

Z-PC LINE

# Seneca Z-NET3

User Guide



## WARNINGS

SENECA guarantees its products according to the current laws and norms and to the control system of the quality. In particular, Z-NET 2 / Z-NET 2 RTU and the libraries developed for telecontrol functions, are realized through standard informatic technologies, allowing the accessibleness toward other supervision systems. Anyway SENECA does not assume any responsibility as regards the integration of its systems with infrastructures realized or furnished by others. The user has the responsibility to guarantee, with his choices, the reliability and the functionality of the whole system of products and services. Besides SENECA does not assume any responsibility as regards the utilization of its softwares for configurations which are not appropriate to the application; SENECA does not guarantee the system enterity for automatic logics and data interchange, developed and controled by the user. Therefore SENECA does not have any responsibility as regards the choices of the persons who develop the whole automation system.

## SUMMARY

<b>1</b>	<b>INTRODUCTION</b>	<b>5</b>
<b>2</b>	<b>Z-PC LINE</b>	<b>6</b>
2.1	Z-TWS 2	6
2.2	Z-RTU	8
2.3	I/O MODULES	9
2.4	CPU VARIABLES MANAGEMENT	9
2.5	Z-NET 2	10
<b>3</b>	<b>Z-NET BASE PROJECT CONFIGURATION</b>	<b>12</b>
3.1	SENECA Z-NET 2 INSTALLATION	12
3.2	SETTING COMMUNICATION WITH CPU	13
3.2.1	CPU'S IP ADDRESS DETERMINATION	14
3.3	TCP/IP LINK CONFIGURATION	15
3.4	SENECA Z-NET 2 CONFIGURATION	17
3.4.1	Z-NET 2 NEW PROJECT CREATION	17
3.4.2	TCP/IP ADDRESS MODIFICATION	19
3.4.3	MODULES INSERTING WIZARD	20
3.4.4	ADD I/O DEVICES	23
<b>4</b>	<b>CPU CONFIGURATION</b>	<b>24</b>
4.1	CONFIGURATIONS WINDOW	24
4.2	SERIAL PORTS WINDOW	25
4.3	MODBUS WINDOW	27
4.4	PARAMETERS WINDOW	27
4.5	PPP WINDOW	29
<b>5</b>	<b>I/O MODULES CONFIGURATIONS</b>	<b>31</b>
5.1	CONFIGURATION WINDOW	31
5.2	PARAMETERS WINDOW	31
5.3	VARIABLES WINDOW	32
<b>6</b>	<b>VARIABLES</b>	<b>33</b>
6.1	VARIABLES WINDOW	33
6.2	I/O VARIABLES	34
6.2.1	ANALOG I/O VARIABLES	34
6.2.2	I/O DIGITAL VARIABLES	35
6.3	PLC VARIABLES	36
6.3.1	PLC ANALOG VARIABLES	36
6.3.2	PLC DIGITAL VARIABLES	37

---

<b>7</b>	<b>TELECONTROL</b>	<b>39</b>
<b>7.1</b>	<b>TELECONTROL FUNCTIONS</b>	<b>39</b>
7.1.1	SENDING DATA LOG FILES	40
7.1.2	SENDING THROUGH EMAIL	40
7.1.3	SEND THROUGH FTP	41
7.1.4	ALARMS	42
7.1.5	CALL ON ALARM	44
7.1.6	SMS COMMANDS	44
<b>7.2</b>	<b>SMS COMMANDS</b>	<b>46</b>
<b>7.3</b>	<b>SMS STATES</b>	<b>46</b>
<b>7.4</b>	<b>DATA LOG</b>	<b>48</b>
7.4.1	COMMON CONFIGURATION	48
7.4.2	DATA LOG CONFIGURATION	49
<b>7.5</b>	<b>SMS ALARMS AND EVENTS</b>	<b>50</b>
7.5.1	DIGITAL ALARMS	50
7.5.2	ANALOG ALARMS	51
<b>7.6</b>	<b>ALARMS AND EVENTS</b>	<b>53</b>
7.6.1	DIGITAL ALARMS	53
7.6.2	ANALOG ALARMS	54
<b>8</b>	<b>DATA RECORDER PROJECT SETTINGS</b>	<b>56</b>
<b>8.1</b>	<b>COMMON SETTINGS</b>	<b>56</b>
8.1.1	PROJECT WITH Z-TWS2	57
8.1.2	PROJECT WITH ONLY I/O MODULES	57
<b>8.2</b>	<b>CHANNELS</b>	<b>59</b>
<b>8.3</b>	<b>CALCULATED CHANNELS</b>	<b>62</b>
<b>8.4</b>	<b>MEASURE UNITS</b>	<b>64</b>
<b>9</b>	<b>ISAGRAF PROJECT</b>	<b>65</b>
<b>9.1</b>	<b>ISAGRAF FILES GENERATION</b>	<b>65</b>
<b>9.2</b>	<b>VARIABLES DATABASE AND I/O BOARDS</b>	<b>66</b>
<b>9.3</b>	<b>LIBRARIES</b>	<b>66</b>
<b>10</b>	<b>APPENDIX A: MODBUS</b>	<b>69</b>
<b>10.1</b>	<b>MODBUS MESSAGE STRUCTURE</b>	<b>69</b>
<b>10.2</b>	<b>MODBUS FUNCTIONS</b>	<b>69</b>

## 1 INTRODUCTION

This guide aims to provide the user all necessary informations to acquire a global knowledge about Seneca Z-PC System.

First of all, the main features of Z-PC line's Modules will be described ( Z-TWS, Z-RTU, I/O modules )

Then we illustrate Seneca Z-NET 2 configuration software, by giving instructions about installation's procedure, the communication with CPU and modules configuration.

Besides the user will be informed about PLC, Variables management and ISaGRAF project.

A wide section is dedicated to Telemetry and telecontrol functionalities.

## 2 Z-PC LINE

The Z-PC line has been developed to offer flexible, cost effective solutions to industry's increasingly diverse applications including distributed control, data acquisition, SCADA & telemetry. An ideal solution for process control and management using distributed I/O on plant and machinery. The I/O modules can be widely distributed or grouped together and accept all standard field signals. Modbus RTU protocol guarantees universal connectivity, so applications are limitless: data acquisition, automation, telemetry control, etc.

Features highlights are:

- Analog I/O modules: universal inputs, PID controller, up to 8 inputs per module;
- Digital I/O: up to 10 inputs or outputs per module for direct control of alarms, indicators, valves, interlocks etc.;
- Wireless Radio link modules, (Z-link, license free UHF transceivers) & standard serial converters;
- Z-TWS Control module features "soft PLC" capability, web server functionality, a 32 bit processor, 16MB flash and 8 MB RAM.

### 2.1 Z-TWS 2

Z-TWS 2 represents a powerful instrument of web automation and it can be used in many different configurations and architectures.

For example Z-TWS may be utilized in all industrial automation applications which can need large calculus capabilities, in industrial processes control and remote telecontrol.

Seneca offers 3 TWS different types, in order to satisfy a great number of user's needs and requests:

- **Bridge TCP/IP:** it is the simplest TWS's version. In this case Z-TWS acts as a bridge between PC and I/O modules. So it offers the use of a RS485 Serial Port to communicate with I/O modules, a RS232 serial port and a Ethernet port. There are not ISaGRAF libraries on it.
- **µPLC:** it is a halfway version which allows the management of small automation processes. It has the use of the same communicatin port of Bridge TCP/IP but it allows some base Isagraf functionalities.
- **Standard TWS:** it is the complete versione which will be accurately described afterward.

Standard Z-TWS is provided of four completely programmable serial ports::

- **Port 0:** RS232 which can be utilized to visualize TWS's debug or to activate a data acquisition task.
- **Port 1:** RS485 MASTER. It is used in order to communicate with I/O modules ( through ModBus RTU protocol )
- **Port 2:** Serial Port which can be set through a dip-switch as RS232 or RS485.
- **Port 3:** RS485, it can be used in order to activate an other data acquisition task
- **Ethernet Port**

Standard TWS is structured by four functional blocks:

- **PLC:** Z-TWS may be used as an integrated PLC in order to locally control small and medium automation's plants.
- **Web Server:** It allows the integration of Z-TWS system with Intranet and Internet nets

- **Datalogger:** it allows trends ( variables stored at fixed intervals of time ) and so signals' memorization.
- **RTU:** it allows to connect remotely to Z-TWS

## 2.2 Z-RTU

Z-RTU ( Remote Terminal Unit ) is an instrument with the same CPU of Z-TWS: so it is able to offer the same functionalities .

Z-RTU represents a safe, compact and versatile solution for remote control, datalogging and I/O management.

Besides it provides a considerable number of I/O channels and an internal modem ( GSM or PSTN to user' choice ).

Z-RTU is provided of the following communication ports:

- Ethernet Port for connection through TCP/IP protocol.
- PORT 0: serial port RS232 which can be used to control Z-RTU debug.
- PORT 1: serial port RS485 Modbus RTU Master to connect with I/O slave modules.
- PORT 2: serial port which can be programmed as RS232 or RS485. On the second case ( RS485 ) port 2 may be used to activate a data acquisition task through ModBus RTU protocol.
- Internal Port utilized in order to connect to intenal modem.

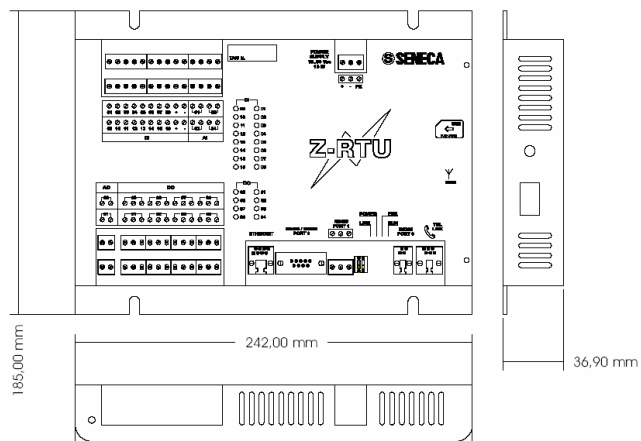
Seneca offers the possibility to choose between two different Z-RTU versions:

- **Light Version:** Z-RTU uses the same CPU of standard TWS but it is provided of only one I/O module called base.
- **Full Version:** Compared with the light version, full version is provided of another I/O module ( extension ) which is exactly equal to the base one.

In this way I/O density is doubled.

The base configuration ( Light version ) manages the following I/O channels ( all galvanically isolated ):

- 8 digital inputs ( with internal or external alimentation )
- 2 analog inputs ( 14 bits resolution and loop's alimentation to user's choice ).
- 4 digital outputs ( SPDT rele and individually isolated )
- 2 analog outputs ( voltage or current ).



**Figure 2.2: Z-RTU Dimensions**

## 2.3 I/O MODULES

Z-PC line includes a well-provided set of instruments used to manage input and output signals which are digitalized and converted to serial protocol.

Z-PC I/O modules communicate with Z-TWS or Z-RTU through serial ports RS485 and data exchange is controlled through Modbus RTU protocol.

Digital modules allows the acquisition of every kind of contacts issued from buttons, rele, proximity, alarms, reed, npn, pnp and provided of 16 or 32 bits counters.

Some modules allow the management of Universal Inputs: voltage, current, resistance, thermocouples and thermoresistances.

The analog line is completed by the PID regulator and by the module of outputs regulation.

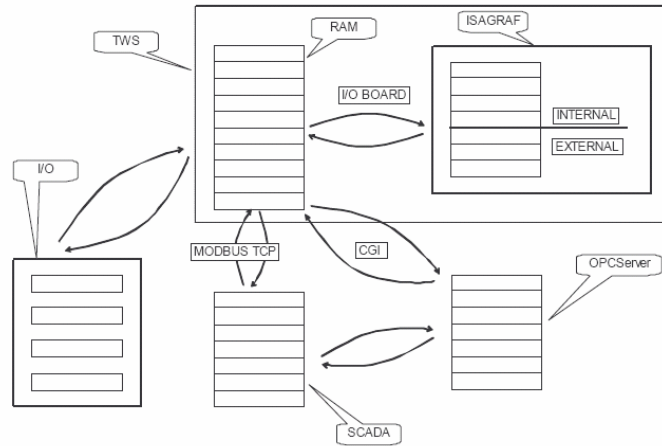
The following list shows all Z-PC line's modules with the associated I/O signals.

