USER MANUAL

MULTIPROTOCOL “KEY” GATEWAYS SERIES

MODBUS SERIAL / ETHERNET GATEWAYS AND SERIAL DEVICE SERVER

SENECA S.r.l.
Via Austria 26 – 35127 – Z.I. - PADOVA (PD) - ITALY
Tel. +39.049.8705355 – 8705355 Fax +39 049.8706287
www.seneca.it

ORIGINAL INSTRUCTIONS

CAUTION

SENECA does not guarantee that all specifications and/or aspects of the product and firmware, included in them, will meet the requirements of the actual final application even if the product referred to in this documentation is in compliance with the technological state of the art.
The user assumes full responsibility and/or risk with regard to the configuration of the product to achieve the intended results in relation to the specific installation and/or end application.
SENECA may, with prior agreement, provide consultancy services for the successful completion of the final application, but under no circumstances can it be held responsible for its proper functioning.
The SENECA product is an advanced product, the operation of which is specified in the technical documentation supplied with the product itself and/or can be downloaded, if desired prior to purchase, from the www.seneca.it website.
SENECA has a policy of continuous development and accordingly reserves the right to make and/or introduce - without prior notice - changes and/or improvements to any product described in this documentation.
The product described in this documentation may solely and exclusively be used by personnel qualified for the specific activity and in accordance with the relevant technical documentation, with particular attention being paid to the safety instructions.
Qualified personnel means personnel who, on the basis of their training, competence and experience, are able to identify risks and avoid potential hazards that could occur during the use of this product.
SENECA products may only be used for the applications and in the manner described in the technical documentation relating to the products themselves.
To ensure proper operation and prevent the occurrence of malfunctions, the transport, storage, installation, assembly, maintenance of SENECA products must comply with the safety instructions and environmental conditions specified in this documentation.
SENECA's liability in relation to its products is governed by the general conditions of sale, which can be downloaded from www.seneca.it.
Neither SENECA nor its employees, within the limits of applicable law, will in any case be liable for any lost profits and/or sales, loss of data and/or information, higher costs incurred for goods and/or replacement services, damage to property and/or persons, interruption of activities and/or provision of services, any direct, indirect, incidental, pecuniary and non-pecuniary, consequential damages in any way caused and/or caused, due to negligence, carelessness, incompetence and/or other liabilities arising from the installation, use and/or inability to use the product.

CONTACT US	
Technical support	supporto@seneca.it
Product information	commerciale@seneca.it

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

Document revisions

DATE	REVISION	NOTES	AUTHOR
21/03/2022	01	First revision for new hardware and new firmware.	MM
25/07/2022	02	Removed wording related to tags	MM
06/02/2023	03	Added new model Z-KEY-2ETH	MM
19/07/2024	05	Added new functions from firmware rev. 152 Now all Z-KEY-0 and Z-KEY-2ETH functions are available also on R-KEY-LT	MM
13/02/2025	07	Added new functions for new hardware “Flex” Parts of the manual rewritten for uniformity with other protocols.	MM
24/02/2025	08	Added chapter on the meaning of LEDs Rewritten the chapter “description”	MM
26/03/2025	09	Added new functions from firmware rev 217	MM
05/05/2025	10	Added info on IO Master mode from rev 221 firmware	MM
11/09/2025	11	Added new functions firmware rev 222	AC - MM

This document is the property of SENECA srl.
Copies and reproduction are prohibited unless authorised.

TABLE OF CONTENTS

1. PRELIMINARY WARNINGS 7
1.1. DESCRIPTION 7
1.2. Modbus protocol 8
1.3. FEATURES OF THE "KEY" SERIES COMMUNICATION PORTS 8

2. DEVICE HARDWARE REVISION 9

3. FLEX TECHNOLOGY FOR PROTOCOL CHANGE 10
3.1. CHANGING PROTOCOLS WITH THE SENECA DISCOVERY DEVICE SOFTWARE 11

4. LED MEANING 12
4.1. LED MODEL Z-KEY-0 (MODBUS) 12
4.2. LED MODEL R-KEY-LT-0 (MODBUS) 13
4.3. LED MODEL Z-KEY-2ETH-0 (MODBUS) 14

5. ETHERNET PORT 15

6. FIRMWARE UPDATE 16

7. MODBUS GATEWAY ETHERNET TO SERIAL MODE 17
7.1. How it works 18

8. ETHERNET TO SERIAL MODBUS TAGS GATEWAY MODE 19
8.1. Master PORTS 20
8.2. 1 Master port and 1 Slave port (Z-KEY / z-key-2eth models only) 21
8.2.1. How it works 22
8.3. Slave Port 1 and port 2 23
8.3.1. How it works 23
8.4. Modbus Client 24
8.5. SIMPLIFIED TAG DIAGNOSTICS 25
8.6. EXTENDED TAG DIAGNOSTICS 26

9. SERIAL DEVICE SERVER MODE AND TCP SERVER SERIAL DEVICE 28
9.1. How it works 29
9.2. Seneca ETHERNET TO SERIAL CONNECT 29
9.2.1. Installing the Seneca Serial to Ethernet Connect driver 30
9.2.2. Select the Com port for Seneca Serial to Ethernet Connect 32
9.2.3. MAINTENANCE OF VIRTUAL SERIAL PORTS 34
9.2.4. CHANGING THE NAME OF THE COM PORTS 35

10. GATEWAY SERIAL TO ETHERNET MODE 37
10.1. HOW IT WORKS 37

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

11. MODBUS GATEWAY SERIAL TO ETHERNET VIRTUAL ID MODE 38
11.1. HOW IT WORKS 38

12. MODBUS IO DEVICE MASTER MODE 39
12.1. HOW IT WORKS 39

13. MODBUS IO DEVICE SLAVE MODE 40
13.1. HOW IT WORKS 40

14. "-0" GATEWAY WEBSERVER 41
14.1. STEP BY STEP GUIDE FOR THE FIRST ACCESS TO THE WEBSERVER 41

15. WEBSERVER DEVICE CONFIGURATION 42
15.1. SETUP PAGE 42
15.1.1. GENERAL CONFIGURATION PARAMETERS 43
15.2. VIRTUAL ADDRESSES SETUP PAGE 48
15.3. SETUP TAG PAGE (MODBUS TAGS GATEWAY MODE ONLY) 49
15.3.1. BUTTONS AND TAG CLONE 49
15.3.2. FIELDS MEANING 50
15.3.3. Real-time view of the Modbus Gateway 53
15.4. TAG/MODBUS COMMAND SETUP PAGE (FOR MODBUS IO DEVICE MASTER MODE ONLY) 54
15.4.1. BUTTONS AND COMMAND CLONE 54
15.4.2. FIELDS MEANING 55
15.4.3. Real-time view of the Modbus Gateway 57

16. RESETTING THE DEVICE TO ITS FACTORY CONFIGURATION 58

17. TEMPLATE EXCEL 59

18. SERIAL TRAFFIC MONITOR 60

19. INSTALLING MULTIPLE DEVICES IN A NETWORK USING THE "DHCP FAIL ADDRESS" 61

20. THE DB9 RS232 CABLE 61

21. EXTEND THE RS485 BUS ON ETHERNET: SERIAL MODBUS ON ETHERNET AND THEN ETHERNET ON SERIAL 61

22. SUPPORTED MODBUS COMMUNICATION PROTOCOLS 62
22.1. Supported Modbus function codes 62

23. INFORMATION ABOUT MODBUS REGISTERS 63

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

23.1. NUMBERING OF "0-BASED" OR "1-BASED" MODBUS ADDRESSES 64
23.2. NUMBERING OF MODBUS ADDRESSES WITH "0-BASED" CONVENTION 64
23.3. NUMBERING OF MODBUS ADDRESSES WITH "1 BASED" CONVENTION (STANDARD) 65
23.4. BIT CONVENTION WITHIN A MODBUS HOLDING REGISTER 65
23.5. MSB and LSB BYTE CONVENTION WITHIN A MODBUS HOLDING REGISTER 66
23.6. REPRESENTATION OF A 32-BIT VALUE IN TWO CONSECUTIVE MODBUS HOLDING REGISTERS 66
23.7. TYPE OF 32-BIT FLOATING POINT DATA (IEEE 754) 67

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

1. PRELIMINARY WARNINGS

⚠ ATTENTION!

This user manual extends the information from the installation manual to the configuration of the device. Use the installation manual for more information.

⚠ ATTENTION!

In any case, SENECA s.r.l. or its suppliers will not be responsible for the loss of data/revenue or consequential or incidental damages due to negligence or bad/improper management of the device, even if SENECA is well aware of these possible damages.
SENECA, its subsidiaries, affiliates, group companies, suppliers and distributors do not guarantee that the functions fully meet the customer's expectations or that the device, firmware and software should have no errors or operate continuously.

1.1. DESCRIPTION

The Z-KEY-0, R-KEY-LT-0, Z-KEY-2ETH-0 products allow the conversion of data coming from the Modbus serial bus to the Modbus ethernet bus and vice versa.
The devices can also be configured as serial device servers (i.e. they perform an ethernet / serial conversion transparent to the protocol).

In particular, the following operating modes are possible:

- *Modbus Gateway Ethernet to Serial (PORT#1)*
- *Modbus Gateway Ethernet to Serial (PORT#2) (2 serial ports models only)*
- *Modbus Gateway Ethernet to Serial (PORT#1 AND PORT#2) (2 serial ports models only)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 MASTER)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 MASTER PORT#2 SLAVE) (2 serial ports models only)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 SLAVE PORT#2 MASTER) (2 serial ports models only)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 SLAVE)*
- *Serial device server (PORT #1)*
- *Serial device server (PORT #2) (2 serial ports models only)*
- *Modbus Gateway Serial to Ethernet (PORT#1 AND PORT#2)*
- *TCP Server Serial Device (PORT #1)*
- *TCP Server Serial Device (PORT #2)*
- *Modbus Gateway Serial to Ethernet Virtual ID (PORT#1 AND PORT#2)*
- *Modbus IO Device Master*
- *Modbus IO Device Slave*

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

1.2. MODBUS PROTOCOL

The supported Modbus protocols are:
Modbus RTU Master
Modbus RTU Slave
Modbus ASCII Master
Modbus ASCII Slave
Modbus TCP-IP Server
Modbus TCP-IP Client
For further information on these protocols, see the Modbus specification website:
http://www.modbus.org/specs.php

1.3. FEATURES OF THE "KEY" SERIES COMMUNICATION PORTS

PRODUCT	ETHERNET PORTS	SERIAL PORT # 1 RS232/RS485 CONFIGURABLE	RS485 SERIAL PORT # 2	ISOLATED SERIAL PORTS
Z-KEY-0	1	1	1	Yes, both ports
R-KEY-LT-0	1	1	NO	NO
Z-KEY-2ETH-0	2	1	1	Yes, both ports

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

2. DEVICE HARDWARE REVISION

With a view to continuous improvement, Seneca updates and makes the hardware of its devices increasingly more sophisticated. It is possible to know the hardware revision of a product via the label on the side of the device.
An example of an R-KEY-LT product label is the following:

The label also shows the firmware revision present in the device (in this case 2.0.1.0) at the time of sale, the hardware revision (in this case) is E00.
To improve performance or extend functionality, Seneca recommends updating the firmware to the latest available version (see the section dedicated to the product on www.seneca.it).

An internal Webserver is also available for configuration and display of values in real time.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

3. FLEX TECHNOLOGY FOR PROTOCOL CHANGE

Starting from the hardware revision indicated in the following table, the KEY series devices include Flex technology.

GATEWAY	*FLEX TECHNOLOGY SUPPORTED BY HARDWARE REVISION*
Z-KEY	"G00"
R-KEY-LT	"E00"
Z-KEY-2ETH	"C00"

Flex allows you to change the combination of industrial communication protocols supported by the gateways at will from a list of available ones, the development is continuously updated, for a complete list refer to the page: https://www.seneca.it/flex/

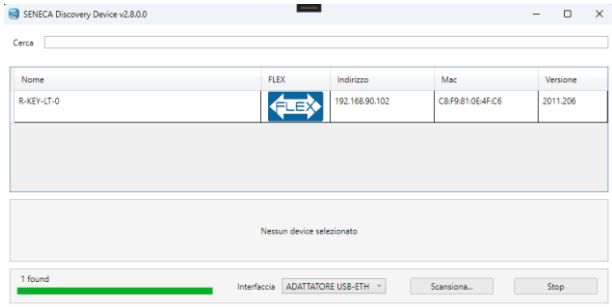
Some examples of supported protocols are:

The gateway then becomes "universal" and compatible with Siemens or Rockwell or Schneider systems etc... without the need to purchase different hardware.

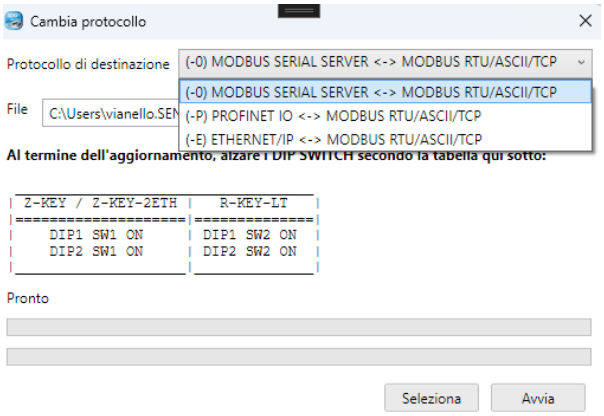
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

3.1. CHANGING PROTOCOLS WITH THE SENECA DISCOVERY DEVICE SOFTWARE

From revision 2.8 the Seneca Discovery Device software identifies the devices that support the "Flex" technology:



For example, in the case in the figure it is possible to press the "Change Protocol" button and select the destination protocol from those in the list:



At the end of the operation, bring (only at the first power-on) the dip switches 1 and 2 to "ON" to force the device to default (see also the chapter "RESETTING THE DEVICE TO ITS FACTORY CONFIGURATION").

Always refer to the user manual of the communication protocol installed in the device by downloading it from the Seneca website.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

4. LED MEANING

The devices are equipped with LEDs whose meaning is as follows:

4.1. LED MODEL Z-KEY-0 (MODBUS)

LED	***STATUS***
PWR	***Steady on***: device powered and IP address set ***Flashing***: IP address not yet set ***Off***: device not powered
COM	Not used
TX1	***Flashing:*** data transmission on serial port #1 ***Off:*** no transmission on serial port #1
RX1	***Flashing:*** data reception on serial port #1 ***Steady on:*** check wiring on serial port #1 ***Off:*** no reception on serial port #1
TX2	***Flashing:*** data transmission on serial port #2 ***Off:*** no transmission on serial port #2
RX2	***Flashing:*** data reception on serial port #2 ***Steady on:*** check wiring on serial port #2 ***Off:*** no reception on serial port #2
ETH ACT (GREEN)	***Flashing:*** presence of data on ethernet port ***Steady on:*** ethernet port connected but no data present ***Off:*** check wiring of the ethernet port
ETH LNK (YELLOW)	***Steady on:***ethernet cable connected ***Off:*** check the wiring of the ethernet port

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

4.2. LED MODEL R-KEY-LT-0 (MODBUS)

LED	*STATUS*
PWR	***Steady on***: device powered and IP address set ***Flashing***: IP address not yet set ***Off***: device not powered
COM	Not used
TX	***Flashing:*** data transmission on serial port ***Off:*** no transmission on serial port
RX	***Flashing:*** data reception on serial port ***Steady on:*** check wiring on serial port ***Off:*** no reception on serial port
ETH ACT (GREEN)	***Flashing:*** presence of data on ethernet port ***Steady on:*** ethernet port connected but no data present ***Off:*** check wiring of the ethernet port
ETH LNK (YELLOW)	***Steady on:***ethernet cable connected ***Off:*** check the wiring of the ethernet port

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

4.3. LED MODEL Z-KEY-2ETH-0 (MODBUS)

LED	*STATUS*
PWR	***Steady on***: device powered and IP address set ***Flashing***: IP address not yet set ***Off***: device not powered
COM	Not used
TX1	***Flashing:*** data transmission on serial port #1 ***Off:*** no transmission on serial port #1
RX1	***Flashing:*** data reception on serial port #1 ***Steady on:*** check wiring on serial port #1 ***Off:*** no reception on serial port #1
TX2	***Flashing:*** data transmission on serial port #2 ***Off:*** no transmission on serial port #2
RX2	***Flashing:*** data reception on serial port #2 ***Steady on:*** check wiring on serial port #2 ***Off:*** no reception on serial port #2
ET1	***Flashing:*** presence of data on ethernet port #1 ***Steady on:*** ethernet port #1 connected but no data present ***Off:*** check wiring of ethernet port #1
ET2	***Flashing:*** presence of data on ethernet port #2 ***Steady on:*** ethernet port #2 connected but no data present ***Off:*** check wiring of ethernet port #2

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

5. ETHERNET PORT

The factory configuration of the Ethernet port is:

STATIC IP: 192.168.90.101
SUBNET MASK: 255.255.255.0
GATEWAY: 192.168.90.1

Multiple devices must not be inserted on the same network with the same static IP.

⚠ ATTENTION!

DO NOT CONNECT 2 OR MORE FACTORY-CONFIGURED DEVICES ON THE SAME NETWORK, OR THE DEVICE WILL NOT WORK
(CONFLICT OF IP ADDRESSES 192.168.90.101)

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

6. FIRMWARE UPDATE

In order to improve, add or optimize the functions of the product, Seneca releases firmware updates on the device section on the www.seneca.it website

The firmware update is performed using Seneca tools or the webserver.

ATTENTION!

