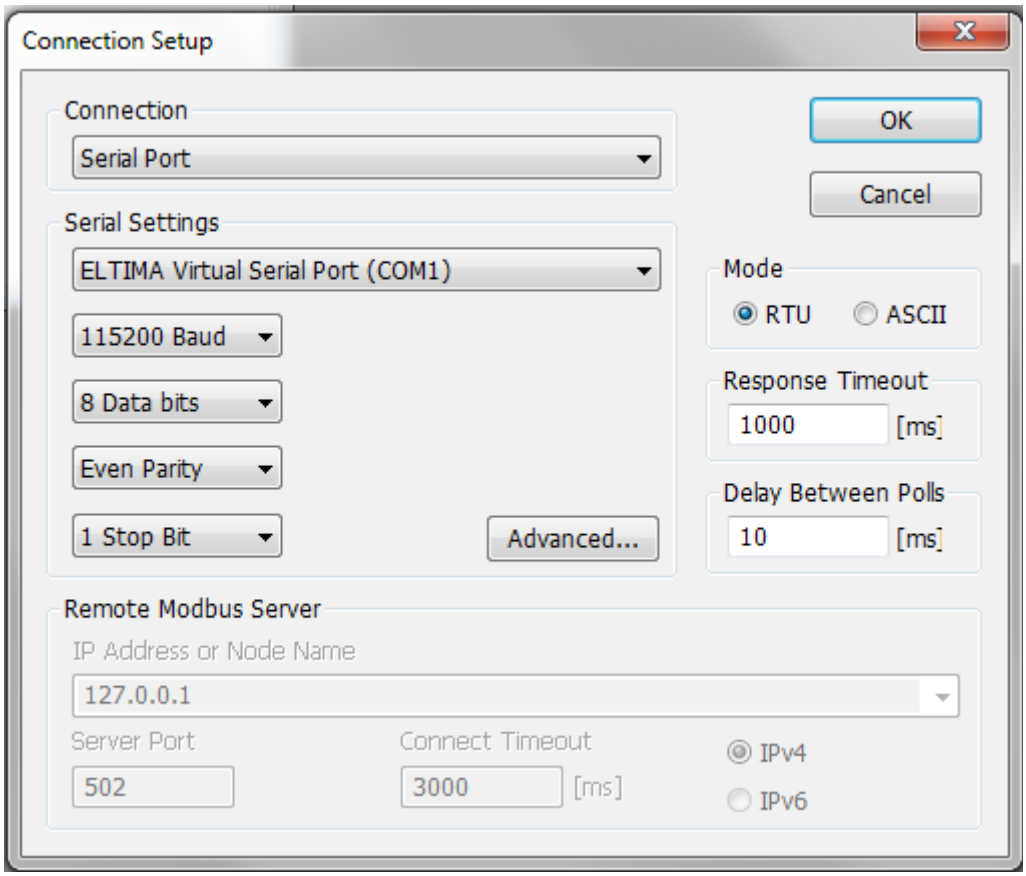
On the software main window, select *Connection → Connect*.

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.



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.

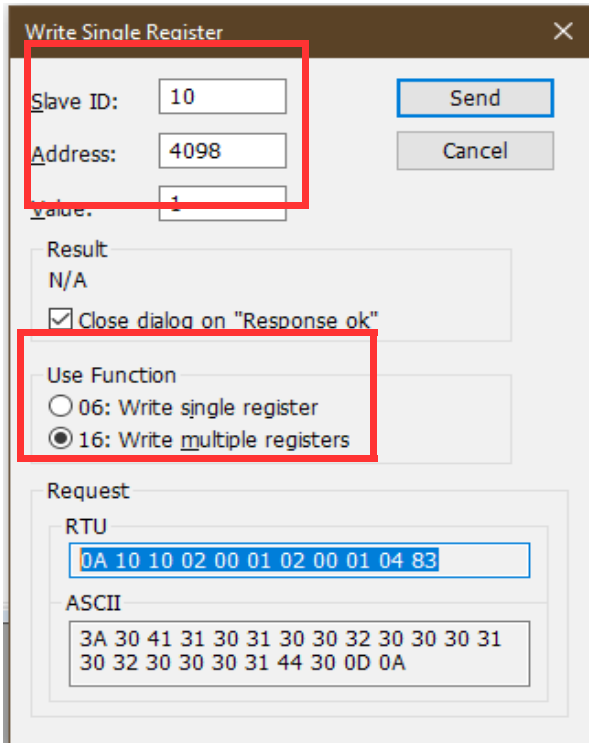
Once a cell is selected, a new window gets opened. In this window, the command that would be sent to the ONE device can be written. From here it can be sent to other ONE devices connected to the network. The required values are:

- *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.[2]
- *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.*

2 **NOTE:** Under the exceptions discussed onto 2.4 Special register addressing (flat-mode).