- **Z-D-IN**            5 digital inputs
- **Z-D-OUT**         5 digital outputs
- **Z-10-D-IN**        10 digital inputs
- **Z-10-D-OUT**      10 digital outputs
- **Z-DAQ**            Universal Analog Input
- **Z-4TC**            4 inputs for thermocouples
- **Z-4RTD**          4 inputs for thermoresistances
- **Z-3RTD**          3 inputs for thermoresistances
- **Z-4AI**            4 analog voltage or current inputs
- **Z-3AO**            3 voltage or current analog outputs
- **Z-PID**            Analog IN/OUT provided of PID regulation functionalities

Besides Z-PC line includes serial converters and radiomodems.

## 2.4 CPU VARIABLES MANAGEMENT

Variables management is described on the following scheme:



Some configuration files may be automatically created in order to allow variables acquisition through ISaGRAF, OPCServer and Movicon ( Scada ).

Variables acquisition may be performed through the following protocols:

- A. **Modbus TCP/IP**: through Ethernet Port supervision is allowed by SCADA and ISaGRAF .
- B. **HTTP**: through CGI functions.

## 2.5 Z-NET3

Seneca Z-NET 3 configuration software, based on IEC 61131 standard, has been created ad hoc for Z-PC line's modules.

This software is able to create a configuration project for a Z-TWS2 or Z-RTU module and all the I/O devices connected to it.

Through Z-NET 3 is possible to generate, modify and automatically download all CPU's configurations files.

User may also configure all the I/O devices connected to the CPU and visualize in real time the value of IN/OUT signals.

Besides Z-NET 3 allows the automatic and transparent configuration of CPU for variables management.

In particular it creates automatically variables configurations for :

- Variables Acquisition in RAM
- ISaGRAF Variables Database
- OPC Server Variables list
- SCADA ( MOVICON ) Variables list

It is also possible to configure Trend variables with their sampling time, log variables, telecontrol and telemonitoring functionalities.

Z-NET 2 can work on Windows NT ( SP4 ), Windows 2000 (SP1), Windows XP platforms and it has been developed by using microsoft.NET Framework enviroment which would have been duly preinstalled.

Therefore Z-NET 2 can be considered as the global configurator of Z-PC line's modules.

The following chapters will deal with these arguments:

- Chapter 3: Installation and configuration of Z-NET 2 system. It will be described how to create a project, how to connect to the CPU and the

modality to insert I/O modules.

- Chapter 4 : Modality of CPU configuration
- Chapter 5: Variables definition
- Chapter 6: Description of telecontrol and telemonitoring functionalities
- Chapter 7: ISaGRAF files generation and PLC project's management..

### 3 Z-NET BASE PROJECT CONFIGURATION

In the present chapter the user can find all the following instructions, necessary to correctly utilize Z-NET 2 configuration software:

- Z-NET 2 installation.
- Modalities to establish communication with Z-TWS 2 or Z-RTU
- Modalities to create a complete Z-NET 2 project.

#### 3.1 SENECA Z-NET 2 INSTALLATION

Seneca provides **Seneca Package CD** which allows installation of all softwares necessary to a complete management of CPU potentialities.

Through Seneca Package CD users may install the following softwares:

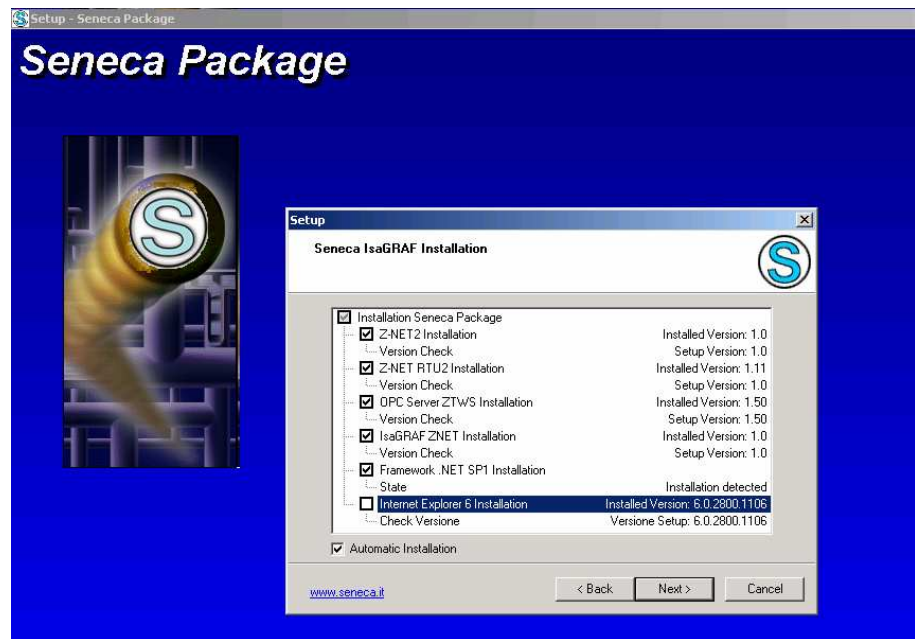
- **Seneca Z-NET 2** : CPU and I/O devices configuration.
  - **Seneca Z-NET 2 RTU**: General configuration of Z-TWS 2 or Z-RTU station and connection's parameters.
  - **ISaGRAF Z-TWS**: PLC application's management.
  - **OPC Server TWS**: Acquisition of OPC variables for telecontrol's functions
- Z-NET 2's working requires the presence of Microsoft.NET Framework 1.1 which can be installed through Seneca Package CD.

At the launch of the installation package, the following window is displayed:

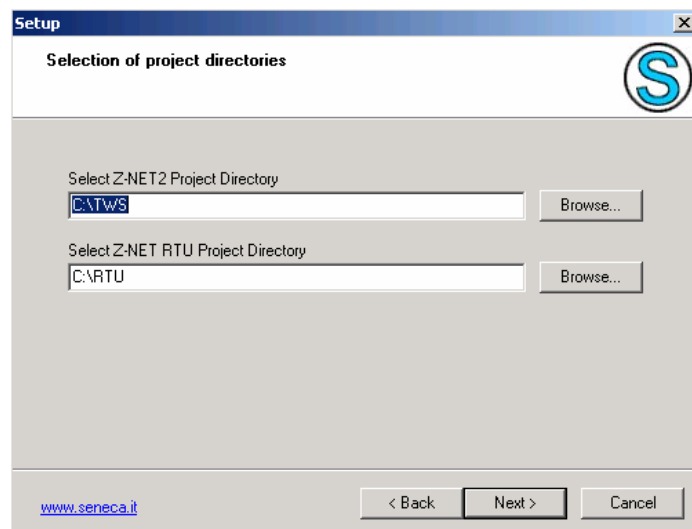


If user desires to continue it is necessary to click on NEXT.

So in the following window user may select the softwares to install:



In order to continue installation click on **Next** while to abort click on **Cancel**. The following window will appear, on which user can define the directories where Z-NET 2 and Z-NET2- RTU projects will be saved.



By clicking on **Install** in the next dialogue window, the installation of the selected softwares will start.



### 3.2 SETTING COMMUNICATION WITH CPU

In this section there will be described the following arguments:

- Procedure to determine network's parameters of CPU ( Z-TWS 2 or Z-RTU 2 )
- Procedure to make network's parameters of CPU compatible with user's local net TCP/IP.

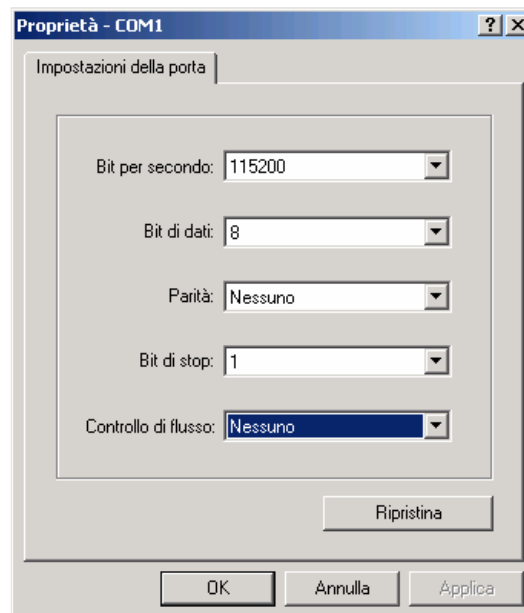
### 3.2.1 CPU's IP Address determination

Z-RTU and Z-TWS 2 are configured with the following standard network's parameters:

**IP Address: 192.168.90.101**  
**Net mask : 255.255.255.0**

Anyway if the device has already been programmed it can have a different IP Address. So in order to find CPU's correct IP address, the user may follow the next instructions:

- Connect CPU serial Port # 0 with one of PC serial ports ( COM 1 or COM 2, etc. )
- Launch Windows communication software **Hyperterminal**
- Create a new connection
- Select PC's serial port to use ( user can choose every **free** serial Port )
- Configure properties window of the chosen serial Port, exactly as the following one:



So PC's serial Port has to be set with the following features:

BaudRate( bit/sec): 115200  
Data Bits: 8  
Parity: None  
Flow Control: None

Through the HyperTerminal section it will be possible to visualize all network's parameters of the connected CPU.

```

MONITOR - HyperTerminal
File Modifica Visualizza Chiama Trasferimento ?
hw_addr: ff:ff:ff:ff:ff:ff
client IP: 192.168.80.102
my IP: 192.168.80.102
server IP: 0.0.0.0
gateway IP: 0.0.0.0
options:
  subnet mask: 255.255.255.0
  IP broadcast: 192.168.80.255
  gateway: 0.0.0.0
TWS_STATE_ACQUISITION_STOPPED
FTP_SERVER_STARTED
=====
=== Tiny Web Server - Version 2.02L ===
=====
ISaGRAF: Comm.thread on eth.port: 1100
ISaGRAF: Comm.thread on eth.port: 502
ISaGRAF: Slave 1 started
timeout set to 5 seconds
send (\dAT\d~M)
block 1 defined: Start:Z2HREG40002, Length: 18 bytes
block 2 defined: Start:Z2HREG40015, Length: 6 bytes
block 3 defined: Start:Z2HREG40011, Length: 8 bytes
Thu Jun 09 18:02:04 2005
Warning: Acquisition Started
TWS_STATE_NORMAL
restore from disk: file:/ISAGRAF/ISA11: size=22704 successful
Connesso a 0.12.36 | Auto detect | 115200 8-N-1 | SCORR | MAIUSC | NUM | Acquisisci | Eco stampante

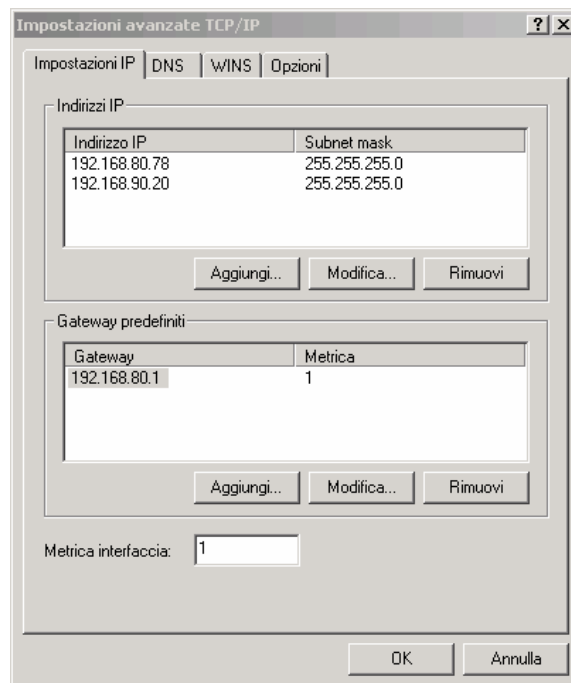
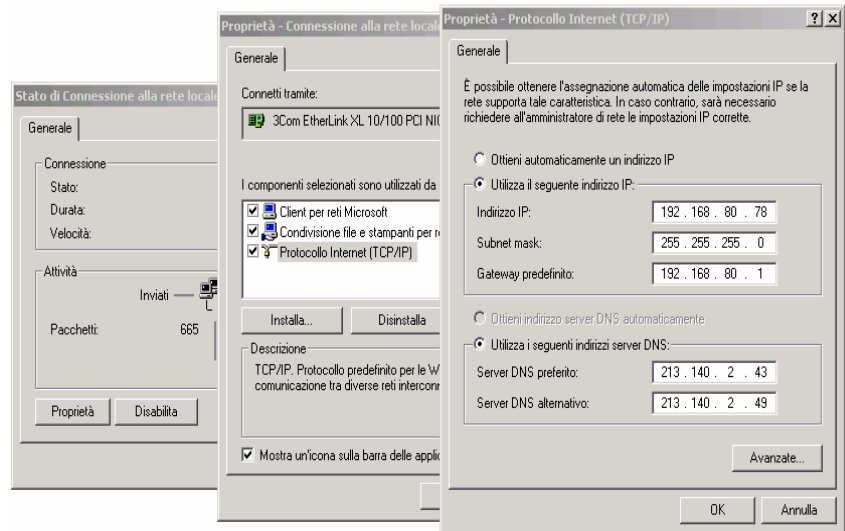
```