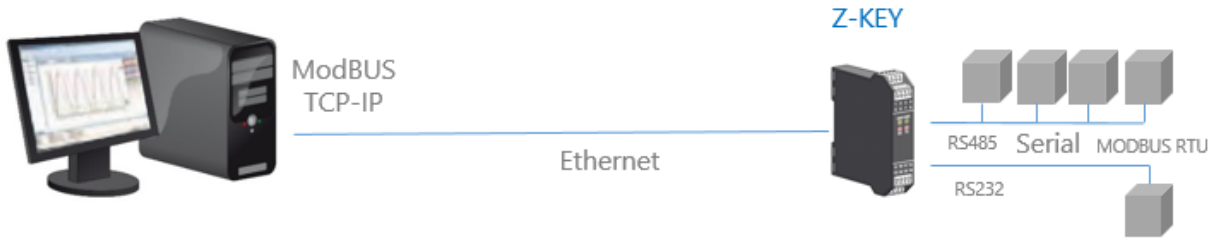
NOT TO DAMAGE THE DEVICE DO NOT REMOVE THE POWER SUPPLY DURING THE FIRMWARE UPDATE OPERATION.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

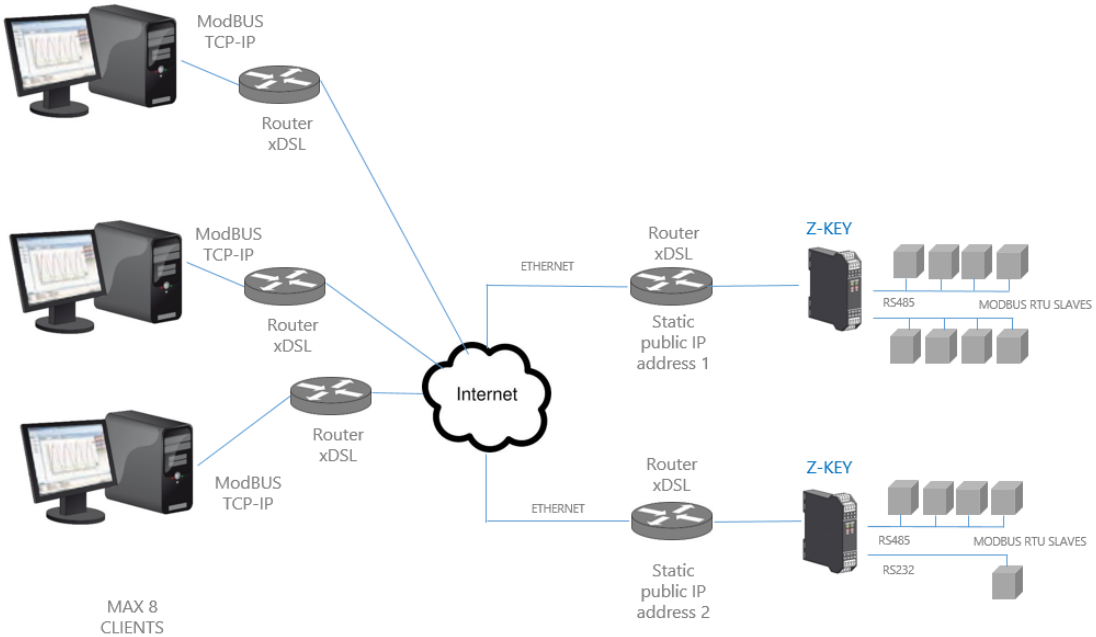
7. MODBUS GATEWAY ETHERNET TO SERIAL MODE

In this scenario a Modbus TCP-IP Client is connected via Ethernet to the Seneca Gateway which has one or more Modbus RTU Slaves (for example the Seneca Z-PC series) connected to its RS232/RS485 serial ports.

This figure shows an example of a LAN connection:



Remote communication via the Internet with several Modbus TCP-IP Clients is also possible:

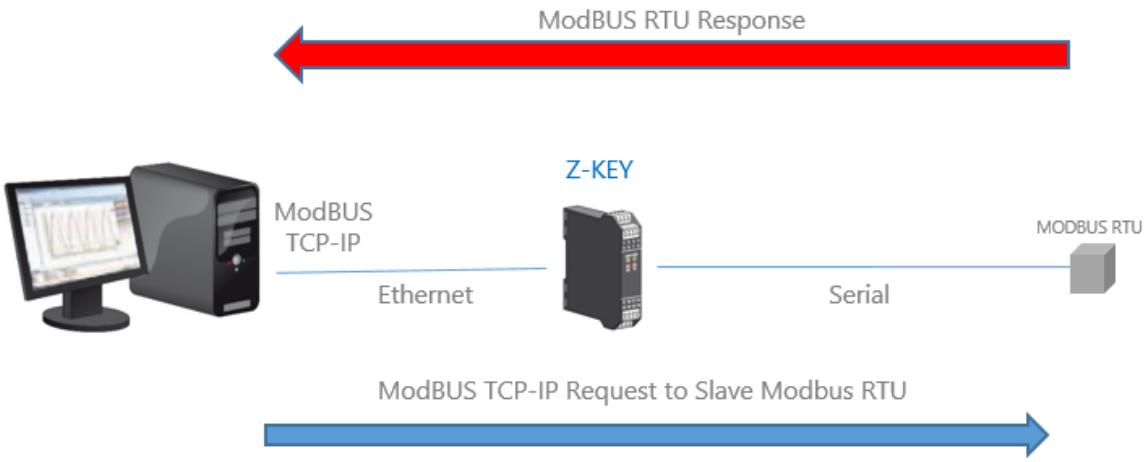


A maximum of 8 simultaneous Modbus TCP-IP Clients are supported.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

7.1. HOW IT WORKS

The Modbus from Ethernet to Serial Gateway is the simplest way to communicate with Modbus RTU Slaves via an Ethernet connection.
It is not necessary to indicate which registers should be requested, because the conversion from Ethernet to serial is performed in real time and transparently.
The Gateway only requires the Network and Serial Configuration (baud rate, parity, etc.).



The Modbus TCP Client requests the reading/writing of a Modbus register via Ethernet, the gateway converts the request towards the serial slaves in Modbus RTU/ASCII and the Modbus RTU/ASCII response of the interrogated slave is subsequently converted towards the Modbus TCP Client.

The Z-KEY and Z-KEY-2ETH gateways have No. 2 serial ports, it is possible to set the function on a single port or on both, in this case the requests of the Client are replicated on both ports (so as to be able to use slaves with different baud rates).

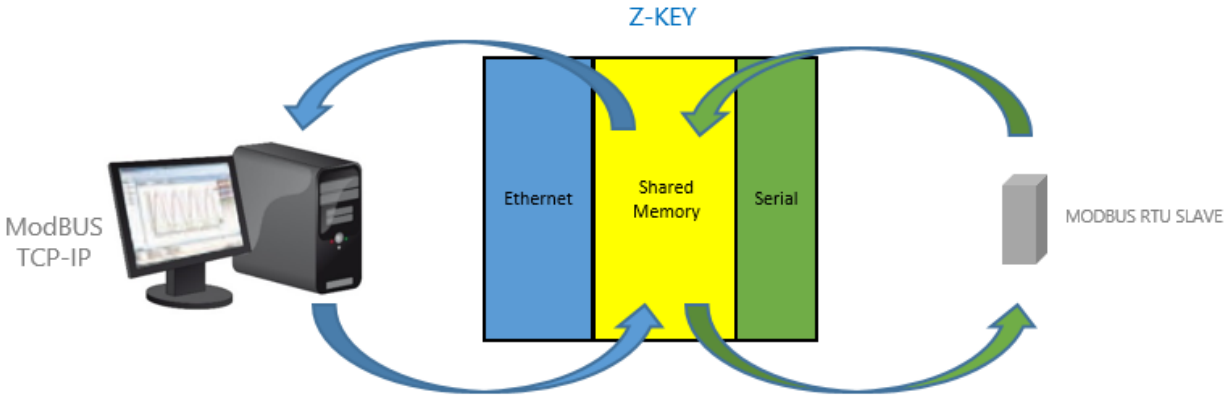
⚠ ATTENTION!

In this operating mode it is not possible to connect two serial Modbus slaves with the same Modbus address.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

8. ETHERNET TO SERIAL MODBUS TAGS GATEWAY MODE

In a Modbus network it is essential to read/write the registers of the various Modbus Slave devices as quickly as possible.
In the Gateways it is possible to configure a maximum of 500 tags (1 tag = 1 variable which can consist of one or more Modbus registers depending on the data format), these values are stored in a shared memory accessible from the Ethernet and/or from the Serials.
The shared memory is updated as quickly as possible by serial communication, so when a register is requested from the Ethernet port, the values are read directly from the Shared Memory without the need to wait for the Slave response time

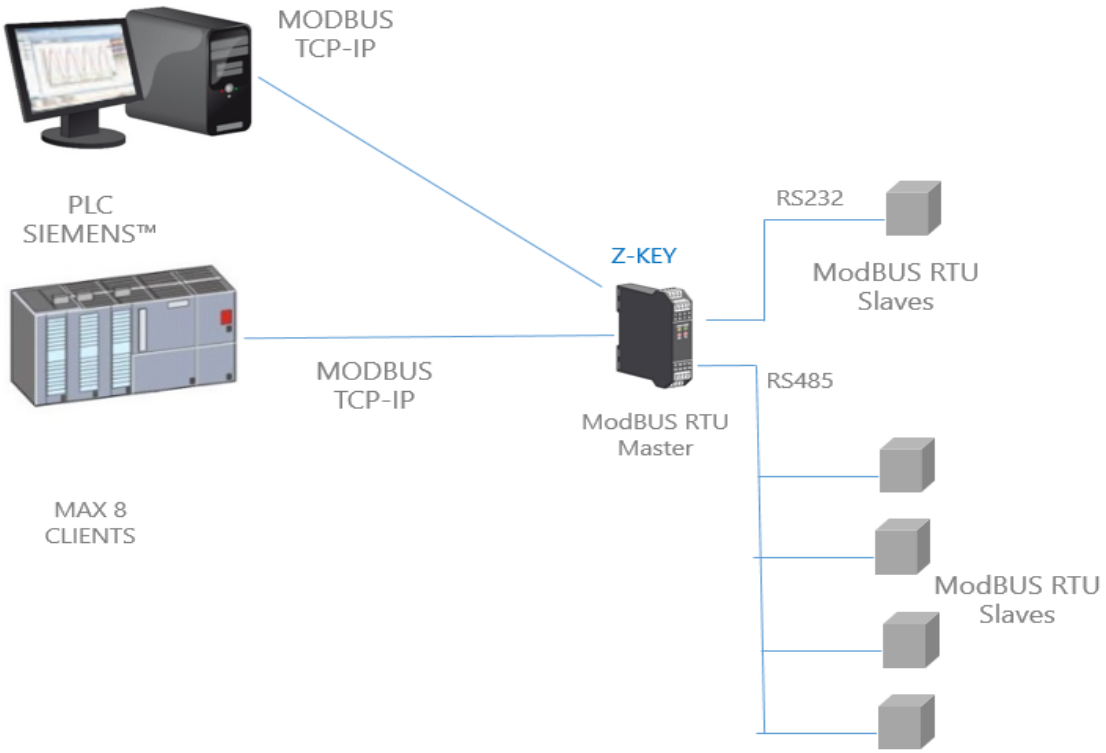


Another advantage is that the shared memory is also accessible from the webserver.
In this mode, the data acquired by several Modbus Slaves are grouped under a single Modbus address, this makes software development on the Ethernet side much easier and drastically reduces the polling time of the serial bus. The number of slaves to be managed can grow by using more than one Gateway.
Serial side readings/ writings are automatically optimized using multiple register readings/writings. With this optimization, the bus speed increases dramatically.
TAGs from Modbus TCP-IP Server devices are also supported (up to a maximum of 10 devices).
When a serial Modbus device generates a time-out error (for example due to a fault), in order to speed up the acquisition cycle, it is quarantined for a configurable time.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

8.1. MASTER PORTS

In this scenario a Modbus TCP-IP Client is connected via Ethernet to the Seneca Gateway and one or more serial Modbus Slaves (for example the Seneca Z-PC series) are connected to its RS232/RS485 serial ports.
The Modbus gateway is used to optimize network performance and Modbus TCP-IP PLC software.
This figure shows an example of use of the mode with Master Port 1 and Port 2:



ATTENTION!

The R-KEY model has just one serial port.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

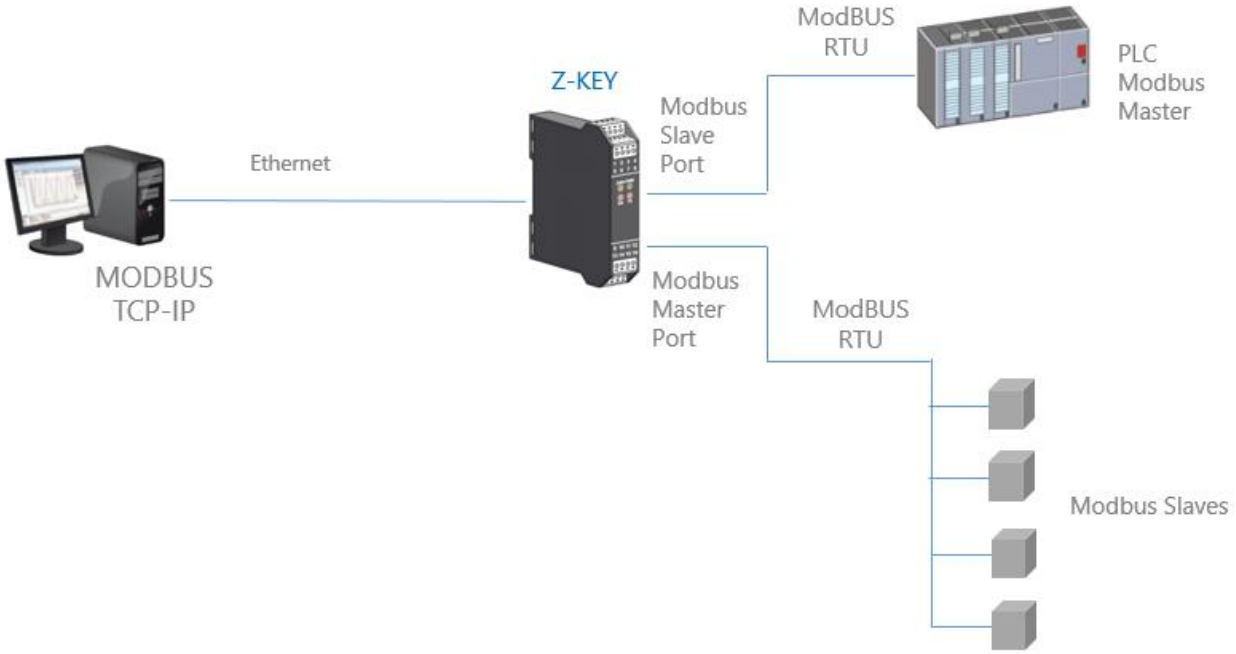
8.2. 1 MASTER PORT AND 1 SLAVE PORT (Z-KEY / Z-KEY-2ETH MODELS ONLY)

In this scenario a Modbus TCP-IP Client is connected via Ethernet to the Seneca Gateway, a PLC with only the Modbus Master serial protocol is connected to one of the serial ports of the Gateway configured as Slave. From this port, the data acquired by the Modbus Slaves connected to the serial port configured as Master are available for the PLC.

The PLC can read and write the data relating to the Modbus Slave registers.

The same operations can also be performed via Ethernet from the Modbus TCP Client (for example a PC).

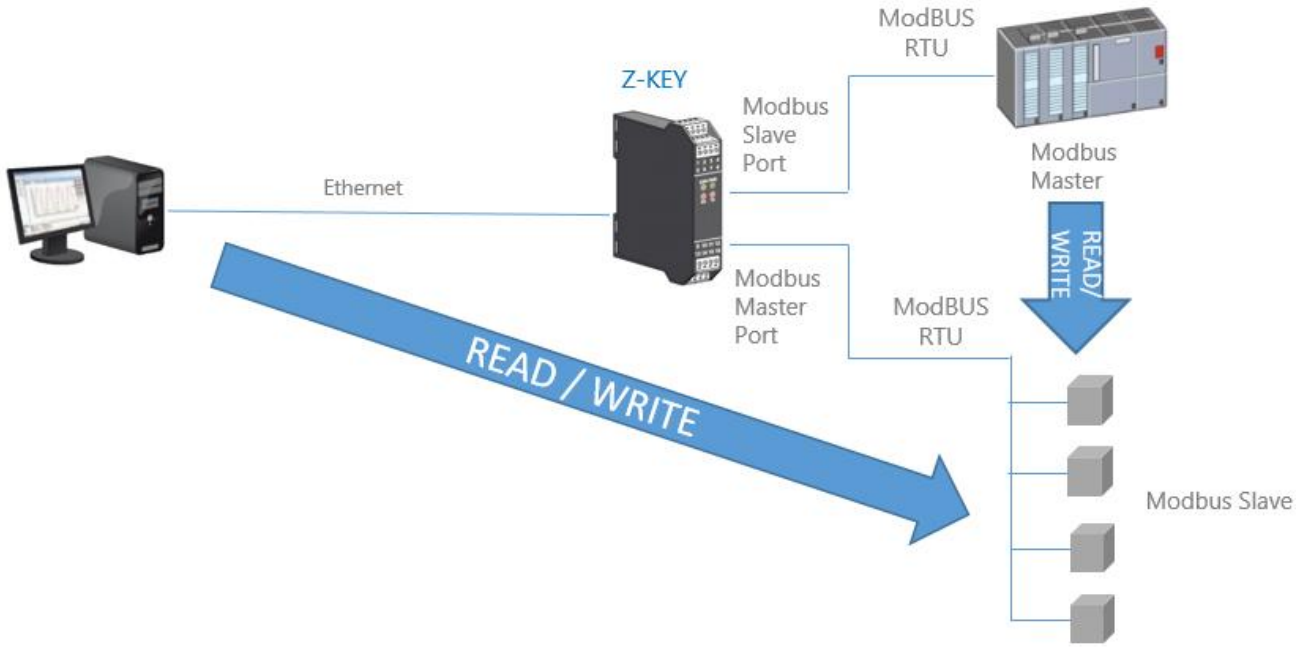
This figure shows an example of this scenario:



ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

8.2.1. *HOW IT WORKS*

Many PLCs only implement the serial Modbus Master protocol (because they do not have an Ethernet port). In this scenario, the values of the Modbus serial Slaves must be read/written by both the PLC via the Slave serial port and by a PC via the Ethernet port.

