USER MANUAL

Z-KEY-0

R-KEY-LT

MODBUS GATEWAY
AND SERIAL DEVICE SERVER

SENECA s.r.l.

Via Austria, 26 - 35127 -PADOVA - ITALY

Tel. +39.049.8705355 - 8705359 Fax. +39.049.8706287

Website: www.seneca.it

Technical service: supporto@seneca.it (IT), support@seneca.it (Other)

Commercial information: commerciale@seneca.it (IT), sales@seneca.it (Other)

MI00579-2

Date	Revision	Note
21/03/2022	01	First revision for new hardware and new firmware.
25/07/2022	02	Removed wording concerning tags

Contents

1. GLOSSARY

- *Modbus RTU*

An open protocol for industrial serial communication developed by Modicon Inc. (AEG Schneider Automation International S.A.S.). Simple and robust, it has de facto become a standard communication protocol.

For further information http://www.modbus.org/specs.php

- *Modbus ASCII*

A variant or Modbus RTU protocol where communication is in ASCII format.

For further information http://www.modbus.org/specs.php

- *MODBUS TCP-IP*

Variant of the Modbus RTU protocol for Ethernet interface, transmission takes place via TCP/IP packets.

For further information http://www.modbus.org/specs.php

- *MODBUS GATEWAY ETHERNET TO SERIAL*

A device that translates, in real time, from the Modbus TCP-IP Ethernet protocol to the Modbus RTU or ASCII serial protocol.

- *MODBUS GATERWAY SERIAL TO ETHERNET*

A device that translates, in real time, from the Modbus RTU or ASCII serial protocol to the Ethernet Modbus TCP-IP protocol.

- *MODBUS TAGS GATEWAY*

A device that translates the Modbus TCP-IP protocol into Modbus RTU or ASCII serial protocol using a buffered memory (called shared memory). The shared memory is accessible by multiple devices through Modbus TCP-IP, Modbus RTU, Webserver etc. but it must be configured.

This mode allows you to speed up and simplify a complex Modbus network.

- *SERIAL DEVICE SERVER (REMOTE SERIAL PORT)*

A device that allows devices with an RS-232 or RS-485 serial interface to connect and communicate with a PC via a LAN network. This makes it possible to use Ethernet instead of serial cables, minimizing the clutter of the workstation and also allowing you to place serial devices away from the computers they are used with. A serial device server is virtually transparent to any serial protocol. This mode is also called remote serial port because it allows you to extend the serial port through the Ethernet network.

- *MODBUS MASTER AND MODBUS SLAVE*

The Master is connected to one or more slaves, the slave waits for a register request from the Master. Only one Master is allowed. As a remedy to this limitation, a Modbus gateway is required.

- *MODBUS TCP-IP CLIENT AND MODBUS TCP-IP SERVER*

The Client (called Master in the Modbus RTU protocol) establishes a connection with the server (called Slave in the Modbus RTU/ASCII protocol). The server waits for a connection from the Client. Once the connection is established, the Server answers to the Client's requests.

- *WEB SERVER*

A software that saves, processes and supplies web pages for clients. The web clients can be PCs, Smartphones or Tablets equipped with a browser (Chrome, Internet Explorer, Firefox etc.).

2. INTRODUCTION

Z-KEY consists of:
No. 2 independent serial ports (one RS485 port and one configurable RS232/RS485 port)
No. 1 Fast Ethernet 100 MBits

R-KEY-LT consists of:
No. 1 RS232/RS485 serial port
No. 1 Fast Ethernet 100 MBits

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

The Z-KEY device can be configured in the following modes:

- *Modbus Gateway Ethernet to Serial (PORT#1)*
- *Modbus Gateway Ethernet to Serial (PORT#2)*
- *Modbus Gateway Ethernet to Serial (PORT#1 AND PORT#2)*
- *Modbus Gateway Serial to Ethernet (PORT#1 AND PORT#2)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 MASTER)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 MASTER PORT#2 SLAVE)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 SLAVE PORT#2 MASTER)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 SLAVE)*
- *Serial Device Server (PORT #1)*
- *Serial Device Server (PORT #2)*

The R-KEY-LT device can be configured in the following modes:

- *Modbus Gateway Ethernet to Serial (PORT#1)*
- *Modbus Gateway Serial to Ethernet (PORT#1)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 MASTER)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 SLAVE)*
- *Serial Device Server (PORT #1)*

3. ETHERNET COMMUNICATION

The default factory configuration for the Ethernet port is as follows:

- *IP static address 192.168.90.101*
- *Subnet 255.255.255.0*
- *Gateway 192.168.90.1*

ATTENTION!

BEFORE CONNECTING THE DEVICE TO THE ETHERNET NETWORK, MAKE SURE THAT THE IP ADDRESS 192.168.90.101 IS NOT USED BY ANOTHER ETHERNET DEVICE!

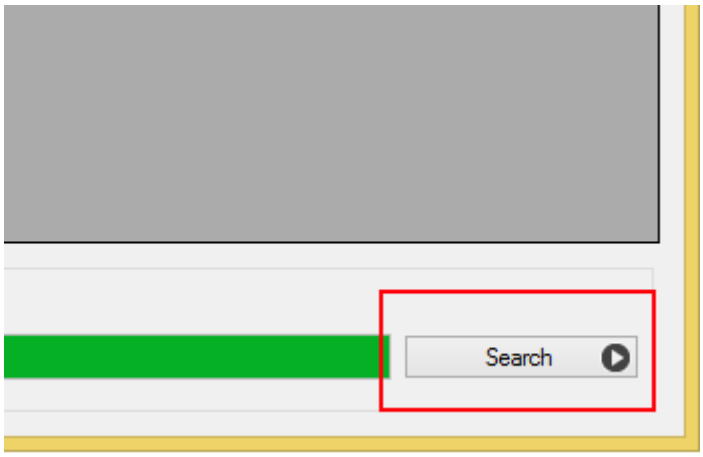
ATTENTION!

DO NOT CONNECT MORE THAN ONE DEVICE WITH THE SAME IP ADDRESS ON THE SAME NETWORK!

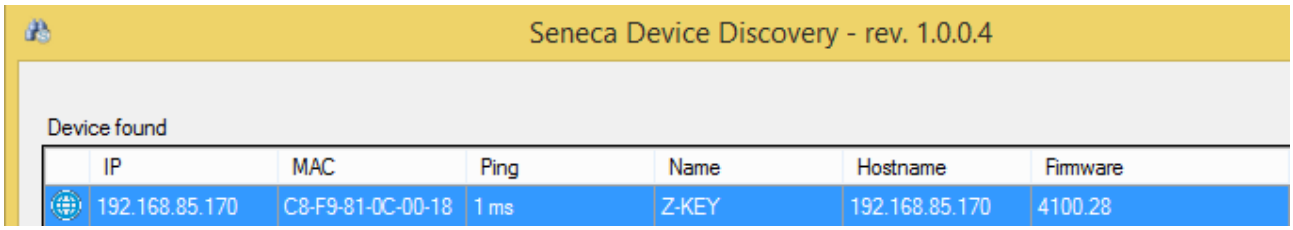
3.1. SEARCH FOR DEVICES ON THE NETWORK WITH SENECA DISCOVERY DEVICE SOFTWARE

From the Seneca website (www.seneca.it) you can download the Seneca Discovery Device utility for free to get the current IP address of the devices in your network.

Click Search to start the search process:



A list of found devices will be displayed:



At this point it is possible to change the IP address of the device directly from the software in order to make it compatible with your network.

4. THE WEB SERVER

The Gateway integrates a Webserver for the configuration and display of values in real time.

To access the web pages, type the IP address of the device on a browser

http://192.168.90.101 (default IP).

The web server will allow access through a username and password (admin, admin are the default login credentials)

5. MODBUS RTU, MODBUS ASCII AND MODBUS TCP-IP PROTOCOLS

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

5.1. SUPPORTED MODBUS FUNCTION CODES

The following Modbus functions are supported:

Read Coil (function 1)

Read Discrete Input (function 2)

Read Holding Registers (function 3)

Read Input Registers (function 4)

Write Single Coil (function 5)

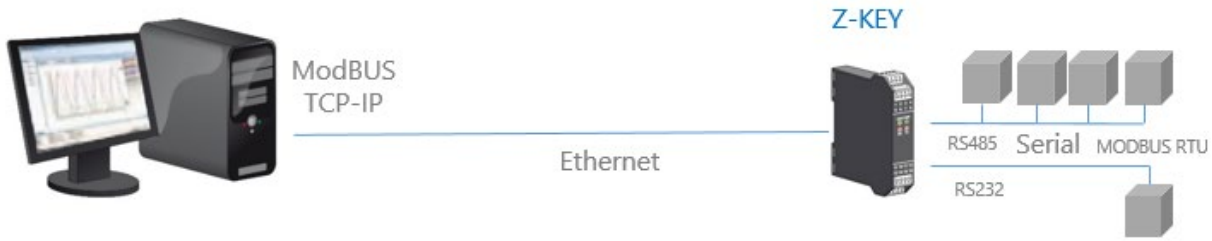
Write Single Register (function 6)

Write Multiple Registers (function 16)

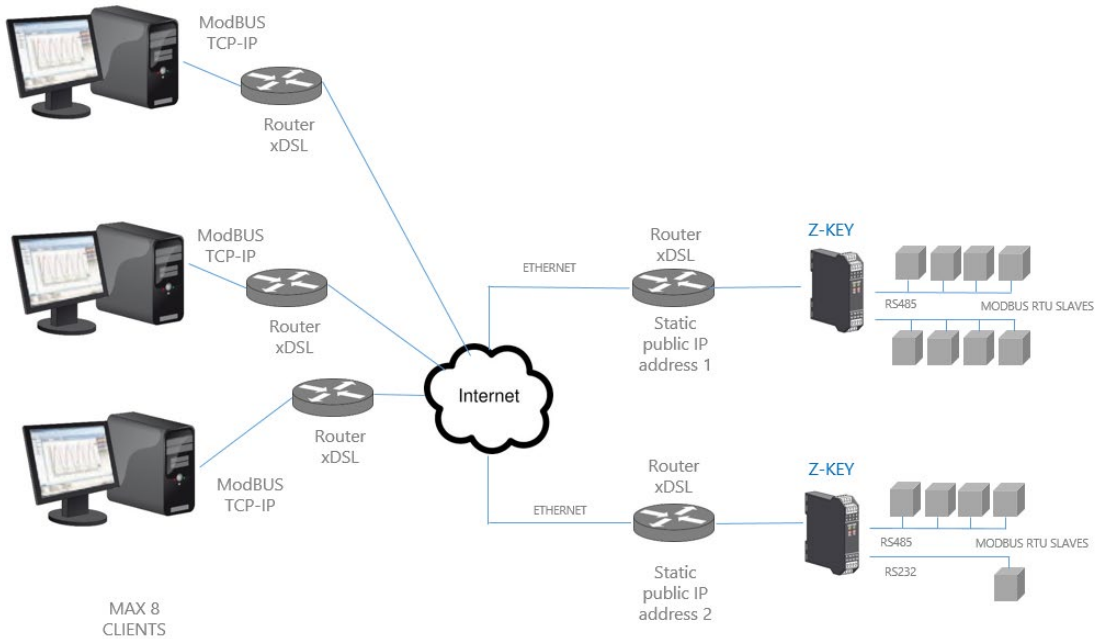
6. Modbus Gateway mode from Ethernet to Serial