### 3.3 TCP/IP LINK CONFIGURATION

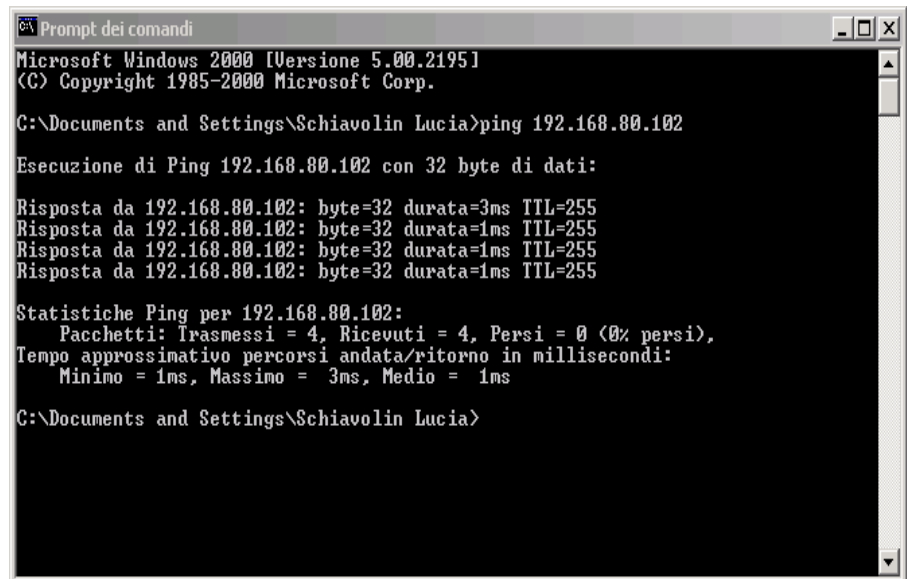
The following step is to verify if CPU's IP Address is compatible with user's local network ( LAN ).

The user may follow these steps:

- 1) Click on **Computer Resources**
- 2) Select **Control Panel**
- 3) Select **Network and remote connections**
- 4) Select **Network LAN connection**
- 5) Click on **Properties**
- 6) Select **TCP/IP Protocol**
- 7) Click on **Advanced**
- 8) then click on **ADD IP ADDRESS**
- 9) Add a new IP ADDRESS, for example: 192.168.90.78 which is compatible with TWS's IP Address ( not equal ).



If the user desires to check the effective compatibility between the two IP Addresses, he can launch the ping command from the DOS prompt. If the device is actually communicating with local LAN, the following window will be obtained:



```
Prompt dei comandi
Microsoft Windows 2000 [Versione 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Schiavolin Lucia>ping 192.168.80.102

Esecuzione di Ping 192.168.80.102 con 32 byte di dati:

Risposta da 192.168.80.102: byte=32 durata=3ms TTL=255
Risposta da 192.168.80.102: byte=32 durata=1ms TTL=255
Risposta da 192.168.80.102: byte=32 durata=1ms TTL=255
Risposta da 192.168.80.102: byte=32 durata=1ms TTL=255

Statistiche Ping per 192.168.80.102:
    Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 1ms, Massimo = 3ms, Medio = 1ms

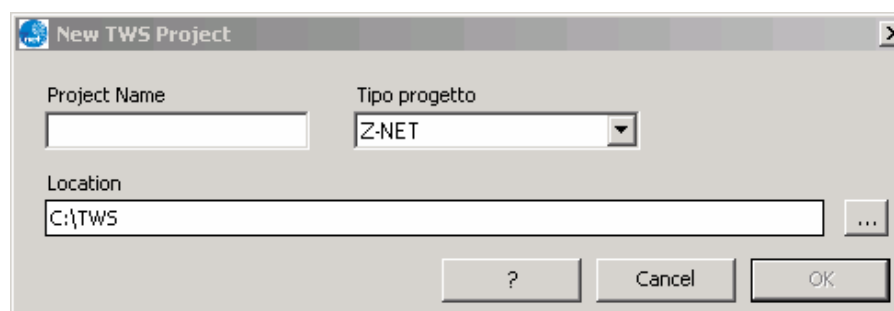
C:\Documents and Settings\Schiavolin Lucia>
```

### 3.4 SENECA Z-NET 2 CONFIGURATION

In this paragraph we are going to explain the modalities to create and configure a Z-NET 2 project.

#### 3.4.1 Z-NET 2 New Project Creation

Once all network's parameters of CPU have been detected, it will be possible to create a Z-NET 2 configuration project. In order to create a new project it is sufficient to click on menu **File|New..** and the following window will be opened:



The project will be saved in the directory, automatically created during the installation's phase. Anyway through the selection button a different destination directory may be chosen for the project.

Once the new project has been defined, the principale window of Z-NET 2 project will be opened and at the same time the following configuration mask will appear:

The screenshot shows a window titled "Configuration Wizard - TWS" with a close button in the top right corner. The main title is "TWS Project Configuration" and the subtitle is "Main ZTWS Settings". There is a small icon of a purple wizard hat in the top right corner of the window. The window is divided into three main sections:

- TCP/IP Address:** Contains two input fields: "Address" with the value "192.168.90.101" and "Timeout (mSec)" with the value "20000".
- FTP:** Contains two input fields: "User Name" with the value "abc" and "Passwor" (truncated) with the value "abc".
- Communication Parameters:** Contains two input fields: "Baud Rate" with a dropdown menu showing "38400" and "Timeout (mSec)" with the value "200".

At the bottom of the window, there are four buttons: "Cancel", "< Back", "Next >", and "Finish".

This window contains the most important configuration parameters of CPU:

Through **TCP/IP ADDRESS** section, user decides CPU's network's parameters:

**Address:** IP Address of the Z-RTU or Z-TWS 2 which will be utilized in the current project.

**Timeout (mSec):** Max Time in mSec, in which communication must be established.

Through **FTP** section, the user may define the password to access by FTP to CPU's files.

**User Name:** User Name for FTP access to CPU ( Z-RTU or Z-TWS 2 ). The standard user name is abc.

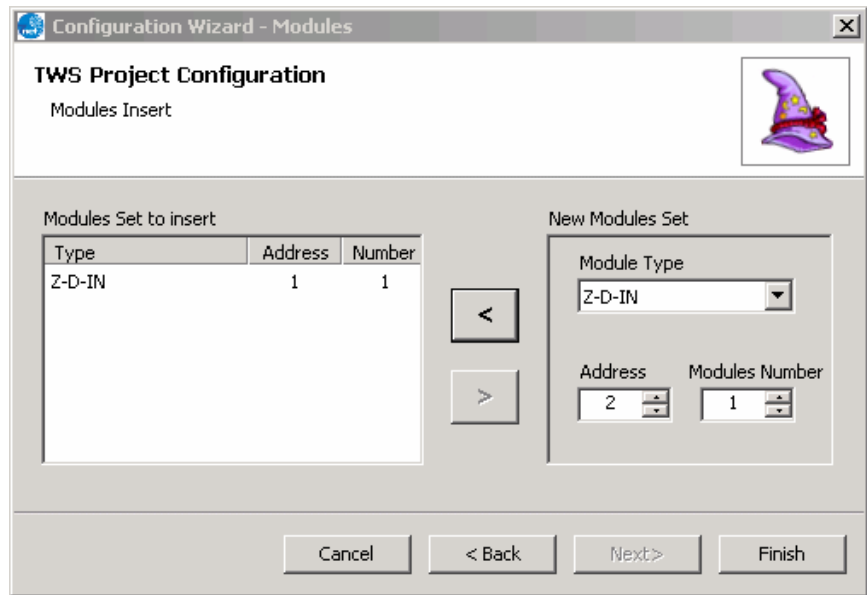
**Password:** Password for FTP access to CPU ( Z-RTU or Z-TWS 2 ). The standard password is abc.

At last in **Communication Parameters** section, the main parameters of serial communication ( between CPU and the linked I/O devices through Port 1 ) are decided.

**Baudrate:** baudrate of serial port RS485, Port 1. The allowed values are: 4800, 9600, 19200, 38400, 57600 e 115200 baud.

**Timeout (mSec):** Maximum Time in mSec in which serial communication with I/O modules must be established.

Once the data have been entered and saved by clicking on **Finish**, Z-NET 2 will recognize the CPU with the IP Address specified on the project. To verify the effective connection between PC and CPU, the user may select **Online|Data and Time**: if the reading operation succeeds CPU is correctly connected. By clicking on **Next** button it will be possible to insert I/O devices:

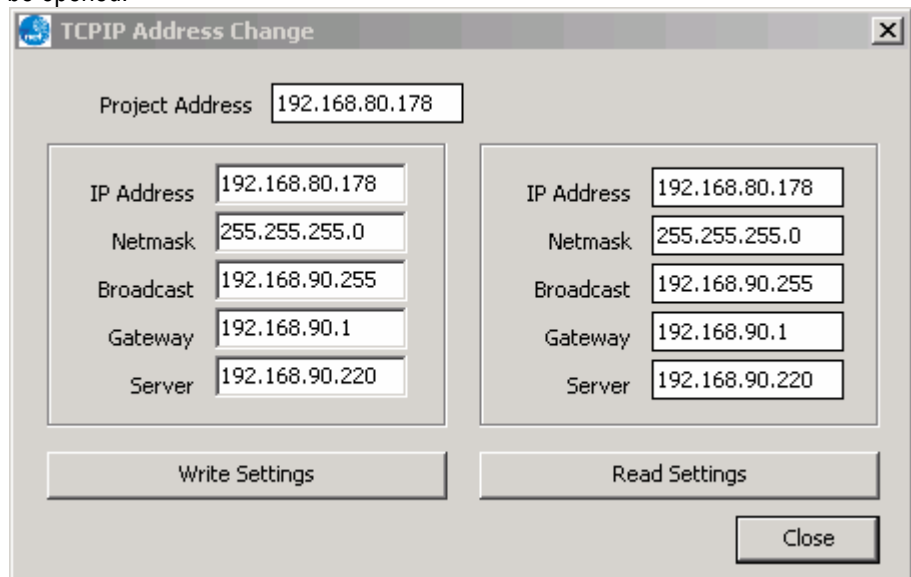


The user may select between all Z-PC line modules and has to assign them an address from 2 to 255. This modality of insertion may be used only if module's address and baudrate are known and set. Besides device's baudrate must be equal to the baudrate of CPU 's serial port to which I/O module will be linked.