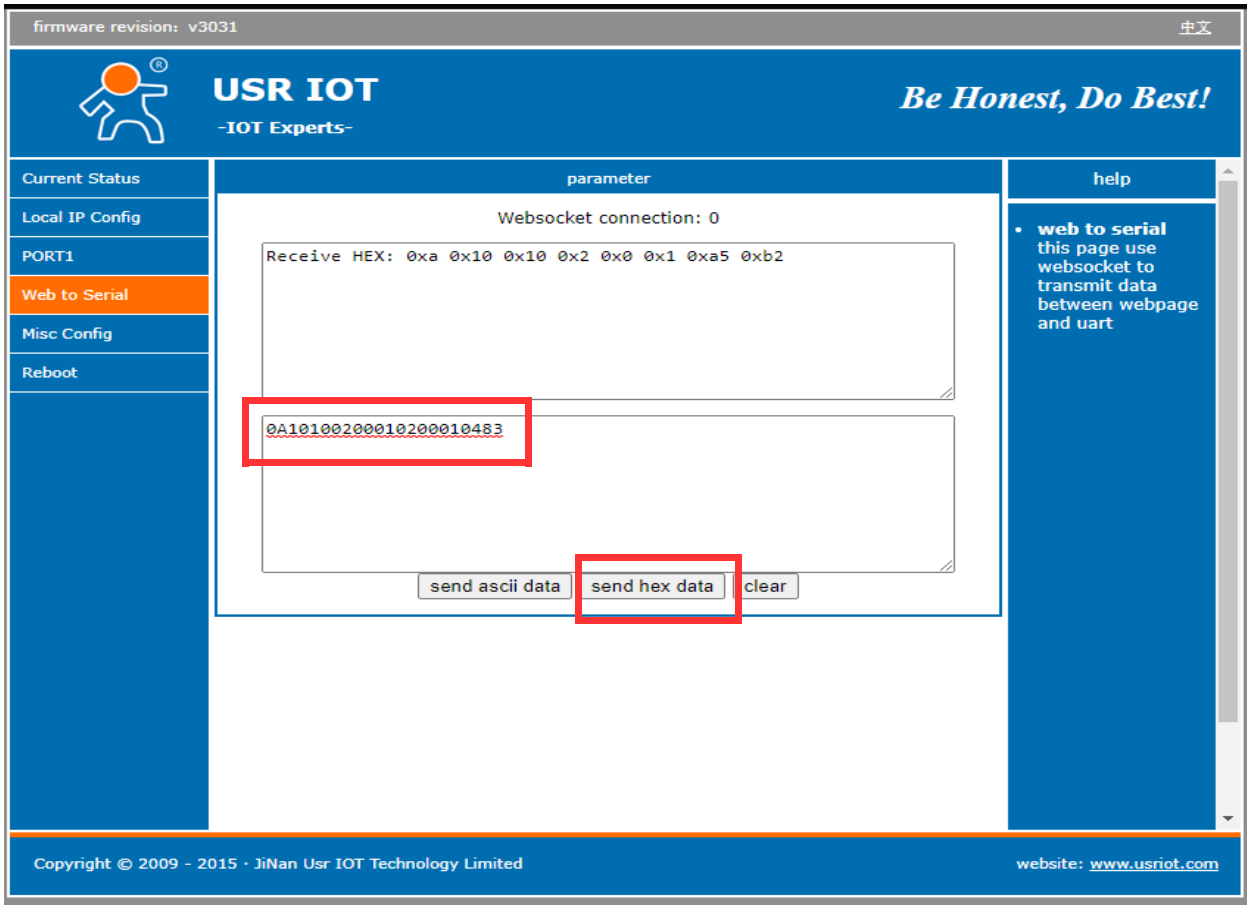
Picture 8: TEST function for ONE Series from Modbus Poll

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).

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.



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 an 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.

However, a series of key points will be given below 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.
- 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. In ANNEX IV: ONE parameter examples some example commands are given, 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, wich 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" (remember register address decrement explained in point 2.4).
    - 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 were 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.

ANNEX IV: ONE parameter examples

Some basic variable parameters are given to apply and check the proper performance of the remote communication between the controller device with Modbus Poll and ONE. Available commands may vary for each firmware version.