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:



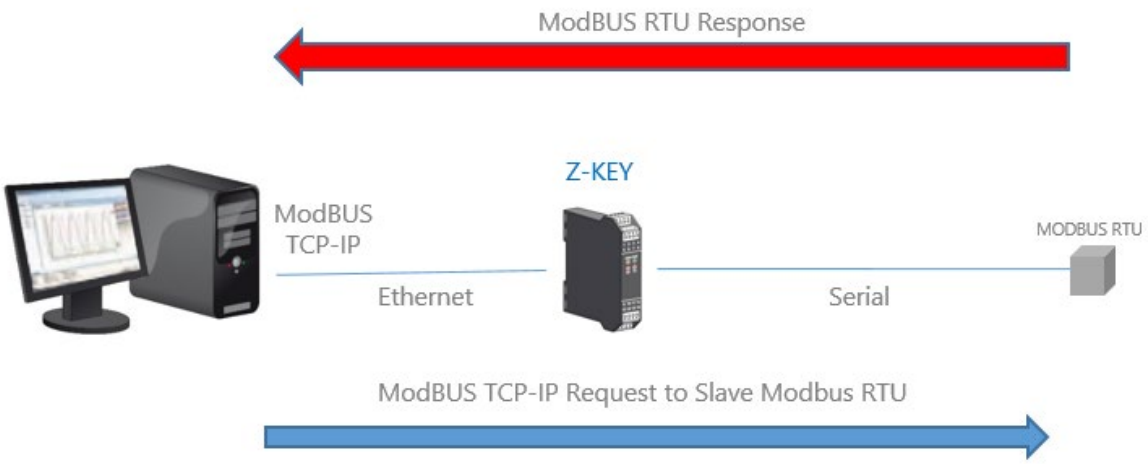
A maximum of 8 simultaneous Modbus TCP-IP Clients are supported.

6.1. 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 gateway has 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!

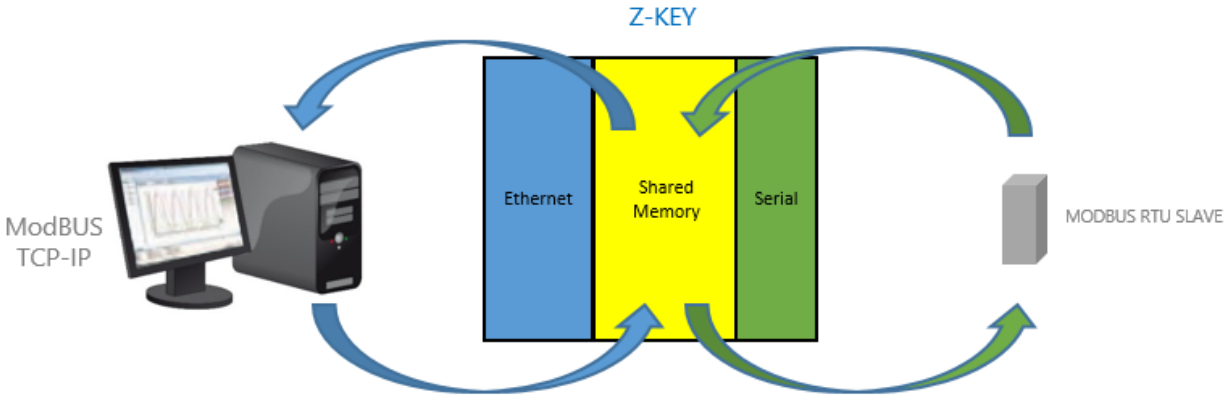
In this operating mode it is not possible to connect two serial Modbus slaves with the same Modbus address.

7. 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).

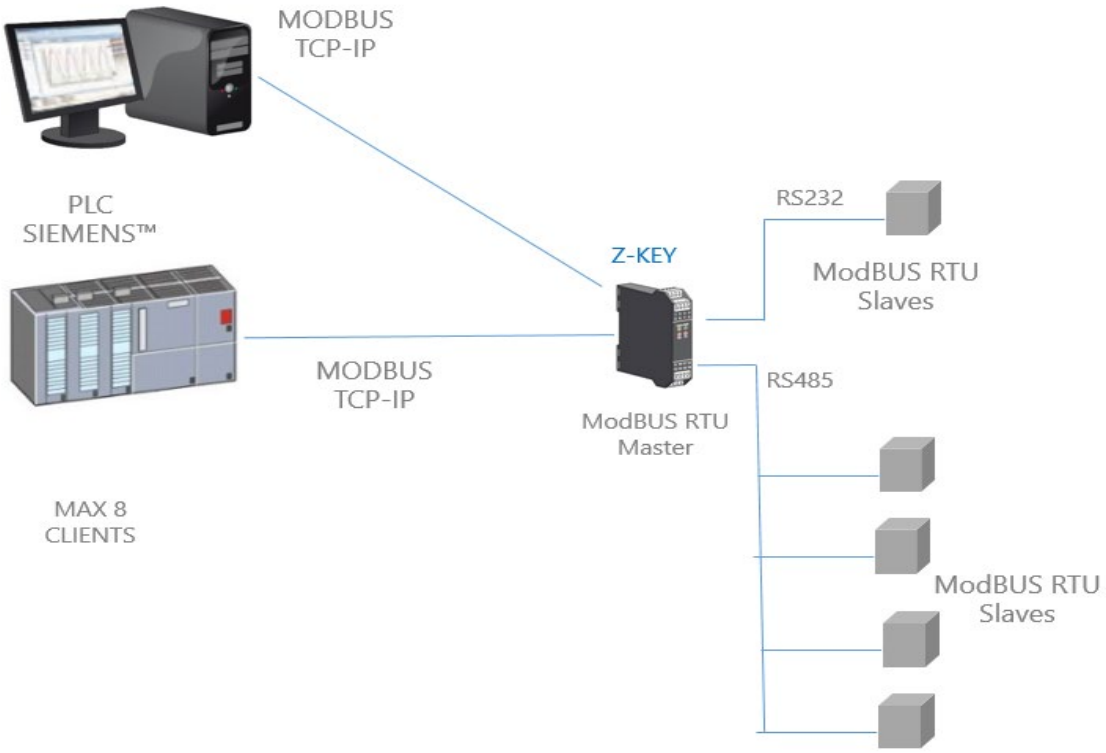
When a serial Modbus device generates a time-out error (for example due to a fault), in order to speed up the acquisition cycle, it is quarantined for a configurable time.

7.1. 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.

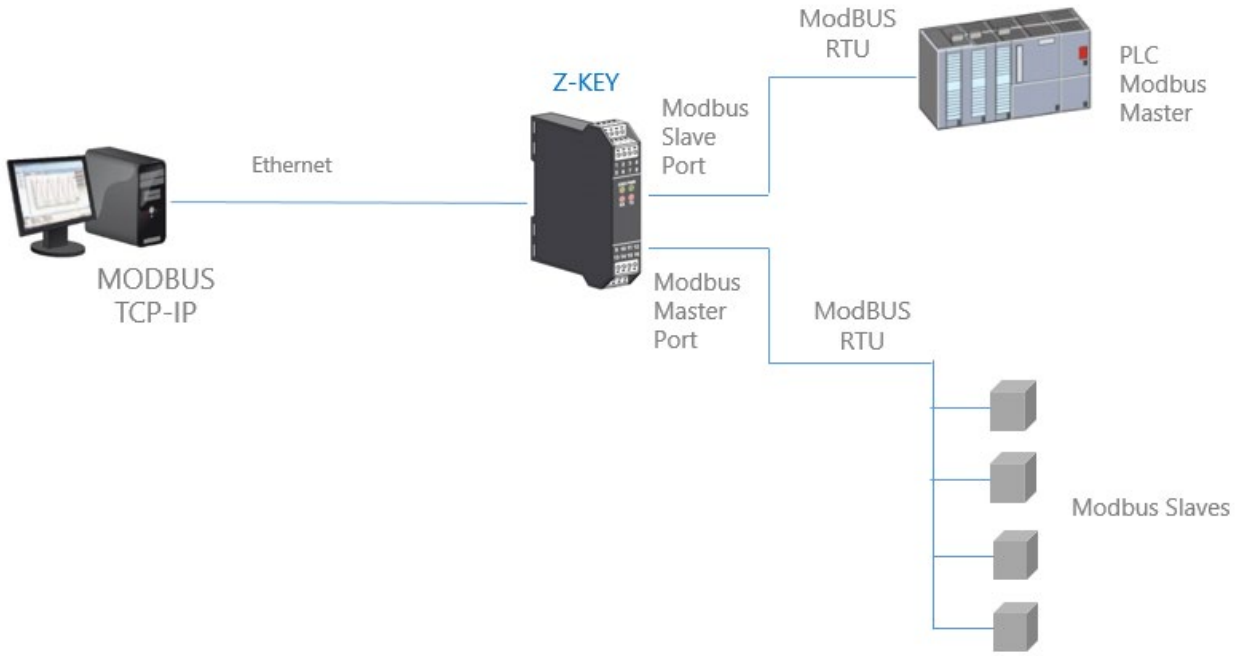
7.2. 1 MASTER PORT AND 1 SLAVE PORT (Z-KEY MODEL ONLY)

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.

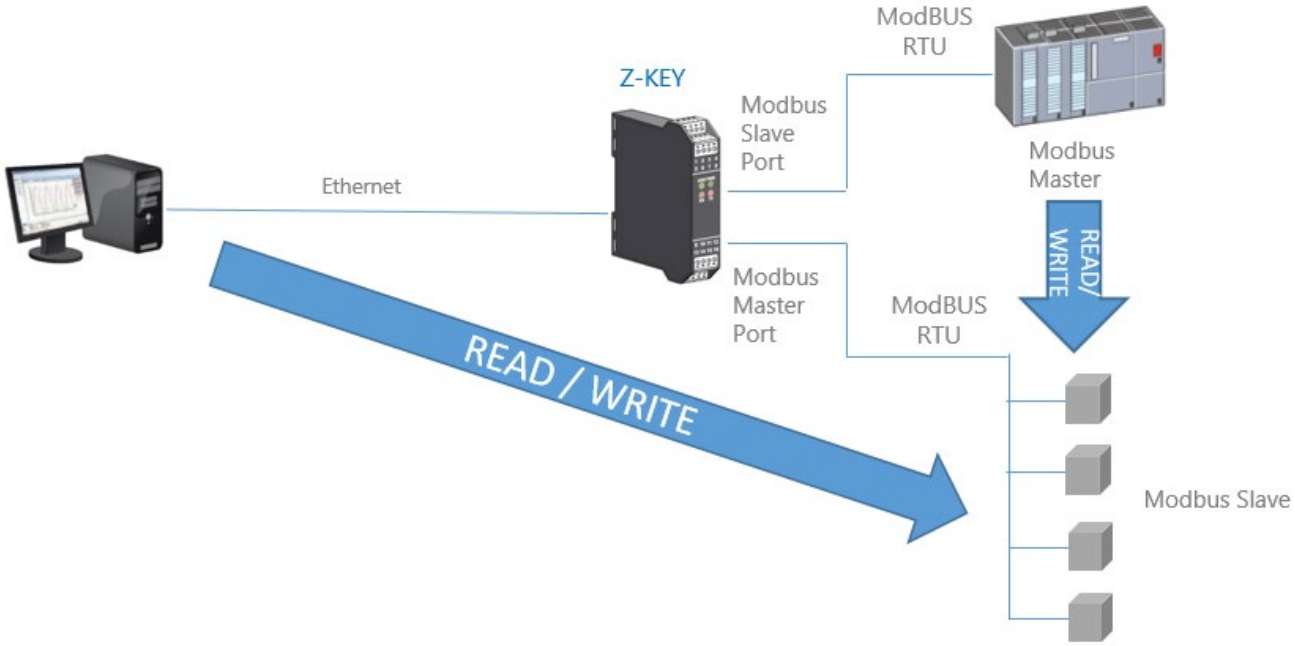
The same operations can also be performed via Ethernet from the Modbus TCP Client (for example a PC).

