USER MANUAL

Z-TWS5

SENECA s.r.l.

Via Austria, 26 – 35127 – Z.I. CAMIN – PADOVA – ITALY

Tel. +39.049.8705359 – 8705408 Fax. +39.049.8706287

Web site: www.seneca.it

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

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

MI004390

Seneca Z-PC Line module: **Z-TWS5**

Table of contents

1. Preliminary information / Informazioni preliminari

WARNING!

IN NO EVENT WILL SENECA OR ITS SUPPLIERS BE LIABLE FOR ANY LOST DATA, REVENUE OR PROFIT, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, REGARDLESS OF CAUSE (INCLUDING NEGLIGENCE), ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE Z-TWS5, EVEN IF SENECA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SENECA, ITS SUBSIDIARIES AND AFFILIATES COMPANY OR GROUP OF DISTRIBUTORS AND SENECA RETAILERS NOT WARRANT THAT THE FUNCTIONS WILL MEET YOUR EXPECTATIONS, AND THAT Z-TWS5, ITS FIRMWARE AND SOFTWARE WILL BE FREE FROM ERRORS OR IT OPERATES UNINTERRUPTED.

SENECA SRL CAN MODIFY THE CONTENTS OF THIS MANUAL IN ANY TIME WITHOUT NOTICE TO CORRECT, EXTEND OR INTEGRATING FUNCTION AND CHARACTERISTICS OF THE PRODUCT.

ATTENZIONE!

IN NESSUN CASO SENECA O I SUOI FORNITORI SARANNO RITENUTI RESPONSABILI PER EVENTUALI PERDITE DI DATI ENTRATE O PROFITTI, O PER CAUSE INDIRETTE, CONSEQUENZIALI O INCIDENTALI, PER CAUSE (COMPRESA LA NEGLIGENZA), DERIVANTI O COLLEGATE ALL' USO O ALL' INCAPACITÀ DI USARE Z-TWS5, ANCHE SE SENECA AVVISATA DELLA POSSIBILITÀ DI TALI DANNI.

SENECA, LE SUSSIDIARIE O AFFILIATE O SOCIETÀ DEL GRUPPO O DISTRIBUTORI E RIVENDITORI SENECA NON GARANTISCONO CHE LE FUNZIONI SODDISFERANNO FEDELMENTE LE ASPETTATIVE E CHE Z-TWS5, IL SUO FIRMWARE E SOFTWARE SIA ESENTE DA ERRORI O CHE FUNZIONI ININTERROTTAMENTE.

SENECA SRL PUO' MODIFICARE IL CONTENUTO DI QUESTO MANUALE IN QUALUNQUE MOMENTO E SENZA PREAVVISO AL FINE DI CORREGGERE, ESTENDERE O INTEGRARE FUNZIONALITA' E CARATTERISTICHE DEL PRODOTTO.

2. Features

Z-TWS5 is a programmable, communication oriented PLC.

The device is based on a 32bits ARM-Cortex-A8 processor, equipped with the Windows CE operating system (WinCE 6.0).

The Z-TWS5 Codesys PLC is programmable according to the IEC61131-3 standard, by means of the Codesys development environment.

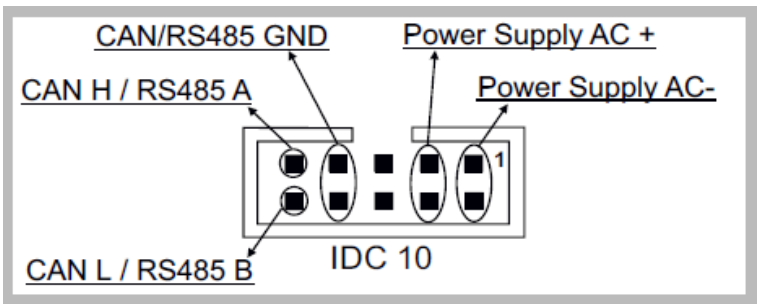
3. Technical specifications

COMMUNICATION PORTS	
RS 485	Maximum Baud rate 115 Kbps COM 4 (screw terminals 4-5-6) COM 3 (screw terminals 10-11-12) COM 2 (screw terminals 1-2-3 or IDC10 connector) COM 1 (removable 4 pin connector USB mini, as an alternative to RS232)
RS 232	Maximum Baud rate 115 Kbps COM 1 (removable 4 pin connector Usb mini, as an alternative to RS485)
CAN	CAN bus port 2.0A and 2.0B (COM-0) (IDC10 connector)
Ethernet 1 and Ethernet 2	Ethernet 10/100 Mbps Two RJ45 connectors on front-panel Maximum connection length 100 m.
USB #1 HOST	Plug-in: USB type A
USB #2 On The Go	Plug-in: micro USB
CPU AND MEMORY	
Microprocessor	ARM Cortex A8 600Mhz
Memories	256 Mbytes of RAM DDR2 128 Mbyte of FLASH 64 Kbytes of FeRAM

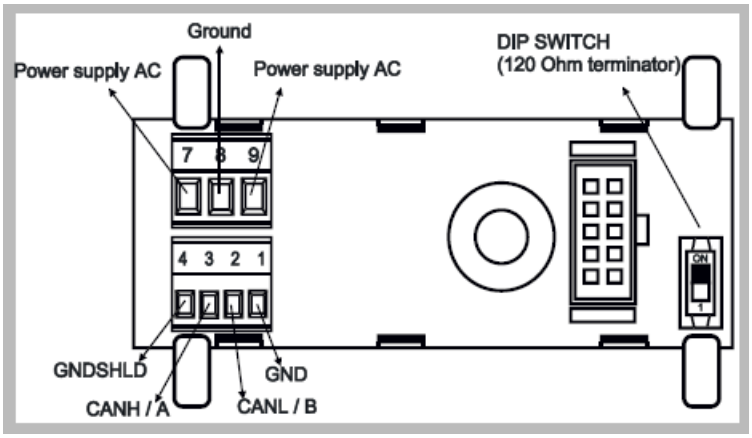
Slot for external memory	Micro SD card: max 32 Gbytes
POWER SUPPLY	
Power supply	10..40 Vdc or 19..28 Vac @ 50..60 Hz
Consumption	Max 6 W
ENVIRONMENTAL CONDITIONS	
Temperature	-0..+55 °C
Humidity	30..90 % @ 40 °C not condensing
Storage temperature	-20..+85 °C
Degree protection	IP20
CONNECTIONS	
Connections	Removable 3 way screw terminals, 5.08 pitch Rear IDC10 connector for DIN 46277 rail Removable 4 pin connector, two RJ45 connectors, USB and micro USB connectors Plug in: micro SD card
BOX / DIMENSIONS	
Dimensions	L:100 mm; H:112 mm; W:35 mm
Case	Nylon 6 with 30% fiberglass field, self extinguishing class V0, black color

4. Electrical Connections

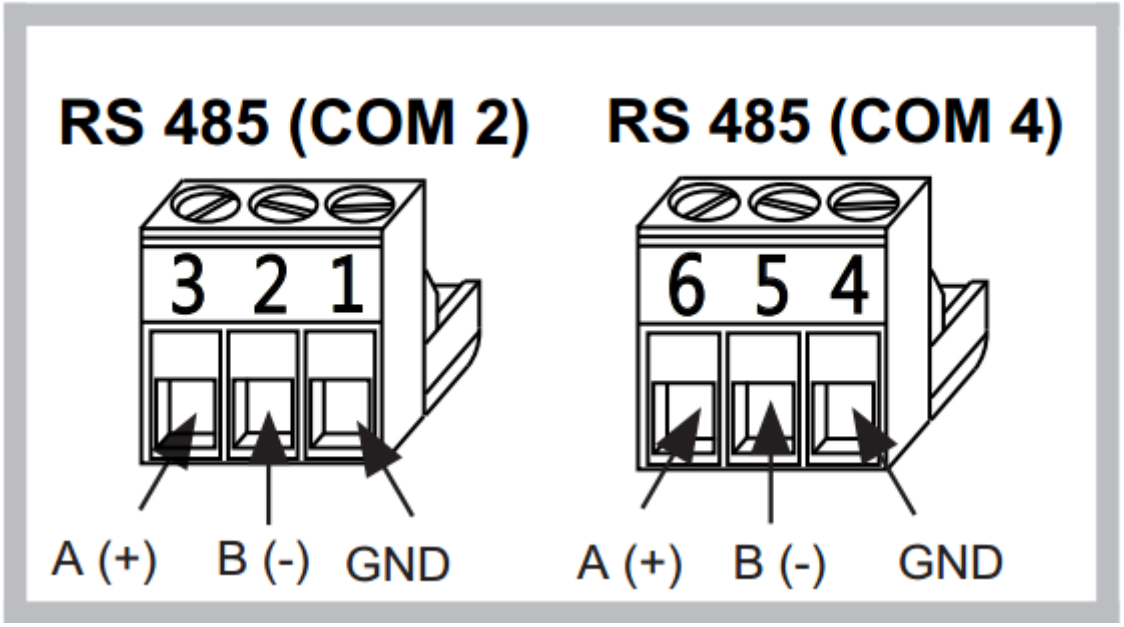
Power Supply and Modbus interface are available by using the bus for the Seneca DIN rail, by the rear IDC10 connector or by Z-PC-DINAL1-35 accessory. The following picture shows the meaning of the IDC10 connector pins. Power supply is available only from the rear connector.