### 3.4.2 TCP/IP ADDRESS MODIFICATION

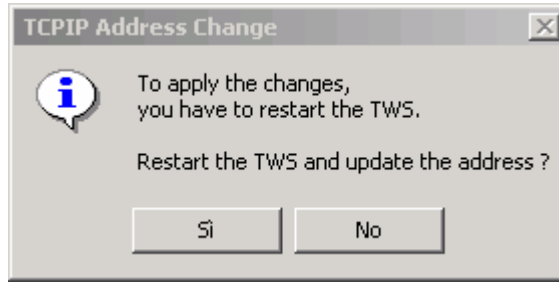
A very common necessity is to change CPU 's IP ADDRESS in order to differentiate it from the standard one.

It is sufficient to select **Online|TCP/IP Address Change** and the following window will be opened:



Once the new IP Address has been inserted in **IP Address**, by clicking on **Write Settings** Z-NET 2 warns the user that to effectly apply the changes, CPU must be

restarted.



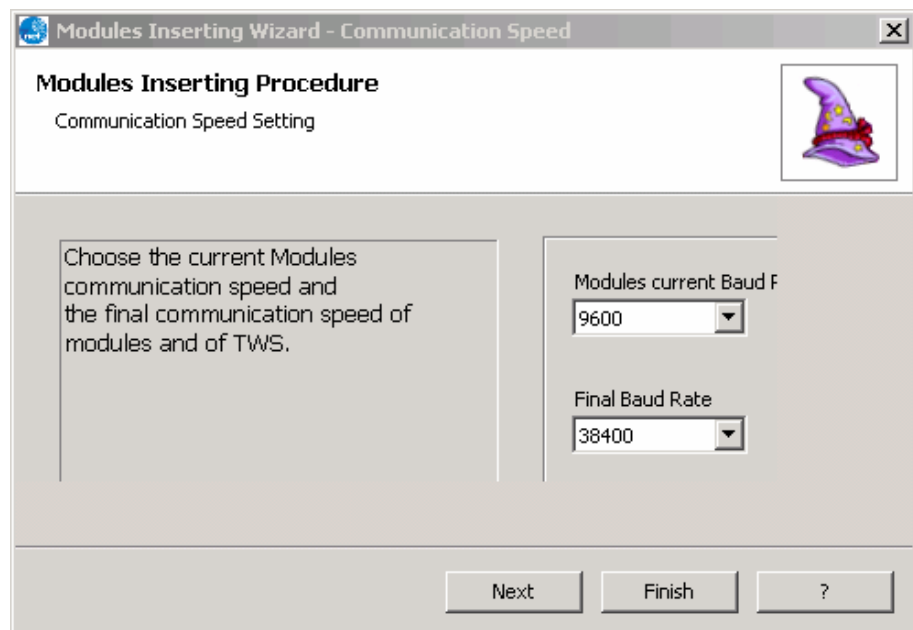
### 3.4.3 Modules Inserting Wizard

Before starting **Modules Inserting Wizard** procedure, it is necessary to know modules original address and baudrate. Z-PC I/O devices are sold by Seneca with standard communication's parameters: address 1 and baudrate 38400. However if a module has already been programmed it is not always possible to go back to the actual communication parameters. For this reason I/O devices are endowed of a dip-switch; if it is in OFF position module's Baudrate is fixed to 9600 and address to 1 while if it is in ON position, communication parameters are the ones memorized in RAM or EEPROM. Therefore, if user is not sure of communication's parameters values, it is advisable to set up to OFF position the correspondent dip-switch, before starting insert procedure.

The most efficient insert way is the **Module Wizard** one.

In insertion phase it is advisable to use only PORT 1 and to free all the other serial ports. At the end of modules' insert user may program the other ports and assignate modules ( addresses ) to them on his choice.

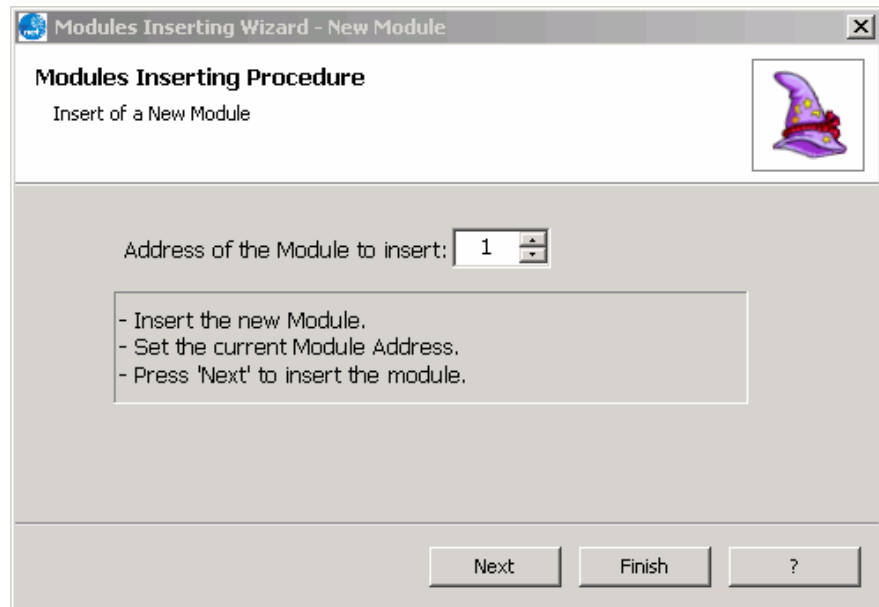
By selecting **Project|Modules Insert Wizard** the following window will be opened:



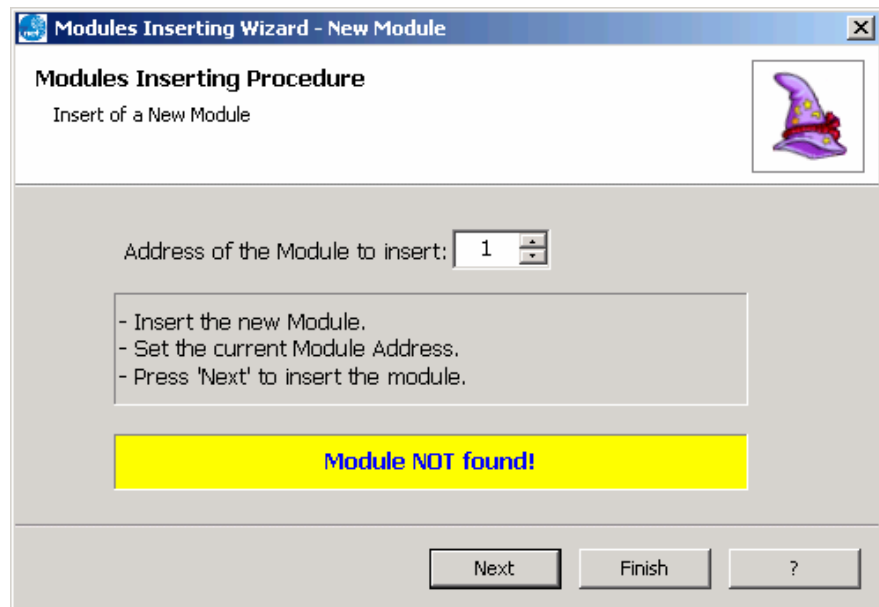
If communication's dip-switch is in OFF position, Modules current BaudRate must be 9600, otherwise user must insert the **known** value which is memorized in

EEPROM.

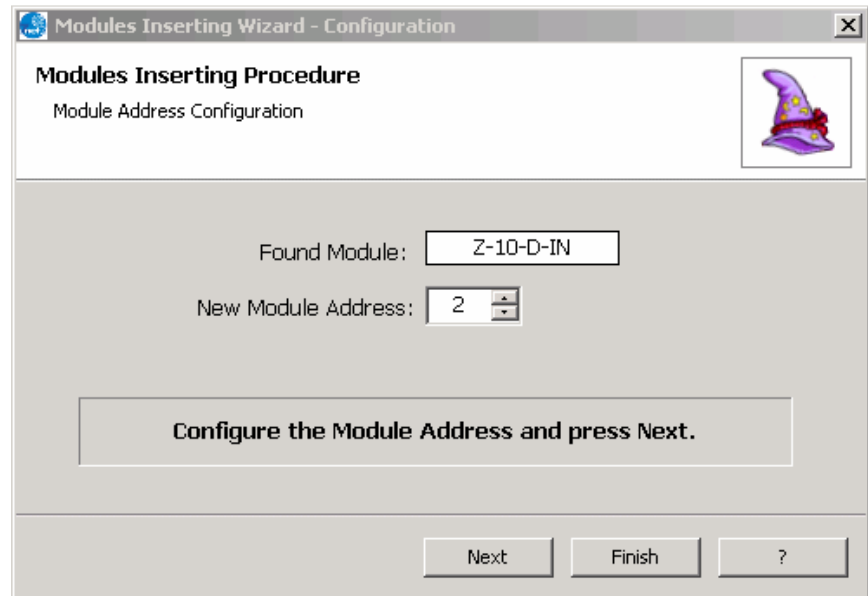
At this point it is necessary to insert the module's current address ( 1 if dip-switch position is OFF ):



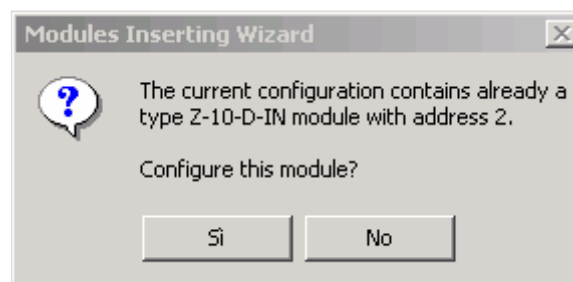
If address or baudarate values are wrong, module cannot be found and Z-NET 2 shows the following warning:



If the given parameters are correct, the CPU identifies the device and continues with the next phase in which user assigns module's new address. Since CPU has several serial Ports, it is very important to allocate connected modules between them. For example user may decide that PORT 1 modules have addresses from 1 to 30 while the I/O devices linked to PORT 3 own addresses from 30 to 254.

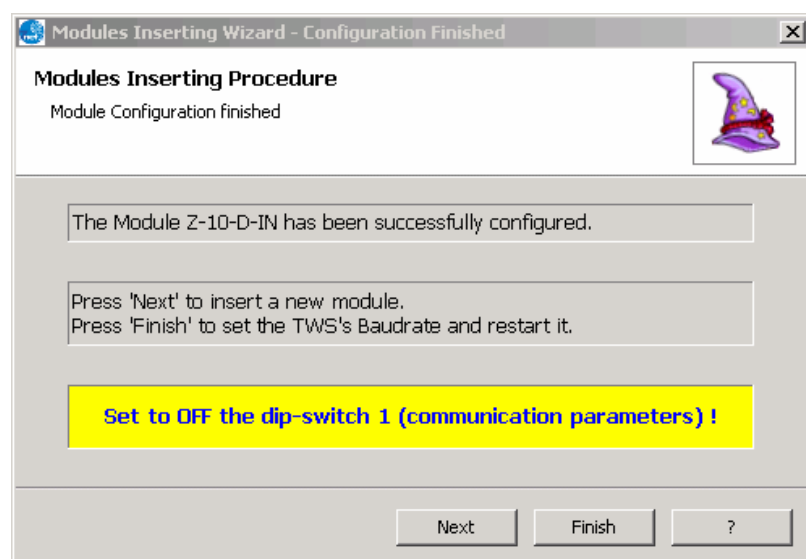


If user tries to give an already assigned address, the following window will be visualized:



By clicking on **YES**, the new module will be configured with address 2 while the first device will be canceled; by clicking on **NO** another address must be assigned to the new module.