This figure shows an example of this scenario:



7.2.1. 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.



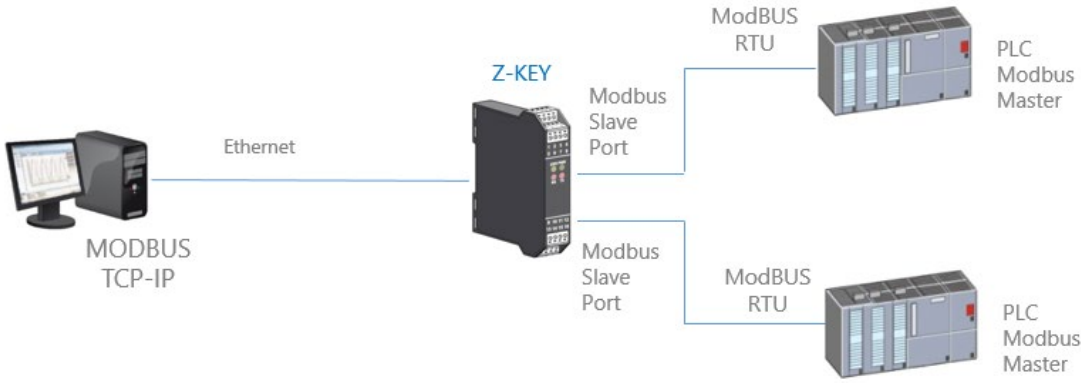
This function is possible because the Seneca Gateway uses an internal memory shared between the serial ports and the Ethernet port, where the data acquired from the serial Modbus Slave network are saved.

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.

7.3. SLAVE PORT 1 AND PORT 2

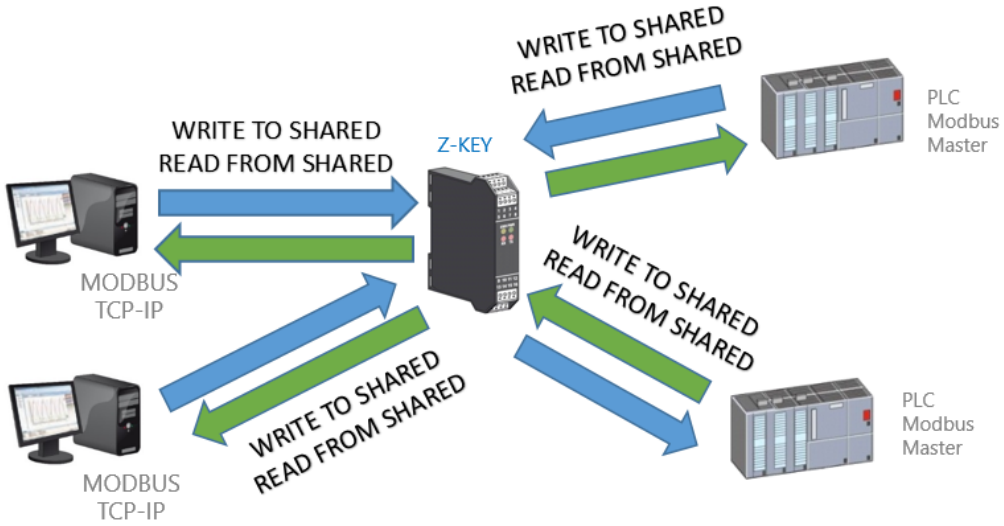
In this scenario a Modbus TCP-IP Client is connected via Ethernet to the gateway, two PLCs that support only the Modbus Master serial protocol are connected to the two serial ports of the gateway, configured as Slave. Through these ports the PLCs can communicate with each other.

This figure shows an example of this scenario:



7.3.1. HOW IT WORKS

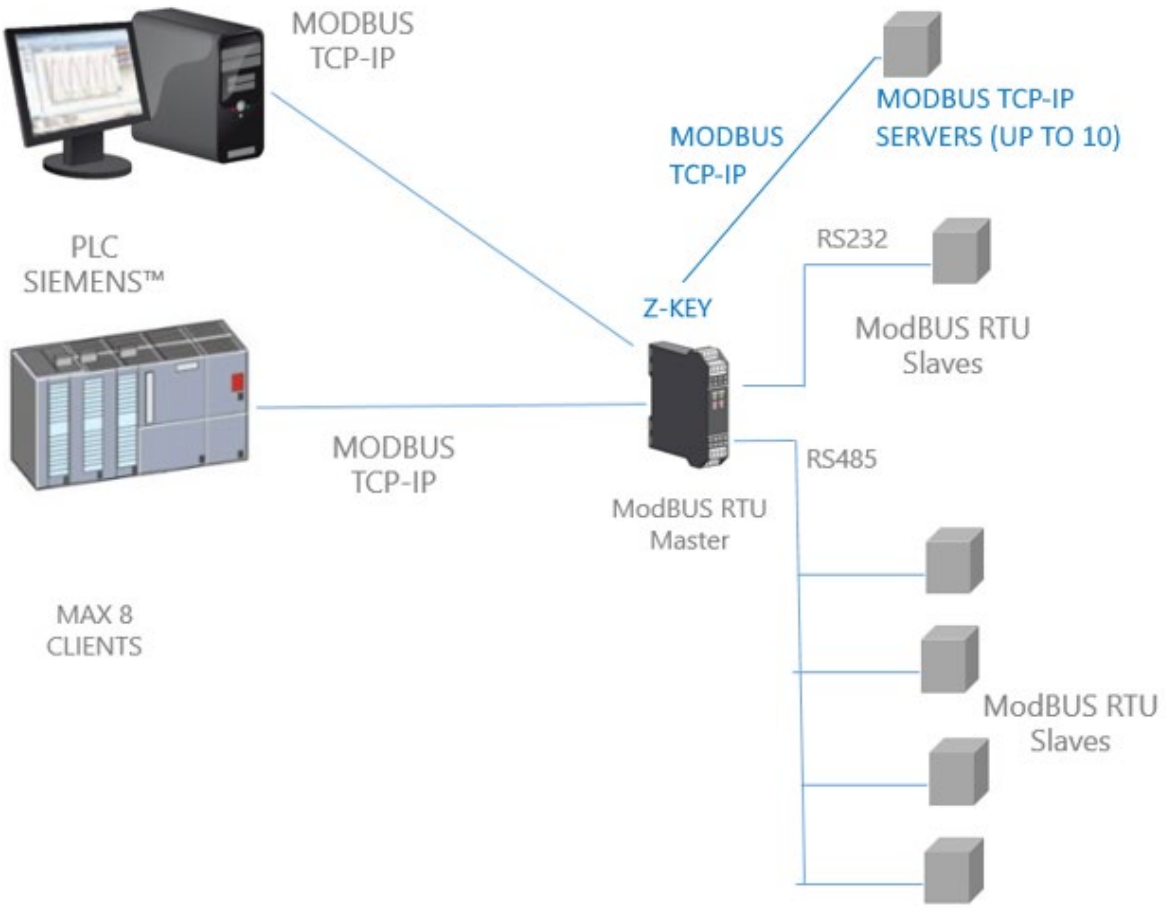
Many PLCs only implement the serial Modbus Master protocol (because they do not have an Ethernet port). If it is necessary to exchange Modbus registers between PLCs and PCs, the shared memory can be used.



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.

7.4. 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 (maximum 10); refer to the following figure:



7.5. 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 0 1 0 0 1

49002 0 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).

7.6. 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

For example:

49033 0x0000

49034 0x0002

Means:

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

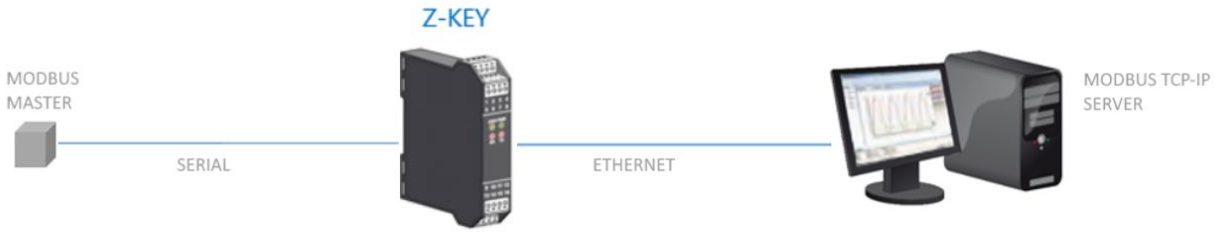
49284 42

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

8. SERIAL TO ETHERNET GATEWAY

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 networks.

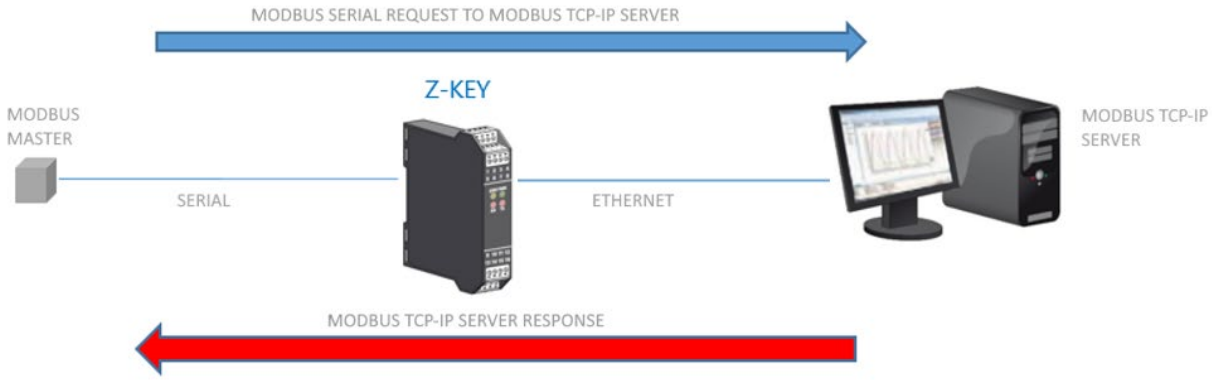


8.1.1. HOW IT WORKS

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).