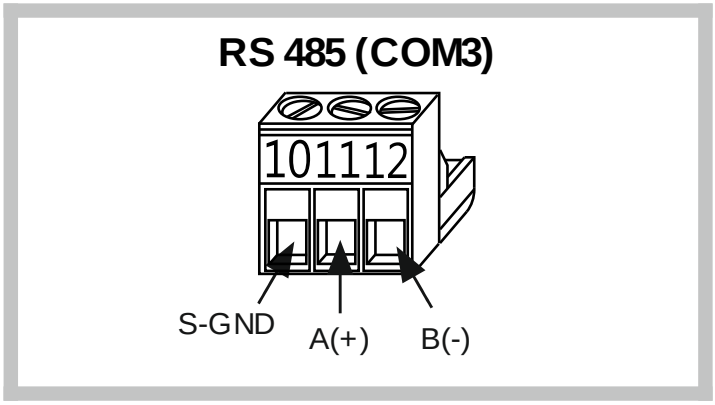
If ***Z-PC-DINAL1-35*** accessory is used, the power supply signals and communication signals may be provided by the terminals block into the DIN rail support. In the following figure the meaning and the position of the terminal blocks are shown. The DIP-switch that sets the 120 Ω terminator is used only for CAN communication.
GNDSHLD: Shield to protect the connection cables (recommended).



The Z-TWS5 has three RS 485 serial ports for Modbus communication: COM 4 , COM3 and COM 2. The RS485 connection for COM 2 can be set up by means of the corresponding screw terminals or by the IDC10 connector. To select RS 485 on IDC10 connector, put the SW1 DIP-switch on OFF position.



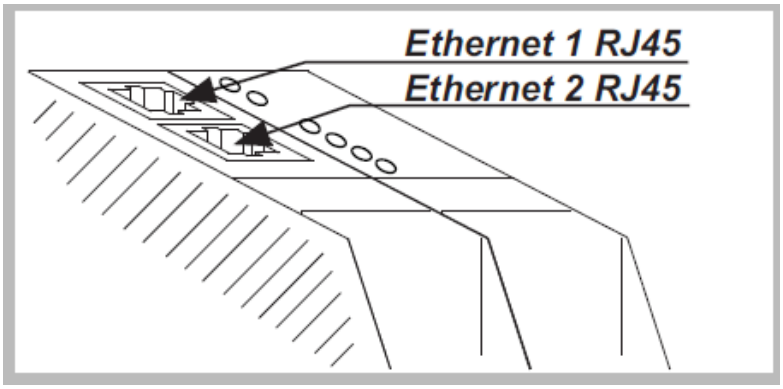
The Z-TWS5 has RS-485 COM3 port available at screw terminals 10-11-12.Warning!.
As you can see from the next figure the signals A(+) and B(-) has the sequence inverted!



The Z-TWS5 has a USB HOST type A connector, that can be used as an external USB memory.

The Z-TWS5 has a USB On The Go connector, with micro-USB plug-in.

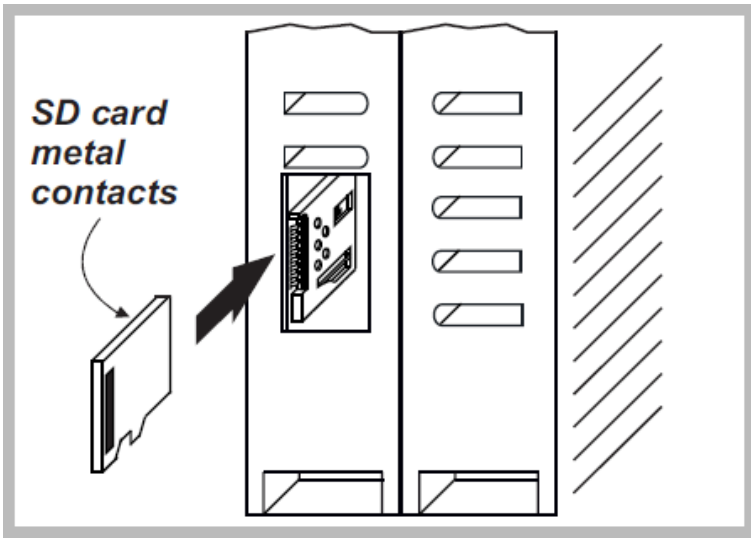
The Z-TWS5 has two ethernet ports with RJ45 connectors on the front panel. The two ports are internally connected in HUB/SWITCH mode. The two ports have the same MAC ID.



Through an USB special cable connector, the Z-TWS5 provides a serial RS232 port or, as an alternative, RS485 port.
In order to select the RS232 port on the 4 pin removable connector, we must select by software. To select the RS485 port on the 4 pin removable connector, , we must select by software too.
The cable length for the RS232 interface must be less than 3 meters.

The Z-TWS5 has a plug-in connector for micro SD card placed in the side part of the case.
To insert the SD card into the connector, be sure that the SD card is oriented with metal contacts facing towards left (with reference to the figure).

The SD card can have each class.