For more information about the modifiable ONE variables and their addresses, please ask LDA Audiotech Support department.

The following values are some of the variables in firmware version v02.00.01.00:

Variable name	Description	Hex	Dec	Modbus command examples
DAM_MASTER_PA_VOL	Master PA volume. Requires value [0, -100]. Type: Integer, Read level: 1, Write level: 1 Item size: 1 block, Total size: 1 block Index: n/a	1500	5376	0A1014FF0001020000956E (Sets master PA volume to 0dB)
DEV_ACT_TEST	Launches device test. Type: Action, Read level: n/a, Write level: 0 Item size: 1 block, Total size: 1 block Index: n/a	1003	4099	0A1010020001020000C543 (Runs device test)
SYS_EMG_ACTIVE	Checks system EMG status. Type: Boolean, Read level: 0, Write level: n/a Item size: 1 block, Total size: 1 block Index: n/a	28CE	10446	0A0328CD00011D2E (Checks System EMG status)
DEV_ACT_ACK	Launch device status acknowledge. Type: Text string, Read level: n/a, Write level: 1 Item size: 1 block, Total size: 1 block Index: n/a	1002	4098	0A1010010001020000C570 (Runs device ACK operation)
CIE485_LINK_TIMEOUT	CIE RS-485 bus link timeout for failure detection. Type: Integer, Read level: 2, Write level: 3 Item size: 1 block, Total size: 1 block Index: n/a	09D3	2515	0A1009D2000102001EDE1A (Sets link timeout to 30 seconds)

Tabla 3: Status commands

Variable name	Description	Hex	Dec	Modbus command examples
SYS_ACT_EMG_REMOTE	Activates system EMG (remote actor). Type: Action, Read level: n/a, Write level: 2 Item size: 1 block, Total size: 1 block Index: n/a	1210	4624	0A10120F0001020000E79E (Activates system EMG)
SYS_ACT_RST	Launches device status reset. Type: Action, Read level: n/a, Write level: 2 Item size: 1 block, Total size: 1 block Index: n/a	1001	4097	0A1010000001020000C4A1 (Performs RST operation on system)
SYS_ACT_EVAC_ZONE_REMOTE	Launches Evac on the selected zone (remote actor). Requires value [1, zone quantity]. Frame for Z1. Type: Action, Read level: n/a, Write level: 2 Item size: 1 block, Total size: 64 blocks Index: 1,60	1216	4630	0A1012150001020012464 (Launches EVAC on zone 1)
	Frame for Z2.	1217	4631	0A1012160001020012457 (Launches EVAC on zone 2)
	Frame for Z3.	1218	4632	0A1012170001020012586 (Launches EVAC on zone 3)
SYS_ACT_ALERT_ZONE_REMOTE	Launches Alert on the selected zone (remote actor). Requires value [1, zone quantity]. Frame for Z1. Type: Action, Read level: n/a, Write level: 2 Item size: 1 block, Total size: 64 blocks Index: 1,60	1256	4694	0A1012550001020012AA4 (Launches ALERT on zone 1)
	Frame for Z2.	1257	4695	0A1012560001020012A97 (Launches ALERT on zone 2)
	Frame for Z3.	1258	4696	0A1012570001020012B46 (Launches ALERT on zone 3)
SYS_ACT_SILENCE_ZONE_REMOTE	Mute selected zone (remote actor). Requires value [1, zone quantity]. Frame for Z1. Type: Action, Read level: n/a, Write level: 2 Item size: 1 block, Total size: 64 blocks Index: 1,60	1296	4758	0A1012950001020013BA4 (Silences zone 1)
	Frame for Z2.	1297	4659	0A1012960001020013B97 (Silences zone 2)

Variable name	Description	Hex	Dec	Modbus command examples
	Frame for Z3.	1298	4660	0A1012970001 0200013A46 (Silences zone 3)
SYS_ACT_RESET_ZONE_REMOTE	Reset selected zone (remote actor). Requires value [1, zone quantity]. Frame for Z1. Type: Action, Read level: n/a, Write level: 2 Item size: 1 block, Total size: 64 blocks Index: 1,60	12D6	4822	0A1012D50001 0200013564 (Resets zone 1)
	Frame for Z2.	12D7	4823	0A1012D60001 0200013557 (Resets zone 2)
	Frame for Z3.	12D8	4824	0A1012D70001 0200013486 (Resets zone 3)

Tabla 4: Emergency remote control commands