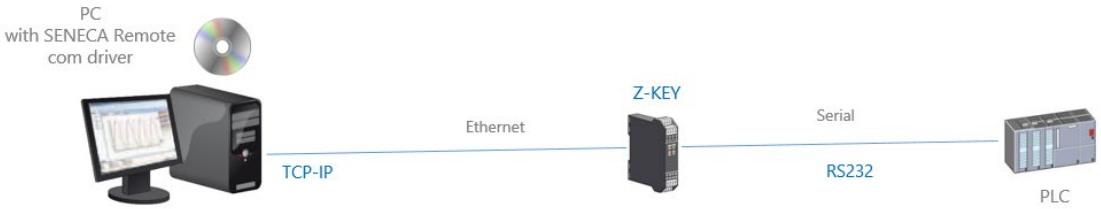
ATTENTION!

It is not possible to connect more than one Modbus server using the same Station address

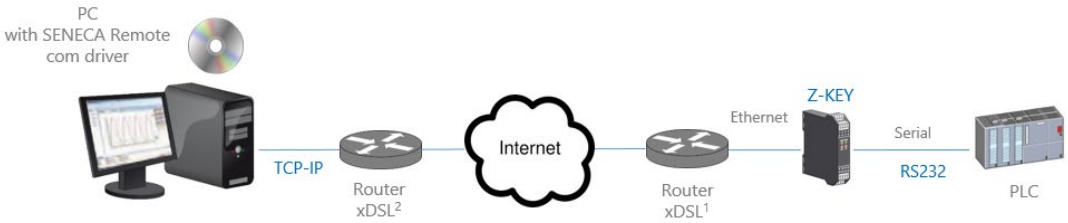
9. SERIAL DEVICE SERVER

In this mode 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.):



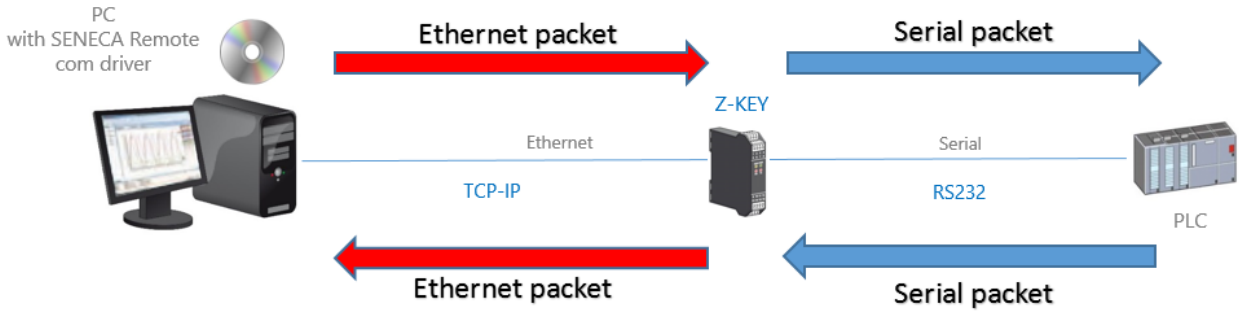
For internet communication, the router connected to the gateway must have a static IP address.

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

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.



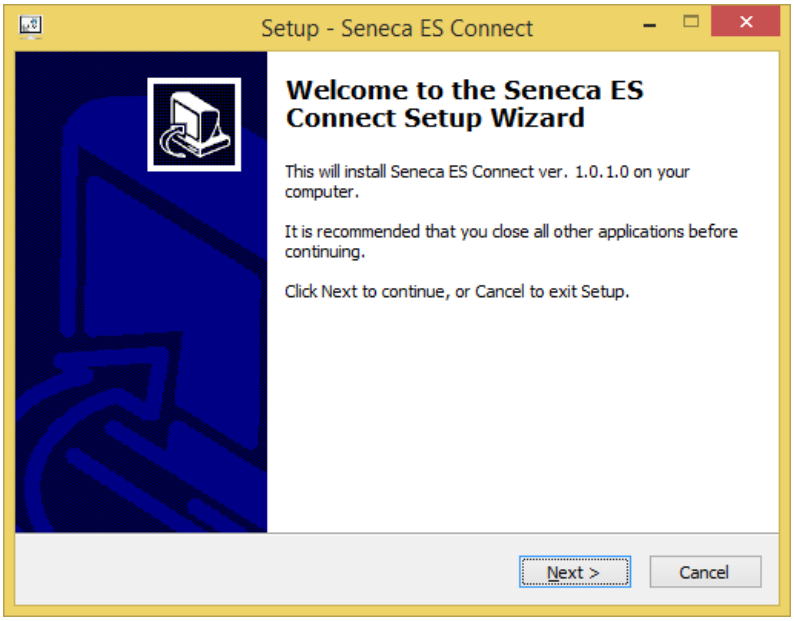
9.2. SENECA ETHERNET TO SERIAL CONNECT

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

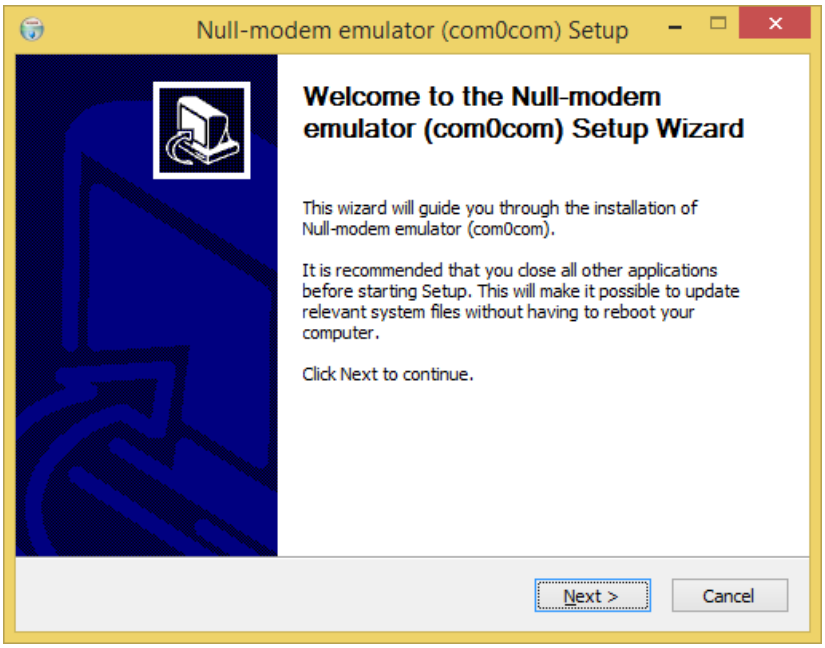
9.2.1. INSTALLING THE SENECA SERIAL TO ETHERNET CONNECT DRIVER

Seneca Ethernet to Serial Connect works on windows vista™, windows 7™, windows 8™, windows 10™ and windows 11™.

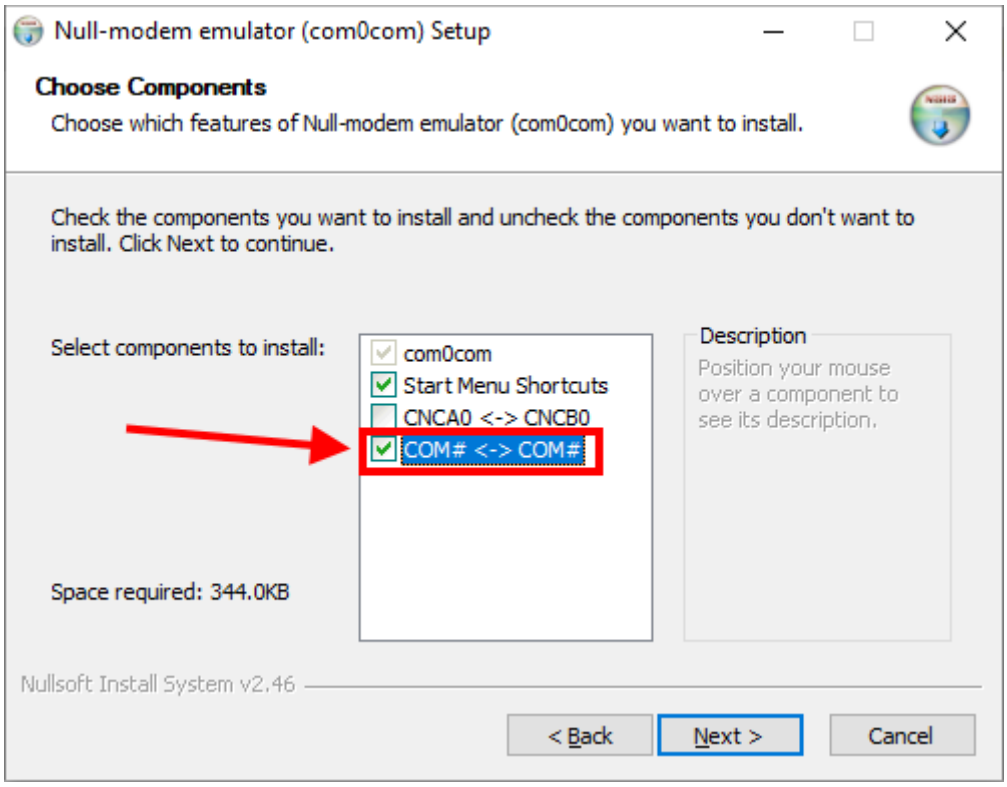
Double-click on the installation program:



Then the com0com driver will be installed:



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



Wait for the installation to finish.