5. LEDs signaling

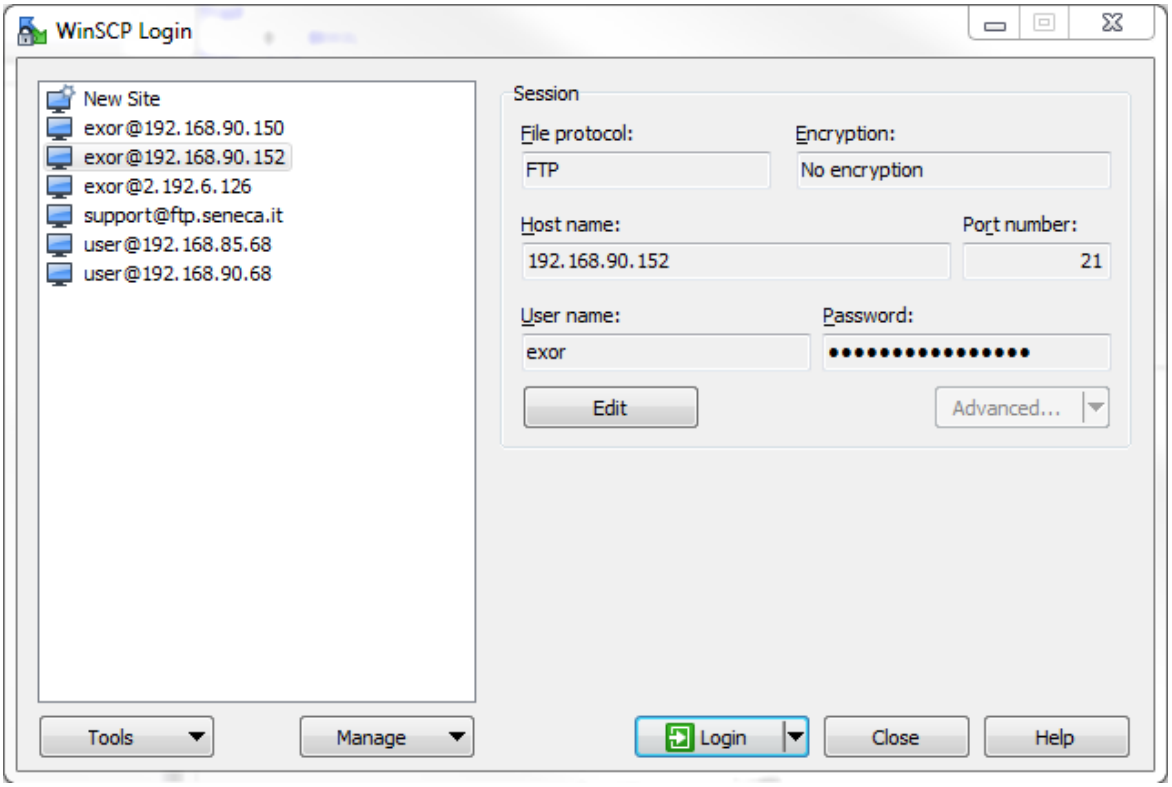
LED	STATUS	LED meaning
PWR Green	ON	The module is power on
RUN Red	Blinking	The module is ready for use
LINK1 Yellow	ON OFF	Ethernet 1 connection detected Ethernet 1 connection absent
ACT1 Green	Blinking OFF	There is data activity (Ethernet 1) There is no data activity (Ethernet 1)
LINK2 Yellow	ON OFF	Ethernet 2 connection detected Ethernet 2 connection absent
ACT2 Green	Blinking OFF	There is data activity (Ethernet 2) There is no data activity (Ethernet 2)
RX1-2-4 Red	Blinking ON	Data reception (COM 1-2-4) Check the connection (COM 1-2-4)
TX1-2-4 Red	Blinking ON	Data transmission (COM 1-2-4) Check the connection (COM 1-2-4)

6. FTP

To easily access Z-TWS5 by means of FTP, you can use the WINSCP™ program; you can free download WINSCP™ from:

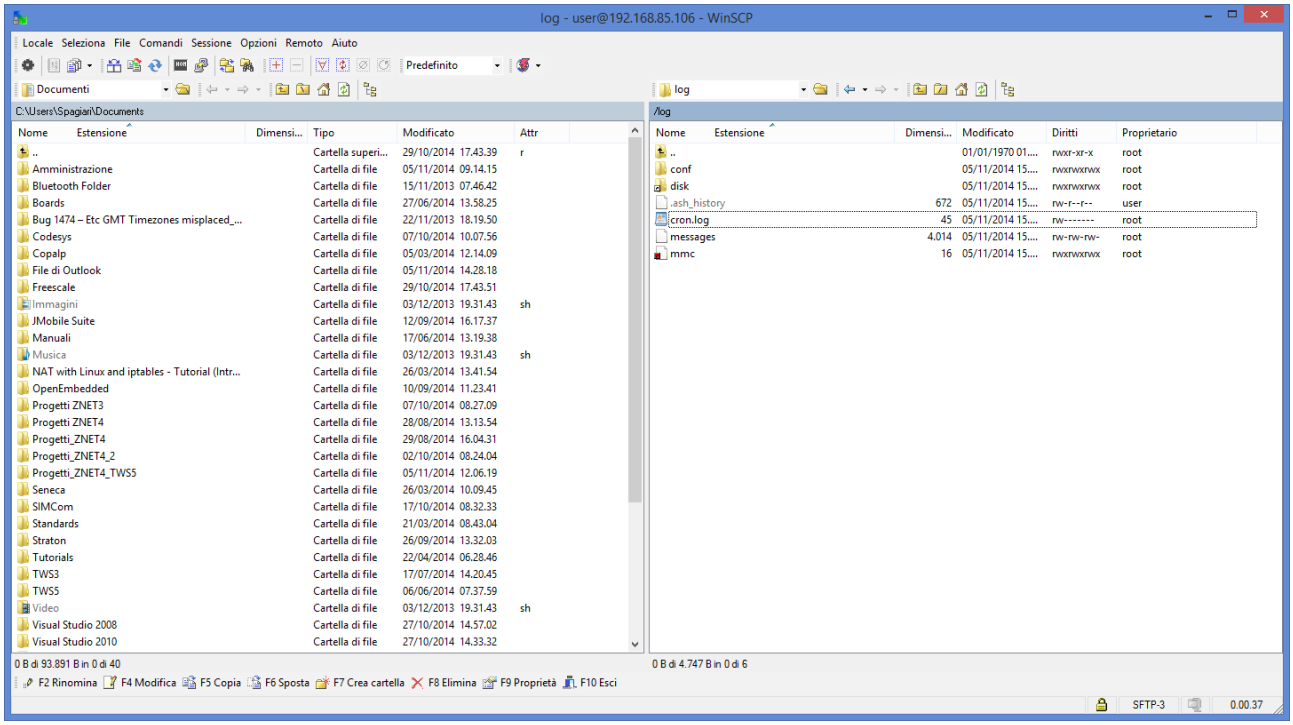
http://winscp.net/eng/download.php

You must set the connection as in the following figure (the screenshot shows a connection to the 192.168.90.152 IP address):



Note that the credentials (username and password) are the same (“exor”, “exor”).

After clicking the “Access” button, you will get a new window, as in the following screenshot; on the right, you can copy and delete files directly to/from the Z-TWS5.



7. Codesys PLC

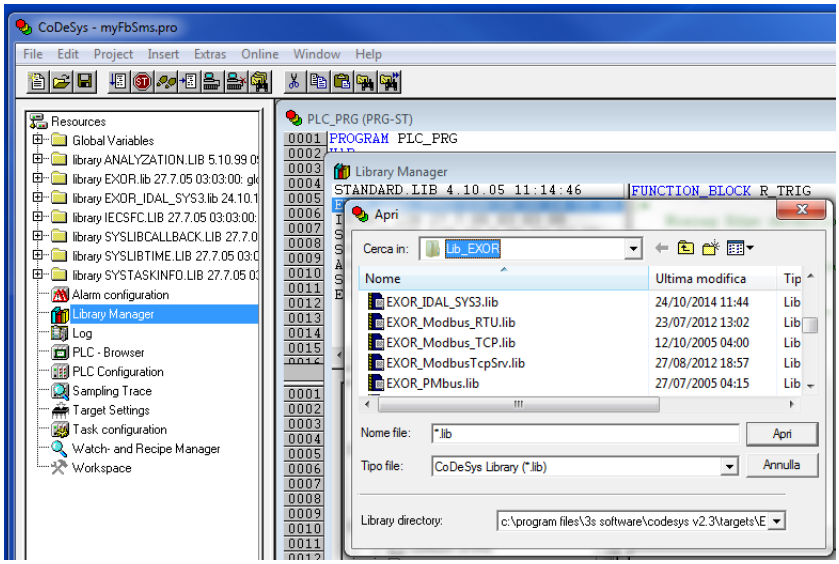
Z-TWS5 Codesys PLC version provides the full support for IEC 61131-3 PLC Standard; an Integrated Development Environment (IDE) is available for Windows™ and Linux PCs.

The Codesys Integrated Development Environment includes several tools such as: a fieldbus configuration tool, an analog signal editor and editors compliant with the five languages of the IEC 61131-3 Standard: Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), Structured Text (ST), Instruction List (IL).

With Codesys IDE, it's simple to write, download and debug IEC 61131-3 code.