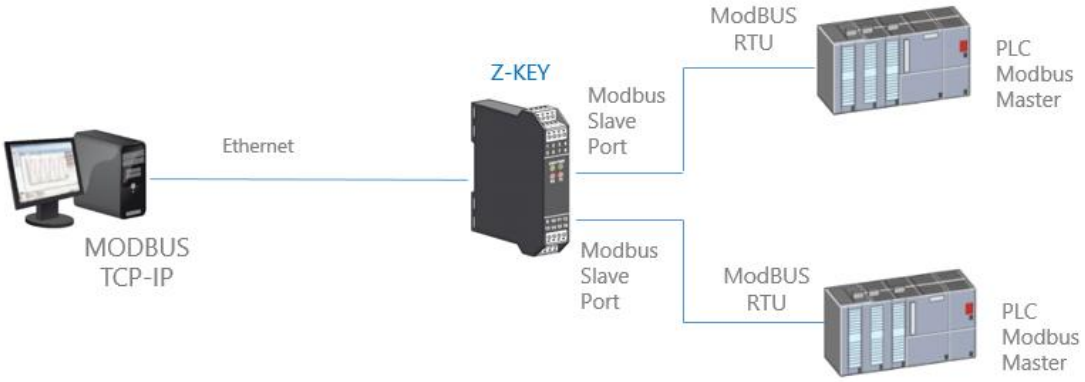


This function is possible because the Seneca Gateway uses an internal memory shared between the serial ports and the Ethernet port, where the data acquired from the serial Modbus Slave network are saved.
The Modbus Master PLC and the PC write/read the registers of the shared memory of the gateway which in real time keeps it updated by communicating with the Modbus Slaves.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

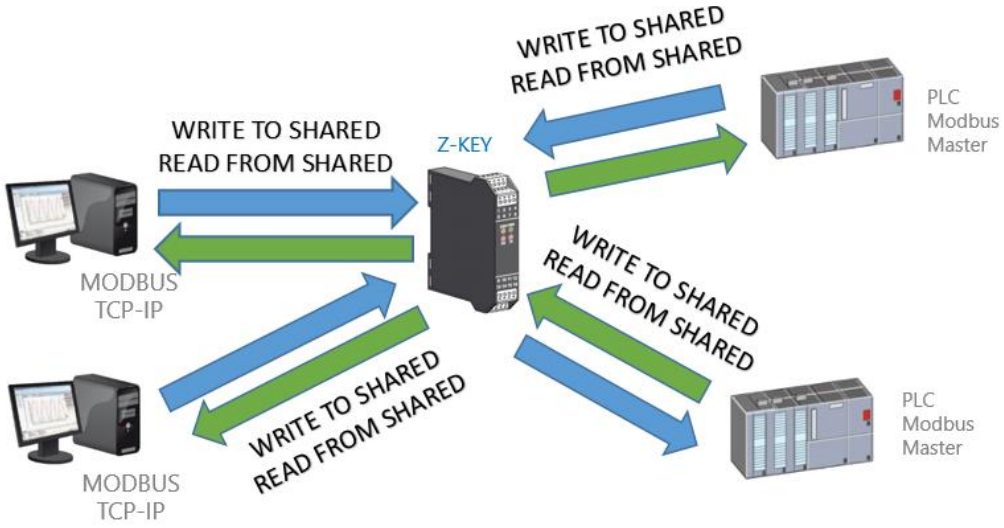
8.3. SLAVE PORT 1 AND PORT 2

In this scenario a Modbus TCP-IP Client is connected via Ethernet to the gateway, two PLCs that support only the Modbus Master serial protocol are connected to the two serial ports of the gateway, configured as Slave. Through these ports the PLCs can communicate with each other.
This figure shows an example of this scenario:



8.3.1. HOW IT WORKS

Many PLCs only implement the serial Modbus Master protocol (because they do not have an Ethernet port). If it is necessary to exchange Modbus registers between PLCs and PCs, the shared memory can be used.

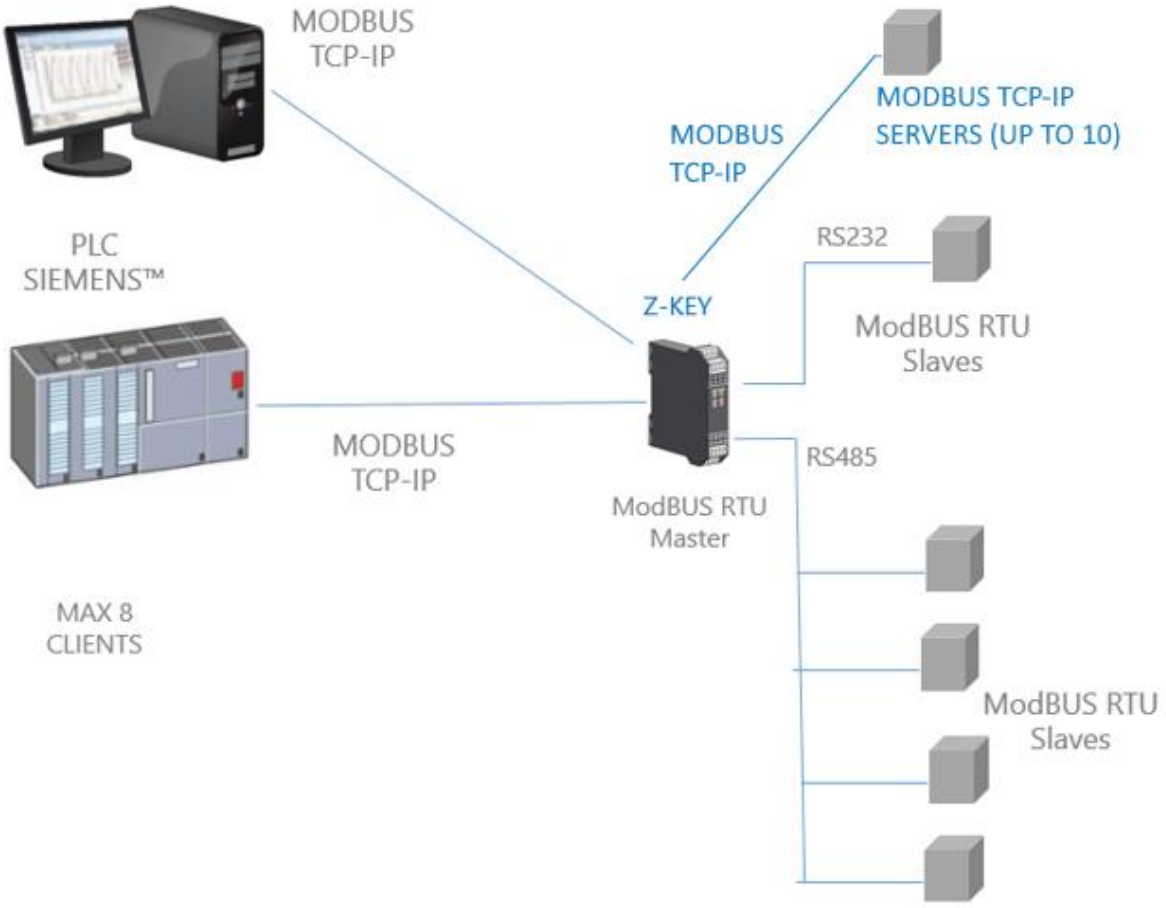


The shared memory of the gateway can be freely read and written by the Ethernet and/or by the serial ports.
The shared memory registers must be defined first with the TAG section of the setup.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

8.4. MODBUS CLIENT

In the functions with Master or Slave ports it is always possible to activate this function. In fact, the shared memory can also be populated by data from Modbus TCP-IP Server.
Please refer to the following figure:



ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

8.5. SIMPLIFIED TAG DIAGNOSTICS

Tag diagnostics is only available in Modbus TAGs Gateway mode.
Tag diagnostics can also be viewed via the Modbus serial and Ethernet ports: via special Modbus registers.
The first Modbus address, from which the simplified diagnostics starts, is by default 49001 (Holding Register 9000).
Each bit represents a tag with the following meaning:
1 = TAG OK
0 = TAG FAIL
The least significant bit is the status of tag no. 1
The next is the status of tag no. 2 and so on ...
For example the reading of the following registers:

49001	0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
49002	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

Means: TAG 1, TAG 4, TAG17, TAG 18 , TAG 19, TAG 20 OK, all the others in FAIL.
At the start, all tags are in a fail state (all 0).

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

8.6. EXTENDED TAG DIAGNOSTICS

Tag diagnostics is only available in Modbus TAGs Gateway mode.
When a tag is in an error state it is possible to get more information using extended diagnostics.
Extended diagnostics reserves 1 byte for each tag (since the limit is 500 tags, there are 500 bytes = 250 Modbus registers for extended diagnostics).
This diagnostics is found at the end of the simplified diagnostics (default starting Modbus address is 49033, Holding register 32).

Each Modbus register contains 2 tags, so for example:
49033 TAG02_TAG01
49034 TAG04_TAG03
…
49282 TAG500_TAG499
49283 LAST_LOOP_TIME_COM1 [x1 ms]
49284 LAST_LOOP_TIME_COM2 [x1 ms]

The meaning of the advanced diagnostics byte is:

BYTE VALUE	***MEANING***	***NOTE***
0	OK	The tag is read/written correctly
1	TIMEOUT	The response of the tag timed out, but will be queried again
2	DELAYED	Too many fails, tag polling is delayed (tag will be interrogated again after the configured quarantine time)
3	EXCEPTION	Modbus exception response but the tag will be queried again
4	CRC ERROR	CRC Modbus exception response but the tag will be queried again

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

For example:

49033 0x0000
49034 0x0002

It means that:

TAGs 1 and 2 are OK (0x00 and 0x00)
TAG 03 is in a delayed state (0x02)
TAG 4 is OK (0x00)

LAST_LOOP_TIME_COMx is a register that contains the last interrogation time of all serial tags (in how many of 10 ms) so, for example:

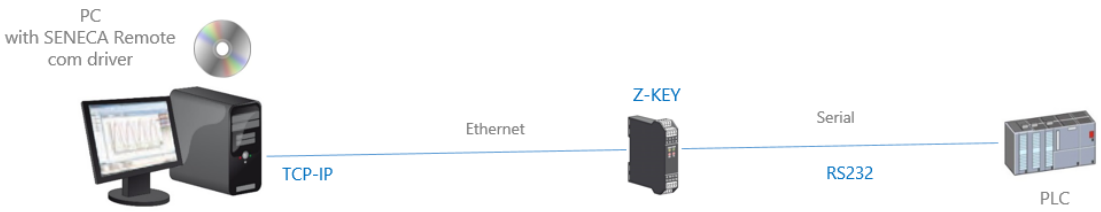
49283 25
49284 42

It means that the serial 1 loop was 250ms, the serial 2 loop was 420ms.

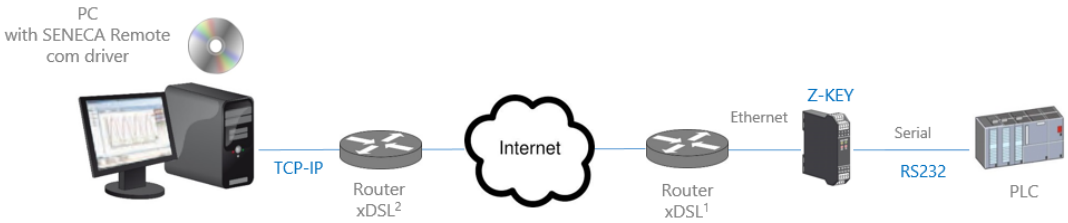
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

9. SERIAL DEVICE SERVER MODE AND TCP SERVER SERIAL DEVICE

These two modes have a similar operation, in particular the "Serial Device Server" mode refers to the management of the COM port according to RCF2217 and based on the Telnet protocol, while the "TCP Server Serial Device" mode is a transparent connection of all traffic .
In these modes a Serial device must be connected to a PC, but a direct connection is not possible (for example the device is too far from the PC)
The Seneca Gateway can be used to extend a serial connection using an Ethernet connection:



It is also possible to make an internet connection (typically for remote maintenance of machines, PLCs, etc.):



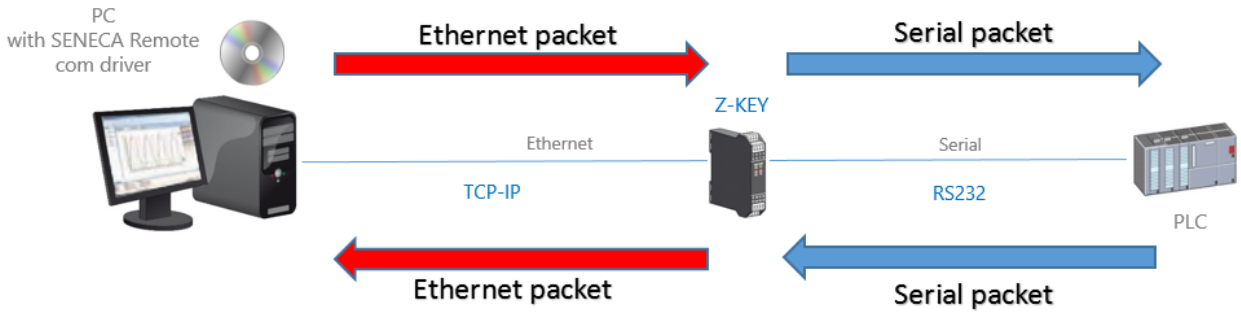
For internet communication, the router connected to the gateway must have a static IP address.

NOTE:
To use the ZModem serial protocol through the internet, remember to activate the crash recovery option of the protocol.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

9.1. HOW IT WORKS

If a PC program only has serial port support but you need to exit via an Ethernet connection, you can install the Serial to Ethernet driver supplied by Seneca free of charge.
With this driver a pair of virtual serial ports is created, by selecting these serial ports as you normally do, you get that the packets will no longer be sent via serial but via Ethernet. At this point the Seneca Gateway will convert the Ethernet traffic into serial through the real serial port, then the serial response will be reconverted to the Ethernet.



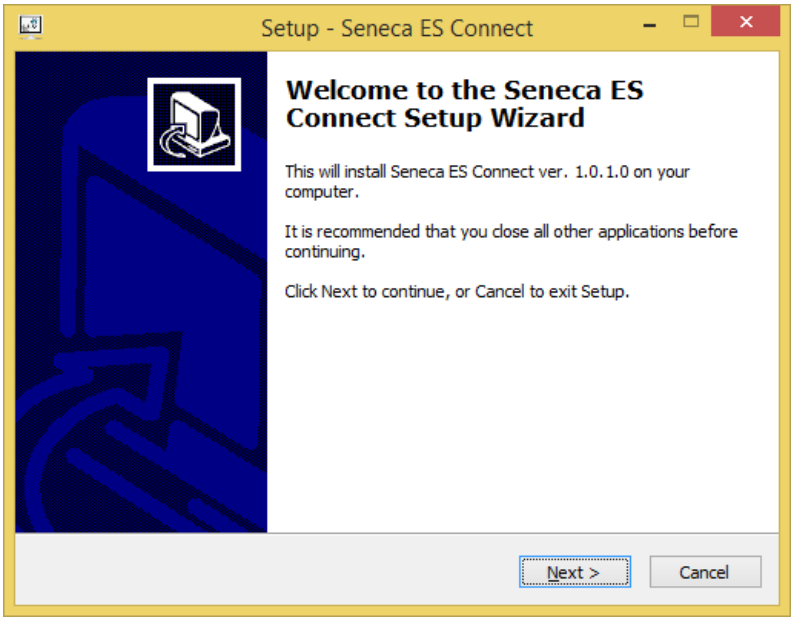
9.2. SENECA ETHERNET TO SERIAL CONNECT

As we have seen, to use the Gateway in serial device server mode or TCP server serial mode, the Seneca Ethernet to Serial Connect software must be installed on the PC.