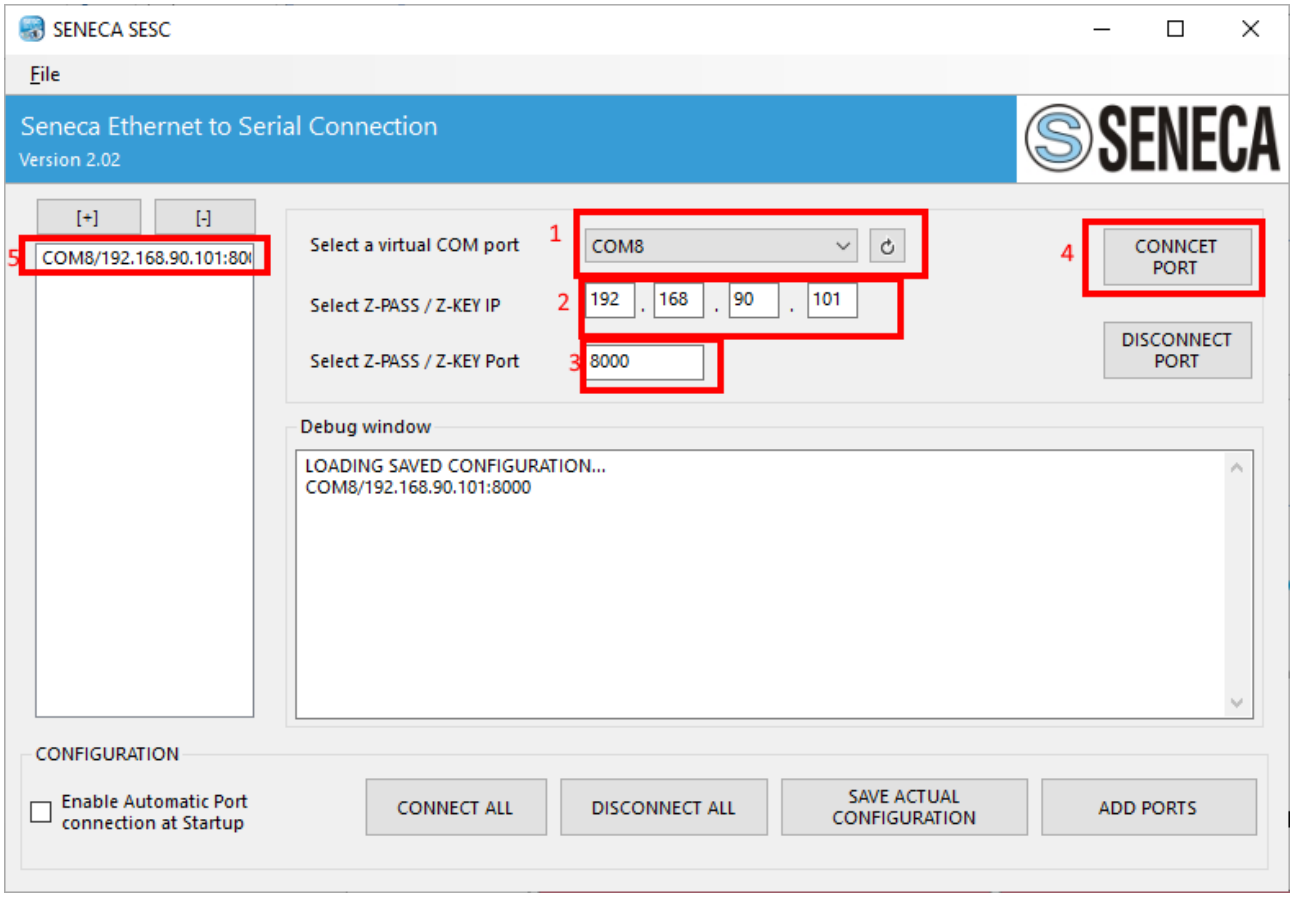
9.2.2. SELECT THE COM PORT FOR SENECA SERIAL TO ETHERNET CONNECT

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).

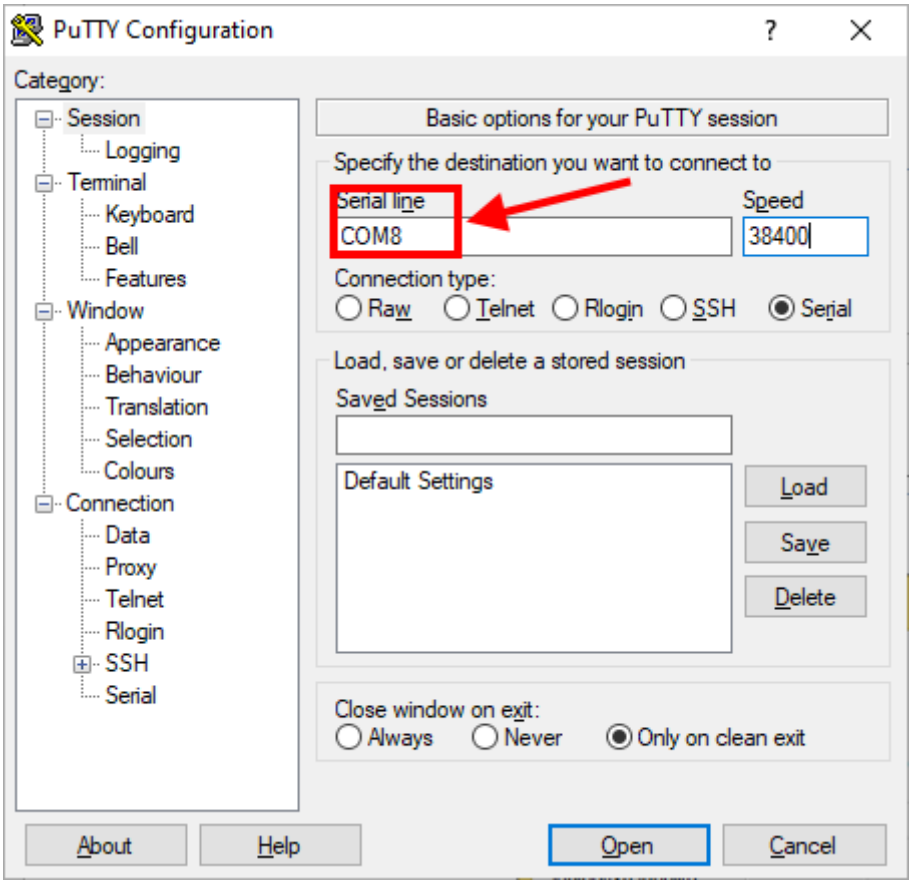
On the SESC interface set the Gateway IP (2) and TCP port (3) configured on the Z-KEY via the web Server.

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:

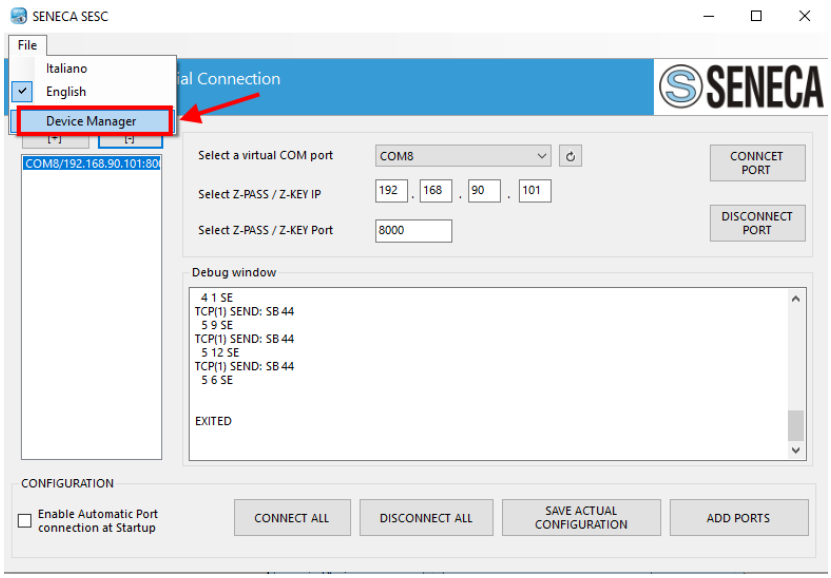


The other buttons allow you to:

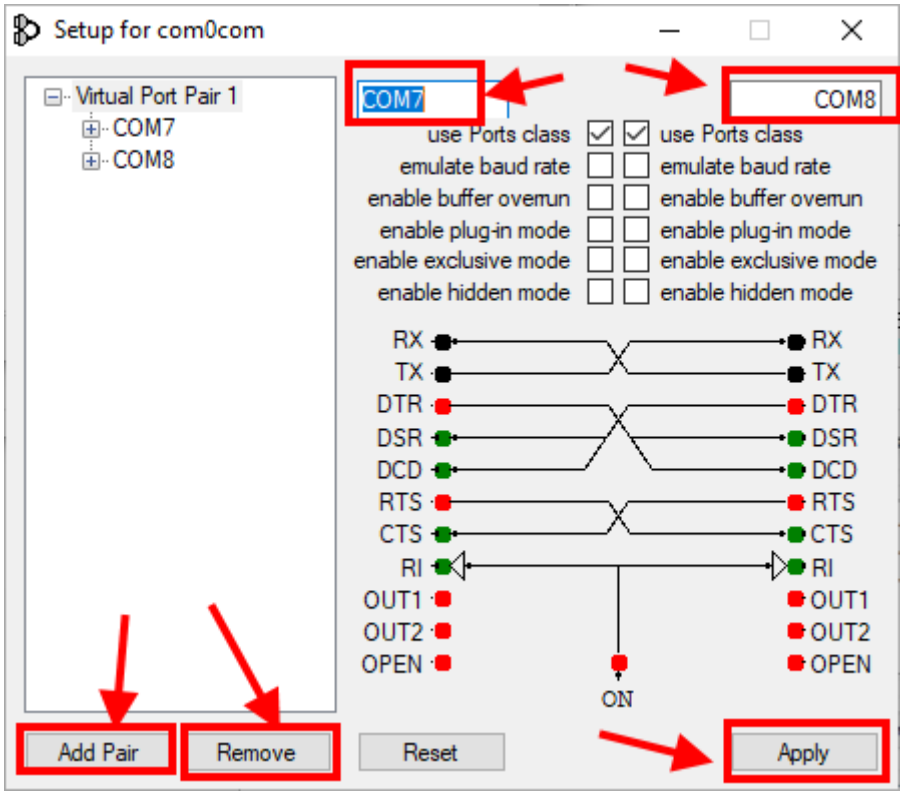
Activate the connection (with the saved configuration) when starting the PC
Connect/Disconnect all configured ports
Save the current configuration.

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:



Here you can:

- Rename virtual serials

- Add a pair of ports

- Remove a pair of ports

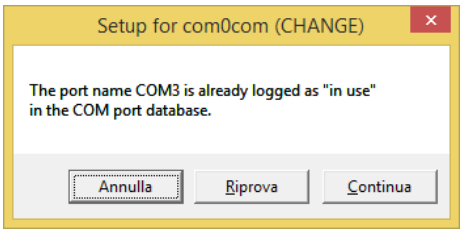
ATTENTION!

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

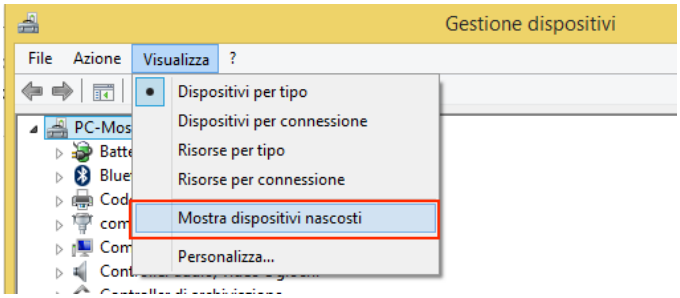
Older software can only use a small range of COM ports (typically 1 to 9), so you may need to change the virtual COM number.

COM may sometimes be marked as "in use":

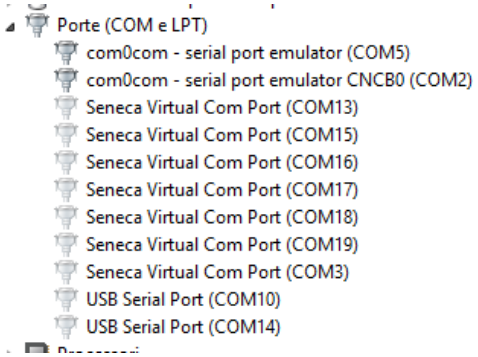


If you need to use this COM number, click on "Continue", then go to "Device Manager".