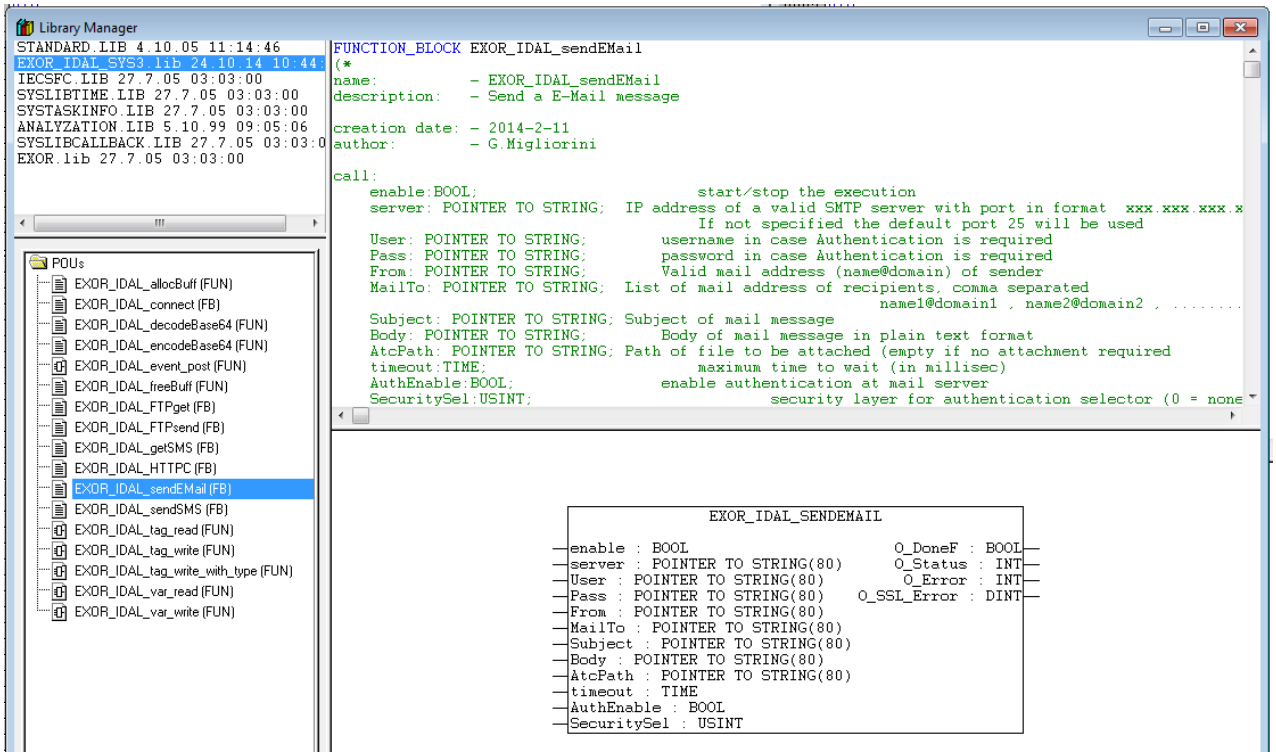
7.1. Writing, downloading and running the first program

First, we must add the Seneca Library (file *EXOR.zip*) to the IDE:
Example scompact in the path: C:\Program Files\3s software\codesys v2.3\targets\



Import the Library (double click on “Library Manger” and left click on one item of library then select “Additional library…” item menu’. Choice “EXOR_IDAL_SYS3.lib”):

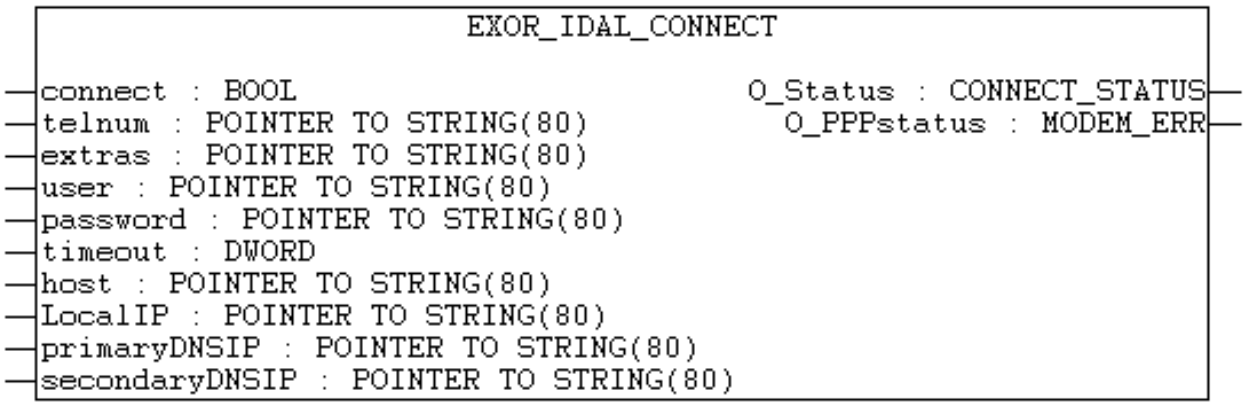
Now, we can use the library.



8. Seneca Function Blocks

To let the users exploit Z-TWS5 features in their IEC 61131-3 programs, Seneca has developed a set of "Function Blocks", supplied with the Seneca library for Codesys.

In this appendix, Seneca FBs are listed, providing a description of input/output parameters and some notes for each of them.



8.1. PPPconnect

The PPP_CONNECT FB performs PPP connection setup or release.

The FB has the following input parameters:

connect: BOOL;

on rising edge start a connection. On falling edge disconnect

telnum: POINTER TO STRING;

telephone number to use in dial-up

extras: POINTER TO STRING;

extra string for modem setup

user: POINTER TO STRING;

username known by server

password: POINTER TO STRING;

password know by server for given username

timeout:DWORD;

timeout in seconds

host: POINTER TO STRING;

string where the IP number of server will be stored

LocalIP: POINTER TO STRING;

string where the local IP assigned by the Host is stored

primaryDNSIP: POINTER TO STRING;

string where the primary DNS IP assigned by the Host is stored

secondaryDNSIP: POINTER TO STRING;

string where the secondary DNS IP assigned by the Host is stored

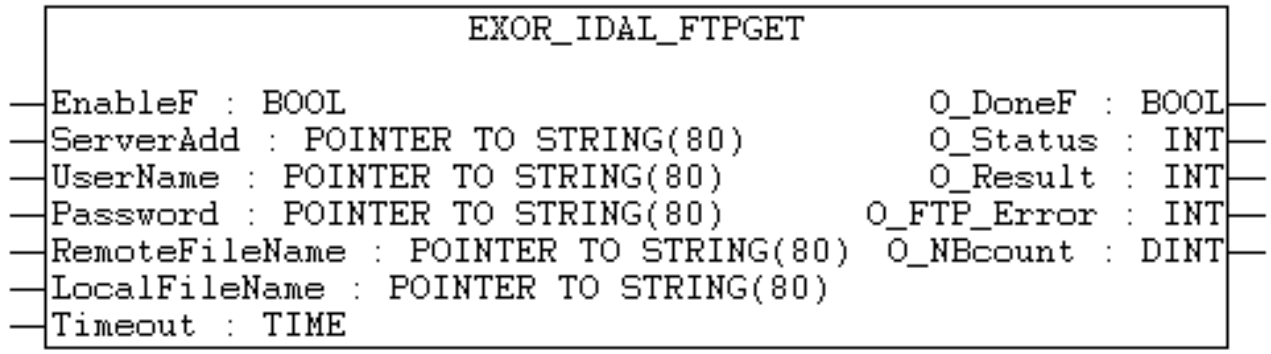
The FB has the following output parameter:

O_Status:

CONNECT_STATUS;

O_PPPstatus:

MODEM_ERR;



8.2. FTPget

Connects to an FTP server and retrieves a file. This FB posts a "get request" to the thread which performs the actual transfer.

The FB has the following input parameters:

EnableF:BOOL;

Rising edge starts operations.

ServerAdd: POINTER TO STRING;

Address of FTP server, for example: "192.168.10.138:21", port parameter (":21" in the example) is optional and defaults to 21.

UserName:POINTER TO STRING;

Username for logging into FTP Server.

Password:POINTER TO STRING;

Password for logging into FTP Server

RemoteFileName: POINTER TO STRING;

Remote File Name (name of the file on FTP Server file system).

LocalFileName: POINTER TO STRING;

Local File Name (name of the file on local file system)

Timeout:TIME;

Timeout for the operation (in millisec)

The FB has the following output parameter:

O_DoneF:BOOL;

Becomes true at the completion of the operation (becomes true also in case of errors).