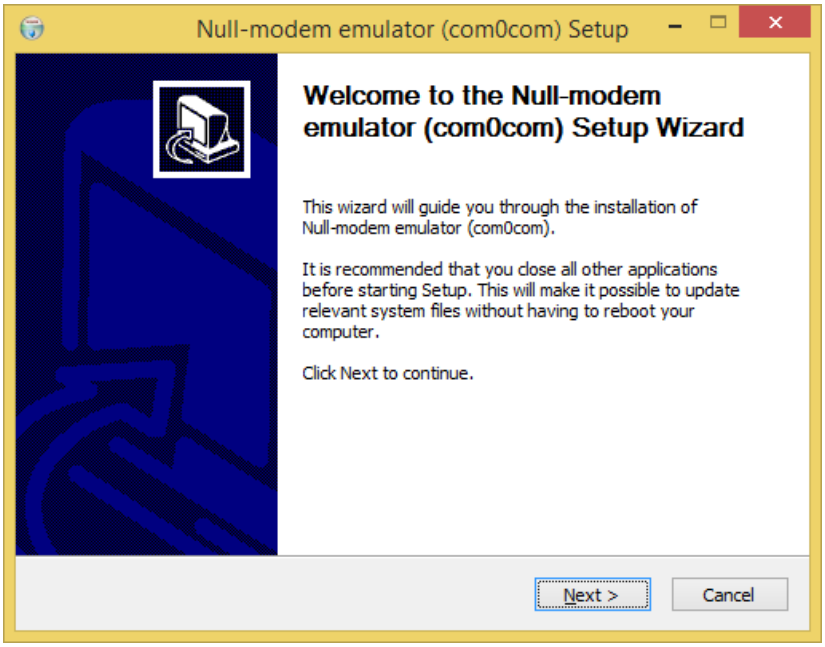
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

9.2.1. INSTALLING THE SENECA SERIAL TO ETHERNET CONNECT DRIVER

Seneca Ethernet to Serial Connect works on windows vista™, windows 7™, windows 8™, windows 10™ and windows 11™.
Double-click on the installation program:

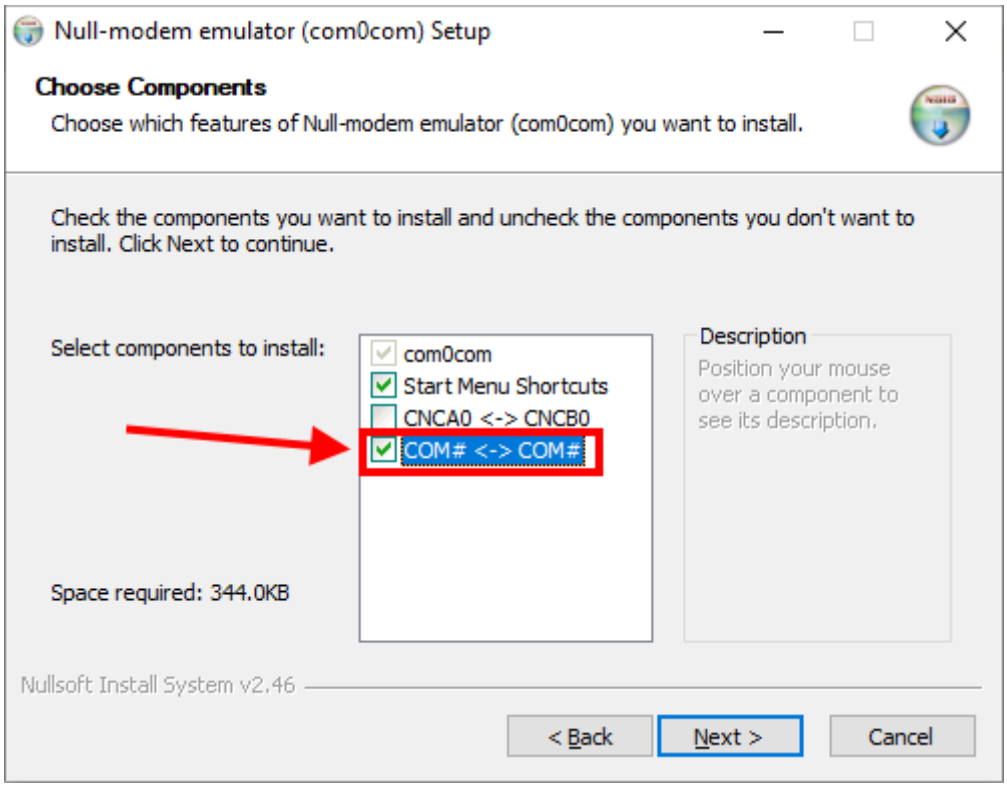


Then the com0com driver will be installed:



Select the names of the virtual ports COM#<->COM#:

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

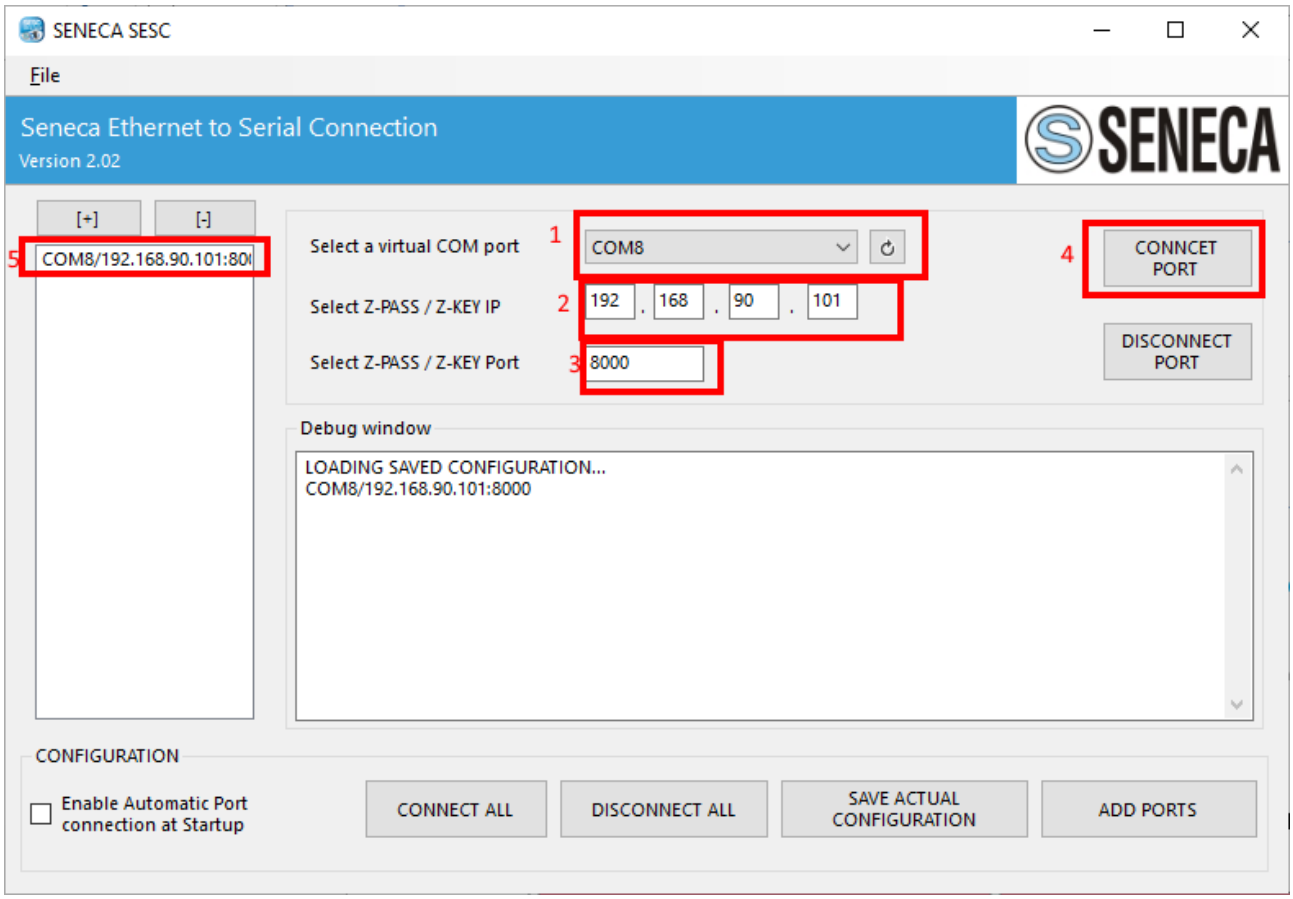


Wait for the installation to finish.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

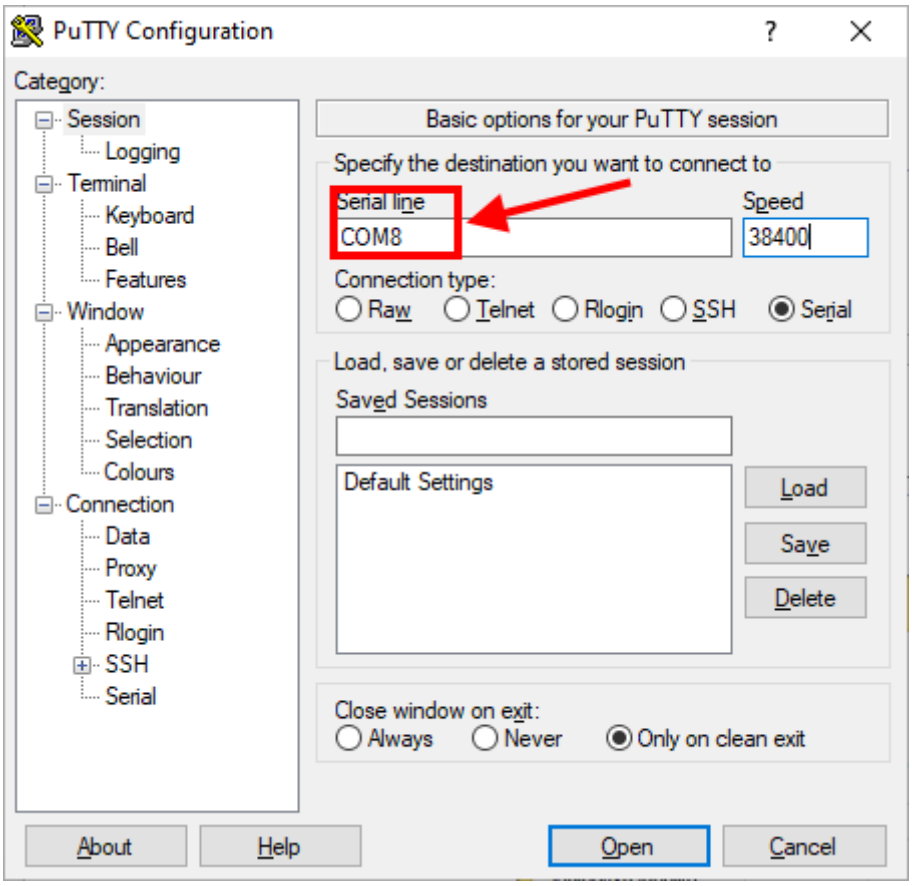
9.2.2. SELECT THE COM PORT FOR SENECA SERIAL TO ETHERNET CONNECT

Once the software has been installed, it can be launched (from the start menu -> Seneca -> Seneca Serial to Ethernet Connect").
The available virtual ports appear at point (1) (in our case the COM8 port is available).
On the SESC interface set the Gateway IP (2) and TCP port (3) configured on the Z-KEY via the web Server.
To start the connection, press the "Connect Port" button (4) COM8 is now connected to the Gateway via the TCP 8000 port:



Please note that Seneca SESC can connect more than one serial device at the same time by adding another port with the "ADD PORTS" button. Each new port needs another gateway, so 2 different gateways are required to connect two serial devices to the same PC.
Now use the same port (COM8 in our example) for the serial software:

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.



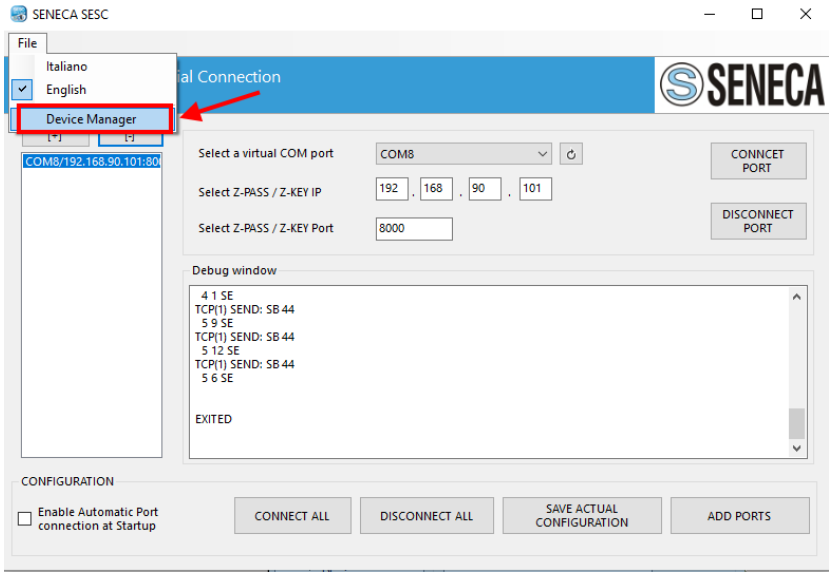
The other buttons allow you to:

Activate the connection (with the saved configuration) when starting the PC
Connect/Disconnect all configured ports
Save the current configuration.

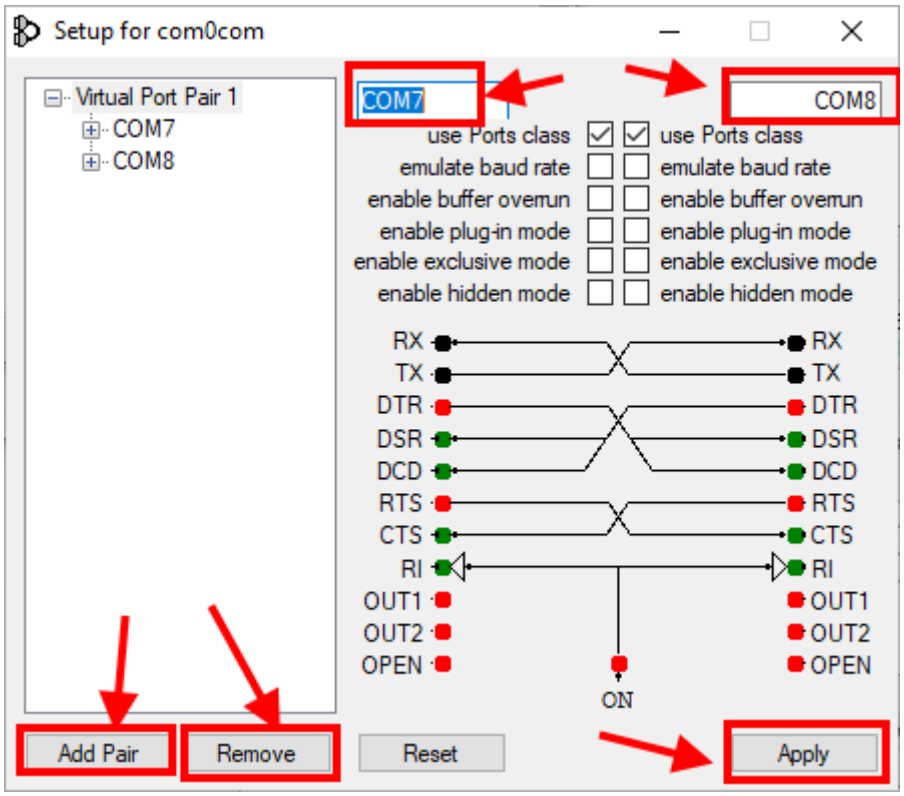
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

9.2.3. *MAINTENANCE OF VIRTUAL SERIAL PORTS*

To carry out maintenance of the virtual serial ports, access File->Device Manager:



At this point the com0com driver setup appears:



ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

Here you can:
- Rename virtual serials
- Add a pair of ports
- Remove a pair of ports

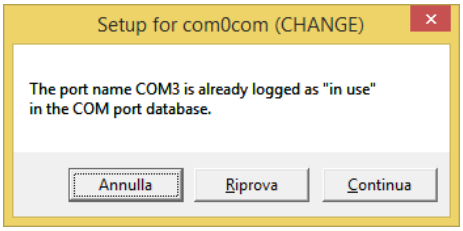
⚠ ATTENTION!

Com0Com always creates a pair of ports, the Seneca Ethernet to Serial Connect software only displays the one to be used for the connection (it is always the second).

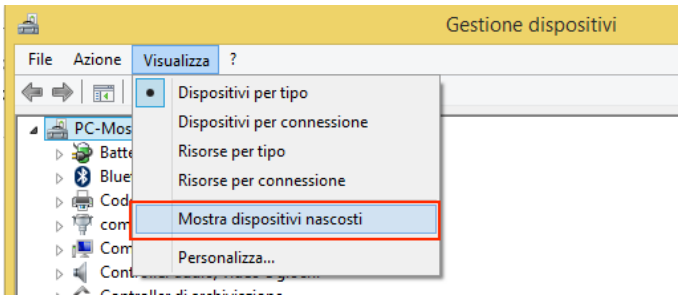
9.2.4. CHANGING THE NAME OF THE COM PORTS

Older software can only use a small range of COM ports (typically 1 to 9), so you may need to change the virtual COM number.