Once all the inserted values are correct, Z-NET 2 visualizes this window:



Now the user has to set up the communication dip switch on ON position  
By clicking on **Next** it will be possible to insert other modules. Click on **Finish** to

conclude insert procedure.

#### 3.4.4 Add I/O devices

Another way to insert I/O modules is to select **Project|Add module** and choose the device to add. In this way it is necessary to know module's address and baudrate. This procedure is often used for modules' direct configuration through PC. It will be sufficient to select **Tools|Serial Port**: in this way PC and I/O device will communicate through the specified COM and the apposite serial converter.

## 4 CPU CONFIGURATION

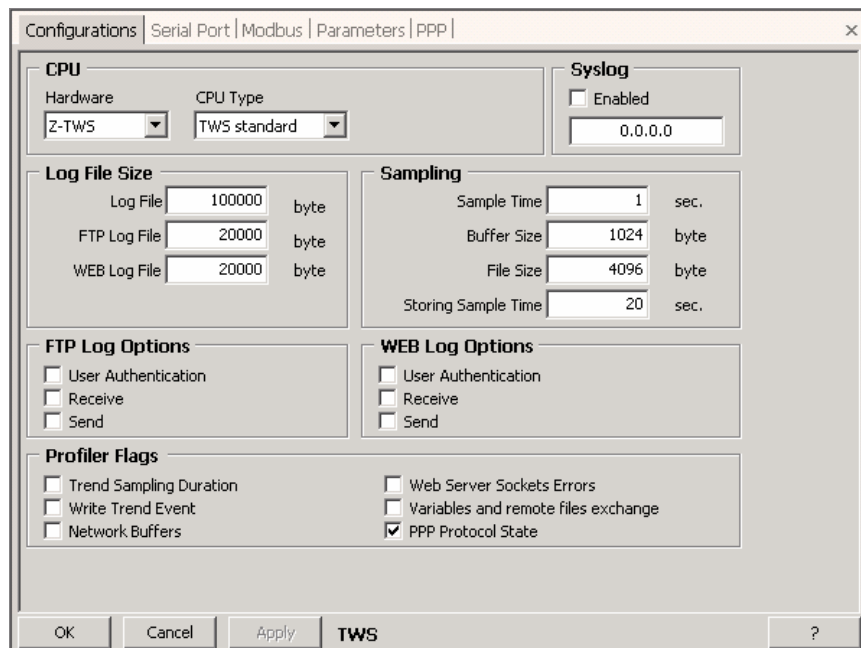
Through Z-NET 2, CPU's configuration is very easy and quick.

CPU's configuration is structured in five parts whose setting-out is generated by the following 5 windows :

- **Configurations:** Setting of the used CPU ( Z-TWS 2 or Z-RTU ), ModBus tasks and initialization parameters of operating system.
- **Serial Port** Configuration of CPU's serial Ports.
- **ModBus:** Parameters for the access to PLC through ModBus protocol .
- **Parameters:** Passwords for the access to CPU's files and variables.
- **PPP:** Parameters for remote connection through PPP protocol

### 4.1 CONFIGURATIONS WINDOW

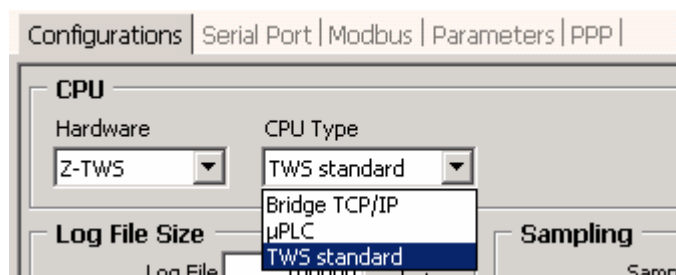
In this window, the user defines the features of the used master device ( Z-TWS 2 or Z-RTU ) and CPU's type. If Z-RTU is utilized, the user must also decide the used I/O version. Besides through this window initialization's parameters of operating system is performed



#### CPU

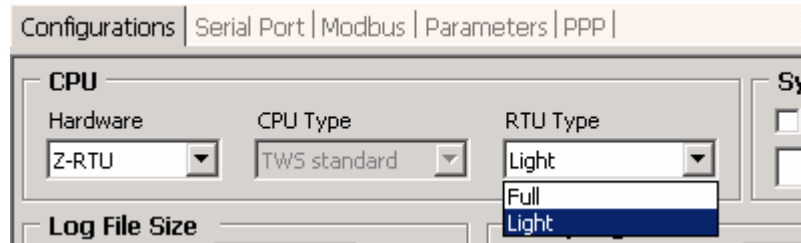
**Hardware:** User selects the master device which will be used: Z-TWS 2 or Z-RTU.

If Z-TWS is selected, users have to choose also CPU's version:



**CPU type:** User may select CPU's type: Bridge TCP/IP, μPLC and standard TWS which are illustreted on Paragraph 2.2.

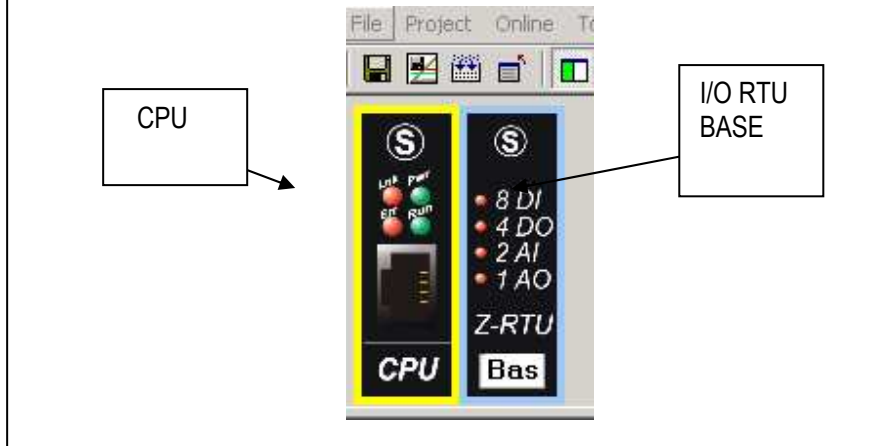
If Z-RTU is selected, Z-NET 2 shows the next configuration mask:



CPU's type is automatically configured as standard TWS but now users have also to choose the used I/O version:

**RTU type:** The two possible options are: *full* ( base + expansion) and *light*.( base ) ( Paragraph 2.2.)

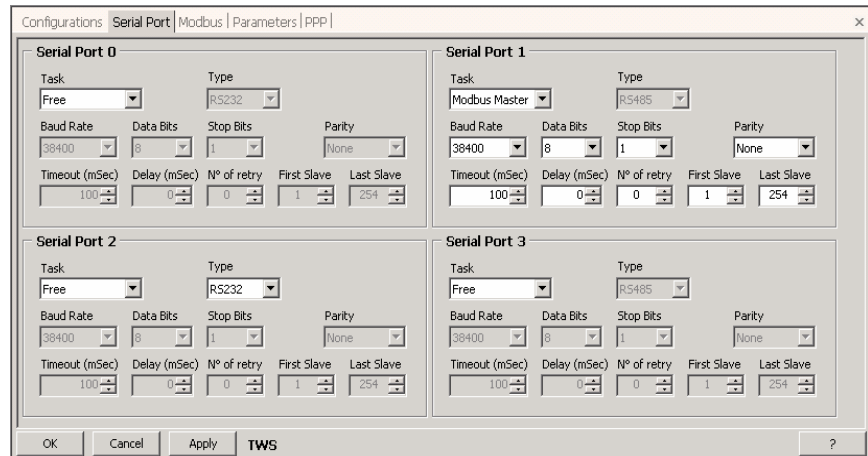
**Note:** If the chosen device is Z-RTU then click on **Apply** to visualize I/O modules ( base and possible expansion).  
At this point base module (light version ) or base + expansion ( full version ) will appear:



**SYSLOG**

## 4.2 SERIAL PORTS WINDOW

Serial Port window allows to program all the serial ports of the master device utilized ( Z-RTU o Z-TWS 2):



If the used CPU is bridge TCP or  $\mu$ PLC, only two serial ports can be programmed ( Port 1 and Port 0 ). Instead on Z-RTU user may configure 3 ports . For every serial ports there are the following configuration parameters:

**Task:** user may select the modalities of port's use: Three options are possible: Free, Modbus Master, ModBus Slave.

If the port is defined as free then all the other parameters are disabled.

**Type:** only for serial Port 2 user may select the type. If the used device is Z-TWS, Port 2 is associated to a dip switch which allows to set it as RS485 or RS232. In the case of Z-RTU, Port 2 must be programmed only by software. The other Ports instead are utilized univocally as RS485 or RS232.

**Baudrate:** Communication rate. User may select the following values: 4800 baud, 9600 baud, 19200 baud, 38400 baud, 57600 baud e 115200 baud.

**Data bits:** user may choose the number of bits to send: 7 or 8

**Stop bits:** user may choose the number of stops bits: 1 or 2.

**Parity:** user may choose the parity's control to use. The options are: *None*, *Even*, *Odd* .

If the port has been configured as Modbus Master, then user may set also the following parameters:

**Timeout (msec):** The timeout which is utilized in the communication between DAM and PLC. ( 0-10000 ms )

**Delay (msec):** Pause between two consecutive communications

**Retry:** Number of retries to establish the communication in case of I/O module 's missed answer

Then there is slave's partition:

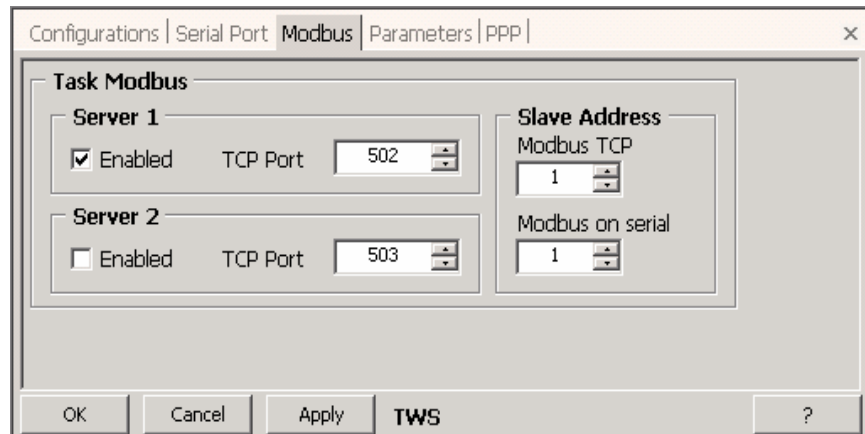
**First Slave:** address of the first I/O module connected to the specified serial port.

**Last Slave:** address of the last I/O module connected to the specified serial port  
It is necessary to assign to every serial ports different slaves' addresses.

### 4.3 MODBUS WINDOW

User may program two ISaGRAF Ethernet sockets which answer to ModBus TCP protocol. So through ModBus TCP it is possible to accede to ISaGRAF variables from two different stations. .

This section allows to define the number of ethernet sockets and the correspondent TCP port.



#### TASK MODBUS

##### SERVER 1

**Enabled:** If checked, it enables server number 1 which is the first ethernet socket.

**TCP Port:** Etherne socket for TCP connection of server 1 ( default value: 502 ).

##### SERVER 2

**Enabled:** If checked, it enables server number 2 which is the first ethernet socket.

**TCP Port:** Etherne socket for TCP connection of server 2 ( default value: 503 ).

#### SLAVE ADDRESS

In this section user defines CPU's addresses for ISaGRAF variables' acquisition.

**Modbus TCP:** Slave address for variables acquisition through ModBus TCP.

**ModBus on Serial:** Slave ModBus Address for variables' acquisition through serial line.

### 4.4 PARAMETERS WINDOW

Through this window the user may configure network's parameters of CPU and all the passwords necessary to accede to files and variables.