O_Status:INT;

Thread Status (see table below).

O_Result:INT;

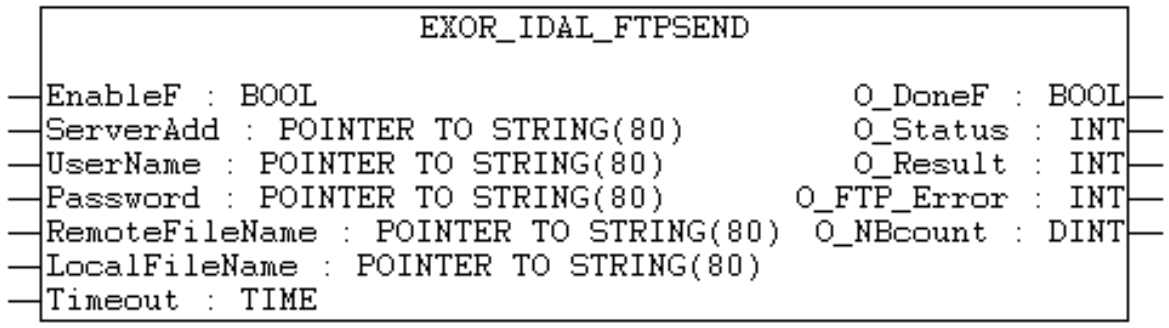
Operation result (see table below).

O_FTP_Error:INT;

FTP Error code (valid in the case "Result Output" is FTP_CLIENT_RESULT_ERR_FTP_ERROR).

O_NBcount:DINT;

Progressive count of transferred bytes.



8.3. FTPsend

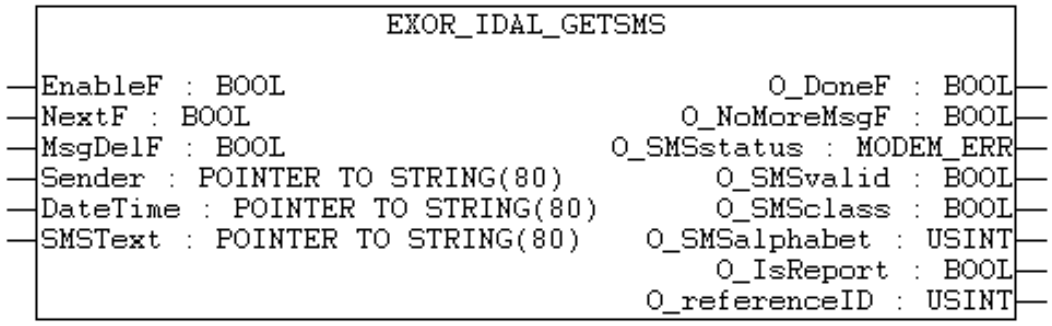
Connects to an FTP server and sends a file. This FB posts a "send request" to the thread which performs the actual transfer.

The FB has the following input parameters:

EnableF:BOOL;
Rising edge starts operations.
ServerAdd: POINTER TO STRING;
Address of FTP server, for example: "192.168.10.138:21", port parameter (":21" in the example) is optional and defaults to 21.
UserName:POINTER TO STRING;
Username for logging into FTP Server.
Password:POINTER TO STRING;
Password for logging into FTP Server
RemoteFileName: POINTER TO STRING;
Remote File Name (name of the file on FTP Server file system).
LocalFileName: POINTER TO STRING;
Local File Name (name of the file on local file system)
Timeout:TIME;
Timeout for the operation (in millisec)

The FB has the following output parameter:

O_DoneF:BOOL;
Becomes true at the completion of the operation (becomes true also in case of errors).
O_Status:INT;
Thread Status (see table below).
O_Result:INT;
Operation result (see table below).
O_FTP_Error:INT;
FTP Error code (valid in the case "Result Output" is FTP_CLIENT_RESULT_ERR_FTP_ERROR).
O_NBcount:DINT;
Progressive count of transferred bytes.



8.4. GETsms

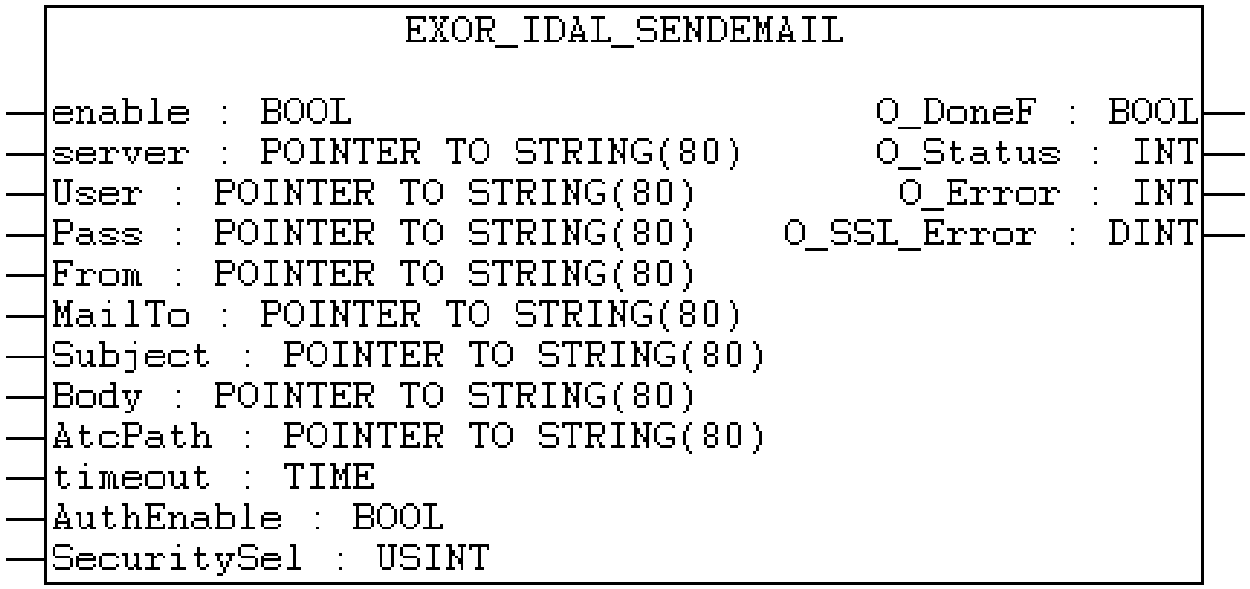
Get SMS from GSM/GPRS modem. Rising edge of "EnableF Input" starts operations. As soon the first SMS has been fetched, the "DoneF Ouput" becomes true and message text is available at "Text Output". At this point, a rising edge on "NextF Input" causes another SMS to be fetched. "DoneF Ouput" becomes false and is rose again as soon the second SMS has been fetched.

The FB has the following input parameters:

EnableF:BOOL;
Rising edge starts get procedure. Falling edge stop operations.
NextF: BOOL;
Rising edge causes next message to be read
MsgDelF: BOOL;
If true then messagge is deleted after being read.
Sender: POINTER TO STRING;
string where to store Telephone number of SMS Sender.
DateTime: POINTER TO STRING;
string where to store SMS Date & Time in received format.
SMSText: POINTER TO STRING;
string where to store SMS Text.