COM may sometimes be marked as "in use":

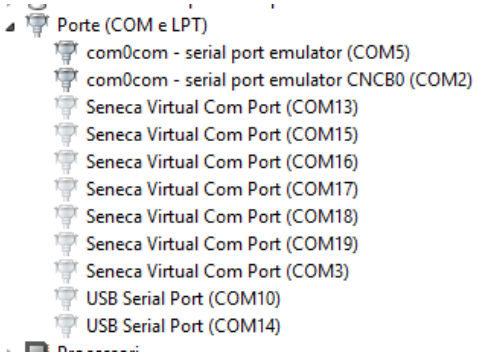


If you need to use this COM number, click on "Continue", then go to "Device Manager".
We must deselect the "in use" flag by uninstalling the port. Since the port is now disconnected, click on "Show hidden devices":



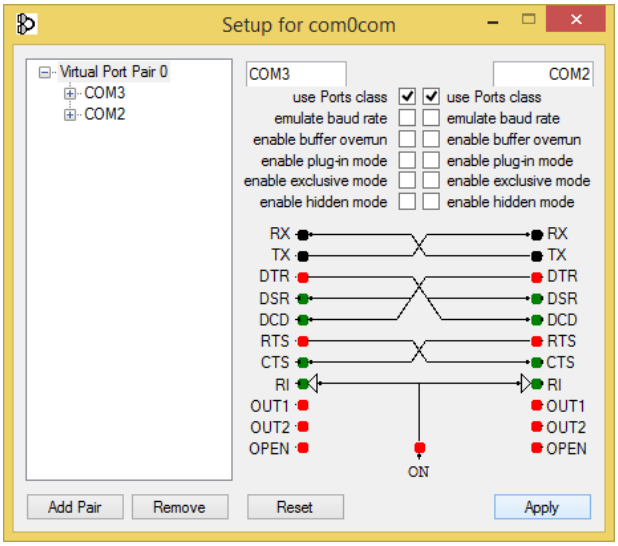
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

Now all the ports that are not in use are shown in transparency (also our COM3):

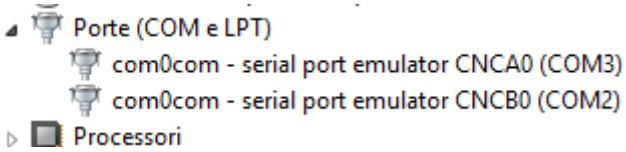


Now select the COM3 port and click on "Uninstall":

Now COM3 is free, and we can use it on the com0com setup:



Then click on "Apply", now the pair COM3, COM2 is created:

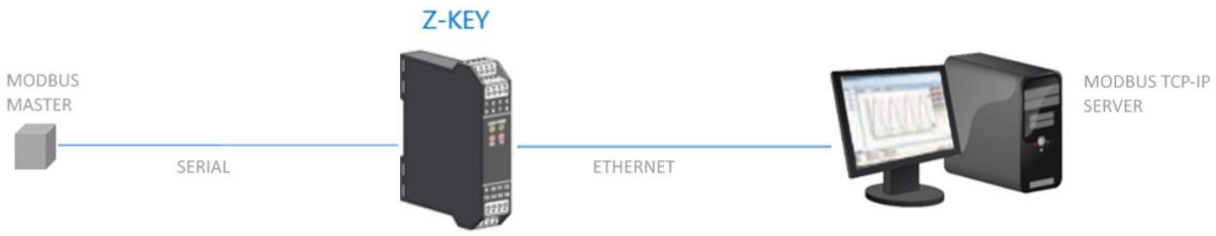


In the Seneca Ethernet to Serial Device software, the port on the right will appear, then COM2

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

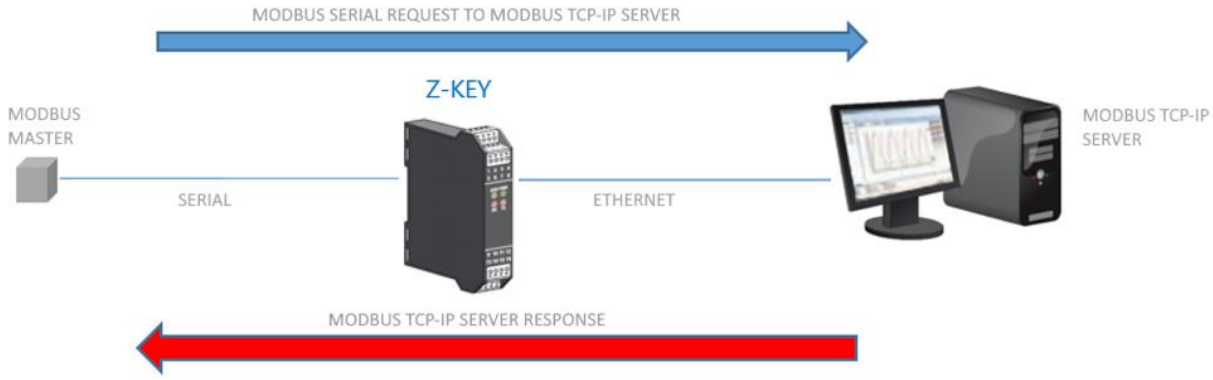
10. GATEWAY SERIAL TO ETHERNET MODE

In this scenario a Modbus serial Master must be connected to one or more Modbus TCP-IP servers. Typically, these are PLCs without an Ethernet port that must be connected to Modbus TCP servers.



10.1. HOW IT WORKS

This mode is the simplest way to create a communication between a serial Modbus Master device with one or more Modbus TCP-IP Servers.
It is not necessary to indicate which registers should be requested because the conversion from Serial to Ethernet is performed in real time and transparently.
The Gateway only requires the Network Configuration, the serial communication parameters (baud rate, parity) and the range of Modbus addresses managed by the single Server (because a server can manage multiple station addresses).



ATTENTION!

It is not possible to connect more than one TCP-IP Modbus server with the same ID Station address. If you need to connect multiple Modbus TCP-IP servers that respond to the same station ID, use the "Modbus Gateway Serial To Ethernet Virtual ID" mode

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

11. MODBUS GATEWAY SERIAL TO ETHERNET VIRTUAL ID MODE

This scenario is similar to the Modbus Gateway Serial to Ethernet mode but allows you to solve the problem (for example) where the Modbus TCP-IP servers all respond to the same station ID (for example 1 and it is not possible to change it).

11.1. HOW IT WORKS

In this mode it is possible to define an association between the station ID requests of the serial master in order to replace the station ID requests on the fly and divert them to another server.
Please refer to the following table:

VIRTUAL MODBUS ADDRESS	MODBUS TCP/IP SERVER	MODBUS ADDRESS
0	DISABLED	0
1	SERVER#1	1
2	SERVER#2	1
3	SERVER#3	1
4	DISABLED	0
5	DISABLED	0
6	DISABLED	0
7	DISABLED	0
8	DISABLED	0
9	DISABLED	0
10	DISABLED	0
11	DISABLED	0
12	DISABLED	0
13	DISABLED	0
14	DISABLED	0
15	DISABLED	0
16	DISABLED	0
17	DISABLED	0

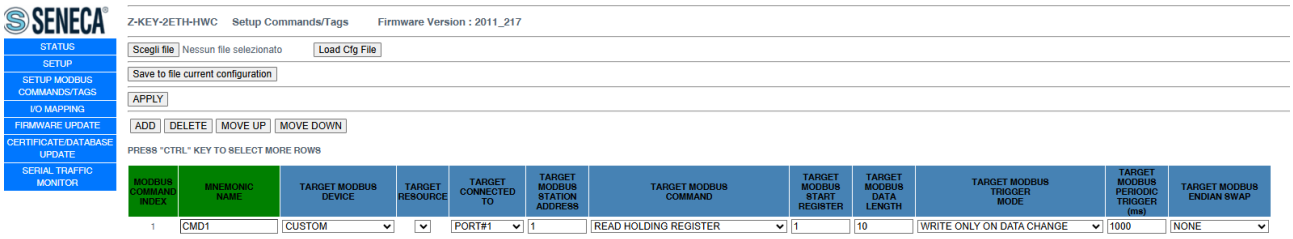
Serial requests with serial station ID 1 are sent with station ID 1 on the modbus tcp-IP server #1
Serial requests with serial station ID 2 are sent with station ID 1 on the modbus tcp-IP server #2
Serial requests with serial station ID 3 are sent with station ID 1 on the modbus tcp-IP server #3

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

12. MODBUS IO DEVICE MASTER MODE

This mode is very similar to the “Modbus Gateway with TAGs” with ports 1 and 2 masters but in this case the optimization of how many registers to request from the single device is done by the user. That is, there is no longer the concept of Tag but rather that of modbus request. This allows you to have absolute control over the serial side modbus commands (even just write commands without any reading for example).

12.1. HOW IT WORKS



The configuration is done by providing the modbus command and the number of registers to use.
Unlike the Tag mode, here the values are copied into the shared memory only as raw data.
However, it is also possible to define only write commands (for example, some slaves do not support read commands and therefore it would not be possible to use the Tag mode).
Up to 512 read bytes and 512 write bytes can be selected.
The reading area is available from modbus addresses with offset 0 to 255 and is accessible depending on the type of data read.
For example if we are reading holding register these will be available from register 40001 to register 40255, if we are reading input register these will be available from register 30001 to register 30255 etc...
The writing area is therefore queued from 40256 to 40511.

So we have that for holding registers:

40001 First 2 bytes read
40002 Second 2 bytes read
40003 Third 2 bytes read
….
40255 Last 2 bytes read

40256 First 2 bytes to write
40257 Second 2 bytes to write
40258 Third 2 bytes to write
….
40511 Last 2 bytes to write

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

The diagnostic registers are the same as in Tag Gateway mode.

13. MODBUS IO DEVICE SLAVE MODE

This mode is very similar to the “Modbus Gateway with TAGs” with slave ports 1 and 2.
In this mode the device provides two areas, one read-only and one write-only.
Each area consists of 512 bytes (256 modbus registers)

13.1. HOW IT WORKS

The reading area is available in the registers:

40001 … 40256 (Holding Registers)
1…256 (Coil)

The writing area is available in the registers:

40257 … 40512 (Holding Registers)
257…512 (Coil)

By writing to the write area, the value is readable in the read area from all ports.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

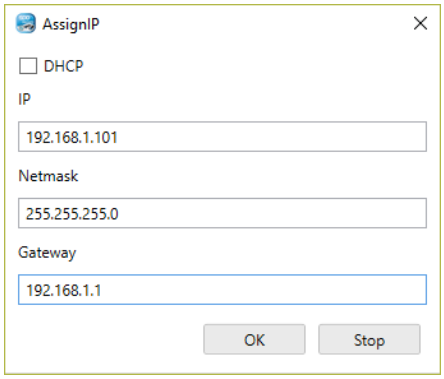
14. "-0" GATEWAY WEBSERVER

14.1. STEP BY STEP GUIDE FOR THE FIRST ACCESS TO THE WEBSERVER

STEP 1: POWER THE DEVICE AND CONNECT THE ETHERNET PORT

SENECA DISCOVERY DEVICE SOFTWARE STEP 2

If you need to change the IP address of the device (default 192.168.90.101), launch the Seneca Discovery Device software and perform the SCAN, select the device and press the "Assign IP" button, set a configuration compatible with your PC, for example:



Confirm with OK. Now the device can be reached via Ethernet from your PC.

STEP 3 ACCESS TO THE CONFIGURATION WEBSERVER

ENTER your access credentials:
user: admin
password: admin

⚠ ATTENTION!

THE WEB BROWSERS WHICH HAVE BEEN TESTED FOR COMPATIBILITY WITH THE DEVICE WEBSERVER ARE:
MOZILLA FIREFOX AND GOOGLE CHROME.
THEREFORE, THE OPERATION WITH OTHER BROWSERS IS NOT GUARANTEED

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15. WEBSERVER DEVICE CONFIGURATION

ATTENTION!

THE WEB BROWSERS WHICH HAVE BEEN TESTED FOR COMPATIBILITY WITH THE DEVICE WEBSERVER ARE:
MOZILLA FIREFOX AND GOOGLE CHROME.
THEREFORE, THE OPERATION WITH OTHER BROWSERS IS NOT GUARANTEED

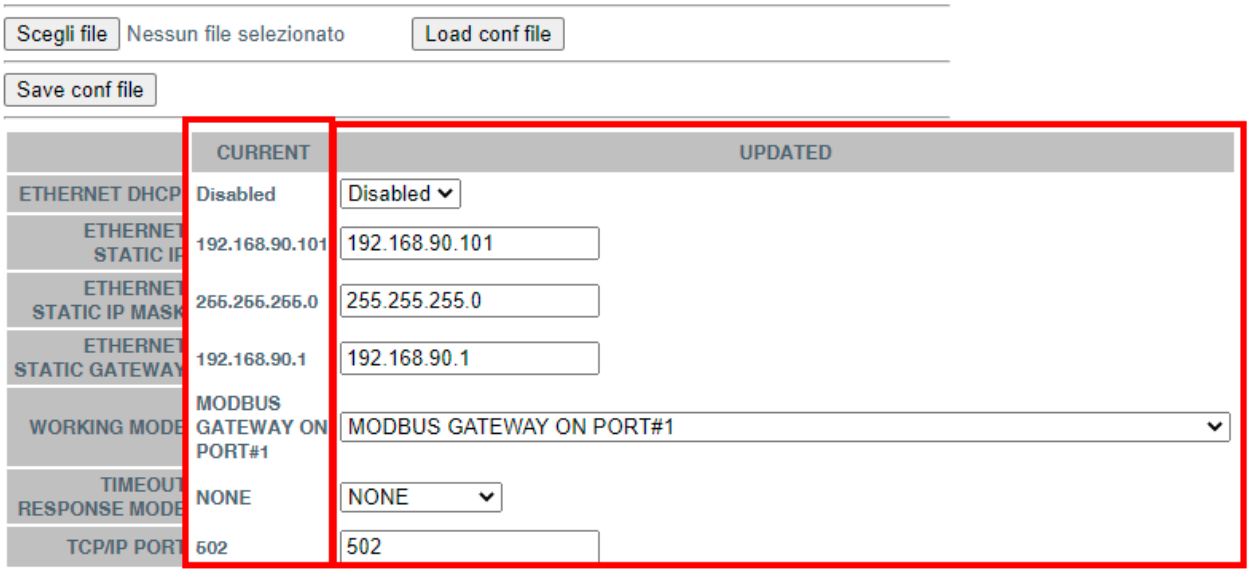
ATTENTION!

AFTER THE FIRST ACCESS CHANGE USER NAME AND PASSWORD IN ORDER TO PREVENT ACCESS TO THE DEVICE TO UNAUTHORIZED PEOPLE.

ATTENTION!

IF THE PARAMETERS TO ACCESS THE WEBSERVER HAVE BEEN LOST, TO ACCESS IT, IT IS NECESSARY TO GO THROUGH THE PROCEDURE TO RESET THE FACTORY-SET CONFIGURATION

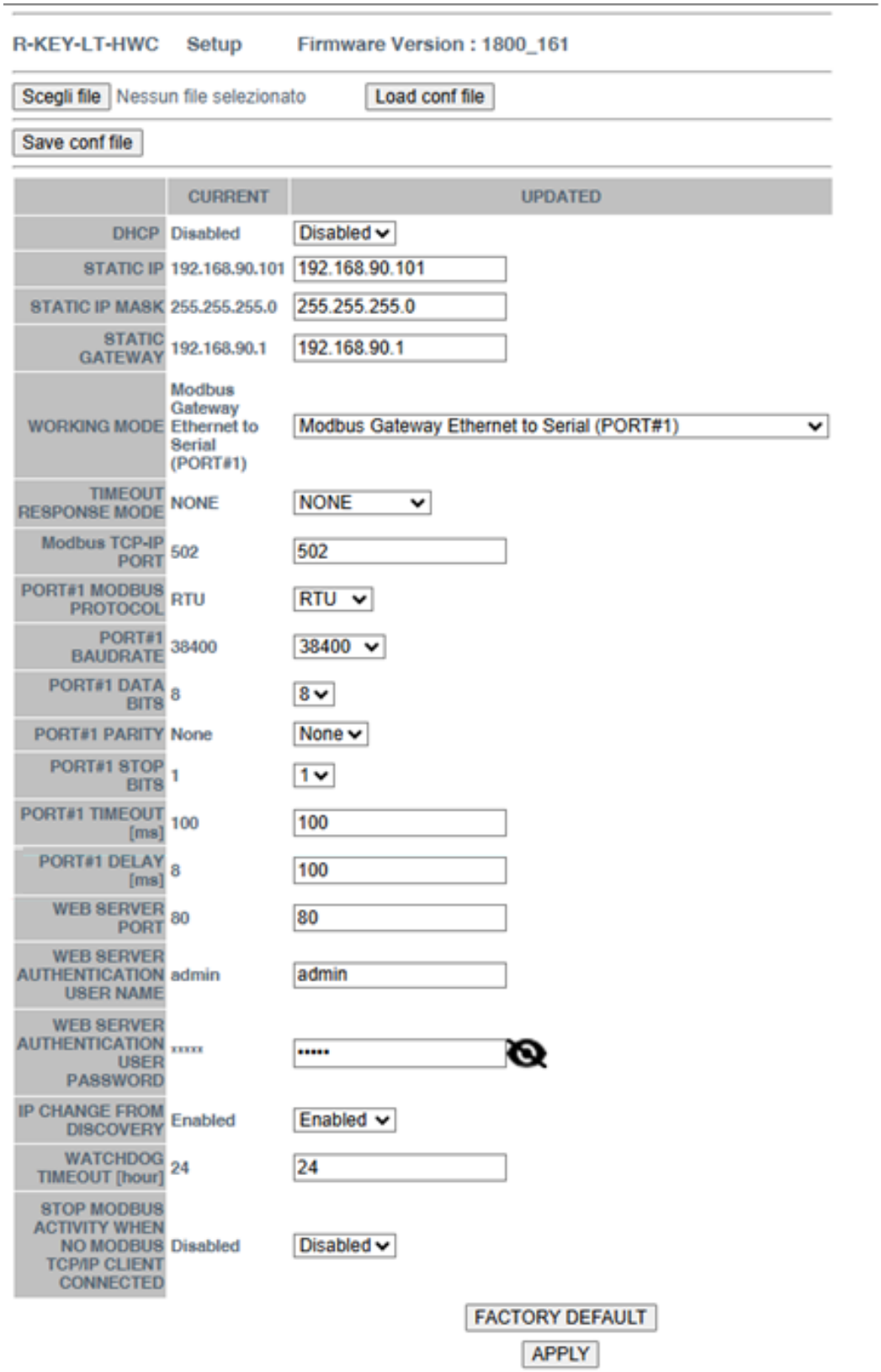
15.1. SETUP PAGE



The first column represents the name of the parameter, the second column “current” is the current value of the parameter. The last column “updated” is used to modify the current configuration.
When a configuration has been entered it is necessary to confirm it with the "APPLY" button, at this point the new configuration is operational.
If you want to restore the default parameters, click on the "FACTORY DEFAULT" button.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15.1.1. GENERAL CONFIGURATION PARAMETERS



R-KEY-LT-HWC Setup Firmware Version : 1800_161

Scegli file | Nessun file selezionato | Load conf file

Save conf file

	CURRENT	UPDATED
DHCP	Disabled	Disabled
STATIC IP	192.168.90.101	192.168.90.101
STATIC IP MASK	255.255.255.0	255.255.255.0
STATIC GATEWAY	192.168.90.1	192.168.90.1
WORKING MODE	Modbus Gateway Ethernet to Serial (PORT#1)	Modbus Gateway Ethernet to Serial (PORT#1)
TIMEOUT RESPONSE MODE	NONE	NONE
Modbus TCP-IP PORT	502	502
PORT#1 MODBUS PROTOCOL	RTU	RTU
PORT#1 BAUDRATE	38400	38400
PORT#1 DATA BITS	8	8
PORT#1 PARITY	None	None
PORT#1 STOP BITS	1	1
PORT#1 TIMEOUT [ms]	100	100
PORT#1 DELAY [ms]	8	100
WEB SERVER PORT	80	80
WEB SERVER AUTHENTICATION USER NAME	admin	admin
WEB SERVER AUTHENTICATION USER PASSWORD	*****	*****
IP CHANGE FROM DISCOVERY	Enabled	Enabled
WATCHDOG TIMEOUT [hour]	24	24
STOP MODBUS ACTIVITY WHEN NO MODBUS TCP/IP CLIENT CONNECTED	Disabled	Disabled

FACTORY DEFAULT

APPLY

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

The general configuration parameters are explained below:

DHCP
Disabled: A static Network Configuration is set up
Enabled: The IP address, IP mask and gateway address are obtained from the DHCP server.
The gateway address can be found by the Seneca Discovery Device software.