The parameters have the following meaning:

#### TCPIP ADDRESS

**Address:** CPU's IP Address.

**Timeout (msec):** Maximum time on which TCP/IP connection must be established.

#### SAMPLING RATE

**Rate (msec):** Period ( msec ) on which CPU interrogates all I/O modules for data reading operations.

Besides the user may define the passwords for CPU's data access. The password for FTP access is obligatory while the other ones can also be omitted.

#### FTP

**User Name:** User name for the access to CPU through FTP. Standard user name is : *abc*

**Password:** Password for access to CPU through FTP. Standard password is : *abc*

#### READ

**User Name :** User name necessary to read CPU's variables.

**Password:** Password necessary to read CPU's variables.

#### WRITE

**User Name :** User name necessary to write CPU's variables.

**Password:** Password necessary to write CPU's variables.

#### READ FILE

**User Name :** User name necessary to read CPU's internal files.

**Password:** Password necessary to read CPU's internal files.

#### WRITE FILES

**User Name :** User name necessary to write CPU's internal files.

**Password:** Password necessary to write CPU's internal files.

#### SYSTEM

**User Name:** User name necessary to utilize the following CGI functions: SetClock, SetIP, TWScontrol, PLCcontrol.

**Password:** User name necessary to utilize the following CGI functions: SetClock, SetIP, TWScontrol, PLCcontrol.

## 4.5 PPP WINDOW

Through this window, user may configure CPU's remote connection and automatically generate OPTIONS and CHAP SECRETS files.

The screenshot shows the PPP configuration window with the following details:

- Options File:**
  - PPP Enabled
  - Set PPP Address
  - Name: TWS
  - Address: 192.168.90.101
  - Netmask: 255.255.255.0
  - Baud Rate: 9600
  - Serial Port: 0 (selected)
  - Log
  - Debug
  - Disconnecting String: TIMEOUT@5@{\d+++}\d@{\dATH0@OK@
  - Initialization String: TIMEOUT@5@{\dAT}\d@OK@\d@{\d+++}\d@{\dATH0@OK@AT&F0@OK@AT&D0@OK@AT&R1@OK@AT&K4@OK@AT50=1@OK@
  - Modem's strings: PSTN
- Chap-secrets File:**

User	Password	Address
user	user	192.168.90.254

  - Buttons: Update, New, Delete
  - User input: user
  - Password input: user
  - Address input: 192.168.90.254

Parameters' meaning:

**PPP Enabled:** if checked, PPP ( Point to Point Protocol ) is enabled and it will be possible to configure all the other parameters.

**Set PPP Address:** If checked, user can manually set CPU's local IP Address. Otherwise Z-NET 2 automatically set CPU's IP Address of the current project.

**Name:** Local system's name. The default name is TWS and it will be used during authentication's phase.

**Address:** CPU's IP Address.

**NetMask:** CPU's Net mask.

**Baudrate:** Modem's communication rate.

**Serial port 0 or 2:** Serial port which will be used to communicate with the linked modem. The user may choose between port 0 and port 2 of Z-TWS 2.

**debug:** If checked, all the sent and received control's packets will be stored in a readable way.

**log:** If checked, all the stored informations will be sent to a log file.

**Disconnecting String:** String to be used at the end of every connection.

**Initialization String:** String to be used in order to initialize a new connection

**Modem's strings:** The user has to specify the modem which will be used: PSTN or GSM. Depending on the selected modem, initialization and disconnecting

string will be different.

In "**Chap-Secrets File**" section, the user will define the parameters for CHAP authentication:

***User***

User name of a client who may connect to CPU through PPP protocol

***Password***

Password of the client.

***Address***

IP Address of the client.

All the three fields are obligatory

By clicking on **Update**, the table on the left will be updated and in every line will be inserted a possible client. To remove a client, click on **Delete**.

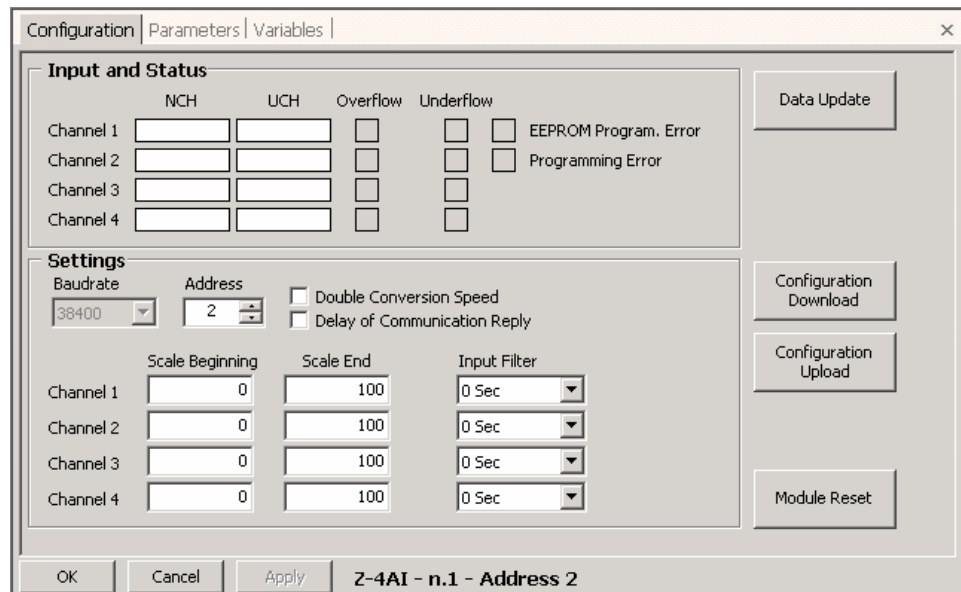
## 5 I/O MODULES CONFIGURATIONS

I/O modules configuration is structured in 3 parts whose setting-up is managed through 3 windows:

- Configuration: The user could program some electrical features of a specific module and also visualize I/O signals' values in real time.
- Parameters: The user may define the parameters of ModBus communication through RS485 serial line.
- Variables: The user may associate some variables to the internal registers of an I/O module.

### 5.1 CONFIGURATION WINDOW

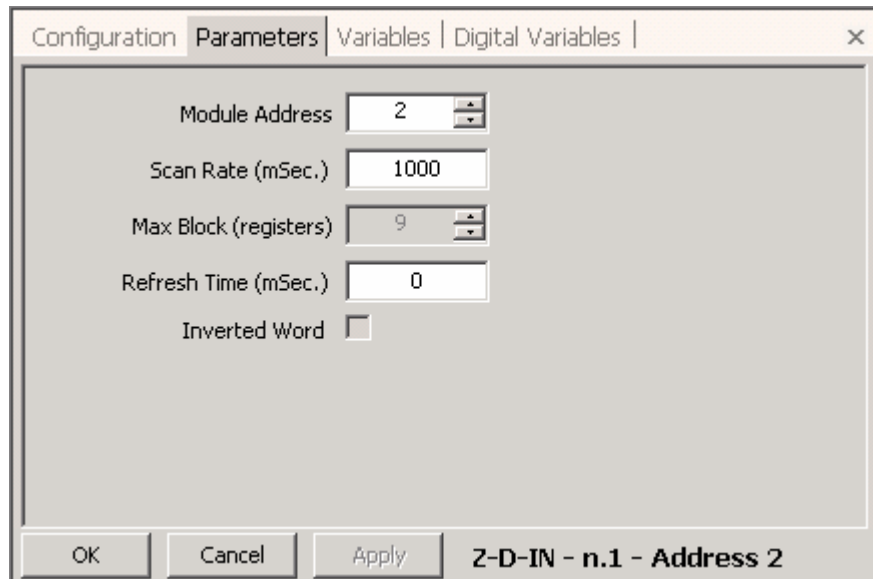
Through this window the user may configure some electrical features of the I/O module and besides visualize I/O signals' values in real time. For example the configuration window of Z-4AI is the following one:



By clicking on **Data Update**, the user may visualize the current values of an electrical signal. Instead if user desires download module's configuration, he has to push on **Configuration Download**. On the contrary by clicking on **Configuration Upload**, module's configuration will be uploaded on the project.

### 5.2 PARAMETERS WINDOW

This window specifies the parameters for ModBus communication through RS485 serial line. An example of a "Parameters Window " is the following one:



Meaning of parameters:

**Module Address:** I/O module's Address.

**Scan Rate ( ms ):** ModBus scan rate for reading and writing operations on the I/O module.

**Max Block ( registers ):** Maximum number of registers which can be contemporaneously accessed on a reading or writing operation.

**Refresh Time ( ms ):** Data Refresh Time.

### 5.3 VARIABLES WINDOW

. If the I/O device controls digital inputs or outputs, then also the **Digital Variables** window is present. The user may decide to visualize only some internal **Enabled** variables or all the internal registers ( by checking on **Show all variables** ) The defined variables could be utilized by PLC or be acquired through supervision's systems. For example the Z-4AI 's "Variables" window is the following one:

Address	Content	Data Type	Enabled	Name	Description	Input
40017	NCH1	Signed	<input checked="" type="checkbox"/>	IN_1		<input checked="" type="checkbox"/>
40018	NCH2	Signed	<input checked="" type="checkbox"/>	IN_2		<input checked="" type="checkbox"/>
40019	NCH3	Signed	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
40020	NCH4	Signed	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
40021	UCH1	Signed	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
40022	UCH2	Signed	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
40023	UCH3	Signed	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
40024	UCH4	Signed	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>

**Variables Window of Z-4AI module**

The user may define a variable by checking **Enabled** and by inserting the desired **Name** and **Description**. If the variable is associated to an Input signal the user must check the **Input** field.

## 6 VARIABLES

Z-PC's system allows variables' acquisition through OPC Server, ModBus TCP/IP and ModBus RTU.

Through Z-NET 2, the user may automatically create ISaGRAF, OPC Server and Movicon variables databases. The base definition has been described in chapter 5 ( **Variables Window** on paragraph 5.3 ), but the complete setting of variables is controled by the **Variables** section, present on the project navigator:

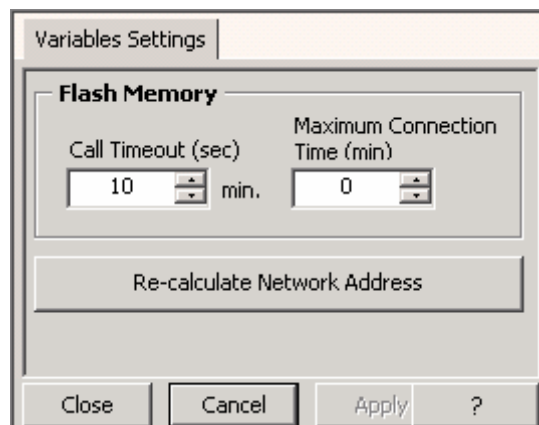


The user may the define the following types of variables:

- **ISaGRAF Variables:** The Z-NET 2 configurator can export the variables defined in the modules and in the CPU as ISaGRAF project configuration file.
- **OPC Server Variables:** Z-NET 2 produces the OPC.ini file which contains all the OPC variables defined in the modules or in the CPU. This file allows the automatic configuration of the OPC Server Z-TWS application.
- **Movicon Variables:** Z-NET 2 configurator can export the variables defined in the modules or in the CPU as a XML configuration file for a Movicon project that uses OPC Server Z-TWS to access to variables.

### 6.1 VARIABLES WINDOW

By clicking on **Variables** the following window will be opened:



Through this window, the user may set the variables to periodically save on the flash memory.

#### FLASH MEMORY

**Refresh Time:** Refresh Time of the variables on the flash memory.

**Variables Number:** Number of variables to save on the flash memory.

**Re-Calculate Network Address:** By clicking on this button, the user may re-calculate the Network Address of flash variables.