We must deselect the "in use" flag by uninstalling the port. Since the port is now disconnected, click on "Show hidden devices":

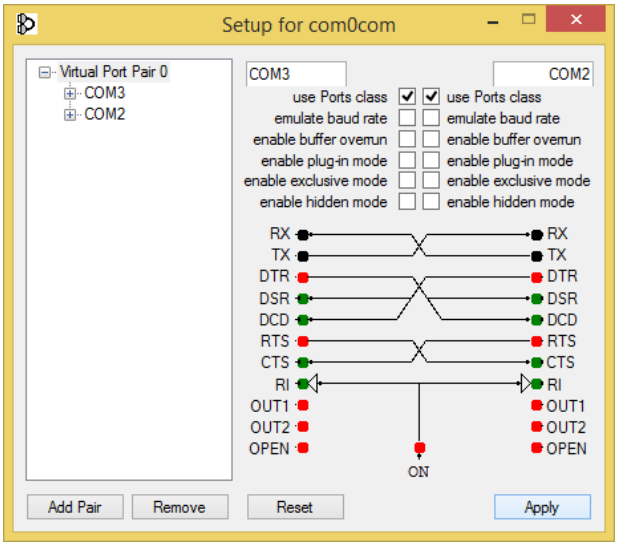


Now all the ports that are not in use are shown in transparency (also our COM3):

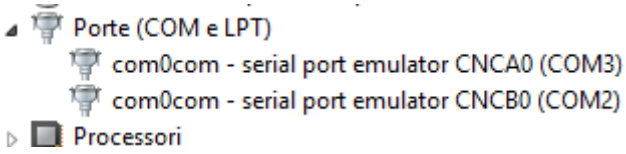


Now select the COM3 port and click on "Uninstall":

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

10. CONFIGURATION VIA WEBSERVER

10.1. ACCESS TO THE WEB SERVER

To access the configuration web server it is necessary to connect to the gateway by entering the IP address, for example using the factory one:

http://192.168.90.101

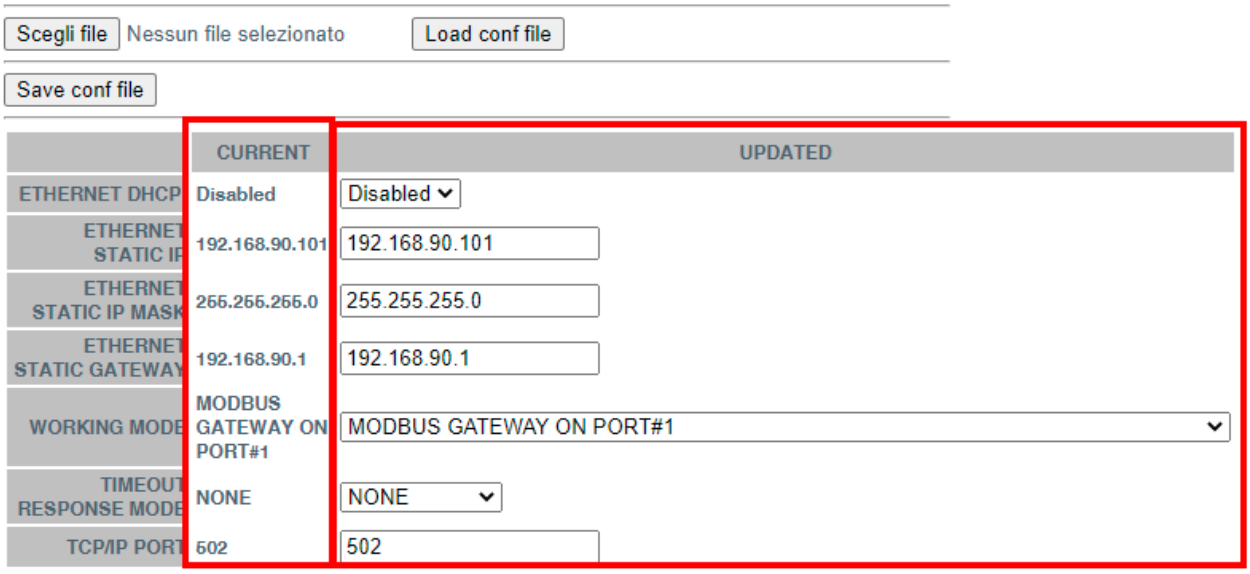
Enter the username and password, for example using the factory login credentials:

User: admin

Password: admin

Now the Webserver can be used to configure the Gateway, the general configuration parameters are available in the "Setup" section of the navigation menu, on the left of the screen.

10.1. SETUP PAGE

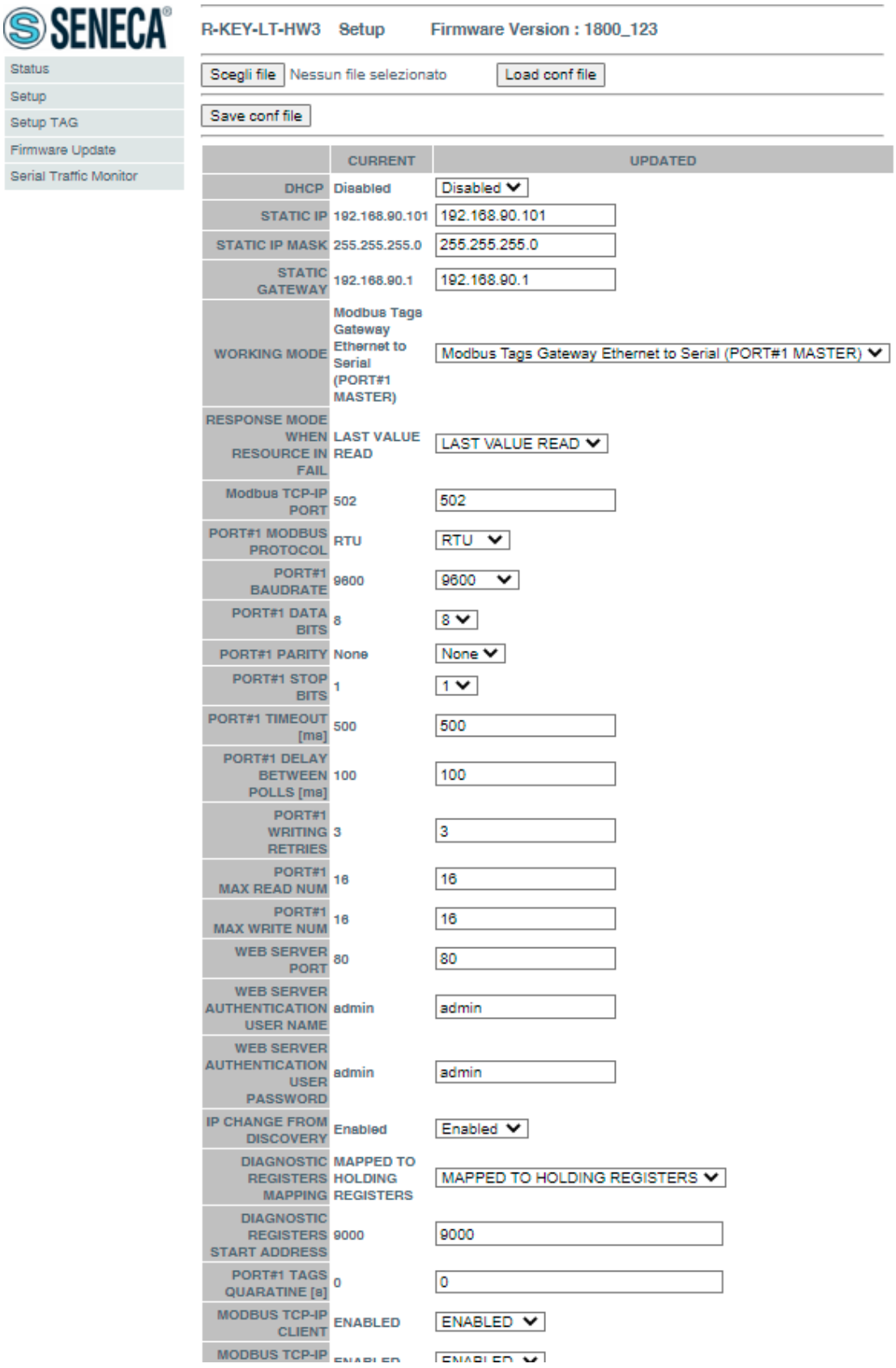


The first column represents the name of the parameter, the second column “current” is the current value of the parameter. The last column “updated” is used to modify the current configuration.

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.

10.1.1. GENERAL CONFIGURATION PARAMETERS



SENECA

Status
Setup
Setup TAG
Firmware Update
Serial Traffic Monitor

R-KEY-LT-HW3 Setup Firmware Version : 1800_123

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 Tags Gateway Ethernet to Serial (PORT#1 MASTER)	Modbus Tags Gateway Ethernet to Serial (PORT#1 MASTER)
RESPONSE MODE WHEN RESOURCE IN FAIL	LAST VALUE READ	LAST VALUE READ
Modbus TCP-IP PORT	502	502
PORT#1 MODBUS PROTOCOL	RTU	RTU
PORT#1 BAUDRATE	9600	9600
PORT#1 DATA BITS	8	8
PORT#1 PARITY	None	None
PORT#1 STOP BITS	1	1
PORT#1 TIMEOUT [ms]	500	500
PORT#1 DELAY BETWEEN POLLS [ms]	100	100
PORT#1 WRITING RETRIES	3	3
PORT#1 MAX READ NUM	16	16
PORT#1 MAX WRITE NUM	16	16
WEB SERVER PORT	80	80
WEB SERVER AUTHENTICATION USER NAME	admin	admin
WEB SERVER AUTHENTICATION USER PASSWORD	admin	admin
IP CHANGE FROM DISCOVERY	Enabled	Enabled
DIAGNOSTIC REGISTERS MAPPING	MAPPED TO HOLDING REGISTERS	MAPPED TO HOLDING REGISTERS
DIAGNOSTIC REGISTERS START ADDRESS	9000	9000
PORT#1 TAGS QUARATINE [s]	0	0
MODBUS TCP-IP CLIENT	ENABLED	ENABLED
MODBUS TCP-IP	ENABLED	ENABLED

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) (only Z-KEY-0 model)*
- *Modbus Gateway Ethernet to Serial (PORT#1 AND PORT#2) (only Z-KEY-0 model)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 MASTER)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 MASTER PORT#2 SLAVE) (only Z-KEY-0 model)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 SLAVE PORT#2 MASTER) (only Z-KEY-0 model)*
- *Modbus Tags Gateway Ethernet to Serial (PORT#1 AND PORT#2 SLAVE)*
- *Serial device server (PORT #1)*
- *Serial device server (PORT #2) (only Z-KEY-0 model)*
- *Modbus Gateway Serial to Ethernet (PORT#1 AND PORT#2) (only Z-KEY-0 model)*

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)

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 (Only for Serial Device Server mode)

Set the silence time after which the Ethernet packet is sent to the serial. This value must be adjusted according to the specific application.

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).