ETHERNET STATIC IP
Static IP address when the DHCP is disabled

ETHERNET STATIC IP MASK
Mask when the DHCP is disabled

ETHERNET STATIC GATEWAY
Gateway address when the DHCP is disabled

WORKING MODE
It selects the operating mode of the Modbus Gateway:

- *Modbus Gateway Ethernet to Serial (PORT#1)*
- *Modbus Gateway Ethernet to Serial (PORT#2) (2 serial ports models only)*
- *Modbus Gateway Ethernet to Serial (PORT#1 AND PORT#2) (2 serial ports models only)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 MASTER)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 MASTER PORT#2 SLAVE) (2 serial ports models only)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 SLAVE PORT#2 MASTER) (2 serial ports models only)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 SLAVE)*
- *Serial device server (PORT #1)*
- *Serial device server (PORT #2) (2 serial ports models only)*
- *Modbus Gateway Serial to Ethernet (PORT#1 AND PORT#2)*
- *TCP Server Serial Device (PORT #1)*
- *TCP Server Serial Device (PORT #2)*
- *Modbus Gateway Serial to Ethernet Virtual ID (PORT#1 AND PORT#2)*
- *Modbus IO Device Master*
- *Modbus IO Device Slave*

TCP/IP PORT
TCP-IP port for Modbus TCP-IP Server protocol (Up to a maximum of 8 clients can be connected to the gateway)

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

PORT#n MODBUS PROTOCOL
Select the Modbus RTU or Modbus ASCII serial protocol

PORT#n BAUDRATE
Select the baudrate of the serial port

PORT#n BIT
Select the number of bits for the serial communication.

PORT#n PARITY
Select the type of parity of the serial port (None, Even or Odd)

PORT#n STOP BITS
Set the number of stop bits of the port (1 or 2), note that if parity is set, only 1 stop bit can be used.

PORT#n TIMEOUT [ms]
Set the waiting time for a response from the Modbus slave serial device, after this time without any response there will be a TIMEOUT.

PORT#n DELAY [ms](Only for Serial Device Server mode)
Allows you to manually change the serial #n silence time between the response of a slave modbus and the next request. Each time the parameter "baud rate" changes, the value will be overwritten by the minimum value recommended. If you use slow slave modbus with more than one client modbus TCP/IP connected to the device, you will need to manually intervene on this parameter (for example, bringing it to 50 ms).

PORT#n DELAY BETWEEN POLLS [ms] (Only for Gateway Tags Modbus mode)
Set the pause between two successive serial Modbus master requests.

PORT#n WRITING RETRIES (Only for Gateway Tags Modbus mode)
Set the number of attempts to write to the TAG(s) before setting the FAIL status.

PORT#n MAX READ NUM (Only for Gateway Tags Modbus mode)
Set the maximum number of registers that can be read with the multiple reading functions (the gateway will optimize readings with this maximum number of registers). It must be adjusted according to the maximum number of registers that can be read at the same time by the slave device.

PORT#1 MAX WRITE NUM (Only for Gateway Tags Modbus mode)
Set the maximum number of registers that can be written with the multiple writing functions (the gateway will optimize writings with this maximum number of registers).

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

WEB SERVER PORT
Set the TCP-IP port for the Webserver.

WEB SERVER AUTHENTICATION USERNAME
Set the username for accessing the Webserver (if the user name and password are left blank, no authentication is required to access the Webserver)

WEB SERVER AUTHENTICATION PASSWORD
Set the password for accessing the Webserver (if the username and password are left blank, no authentication is required to access the Webserver)

⚠ ATTENTION!
CHANGE THE DEFAULT USERNAME AND PASSWORD IN THE WEBSERVER TO RESTRICT ACCESS.

⚠ ATTENTION!

IF THE TWO PARAMETER TEXT BOXES ARE LEFT EMPTY, THE AUTHENTICATION FOR ACCESS IS REMOVED.

WEBSERVER HTTPS
It forces the webserver to use the https secure protocol instead of http one

ETHERNET IP CHANGE FROM DISCOVERY
Set whether a user is authorized to change the IP configuration from the "Seneca Discovery Device" software.

DIAGNOSTIC REGISTERS MAPPING (Only for Gateway Tags Modbus mode)
Set the type of register that will contain simplified and advanced diagnostics. It is possible to select between holding registers or input registers.

***DIAGNOSTIC REGISTER START ADDRESS* (only for Gateway Tags Modbus mode)**
Set the starting address for the diagnostic registers (default offset 9000 -> 49001 in case of holding registers or 39001 in case of input registers)

PORT #n TAGS QUARANTINE [s] PORT (Only for Gateway Tags Modbus mode)
When a TAG is in FAIL it is placed in quarantine and is no longer interrogated for the set time.

MODBUS TCP-IP CLIENT (Only for Gateway Tags Modbus mode)
Enable or not the Modbus TCP-IP clients, the gateway can connect to a maximum of 10 Modbus TCP-IP servers.

TCP-IP PORT SERVER #n (Only if Modbus TCP-IP client or Gateway from Serial to Ethernet are active)
Used to set the TCP-IP server port #n

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

TCP-IP ADDRESS SERVER #n (Only if Modbus TCP-IP client or Gateway from Serial to Ethernet are active)

Used to set the IP address of the #n server

MODBUS TCP-IP CLIENT TIMEOUT [ms] (Only if Modbus TCP-IP client or Gateway from Serial to Ethernet are active)

Used to set the connection time out for Modbus TCP-IP clients.

MODBUS TCP-IP CLIENT DELAY BETWEEN POLLS [ms] (Only if Modbus TCP-IP client is active)

Set the pause between two successive Modbus TCP-IP client requests.

MODBUS TCP-IP CLIENT WRITING RETRIES (Only if Modbus TCP-IP client is active)

Set the number of attempts to write to the TAG(s) before setting the FAIL status.

MODBUS TCP-IP CLIENT MAX READ NUM (Only if Modbus TCP-IP client is active)

Set the maximum number of registers that can be read with the multiple reading functions (the gateway will optimize readings with this maximum number of registers).

MODBUS TCP-IP CLIENT MAX WRITE NUM (Only if Modbus TCP-IP client is active)

Set the maximum number of registers that can be written with the multiple writing functions (the gateway will optimize writings with this maximum number of registers).

SERVER#n START/LAST SLAVE ADDRESS (Only if the Gateway from Serial to Ethernet mode is active)

Used to connect more than one Modbus TCP-IP server to the Modbus serial Master, if the request is in range (server start address/last slave address), the packet is sent to the appropriate server:
For example:
In this configuration:
server#1 has a slave field start = 1 and last = 10
server#2 has a slave field start = 11 and last = 20
if the serial master requests the slave address from 1 to 10 then the packet is sent to server#1
if the serial master requires the slave address from 11 to 20 then the packet is sent to server#2

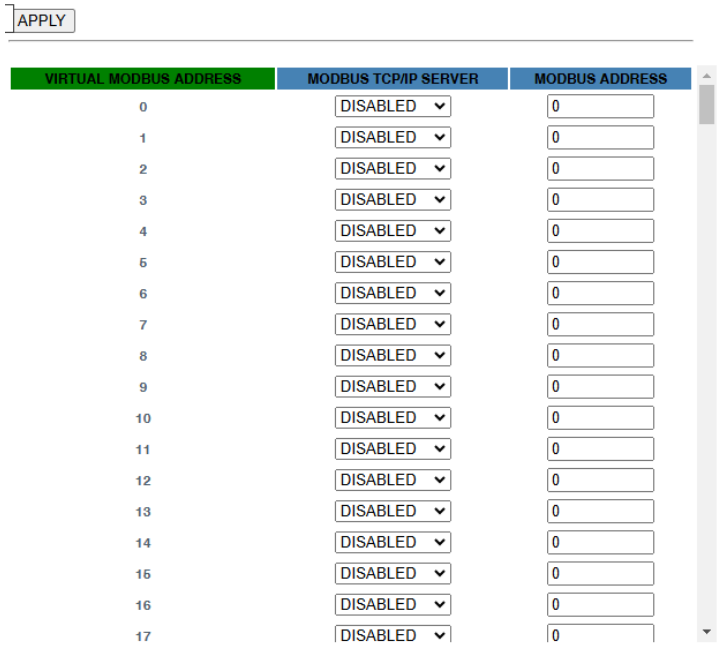
WATCHDOG ENABLE
Enable or disable the time restart of gateway.

WATCHDOG TIMEOUT [hours]
Sets the time in hours after which the gateway will reboot (only if the WATCHDOG ENABLE parameter is enabled).

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15.2. VIRTUAL ADDRESSES SETUP PAGE

Allows you to set the conversion table for virtual addresses (only if the Modbus Gateway Serial to Ethernet Virtual ID mode is selected).



The first Column “Virtual Modbus Address” contains the virtual station address of master request.
In the second Column “Modbus TCP/IP Server” the Modbus TCP-IP server to which the request should be sent is selectable.
The third column “Modbus Address” shows the station address to which that Modbus TCP-IP server responds (usually 1).

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15.3. SETUP TAG PAGE (MODBUS TAGS GATEWAY MODE ONLY)

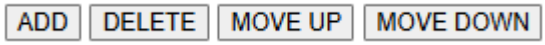
In Modbus Tags Gateway mode it is necessary to define the Modbus tags (i.e. variables), to do this it is possible to use:

- *The webserver*
- *An Excel template downloadable from the Seneca website*

In this chapter we will explain the configuration of the tag from the webserver.
To edit the TAGs via webserver, access the "Setup tag" section of the navigation menu:

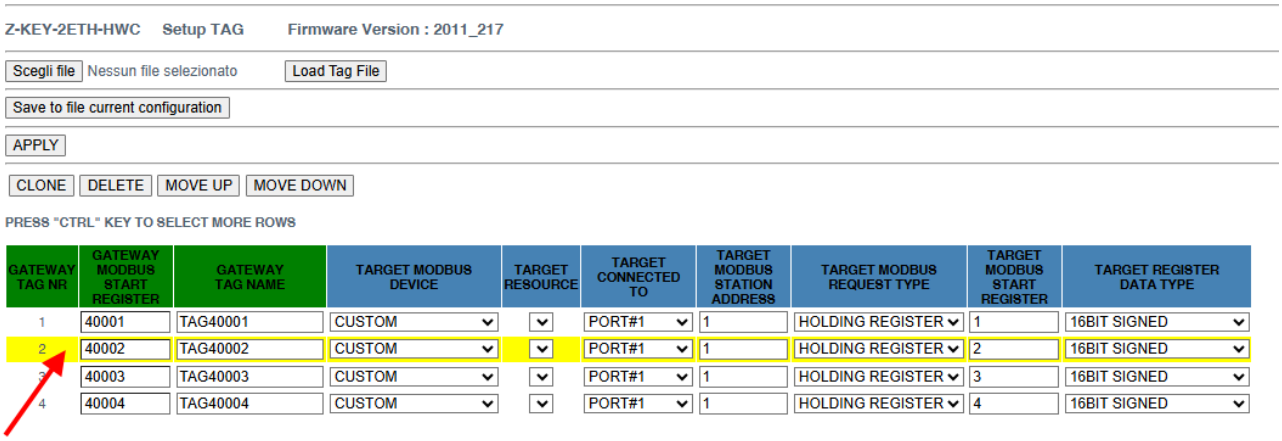
15.3.1. BUTTONS AND TAG CLONE

Using the buttons:

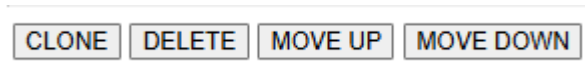


It is possible to add, delete and move a Tag from the list.

It is possible to select a row using the mouse:



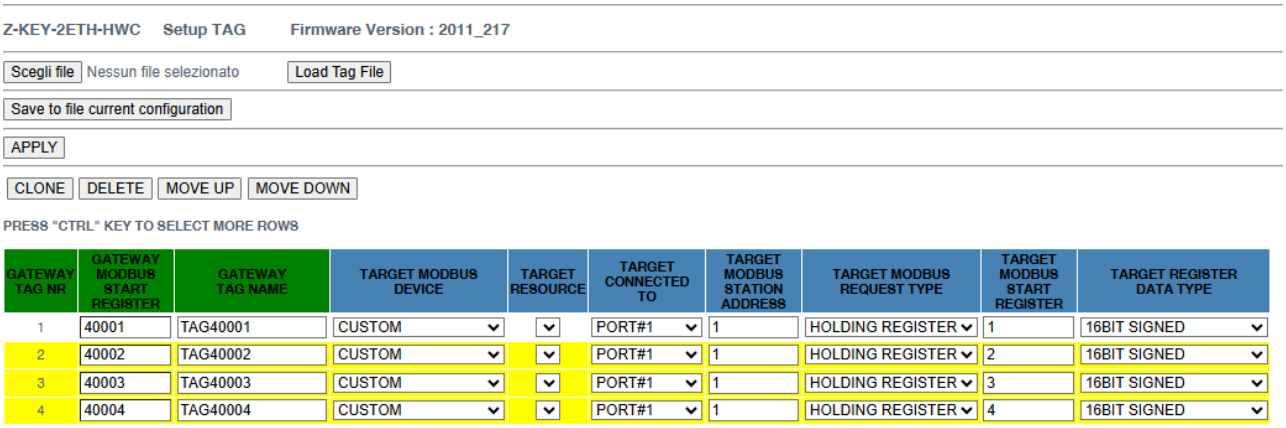
At this point the buttons change to:



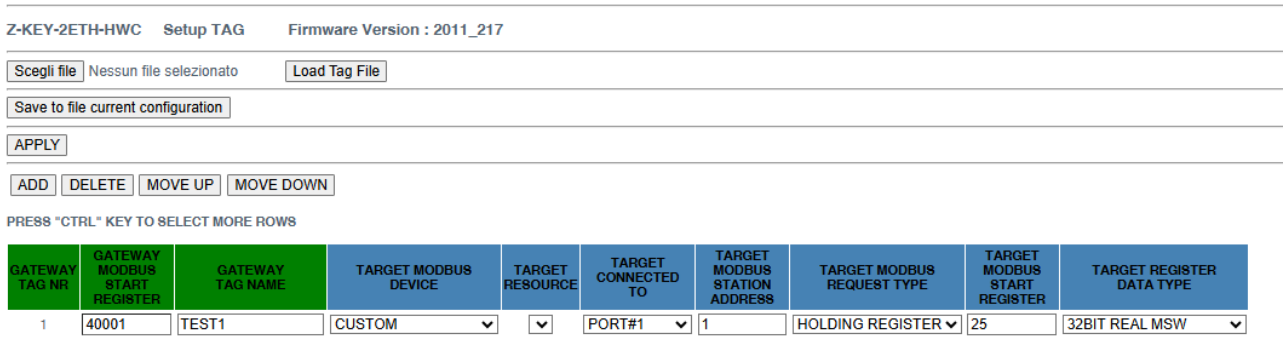
That is, “ADD” is replaced by “CLONE”, by pressing the “CLONE” button the selected TAG is cloned, the position in the modbus memory is automatically calculated based on the type of register.
To return to the starting mode, simply press the mouse again on the selected row.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

By holding down the CTRL button on the keyboard, it is possible to select and clone multiple TAGs at the same time:



15.3.2.*FIELDS MEANING*



GATEWAY MODBUS START REGISTER ADDRESS

Set the address of the Gateway memory location where the TAG is saved, these registers are accessible both from Modbus serial and Modbus TCP-IP.

GATEWAY TAG NAME

Set the mnemonic name of the tag (it will be displayed in the live view)

TARGET MODBUS DEVICE

Select the Modbus RTU slave model from the Seneca device database or select "custom" if you are not using a Seneca Modbus RTU slave.

TARGET RESOURCE

If you are using a Seneca Modbus RTU Slave device select the resource name from the Seneca database.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

TARGET CONNECTED TO PORT#

Select which serial port of the gateway the Modbus RTU slave device is connected to.
(in the case of R-KEY-LT only the COM 1 port is available). Here you can also define a remote Modbus TCP-IP server.

TARGET MODBUS STATION ADDRESS

Defines the Modbus Station Address (also called the Modbus node address) of the slave device.

TARGET MODBUS START REGISTER ADDRESS

Defines the starting register of the TAG to be acquired by the Modbus RTU slave.