The FB has the following output parameters:

O_**DoneF:**BOOL;
TRUE when next message is available
O_**NoMoreMsgF:**BOOL;
TRUE when there are no more SMS to read
O_**SMSstatus:**MODEM_ERR;
status of modem operations (see below)
O_**SMSvalid:**BOOL;
TRUE if a valid SMS was read
O_**SMSclass:**BOOL;
FALSE=NO CLASS, TRUE=CLASS 0
O_**SMSalphabet:**USINT;
0=DEFAULT, 1=8bit, 2=UCS2
O_**IsReport:**BOOL;
TRUE if SMS is a Receive Report
O_**referenceID:**USINT;
reference Number of SMS if O_IsReport=TRUE



8.5. SENDmail

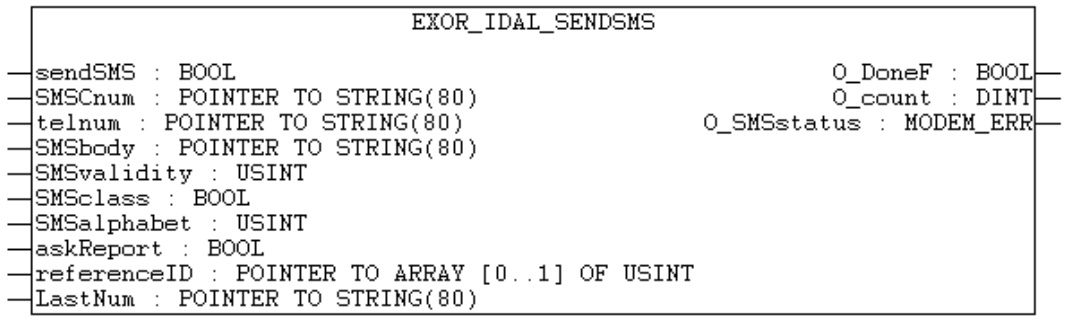
Send a E-Mail message.

The FB has the following input parameters:

enable:BOOL;
start/stop the execution
server: POINTER TO STRING;
IP address of a valid SMTP server with port in format xxx.xxx.xxx.xxx:yyyy.
If not specified the default port 25 will be used
User: POINTER TO STRING;
username in case Authentication is required
Pass: POINTER TO STRING;
password in case Authentication is required
From: POINTER TO STRING;
Valid mail address (name@domain) of sender
MailTo: POINTER TO STRING;
List of mail address of recipients, comma separated name1@domain1 , name2@domain2
Subject: POINTER TO STRING;
Subject of mail message
Body: POINTER TO STRING;
Body of mail message in plain text format
AtcPath: POINTER TO STRING;
Path of file to be attached (empty if no attachment required
timeout:TIME; maximum time to wait (in millisec)
AuthEnable:BOOL;
enable authentication at mail server
SecuritySel:USINT;
security layer for authentication selector (0 = none ; 1 = use SSL; 2 = use TLS) (NOTE: TLS is currently **unsupported**)

The FB has the following output parameter:

O_DoneF:BOOL;
TRUE when finished
O_Status:INT;
status of operation when finished see below table
O_Error:INT;
SMTP error when Status > 2
O_SSL_Error:INT;
SSL error code when Status = 5



8.6. SENDsms

Sends SMS GSM/GPRS modem.

The FB has the following input parameters:

sendSMS:BOOL;
on rising edge start send procedure. On falling edge stop operations
SMSCnum: POINTER TO STRING;
telephone number of SMS service center, including international prefix (es. +30349.....). If NULL string, SMSC is not programmed
telnum: POINTER TO STRING;
telephone number od destination phone, including international prefix. Several destinations can be indicated, sepatared by commas
SMSbody: POINTER TO STRING;
message body (max. 160 chars if default alphabet, 140 if 8bit alphabet, 70 if UCS2 alphabet)
SMSvalidity:USINT;
validity time coded according to SMS standard (es. 169 = 3 days)
SMSclass:BOOL;
FALSE=NO CLASS, TRUE=CLASS 0
SMSalphabet:USINT;
0=DEFAULT, 1=8bit, 2=UCS2
askReport:BOOL;
TRUE if ReceiveReport is requested
referenceID:POINTER TO ARRAY[0..n-1] OF USINT;
string where the reference Number of SMS, generated by modem, is stored the user must define an array with enough elements to store referenceID for each destination
LastNum: POINTER TO STRING;
string containing the current destination number while the operation is performed

The FB has the following output parameter:

```
O_DoneF:BOOL;
        TRUE when completed
O_count:DINT;
        numner of SMS delivered
O_SMSstatus:MODEM_ERR;
        status of modem operations (see below)
```

9. Updating the firmware by a USB pen

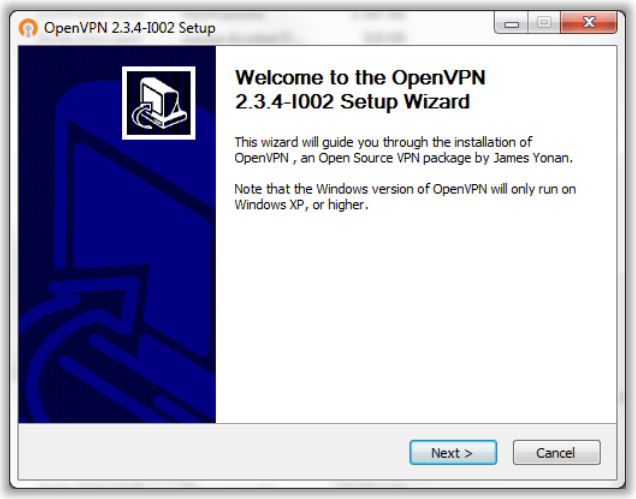
10. Open VPN Configuration

OpenVPN is a full-featured SSL VPN which implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or username/password credentials, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface.

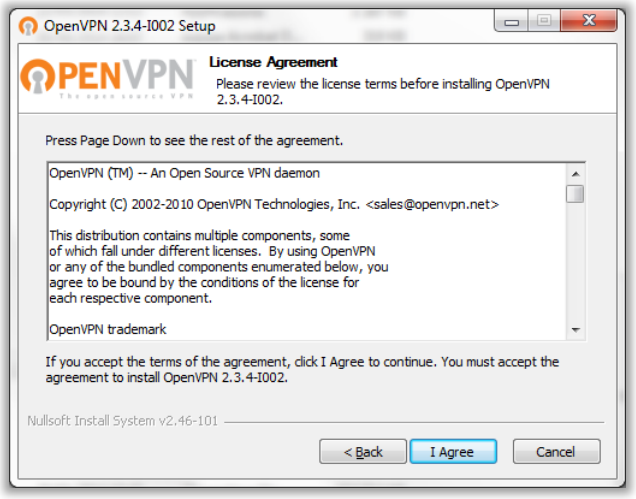
10.1. OpenVPN server installation (for Windows)

OpenVPN server installer for Windows is available for download at
https://openvpn.net/index.php/open-source/downloads.html

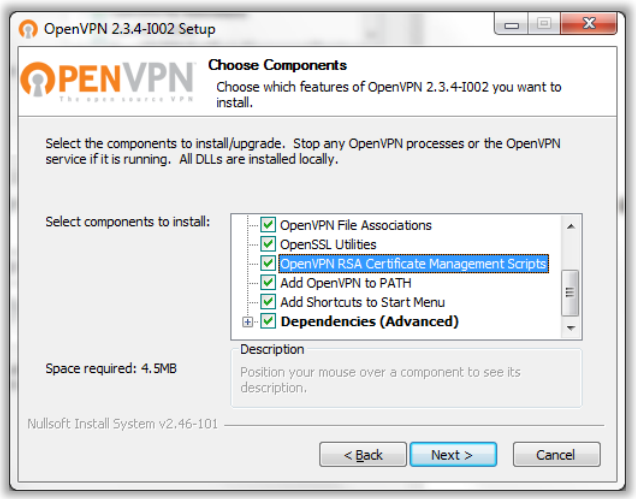
Follow the installation wizard



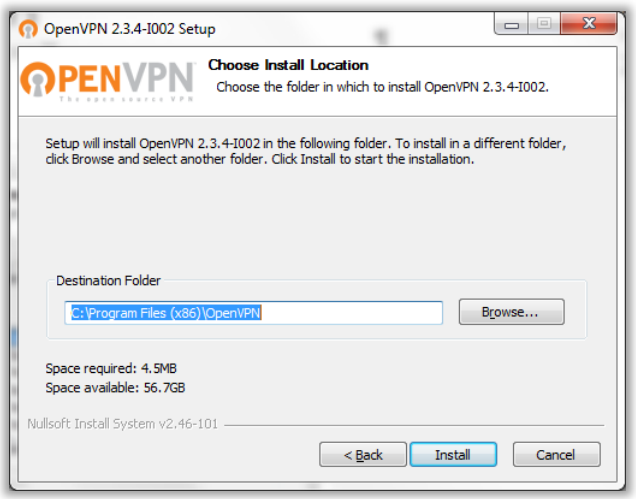
Accept License Agreement



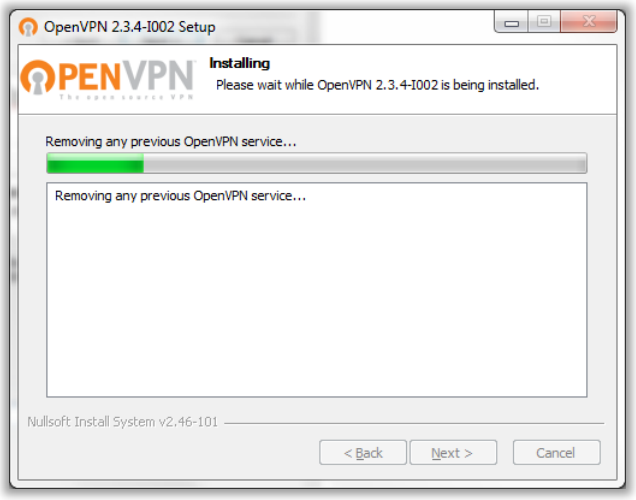
Select “OpenSSL Utilities” and “OpenVPN RSA Certificate Management Scripts” from the check list (they are unchecked by default)