## 6.2 I/O VARIABLES

By clicking on **I/O Variables**, the following window will be opened allowing to complete the definition of all the I/O variables:

Analog Variables   Digital Variables															
Input															
Module	Data Type	Name	Description	MU	Public	IsaGraf	Net. Add.	OPC	OPC Group	Movicon	Scaled	Eng. BS	Eng. ES	Tec. BS	Tec. ES
Z-D-IN	Signed	Mod2_Var40002			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4000	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed	IN_A_1			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4001	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed	IN_A_2			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4002	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed	IN_A_3			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4003	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Output															
Module	Data Type	Name	Description	MU	Public	IsaGraf	Net. Add.	OPC	OPC Group	Movicon	Scaled	Eng. BS	Eng. ES	Tec. BS	Tec. ES
Z-3AO	Signed	OUT_A_1			<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10
Z-10-D-OUT	Signed	Mod5_Var40003			<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10

### 6.2.1 Analog I/O Variables

By selecting **Analog I/O Variables**, the list of all the enabled I/O Variables will appear:

Analog Variables   Digital Variables															
Input															
Module	Data Type	Name	Description	MU	Public	IsaGraf	Net. Add.	OPC	OPC Group	Movicon	Scaled	Eng. BS	Eng. ES	Tec. BS	Tec. ES
Z-D-IN	Signed	Mod2_Var40002			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4000	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed	IN_A_1			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4001	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed	IN_A_2			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4002	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed	IN_A_3			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4003	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Z-4AI	Signed				<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10000
Output															
Module	Data Type	Name	Description	MU	Public	IsaGraf	Net. Add.	OPC	OPC Group	Movicon	Scaled	Eng. BS	Eng. ES	Tec. BS	Tec. ES
Z-3AO	Signed	OUT_A_1			<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10
Z-10-D-OUT	Signed	Mod5_Var40003			<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	0	10000	0	10

In the higher section the user may find Input Analog Variables while in the under part Output variables are listed. To define or modify a variable, the user has to operate directly on table's fields.

**Module:** Variable's Module

**Data Type:** Register's data type ( signed ).

**Name:** Variable's name.

**Description:** Variable's description.

**MU:** Measure Unit of the variable.

**Public:** If checked, Z-NET 2 automatically assigns a Network Address for data acquisition through OPC Server, ModBus TCP/IP and ModBus RTU. Besides a *public* variable is defined in PLC.ini file. In this way the user may export also scaled I/O variables.

**ISaGRAF:** If checked, the variable is defined also as a ISaGRAF variable.

**Net. Add.:** Variable's Network Address. If the user desires to export a scaled I/O variable, a Network Address ( $\neq 0$ ) must be assigned to it.

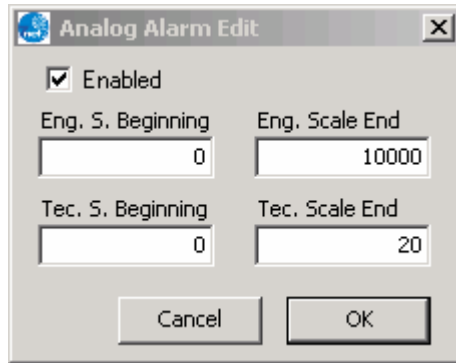
**Input:** If checked, it means that the variable is associated to an input signal.

**OPC:** If checked, the variable is defined also as an OPC variable.

**OPC Group:** The user may select the OPC group ( 1, 2 or 3 ) which the variable will belong to.

**Movicon:** If checked, the variable is defined also as a movicon variable.

**Scaled:** If checked, the user may scale the variable through the following window:



**Enabled:** If checked, a new scale will be defined for the variable.

**Eng.S. Beginning:** The start value of variable's engineeristic scale.

**Eng.S. End:** The end value of variable's engineeristic scale.

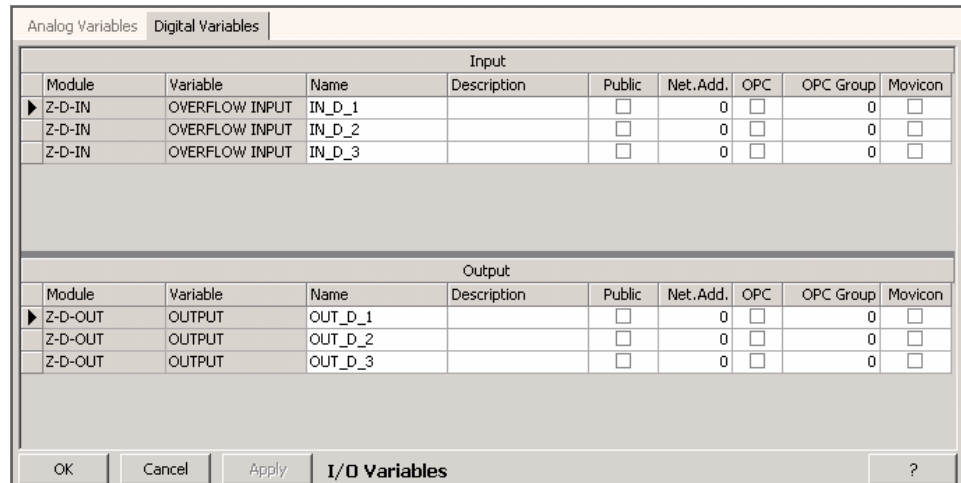
**Tec.S. Beginning:** The start value of variable's technical scale.

**Tec.S. End:** The end value of variable's technical scale.

It is useful to specify that the Engineeristic scale represents the real range of variables while the technical scale defines the new variation range.

## 6.2.2 I/O Digital Variables

By selecting **Digital Variables** the following window will be opened:



In the higher section the user may find Input Digital Variables while in the under part Output variables are listed. To define or modify a variable, the user has to operate directly on table's fields.

**Module:** Variable's Module

**Name:** Variable's name.

**Description:** Variable's description.

**Public:** If checked, Z-NET 2 automatically assigns a Network Address for data acquisition through OPC Server, ModBus TCP/IP and ModBus RTU. Besides a *public* variable is defined in PLC.ini file.

**Net. Add.:** Variable's Network Address. If the user desires to export a digital I/O variable, a Network Address ( ≠ 0 ) must be assigned to it.

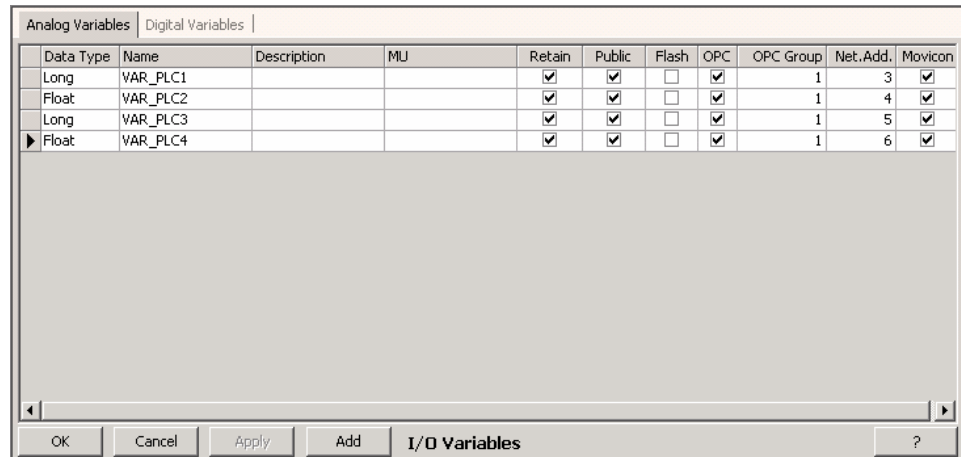
**OPC:** If checked, the variable is defined also as an OPC variable.

**OPC Group:** The user may select the OPC group ( 1, 2 or 3 ) which the variable will belong to.

**Movicon:** If checked, the variable is defined also as a movicon variable.

### 6.3 PLC VARIABLES

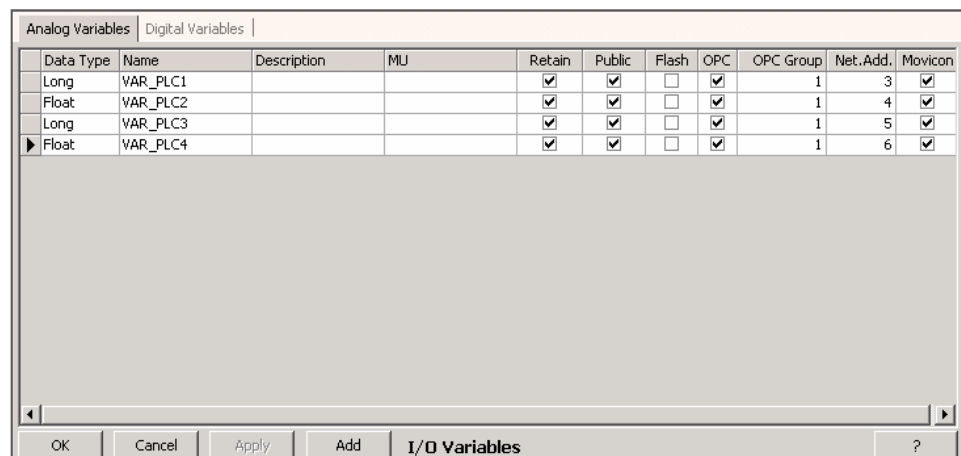
By clicking on **PLC Variables** ( project navigator ), the following window will be opened:



The user may define Analog and Digital variables.

#### 6.3.1 PLC Analog Variables

By checking on **Analog Variables**, the following window will be opened:



User may define a new variable by clicking on **Add** and by setting the following window:

Window's parameters have the following meaning:

**Multiple Variables:** If checked, the user may automatically define more variables :( for example PLC\_1, PLC\_2 etc ).

**Variable Name:** The user insert variable's name.

**Description:** Variable's description.

**Data Type:** Variable's data typer: long or float.

**OPC Group:** The user may select the OPC group ( 1, 2 or 3 ) which the variable will belong to.

**Measure Unit:** Variable's Measure Unit.

**Net. Add.:** Variable's Network Address. If the user desires to export a PLC variable, a Network Address (  $\neq 0$  ) must be assigned to it.

**Public:** If checked, Z-NET 2 automatically assigns a Network Address for data acquisition through OPC Server, ModBus TCP/IP and ModBus RTU. Besides a *public* variable is defined in PLC.ini file.

**Retain:** If checked, variable' s value will be retained also if a lack of power supply occurs.

**Flash:** If checked, the variable will be periodically saved on the flash memory.

**Movicon:** If checked, the variable is defined also as a movicon variable.

**OPC:** If checked, the variable is defined also as an OPC variable.

To modify a variable, the user has to operate directly on the principale table's fields.

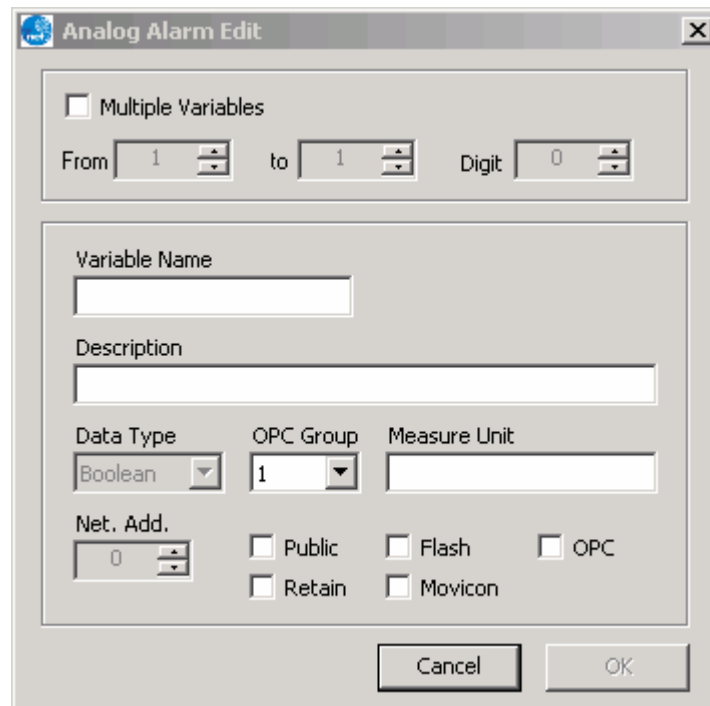
### 6.3.2 PLC Digital Variables

By checking on **Digital Variables**, the following window will be opened:

Analog Variables		Digital Variables								
Data Type	Name	Description	MU	Retain	Public	Flash	OPC	OPC Group	Net.Add.	Movicon
Boolean	PLC_DIG_1			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	7	<input checked="" type="checkbox"/>
Boolean	PLC_DIG_2			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	8	<input checked="" type="checkbox"/>
Boolean	PLC_DIG_3			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	9	<input checked="" type="checkbox"/>
Boolean	PLC_DIG_4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	10	<input checked="" type="checkbox"/>

OK Cancel Apply Add I/O Variables ?