WEB SERVER PORT

Set the TCP-IP port for the Webserver.

WEB SERVER AUTHENTICATION USERNAME

Set the username for accessing the Webserver (if the username 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!

ALWAYS REMEMBER TO 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.

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)

#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

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 restart in case of gateway inactivity. If for a time equal to the "Watchdog Timeout" parameter there has not been a communication to process, the gateway restarts.

WATCHDOG TIMEOUT [min]

Set the gateway idle time in minutes to restart the gateway if the WATCHDOG ENABLE parameter is enabled.

10.2. ADD/DELETE/MODIFY TAGS (ONLY FOR MODBUS TAGS GATEWAY MODE)

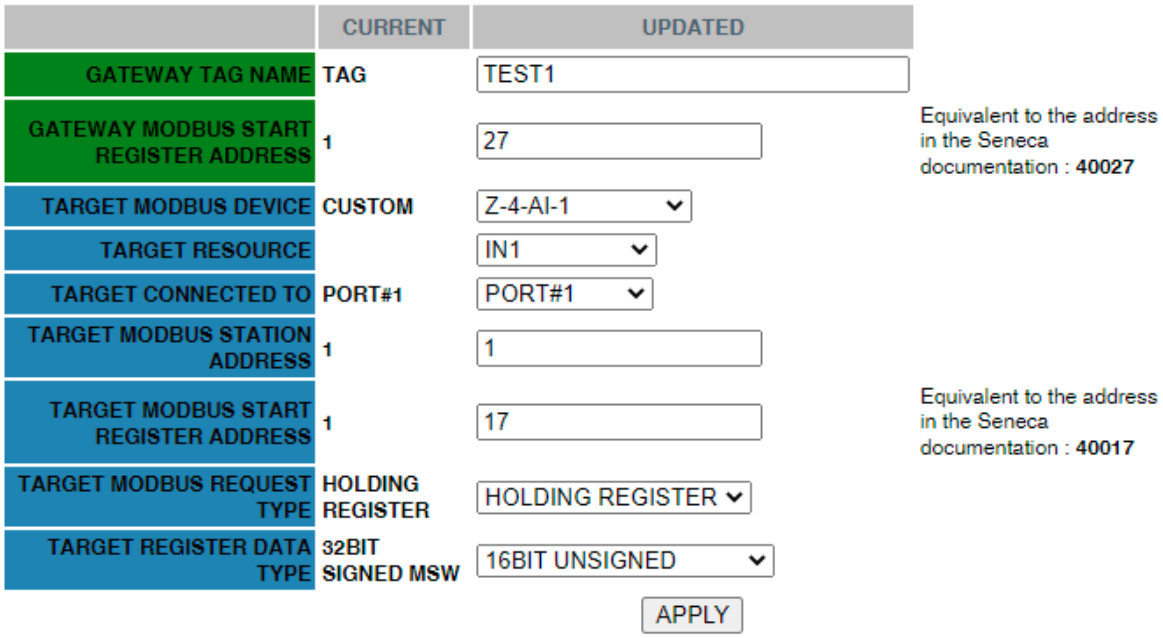
In Modbus Tags Gateway mode it is necessary to define the Modbus tags (i.e. variables), to do this it is possible to use:

- *The webserver*
- *An excel template*
- *The Easy Setup2 software*

In the case of complex configurations it is easier to use the last two.

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:



GATEWAY TAG NAME

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

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.

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 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).

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)

BIT: 1 Boolean Coil or Discrete Input, value true or false.

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:

$$1 = (40017) + ((40016) \times 65536)$$

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

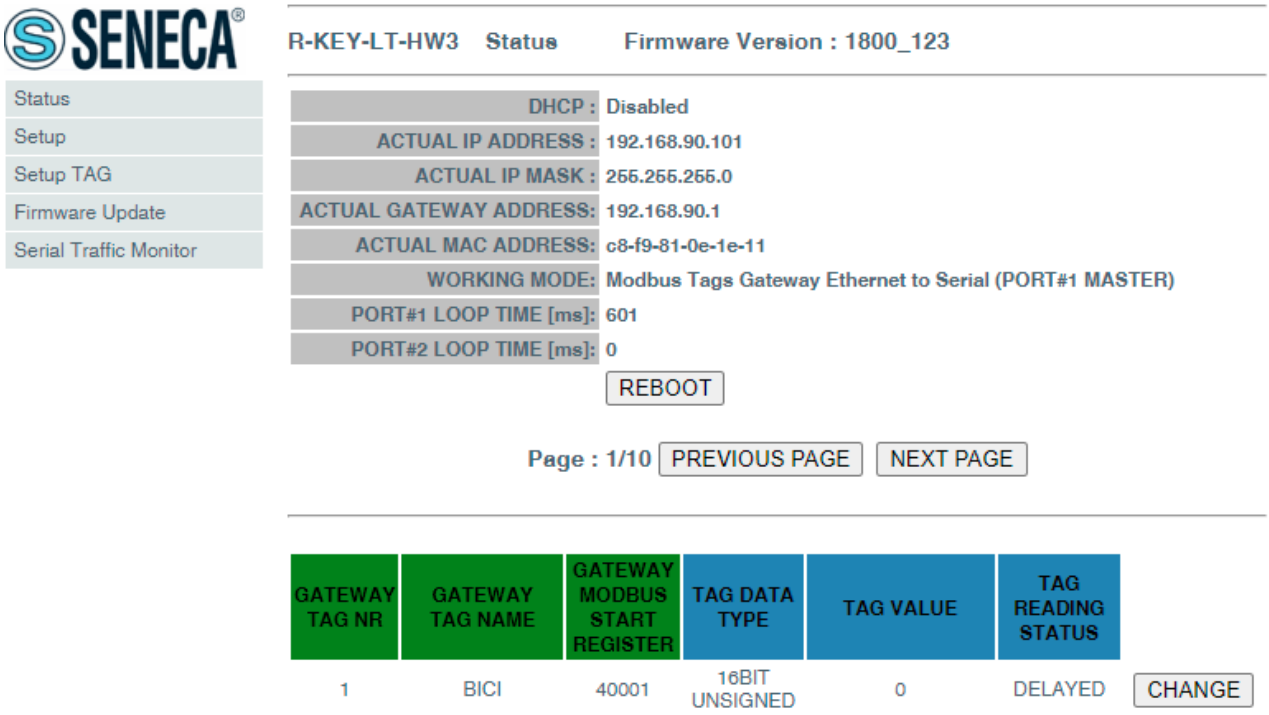
Note that a .cgi file can also be imported/exported from/to the Excel template.

It is also possible to add, modify, delete or move a tag.

10.3. 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.

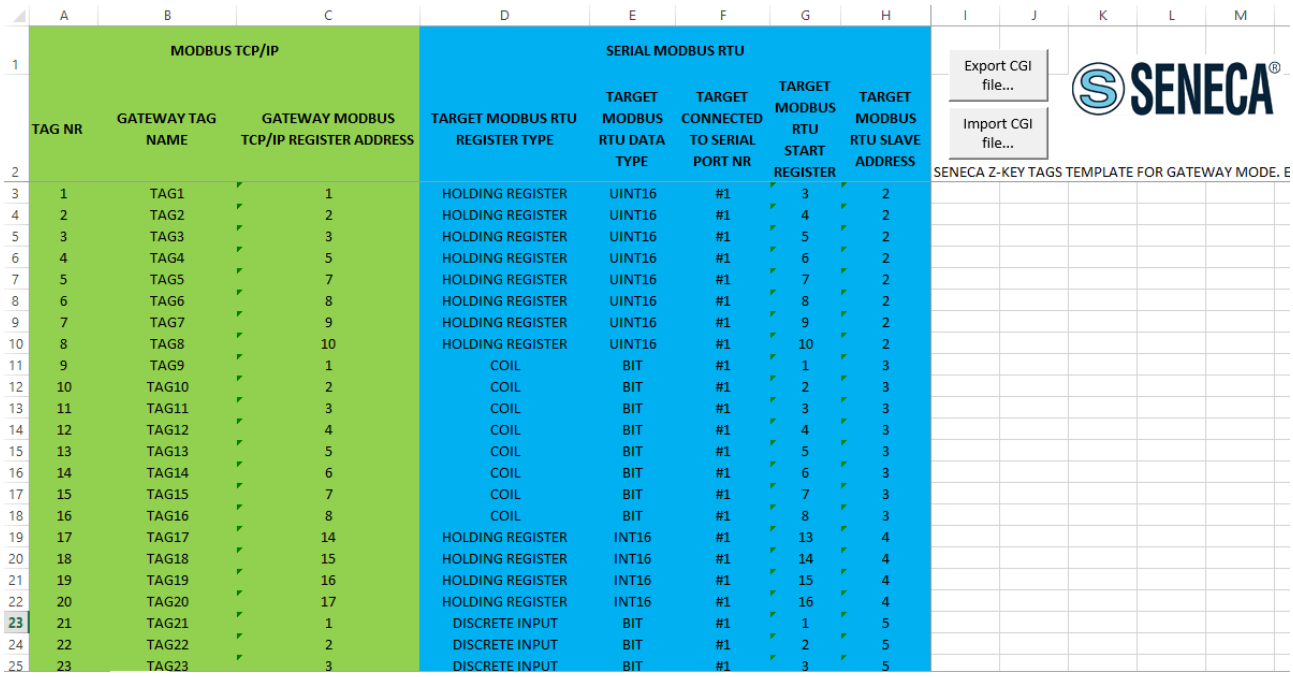


Tag information includes: The name of the TAG, the Modbus address of the TAG Gateway, the value of the Tag and the status of the TAG:
OK = TAG free of errors
FAIL_TO = TAG reading time out
DELAYED = Once the set retry number has been reached, the polling of the tag is delayed (the tag will be interrogated again after the configured quarantine time)
EXC = Modbus protocol exception response

10.4. Using the Microsoft Excel™ template for setting tags

The preparation of the Tag configuration can be a frustrating operation if done with the webserver, so a software and a Microsoft Excel™ Template are available to create a .bin file to import into the gateway.

The model can be freely downloaded from the Seneca website.