TARGET MODBUS REQUEST TYPE

Select the type of Modbus register:
Coil
Discrete Input
Holding Register
Input Register

TARGET REGISTER DATA

Select the type of TAG variable:
16 BIT UNSIGNED: 1 Modbus register, from 0 to 65535
16 BIT SIGNED: 1 Modbus register, from -32768 to +32767
32 BIT UNSIGNED MSW: 2 Modbus registers, whose Modbus register with the lower address contains the most significant word, can assume values from 0 to 4294967295
32 BIT UNSIGNED LSW: 2 Modbus registers, whose Modbus register with the lower address contains the least significant word, can assume values from 0 to 4294967295
32 BIT SIGNED MSW: 2 Modbus registers, whose Modbus register with the lower address contains the most significant word, can assume values from -2147483648 to +2147483647
32 BIT SIGNED LSW: 2 Modbus registers whose Modbus, register with the lower address contains the least significant word, can assume values from -2147483648 to +2147483647
FLOAT MSW: 2 Modbus registers, whose Modbus register with the lower address contains the most significant word, single precision floating point value (IEEE 758-2008)
FLOAT LSW: 2 Modbus registers, whose Modbus register with the lower address contains the least significant word, single precision floating point value (IEEE 758-2008)
64 BIT SIGNED MSW: 4 Modbus registers, whose Modbus register with the lower address contains the most significant word, can assume values from -9223372036854775808 to +9223372036854775807
64 BIT SIGNED LSW: 4 Modbus registers whose Modbus, register with the lower address contains the least significant word, can assume values from -9223372036854775808 to +9223372036854775807
64 BIT UNSIGNED MSW: 4 Modbus registers, whose Modbus register with the lower address contains the most significant word, can assume values from 0 to 18,446,744,073,709,551,615
64 BIT UNSIGNED LSW: 4 Modbus registers whose Modbus, register with the lower address contains the least significant word, can assume values from 0 to 18,446,744,073,709,551,615
BIT: 1 Boolean Coil or Discrete Input, value true or false.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

N.B. This field is automatically filled in if a Seneca slave device has been selected in the "TARGET MODBUS DEVICE" field.

⚠ ATTENTION!

All 32-bit values are stored in 2 consecutive registers, for example:
The 32-bit unsigned MSW TAG 1 Totalizer is stored in the addresses 40016 and 40017:
The most significant word is 40016, the least significant is 40017.
So the 32bit value is obtained from the following relationship:

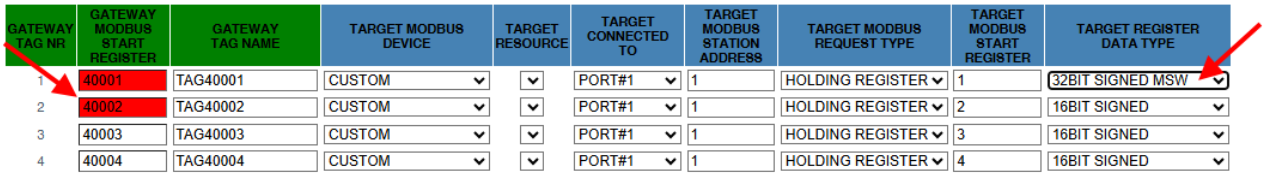
$$Totalizer1 = (40017) + (Reg\ (40016) \times 65536)$$

Tag setup can be imported/exported from/to a ".cgi" file:

Note that a .cgi file can also be imported/exported from/to the Excel template.

If the values entered are incorrect, the relative field is selected in red.
Once all the values have been entered, the fields are saved with the “APPLY” button.

For example, the system signals an error (in red) even when the registers overlap as in this case:



GATEWAY TAG NR	GATEWAY MODBUS START REGISTER	GATEWAY TAG NAME	TARGET MODBUS DEVICE	TARGET RESOURCE	TARGET CONNECTED TO	TARGET MODBUS STATION ADDRESS	TARGET MODBUS REQUEST TYPE	TARGET MODBUS START REGISTER	TARGET REGISTER DATA TYPE
1	40001	TAG40001	CUSTOM		PORT#1	1	HOLDING REGISTER	1	32BIT SIGNED MSW
2	40002	TAG40002	CUSTOM		PORT#1	1	HOLDING REGISTER	2	16BIT SIGNED
3	40003	TAG40003	CUSTOM		PORT#1	1	HOLDING REGISTER	3	16BIT SIGNED
4	40004	TAG40004	CUSTOM		PORT#1	1	HOLDING REGISTER	4	16BIT SIGNED

The 32-bit integer TAG occupies 2 modbus registers (40001-40002 in this case) while the next tag is incorrectly set to 40002.
To make room for the first tag, all the others must be moved:

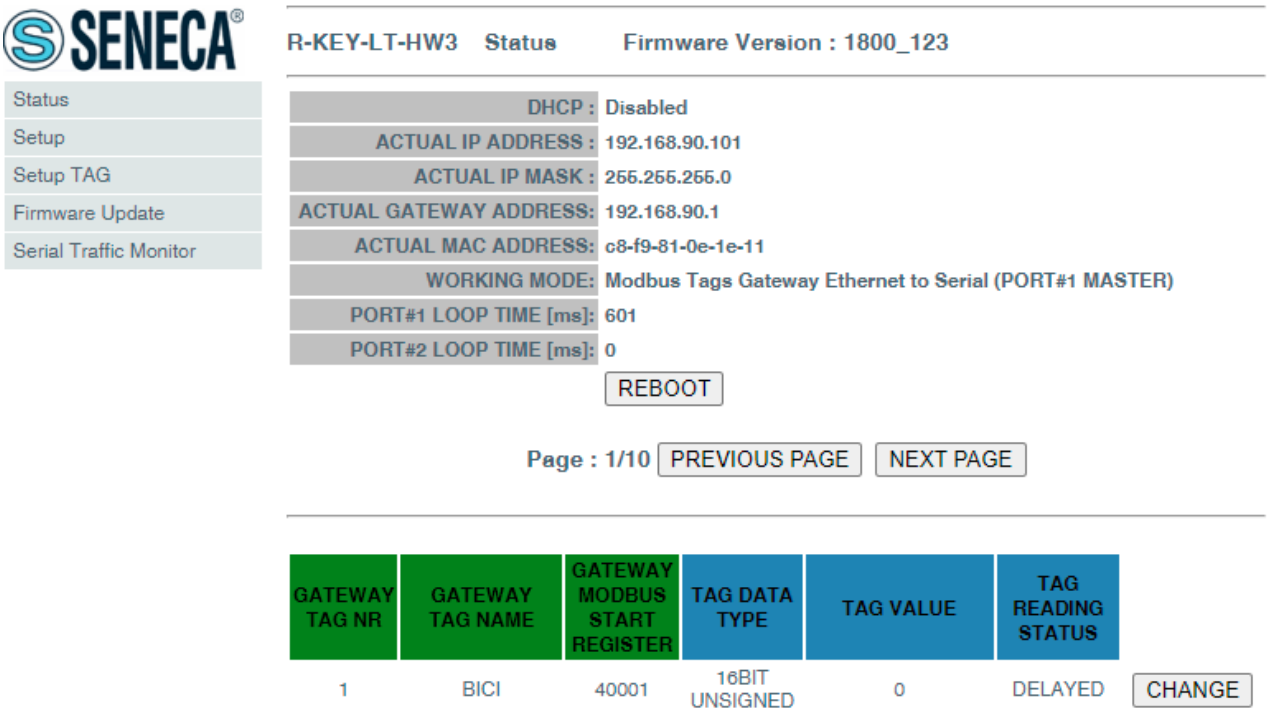
GATEWAY TAG NR	GATEWAY MODBUS START REGISTER	GATEWAY TAG NAME	TARGET MODBUS DEVICE	TARGET RESOURCE	TARGET CONNECTED TO	TARGET MODBUS STATION ADDRESS	TARGET MODBUS REQUEST TYPE	TARGET MODBUS START REGISTER	TARGET REGISTER DATA TYPE
1	40001	TAG40001	CUSTOM		PORT#1	1	HOLDING REGISTER	1	32BIT SIGNED MSW
2	40003	TAG40002	CUSTOM		PORT#1	1	HOLDING REGISTER	2	16BIT SIGNED
3	40004	TAG40003	CUSTOM		PORT#1	1	HOLDING REGISTER	3	16BIT SIGNED
4	40005	TAG40004	CUSTOM		PORT#1	1	HOLDING REGISTER	4	16BIT SIGNED

And in fact, the system no longer signals the error.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15.3.3.*REAL-TIME VIEW OF THE MODBUS GATEWAY*

Once the TAGs are configured, it is possible to view the status of the Modbus communication in real time, from the Status section of the navigation menu.
The live view will show the current network configuration, operation mode and TAG information.



Tag information includes: The name of the TAG, the Modbus address of the TAG Gateway, the value of the Tag and the status of the TAG:

OK = TAG free of errors
FAIL_TO = TAG reading time out
DELAYED = Once the set retry number has been reached, the polling of the tag is delayed (the tag will be interrogated again after the configured quarantine time)
EXC = Modbus protocol exception response

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15.4. TAG/MODBUS COMMAND SETUP PAGE (FOR MODBUS IO DEVICE MASTER MODE ONLY)

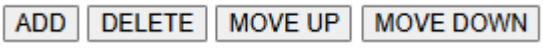
In Modbus IO Device Master mode it is necessary to define the Modbus commands, to do this it is possible to use:

- *The webserver*
- *An Excel template downloadable from the Seneca website*

In this chapter we will explain the configuration of the commands from the webserver.
To edit the commands via webserver, access the "Setup tag" section of the navigation menu:

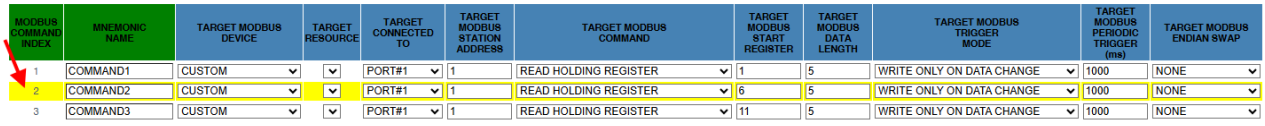
15.4.1. BUTTONS AND COMMAND CLONE

Using the buttons:



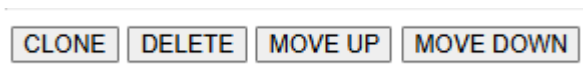
It is possible to add, delete and move a command from the list.

It is possible to select a row using the mouse:



MODBUS COMMAND INDEX	MNEMONIC NAME	TARGET MODBUS DEVICE	TARGET RESOURCE	TARGET CONNECTED TO	TARGET MODBUS STATION ADDRESS	TARGET MODBUS COMMAND	TARGET MODBUS START REGISTER	TARGET MODBUS DATA LENGTH	TARGET MODBUS TRIGGER MODE	TARGET MODBUS PERIODIC TRIGGER (ms)	TARGET MODBUS ENDIAN SWAP
1	COMMAND1	CUSTOM		PORT#1	1	READ HOLDING REGISTER	1	5	WRITE ONLY ON DATA CHANGE	1000	NONE
2	COMMAND2	CUSTOM		PORT#1	1	READ HOLDING REGISTER	6	5	WRITE ONLY ON DATA CHANGE	1000	NONE
3	COMMAND3	CUSTOM		PORT#1	1	READ HOLDING REGISTER	11	5	WRITE ONLY ON DATA CHANGE	1000	NONE

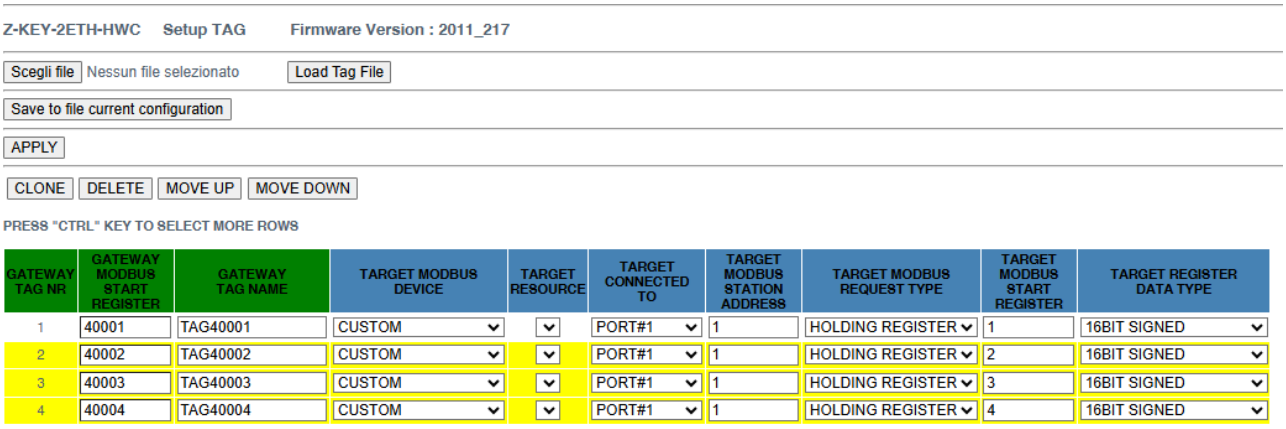
At this point the buttons change to:



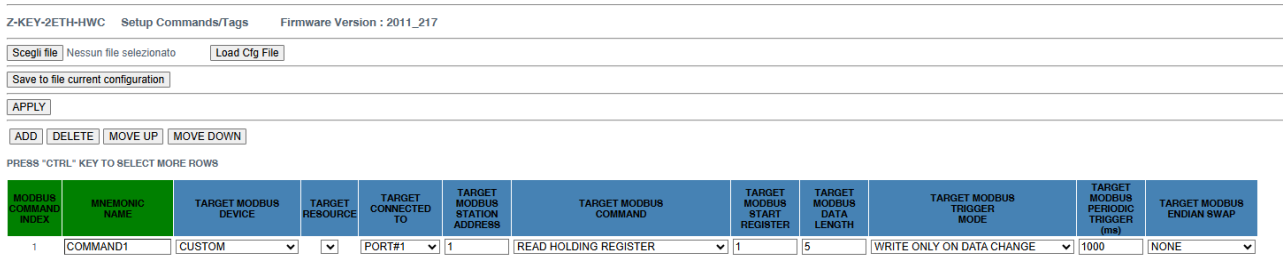
That is, “ADD” is replaced by “CLONE”, by pressing the “CLONE” button the selected command is cloned.
To return to the starting mode, simply press the mouse again on the selected row.

By holding down the CTRL button on the keyboard, it is possible to select and clone multiple commands at the same time:

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.



15.4.2.*FIELDS MEANING*



MNEMONIC NAME

It sets the mnemonic name of the command

TARGET MODBUS DEVICE

It selects the Modbus RTU slave model from the Seneca device database or selects "custom" if you want to use a command that affects multiple Modbus registers at the same time (in example 5).

TARGET RESOURCE

If you are using a Seneca Modbus RTU Slave device select the resource name from the Seneca database.

TARGET CONNECTED TO PORT#

Select which serial port of the gateway the Modbus RTU slave device is connected to.
(in the case of R-KEY-LT only the COM 1 port is available). Here you can also define a remote Modbus TCP-IP server.

TARGET MODBUS STATION ADDRESS

Defines the Modbus Station Address (also called the Modbus node address) of the slave device.

TARGET MODBUS COMMAND

Selects the type of Modbus command to use between:

Read Coils

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

Write Single Coil
Write Multiple Coil
Read Discrete Inputs
Read Holding Register
Write Single Holding Register
Write Multiple Holding Registers
Read Input Register

TARGET MODBUS START REGISTER ADDRESS
Defines the starting register of the Modbus RTU command.

TARGET MODBUS DATA LENGTH
Defines the registers number of the Modbus RTU command

TARGET MODBUS TRIGGER MODE
In the case of a write command, it allows you to select when the command should occur. In the case of a read command, it has no effect.

WRITE ONLY ON DATA CHANGE It executes the command only if the data contained in the command have changed since the last writing.
WRITE ONLY PERIODIC It executes the time write command based on the value of the "Target Modbus Periodic Trigger" parameter.
WRITE PERIODIC OR DATA CHANGE It executes the time write command or if the value has changed since the last writing.

TARGET MODBUS PERIODIC TRIGGER (ms)
It represents the execution time of the write command in [ms]. It has no effect in case of a read command.

TARGET MODBUS ENDIAN SWAP
It changes the order of data saved in memory:

BYTE: It swaps the value of the most significant byte (8 bits) with the least significant one
WORD: It swaps the value of the most significant word (16 bits) with the least significant one
BYTE AND WORD: It swaps the value of the most significant byte (8 bits) with the least significant one and the value of the most significant word (16 bits) with the least significant one.

Commands setup can be imported/exported from/to a ".cgi" file:

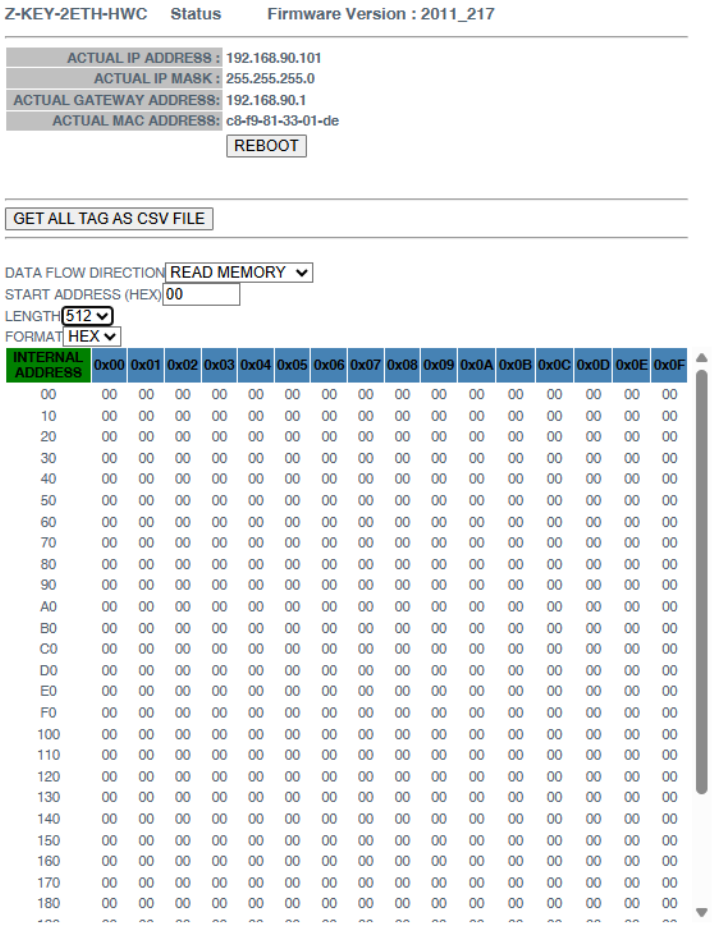
Note that a .cgi file can also be imported/exported from/to the Excel template.