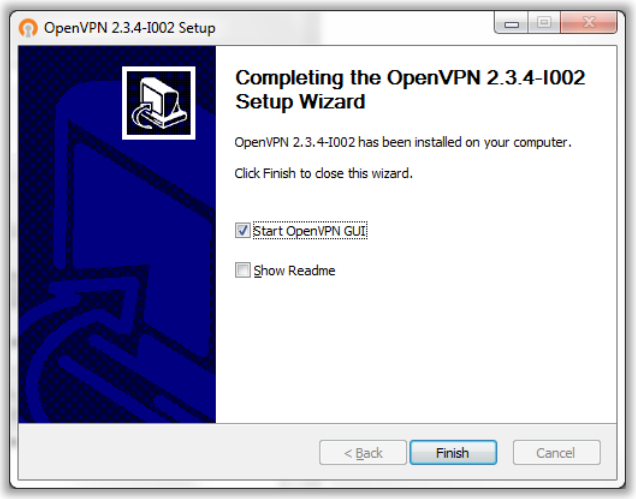
Choose installation folder



Wait installation completion



Installation completed



10.2. OpenVPN server configuration

10.2.1. Overview

The first step in building an OpenVPN 2.x configuration is to establish a PKI (public key infrastructure). The PKI consists of:

- **a separate certificate (also known as a public key) and private key for the server and each client, and**
- **a master Certificate Authority (CA) certificate and key which is used to sign each of the server and client certificates.**

OpenVPN supports bidirectional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established.

Both server and client will authenticate the other by first verifying that the presented certificate was signed by the master certificate authority (CA), and then by testing information in the now-authenticated certificate header, such as the certificate common name or certificate type (client or server).

This security model has a number of desirable features from the VPN perspective:

- **The server only needs its own certificate/key it doesn't need to know the individual certificates of every client which might possibly connect to it.**
- **The server will only accept clients whose certificates were signed by the master CA certificate (which we will generate below). And because the server can perform this signature verification without needing access to the CA private key itself, it is possible for the CA key (the most sensitive key in the entire PKI) to reside on a completely different machine, even one without a network connection.**
- **If a private key is compromised, it can be disabled by adding its certificate to a CRL (certificate revocation list). The CRL allows compromised certificates to be selectively rejected without requiring that the entire PKI be rebuilt.**
- **The server can enforce client-specific access rights based on embedded certificate fields, such as the Common Name.**

Note that the server and client clocks need to be roughly in sync or certificates might not work properly.

10.3. Generate the master Certificate Authority (CA) certificate & key

In this section we will generate a master CA certificate/key, a server certificate/key, and certificates/keys for 3 separate clients.

For PKI management, we will use easy-rsa, a set of scripts which is bundled with OpenVPN 2.2.x and earlier. If you're using OpenVPN 2.3.x, you need to download easy-rsa separately from here:

https://github.com/OpenVPN/easy-rsa

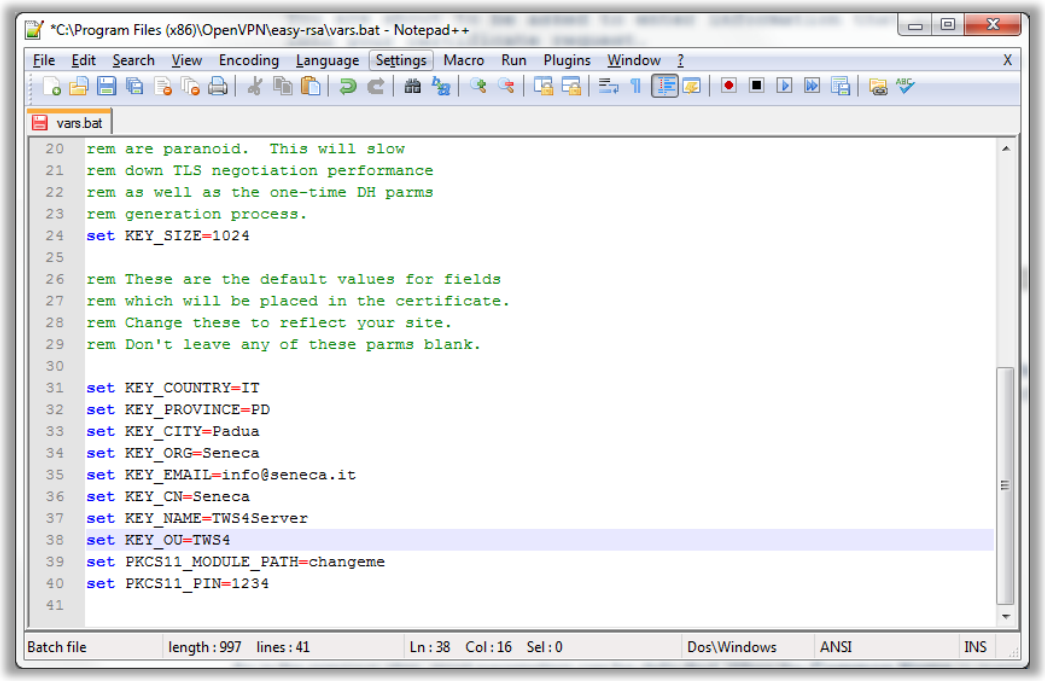
If you are using Windows the openSSL in binary format is is available for download at:

https://indy.fulgan.com/SSL/

If you are using Windows, open up a Command Prompt window and cd to \Program Files\OpenVPN\easy-rsa. Run the following batch file to copy configuration files into place (this will overwrite any preexisting vars.bat and openssl.cnf files):

```
init-config
```

Now edit the vars file (called vars.bat on Windows) and set the KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG, and KEY_EMAIL parameters. Don't leave any of these parameters blank.



Next, initialize the PKI.with commands

```
vars.bat
```

```
clean-all.bat
build-ca.bat
```

The final command (build-ca) will build the certificate authority (CA) certificate and key by invoking the interactive openssl command:

```
Generating a 1024 bit RSA private key
.............++++++
...........++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KG]:
State or Province Name (full name) [NA]:
Locality Name (eg, city) [BISHKEK]:
Organization Name (eg, company) [OpenVPN-TEST]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:OpenVPN-CA
Email Address [me@myhost.mydomain]:
```

Note that in the above sequence, most queried parameters were defaulted to the values set in the vars or vars.bat files. The only parameter which must be explicitly entered is the Common Name. In the example above, I used "OpenVPN-CA".

10.3.1. Generate certificate & key for the server

Next, we will generate a certificate and private key for the server. On Linux/BSD/Unix:

```
build-key-server server
```

As in the previous step, most parameters can be defaulted. When the **Common Name** is queried, enter "server". Two other queries require positive responses, "Sign the certificate? [y/n]" and "1 out of 1 certificate requests certified, commit? [y/n]".