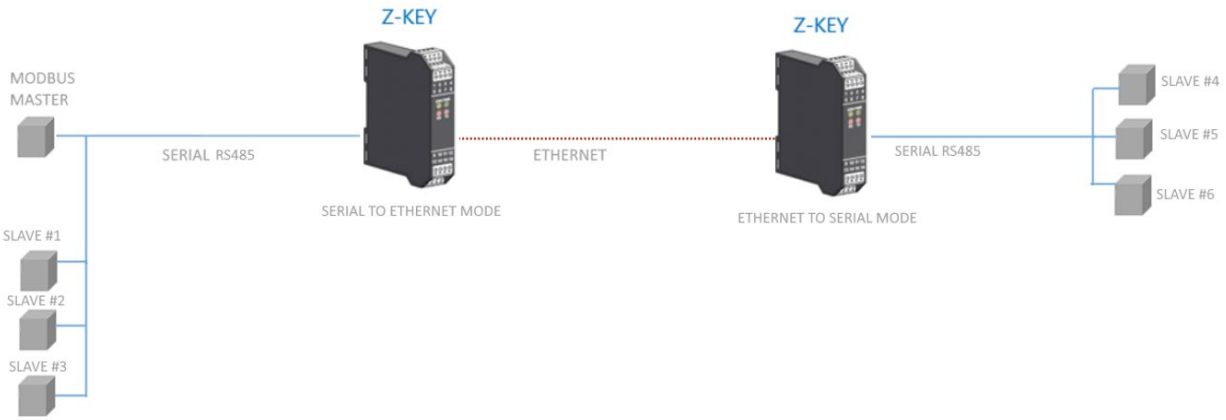


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	MODBUS TCP/IP			SERIAL MODBUS RTU					Export CGI file...		SENECA		
2	TAG NR	GATEWAY TAG NAME	GATEWAY MODBUS TCP/IP REGISTER ADDRESS	TARGET MODBUS RTU REGISTER TYPE	TARGET MODBUS RTU DATA TYPE	TARGET CONNECTED TO SERIAL PORT NR	TARGET MODBUS RTU START REGISTER	TARGET MODBUS 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					

The setup can be exported from Excel to the webserver and vice versa.

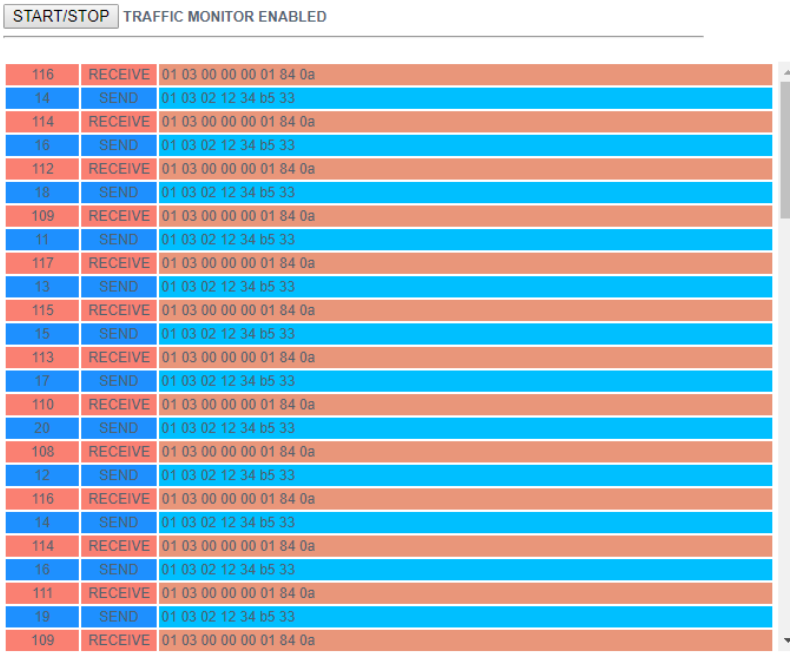
11. EXTEND THE RS485 BUS ON ETHERNET: SERIAL MODBUS ON ETHERNET AND THEN ETHERNET ON SERIAL

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



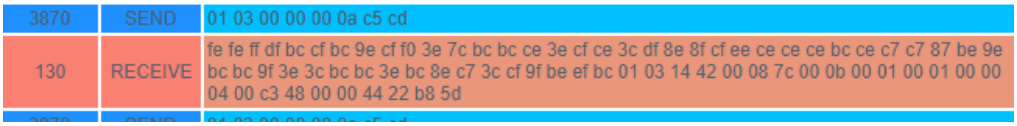
12. SERIAL TRAFFIC MONITOR

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

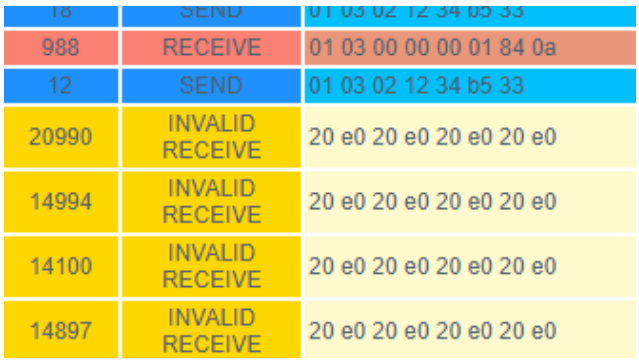


The first column is the delay in milliseconds from the last packet, the second column is the direction of the packet (received from or transmitted to), the last column is the contents of the packet in hexadecimal format. Only the serial ModBUS stream is displayed.

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):



13. 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.

14. THE DB9 RS232 CABLE

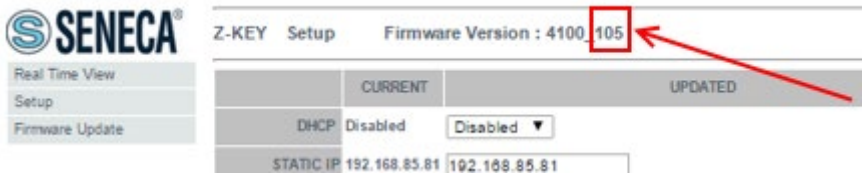
The DB9 CABLE RS232 CABLE can be obtained from Seneca (it can also be purchased from the e-commerce website www.seneca.it) for connection with a DB9 RS232 device.

15. FIRMWARE UPDATE

The devices support firmware updates via the webserver in the “Firmware Update” section of the navigation menu.

The latest firmware can be downloaded from the Seneca website.

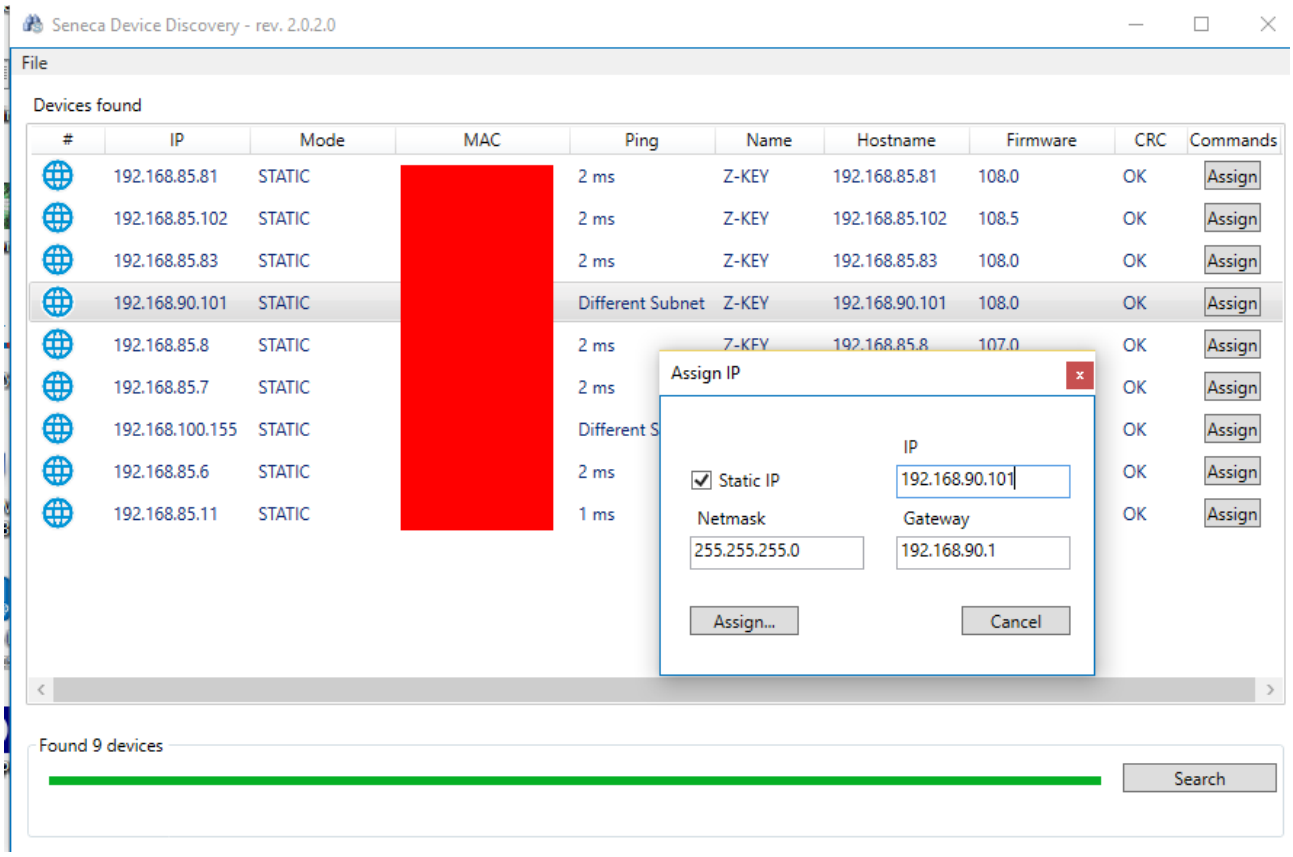
You can check the new firmware revision in the upper corner of the webserver (in the example the revision is 105):



16. SENECA DISCOVERY DEVICE

The Seneca Discovery Device Tool is available free of charge from the Seneca website.

With this software it is possible to search for Seneca devices via the Ethernet port and change the current IP configuration even if the PC has a different subnet.



ATTENTION!

For security reasons, it is possible to disable the IP change function from the "Setup" page of the Webserver.

17. 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)

17.1. 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

17.2. 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".***

17.3. 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

17.4. 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

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

17.5. 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							

17.6. 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.

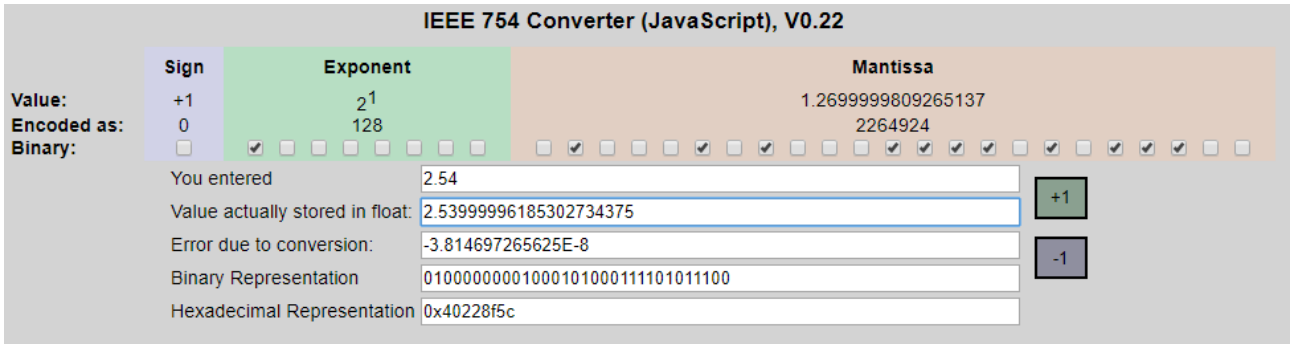
17.7. TYPE OF 32-BIT FLOATING POINT DATA (IEEE 754)

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:

0x40228F5C

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).