Once all the values have been entered, the fields are saved with the "APPLY" button.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

15.4.3.*REAL-TIME VIEW OF THE MODBUS GATEWAY*

Once the commands are configured, it is possible to view the status of the Modbus communication in real time, from the Status section of the navigation menu.
The live view will show the current network configuration, operation mode and command information.



The memory is divided into 512 bytes of Read Memory and 512 bytes of Write Memory, these values are updated in real time.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

16. RESETTING THE DEVICE TO ITS FACTORY CONFIGURATION

The factory configuration resets all parameters to default.

To reset the device to the factory configuration it is necessary to follow the procedure below:

Z-KEY-0 / Z-KEY-2ETH-0:

1) Remove power from the device
2) Turn dip switches 1 and 2 to ON
3) Power up the device and wait at least 10 seconds
4) Remove power from the device
5) Turn dip switches 1 and 2 to OFF
6) At the next restart the device will have loaded the factory configuration

R-KEY-LT-0:

1) Remove power from the device
2) Set dip switches 1 and 2 of SW2 to ON
3) Power up the device and wait at least 10 seconds
4) Remove power from the device
5) Turn 2 SW2 dip switches to OFF.
6) At the next restart the device will have loaded the factory configuration

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

17. TEMPLATE EXCEL

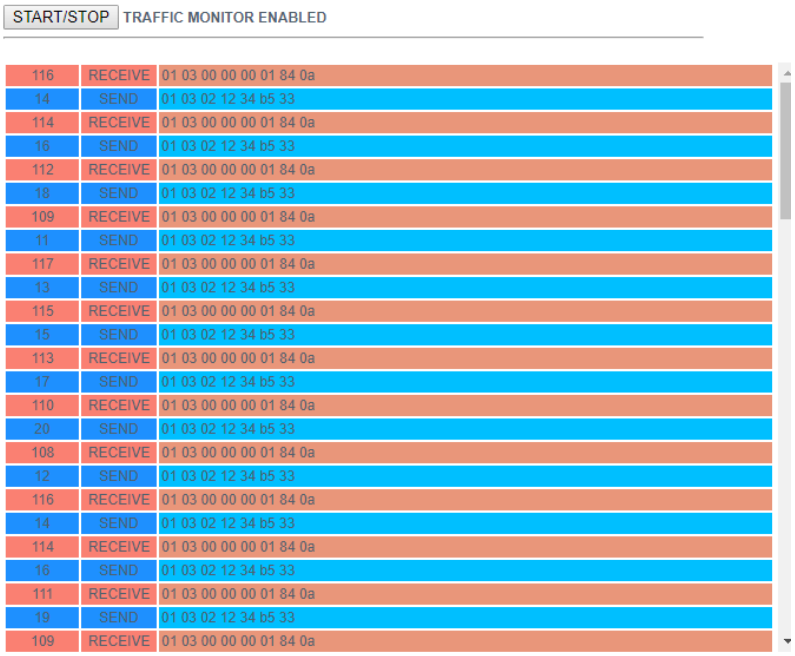
To prepare the configuration of Tags and commands a Microsoft Excel™ template is available to create a .bin file to import into the gateway or vice versa. The model can be freely downloaded from the Seneca website.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	MODBUS TCP/IP			SERIAL MODBUS RTU					Export CGI file...		SENECA		
2	TAG NR	GATEWAY TAG NAME	GATEWAY MODBUS TCP/IP REGISTER ADDRESS	TARGET MODBUS RTU REGISTER TYPE	TARGET MODBUS RTU DATA TYPE	TARGET CONNECTED TO SERIAL PORT NR	TARGET MODBUS RTU START REGISTER	TARGET MODBUS RTU SLAVE ADDRESS	Import CGI file...	SENECA Z-KEY TAGS TEMPLATE FOR GATEWAY MODE. E			
3	1	TAG1	1	HOLDING REGISTER	UINT16	#1	3	2					
4	2	TAG2	2	HOLDING REGISTER	UINT16	#1	4	2					
5	3	TAG3	3	HOLDING REGISTER	UINT16	#1	5	2					
6	4	TAG4	5	HOLDING REGISTER	UINT16	#1	6	2					
7	5	TAG5	7	HOLDING REGISTER	UINT16	#1	7	2					
8	6	TAG6	8	HOLDING REGISTER	UINT16	#1	8	2					
9	7	TAG7	9	HOLDING REGISTER	UINT16	#1	9	2					
10	8	TAG8	10	HOLDING REGISTER	UINT16	#1	10	2					
11	9	TAG9	1	COIL	BIT	#1	1	3					
12	10	TAG10	2	COIL	BIT	#1	2	3					
13	11	TAG11	3	COIL	BIT	#1	3	3					
14	12	TAG12	4	COIL	BIT	#1	4	3					
15	13	TAG13	5	COIL	BIT	#1	5	3					
16	14	TAG14	6	COIL	BIT	#1	6	3					
17	15	TAG15	7	COIL	BIT	#1	7	3					
18	16	TAG16	8	COIL	BIT	#1	8	3					
19	17	TAG17	14	HOLDING REGISTER	INT16	#1	13	4					
20	18	TAG18	15	HOLDING REGISTER	INT16	#1	14	4					
21	19	TAG19	16	HOLDING REGISTER	INT16	#1	15	4					
22	20	TAG20	17	HOLDING REGISTER	INT16	#1	16	4					
23	21	TAG21	1	DISCRETE INPUT	BIT	#1	1	5					
24	22	TAG22	2	DISCRETE INPUT	BIT	#1	2	5					
25	23	TAG23	3	DISCRETE INPUT	BIT	#1	3	5					

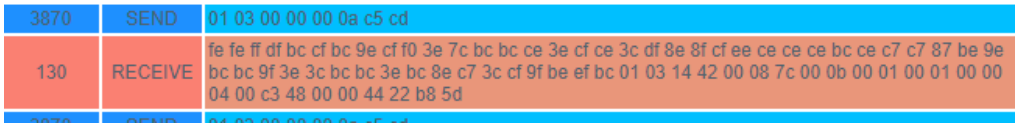
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

18. SERIAL TRAFFIC MONITOR

The Serial Traffic Monitor page of the webserver shows the serial packets that the gateway is receiving and transmitting for line debugging:



The first column is the delay in milliseconds from the last packet, the second column is the direction of the packet (received from or transmitted to), the last column is the contents of the packet in hexadecimal format. Only the serial ModBUS stream is displayed.
The Traffic Monitor shows all packets received from the serial line, for example if it is a serial slave with an incorrect Modbus response:



The Traffic Monitor will also display defective packets in yellow (for example a serial master with wrong baud rate):

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

19. INSTALLING MULTIPLE DEVICES IN A NETWORK USING THE "DHCP FAIL ADDRESS".

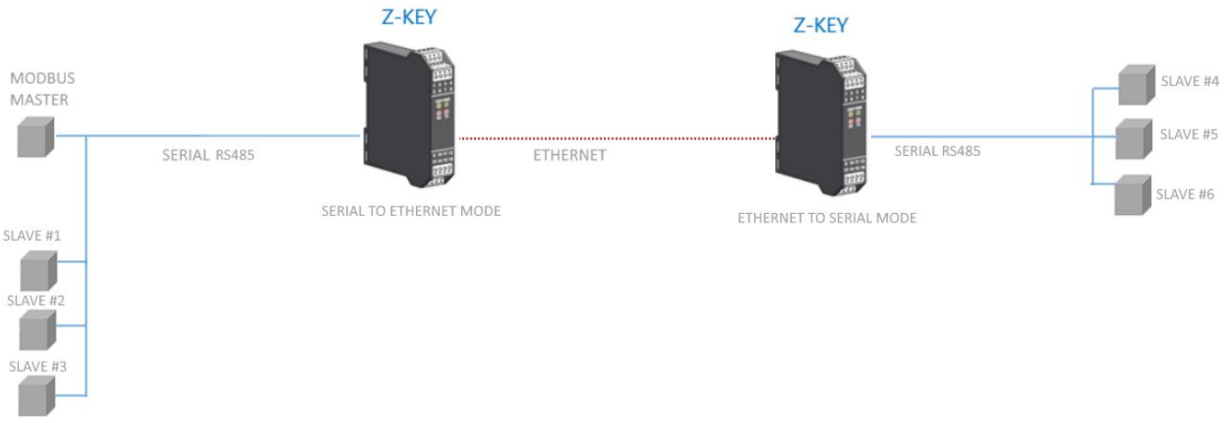
When the Gateway is configured with DHCP enabled but does not receive the DHCP server configuration within 2 minutes then it assumes a fail address.
This fail address is 169.254.x.y where x.y are the last two values from the MAC address.
In this way, if you force all devices to DHCP, you can install on the network even if there is no active DHCP server.
When the fail address has been activated (the relative LED stops flashing), you can launch the "Seneca Discovery Device" software and force the preferred IP address to all devices.

20. THE DB9 RS232 CABLE

The DB9 CABLE RS232 CABLE can be obtained from Seneca (it can also be purchased from the e-commerce website www.seneca.it) for connection with a DB9 RS232 device.

21. EXTEND THE RS485 BUS ON ETHERNET: SERIAL MODBUS ON ETHERNET AND THEN ETHERNET ON SERIAL

It is possible to extend the RS485 bus using the Ethernet or Wi-Fi infrastructure, to obtain this feature at least two gateway devices are required: one configured in Gateway from Serial to Ethernet mode and the other configured in Gateway from Ethernet to Serial mode



ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

22. SUPPORTED MODBUS COMMUNICATION PROTOCOLS

The Modbus communication protocols supported are:

- Modbus RTU/ASCII master/slave (from #1 and #2 serial ports)
- Modbus TCP-IP Client (from the Ethernet port), up to 10 remote TCP-IP Modbus Servers

For more information on these protocols, see the website:
http://www.modbus.org/specs.php.

22.1. SUPPORTED MODBUS FUNCTION CODES

The following Modbus functions are supported:

- Read Coils (function 1)
- Read Discrete Inputs (function 2)
- Read Holding Registers (function 3)
- Read Input Registers (function 4)
- Write Single Coil (function 5)
- Write Single Register (function 6)
- Write multiple Coils (function 15)
- Write Multiple Registers (function 16)

ATTENTION!

All 32-bit variables are contained in 2 consecutive Modbus registers
All 64-bit variables are contained in 4 consecutive Modbus registers

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

23. INFORMATION ABOUT MODBUS REGISTERS

The following abbreviations are used in the following chapter:

MS	Most Significant
LS	Least Significant
MSBIT	Most Significant Bit
LSBIT	Least Significant Bit
MMSW	“Most” Most Significant Word (16bit)
MSW	Most Significant Word (16bit)
LSW	Least Significant Word (16bit)
LLSW	“Least” Least Significant Word (16bit)
RO	Read Only
RW*	Read-Write: REGISTERS CONTAINED IN FLASH MEMORY: WRITABLE ABOUT 10,000 TIMES MAXIMUM
RW**	Read-Write: REGISTERS THAT CAN BE WRITTEN ONLY AFTER WRITING THE COMMAND "ENABLE WRITE CUSTOM ENERGIES = 49616"
UNSIGNED 16 BIT	Unsigned integer register that can assume values from 0 to 65535
SIGNED 16 BIT	Signed integer register that can take values from -32768 to +32767
UNSIGNED 32 BIT	Unsigned integer register that can assume values from 0 to 4294967296
SIGNED 32 BIT	Signed integer register that can take values from -2147483648 to 2147483647
UNSIGNED 64 BIT	Unsigned integer register that can assume values from 0 to 18446744073709551615
SIGNED 64 BIT	Signed integer register that can assume values from -2^63 to 2^63-1
FLOAT 32 BIT	32-bit, single-precision floating-point register (IEEE 754) https://en.wikipedia.org/wiki/IEEE_754
BIT	Boolean register, which can take the values 0 (false) or 1 (true)

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

23.1. NUMBERING OF "0-BASED" OR "1-BASED" MODBUS ADDRESSES

According to the Modbus standard the Holding Registers are addressable from 0 to 65535, there are 2 different conventions for numbering the addresses: "0-BASED" and "1-BASED".
For greater clarity, Seneca shows its register tables in both conventions.

ATTENTION!

CAREFULLY READ THE DOCUMENTATION OF THE MODBUS MASTER DEVICE IN ORDER TO UNDERSTAND WHICH OF THE TWO CONVENTIONS THE MANUFACTURER HAS DECIDED TO USE
SENECA USES THE "1 BASED" CONVENTION FOR ITS PRODUCTS

23.2. NUMBERING OF MODBUS ADDRESSES WITH "0-BASED" CONVENTION

The numbering is:

HOLDING REGISTER MODBUS ADDRESS (OFFSET)	***MEANING***
0	FIRST REGISTER
1	SECOND REGISTER
2	THIRD REGISTER
3	FOURTH REGISTER
4	FIFTH REGISTER

Therefore, the first register is at address 0.
In the following tables, this convention is indicated with ***"ADDRESS OFFSET".***

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

23.3. NUMBERING OF MODBUS ADDRESSES WITH "1 BASED" CONVENTION (STANDARD)

The numbering is that established by the Modbus consortium and is of the type:

HOLDING REGISTER MODBUS ADDRESS 4x	***MEANING***
40001	FIRST REGISTER
40002	SECOND REGISTER
40003	THIRD REGISTER
40004	FOURTH REGISTER
40005	FIFTH REGISTER

This convention is indicated with ***"ADDRESS 4x"*** since a 40000 is added to the address so that the first Modbus register is 40001.
A further convention is also possible where the number 4 is omitted in front of the register address:

HOLDING MODBUS ADDRESS WITHOUT 4x	***MEANING***
1	FIRST REGISTER
2	SECOND REGISTER
3	THIRD REGISTER
4	FOURTH REGISTER
5	FIFTH REGISTER

23.4. BIT CONVENTION WITHIN A MODBUS HOLDING REGISTER

A Modbus Holding Register consists of 16 bits with the following convention:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

For instance, if the value of the register in decimal is
12300
the value 12300 in hexadecimal is:
0x300C

the hexadecimal 0x300C in binary value is:
11 0000 0000 1100

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

So, using the above convention, we get:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0

23.5. MSB AND LSB BYTE CONVENTION WITHIN A MODBUS HOLDING REGISTER

A Modbus Holding Register consists of 16 bits with the following convention:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

LSB Byte (Least Significant Byte) defines the 8 bits ranging from Bit 0 to Bit 7 included, we define MSB Byte (Most Significant Byte) the 8 bits ranging from Bit 8 to Bit 15 inclusive:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
BYTE MSB								BYTE LSB							

23.6. REPRESENTATION OF A 32-BIT VALUE IN TWO CONSECUTIVE MODBUS HOLDING REGISTERS

The representation of a 32-bit value in the Modbus Holding Registers is made using 2 consecutive Holding Registers (a Holding Register is a 16-bit register). To obtain the 32-bit value it is therefore necessary to read two consecutive registers:
For example, if register 40064 contains the 16 most significant bits (MSW) while register 40065 contains the least significant 16 bits (LSW), the 32-bit value is obtained by composing the 2 registers:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
40064 MOST SIGNIFICANT WORD															

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
40065 LEAST SIGNIFICANT WORD															

$$Value_{32bit} = Register_{LSW} + (Register_{MSW} * 65536)$$

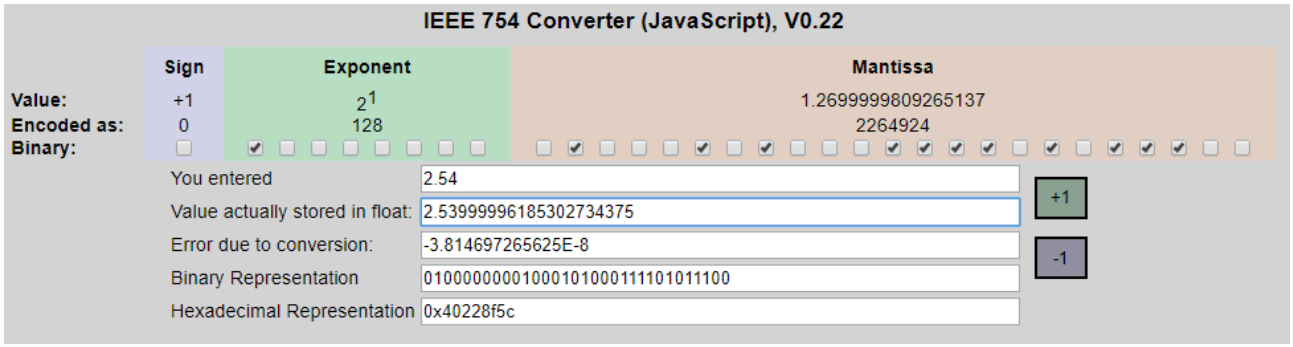
In the reading registers it is possible to swap the most significant word with the least significant word, therefore it is possible to obtain 40064 as LSW and 40065 as MSW.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.

23.7. TYPE OF 32-BIT FLOATING POINT DATA (IEEE 754)

The IEEE 754 standard (https://en.wikipedia.org/wiki/IEEE_754) defines the format for representing floating point numbers.
As already mentioned, since it is a 32-bit data type, its representation occupies two 16-bit holding registers.
To obtain a binary/hexadecimal conversion of a floating point value it is possible to refer to an online converter at this address:

http://www.h-schmidt.net/FloatConverter/IEEE754.html



Using the last representation the value 2.54 is represented at 32 bits as:

0x4022 8F5C

Since we have 16-bit registers available, the value must be divided into MSW and LSW:

0x4022 (16418 decimal) are the 16 most significant bits (MSW) while 0x8F5C (36700 decimal) are the 16 least significant bits (LSW).

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED WITHOUT PRIOR PERMISSION.