10.3.2. Generate certificates & keys for N clients

Generating client certificates is very similar to the previous step. Remember that each client certificate has an expiration date which is 3650 days (10 years) after creation by default. If you would like to change the validity period you have to change “vars.bat” file. After expiration you have to generate a new certificate for that client.

```
build-key client1
build-key client2
build-key client3
……
build-key clientN
```

Remember that for each client, make sure to type the appropriate Common Name when prompted, i.e. "client1", "client2", or "client3". Always use a unique (different) common name for each client.

10.3.3. Generate Diffie Hellman parameters

Diffie Hellman parameters must be generated for the OpenVPN server. On Linux/BSD/Unix:

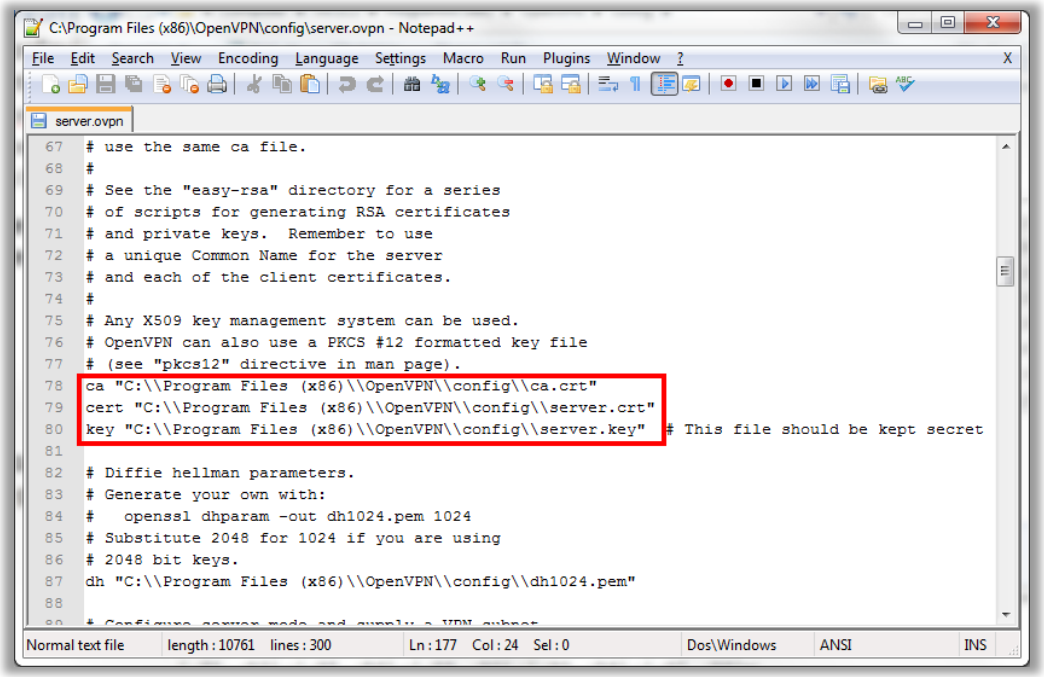
```
build-dh
```

Output:

```
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
..................+.........................................
...................+.............+..................+.........
.....................................
```

10.4. Creating configuration files for server and clients

Server configuration is contained in file “server.ovpn” provided together with this document, copy and paste this file into your “openvpn\config” directory.
Edit “server.ovpn” and change the path to the ca, cert and key if necessary.



Network subnet that will be used for vpn communication is defined with directive

```
server 10.9.7.0 255.255.255.0
```

Remember that the server takes every time .1 as last octect for his IP address. In this example the IP subnet is 10.9.7.0 so the server IP address is 10.9.7.1.

The folder ccd provided together with this document contains the client configuration files for 5 client. Copy and paste the configuration file into your ccd directory. In this example the client IP assignment is as follows:

client1	10.9.7.13	refers to the pair [13, 14]
client2	10.9.7.17	refers to the pair [17, 18]
client3	10.9.7.21	refers to the pair [21, 22]
client4	10.9.7.25	refers to the pair [25, 26]
client5	10.9.7.29	refers to the pair [29, 30]

but you can extend yourself for a large number of clients taking the last octet of ip address for **ifconfig-push** from the pairs below, note that the maximum number of clients is 64 (for windows os):

```
[  1,  2] [  5,  6] [  9, 10] [ 13, 14] [ 17, 18]
[ 21, 22] [ 25, 26] [ 29, 30] [ 33, 34] [ 37, 38]
[ 41, 42] [ 45, 46] [ 49, 50] [ 53, 54] [ 57, 58]
[ 61, 62] [ 65, 66] [ 69, 70] [ 73, 74] [ 77, 78]
[ 81, 82] [ 85, 86] [ 89, 90] [ 93, 94] [ 97, 98]
[101,102] [105,106] [109,110] [113,114] [117,118]
[121,122] [125,126] [129,130] [133,134] [137,138]
[141,142] [145,146] [149,150] [153,154] [157,158]
[161,162] [165,166] [169,170] [173,174] [177,178]
[181,182] [185,186] [189,190] [193,194] [197,198]
[201,202] [205,206] [209,210] [213,214] [217,218]
[221,222] [225,226] [229,230] [233,234] [237,238]
[241,242] [245,246] [249,250] [253,254]
```

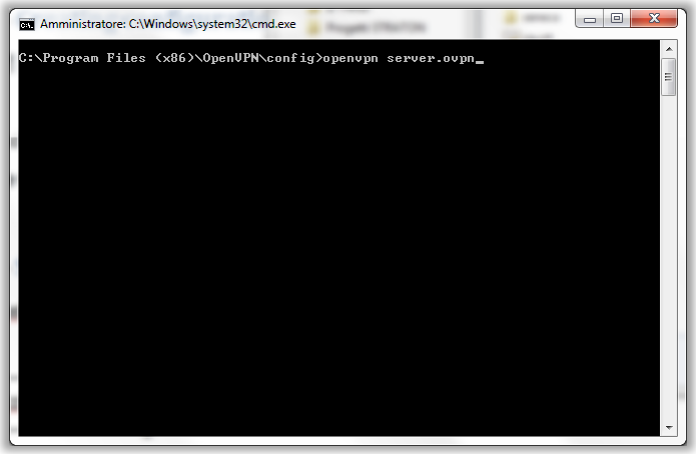
There are 2 important things that must be respected:

- **File name in ccd directory must be the same used in the common name of the client certificate and key (e.g. client1, client2, etc…)**
- **Each file must contain a row that define the IP address of the client (chosen from the pair list above) for example:**
  1.

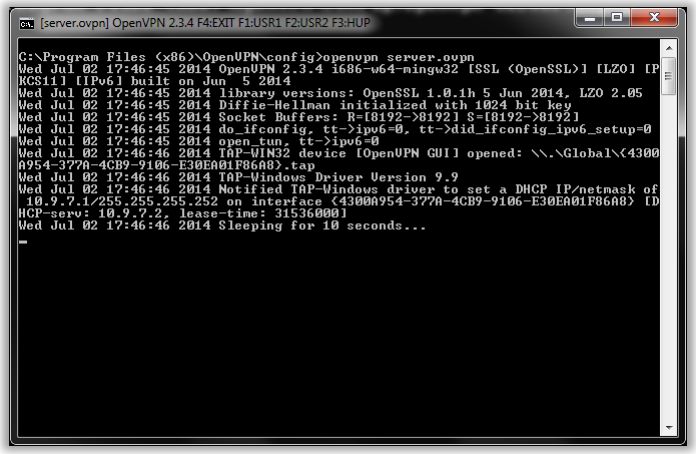
```
ifconfig-push 10.9.7.13 10.9.7.14
```

10.5. Starting up the VPN and testing for initial connectivity

To start OpenVPN Server opens a terminal window (cmd.exe), cd into your OpenVPN\config dir and run openvpn with your server.ovpn config as argument



While running OpenVPN Server will print on console all log informations (es. New client connected, etc...)



10.6. Upload client configuration files on TWS5

Open Seneca Z-NET4 and select the project that you made previously. Select the Cpu Objects and the tabulation Configurations. Enable VPN and set the parameters:

- **Server Vpn**
- **Port Udp**
- **Ca.crt**
- **Client.crt**
- **Client.key**

Confirm the modifications and send the new configuration by special Button.

Reboot the apparatus to make active the new configurations