User may define a new variable by clicking on **Add** and by setting the following window:



The dialog box 'Analog Alarm Edit' contains the following fields and controls:

- Multiple Variables
- From: 1 to 1 Digit: 0
- Variable Name: [Text Field]
- Description: [Text Field]
- Data Type: Boolean (dropdown)
- OPC Group: 1 (dropdown)
- Measure Unit: [Text Field]
- Net. Add.: 0 (spin box)
- Public
- Flash
- OPC
- Retain
- Movicon
- Buttons: Cancel, OK

Window's parameters have the following meaning:

**Multiple Variables:** If checked, the user may automatically define more variables :( for example PLC\_1, PLC\_2 etc ).

**Variable Name:** The user insert variable's name.

**Description:** Variable's description.

**Data Type:** Variable's data type: boolean

**OPC Group:** The user may select the OPC group ( 1, 2 or 3 ) which the variable will belong to.

**Measure Unit:** Variable's Measure Unit.

**Net. Add.:** Variable's Network Address. If the user desires to export a PLC variable, a Network Address (  $\neq 0$  ) must be assigned to it.

**Public:** If checked, Z-NET 2 automatically assigns a Network Address for data acquisition through OPC Server, ModBus TCP/IP and ModBus RTU. Besides a *public* variable is defined in PLC.ini file.

**Retain:** If checked, variable' s value will be retained also if a lack of power supply occurs.

**Flash:** If checked, the variable will be periodically saved on the flash memory.

**Movicon:** If checked, the variable is defined also as a movicon variable.

**OPC:** If checked, the variable is defined also as an OPC variable.

To modify a variable, the user has to operate directly on the principale table's fields.

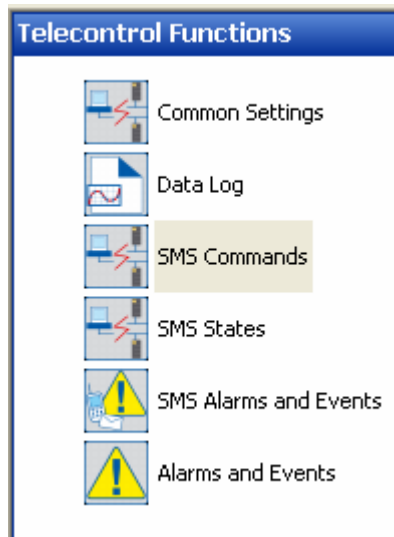
## 7 TELECONTROL

Z-NET 2 allows to easily configure all the important telecontrol functionalities, offered by Z-RTU and Z-TWS 2. The following functions are admitted, La funzionalità *Comandi SMS* sfrutta la potenzialità della CPU di ricevere SMS per attivare dei comandi su variabili digitali definite nel progetto. Invece tramite la funzionalità *Stati SMS* la CPU viene programmata ad inviare tramite SMS informazioni sullo stato ed il valore di alcune variabili digitali.

In ambiente Z-NET3 la sezione per la configurazione delle funzioni di telecontrollo è la seguente

- Log File's sending through email or FTP
- SMS 's sending on alarm's events
- Calling to the telecontrol centre in case of alarm
- Commands on sms riceiving
- Sending of SMS reporting the states of some variables:

The section to accede to telecontrol functions' configuration is the following:



### 7.1 TELECONTROL FUNCTIONS

Through **Telecontrol Functions** section, the user may define the parameters for the following functionalities:

- Log File's sending through email or FTP
- SMS 's sending on alarm's events
- Calling to the telecontrol centre in case of alarm
- Comands on sms riceiving
- Sending of SMS reporting the states of some variables:

By clicking on **Telecontrol Functions**, the user may configure the 3 windows we

are going to describe on the following paragraphs

### 7.1.1 Sending Data Log Files

The user may utilize two different ways to send CPU's Log file: through Email or through FTP.

### 7.1.2 Sending Through Email

By checking on **Send Data Log through Email**, the following window will appear:

The screenshot shows a dialog box titled "Log file sending settings" with tabs for "Alarms", "Call on alarm", and "SMS Commands". The "Working parameters" section includes:
 

- Send Data Log through Email
- Send Data Log through FTP
- Debug
- Delete log file after sent
- Call Timeout (sec): 60
- Maximum Connection Time: 10 min.
- Sending Time: 00.00
- Cycle: 4 hours

 The "Provider parameters" section includes fields for Provider Number, User, Password, Mail Server Address, and Modem String, along with a "Parametri standard" dropdown and "Salva con nome..." and "Elimina" buttons. The "Object and addresses" section includes fields for Object, Sender Email Address, and Receivers Addresses (email addresses separated by a comma). The dialog box has "Close", "Cancel", and "Apply" buttons at the bottom left, and "Telecontrol Functions" and a help icon at the bottom right.

Figura 7.1: Send Data Log through Email

#### WORKING PARAMETERS

In this first section the user may define the following sending general parameters:

**Debug:** If checked, the debug file *debug.txt* will be generated in CPU's Web directory. In this file all sending operations' result will be stored.

**Delete log file after sent :** If checked, log files will be deleted after being sent.

**Time Out:** Maximum Time ( in seconds ) on which CPU must be linked to the Internet Provider.

**Maximum connection Time:** Maximum connection time on which CPU has to complete files' sending.

**Sending Time:** Hour on which files will be sended.

**Period:** Time between two successive sendings ( from 1 to 24 hours )

**PROVIDER PARAMETERS**

This section defines the parameters to allow the connection to the Internet provider.

**Provider Number:** Provider's phone number

**User:** User Name to accede to the selected provider.

**Password:** Password to accede to the selected provider.

**Mail server Address:** IP Address of SMTP server which will be used to send the email.

**Modem String:** String for modem's initialization ( GSM or PSTN )

**Standard Parameters:** the user may select between some standard parameters which allow the connection to some predefined Internet providers. To save the current configuration, the user may select **Save as**. In this way the saved configuration may be selected on other projects.

**OBJECT AND ADDRESSES**

In this section the user may define the remaining sending parameters.

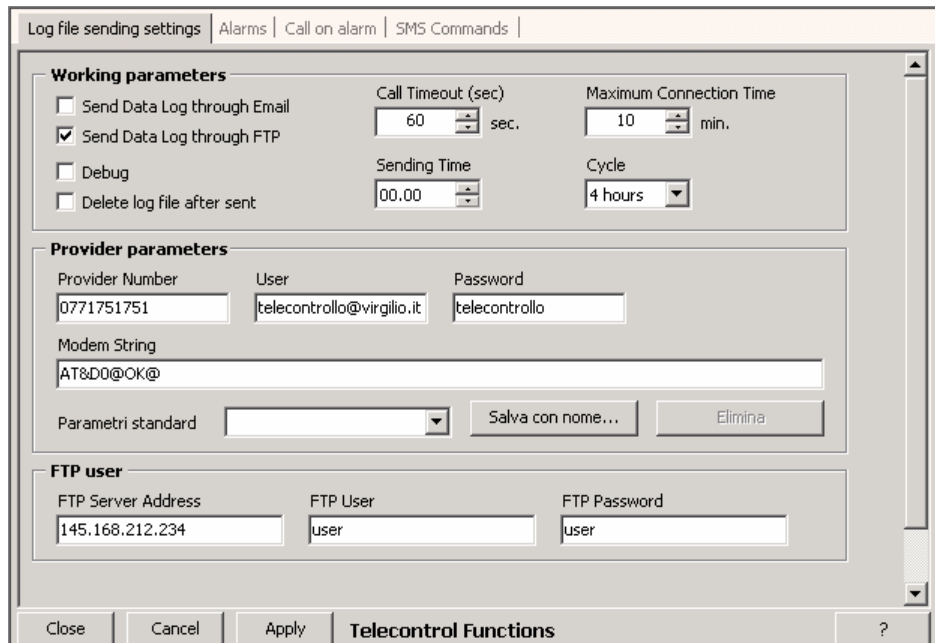
**Object:** Description of email content.

**Sender Email Address:** Sender's Address which will be visualized on the email message.

**Receivers Addresses:** Receivers Addresses to whom email will be sended ( separated by commas )

**7.1.3 Send Through FTP**

The configuration window in case of send through FTP is the following one:



**Fig. 7.2: Send Data Log through FTP**

**WORKING PARAMETERS**

In this first section the user may define the following sending general parameters:

**Send Data Log through Email:** If checked, data Log will be sent through FTP.

**Debug:** If checked debug file *debug.txt* will be generated in CPU's Web directory the. In this file all sending operations' result will be stored.

**Delete log file after sent:** If checked, log files will be deleted after being sent.

**Time Out:** Maximum Time ( in seconds ) on which CPU must be linked to the Internet Provider.

**Maximum connection Time:** Maximum connection time on which CPU has to complete files' sending.

**Sending Time:** Hour on which files will be sent.

**Period:** Time between two successive sendings ( from 1 to 24 hours )

#### PROVIDER PARAMETERS

This section defines the parameters to allow the connection to the Internet provider.

**Provider Number:** Provider's phone number

**User:** User Name to accede to the selected provider.

**Password:** Password to accede to the selected provider.

**Mail server Address:** IP Address of SMTP server which will be used to send the email.

**Modem String:** String for modem's initialization ( GSM or PSTN )

**Standard Parameters:** the user may select between some standard parameters which allow connection to some predefined Internet providers. To save the current configuration, the user may select **Save as**. In this way the saved configuration may be selected on other projects.

#### FTP USER

This section allows to define all the parameters to connect to the FTP server

**FTP Server Address:** IP Address of the FTP Server to which the log files will be sended.

**FTP User:** User Name for the access to the specified FTP Server.

**FTP Password:** Password for the access to the specified FTP Server.

### 7.1.4 Alarms

This configuration window allows to define all the necessary parameters to send sms if an alarm's condition occurs.

SMS	Description	SMS Numbers (separated by a comma)
Group 1	<input checked="" type="checkbox"/> GRUPPO1	00000000
Group 2	<input type="checkbox"/>	
Group 3	<input type="checkbox"/>	
Group 4	<input type="checkbox"/>	
Group 5	<input type="checkbox"/>	
Group 6	<input type="checkbox"/>	
Group 7	<input type="checkbox"/>	
Group 8	<input type="checkbox"/>	

**Sms Message:** Text of the sms.

**SMS Provider Number:** SMS Provider's Number which will be used to send SMS.

**Call Timeout:** Time ( in seconds ) after which SMS must be sent.

**Min interval between two SMS ( min ):** Minimum Time between two successive SMS sendings. In the meantime alarm's events will be joined in the same SMS.

**Timeout SMS max (sec):** Maximum time during which the CPU will

try to send the SMS ( included all the attempts).

**Retry SMS (sec):** Time in seconds, during which the CPU will try a single attempt of SMS sending.

**Sms Format:** There 3 SMS formats.

- **Standard**

*Digital Alarm:*

< SMS Text > dd/mm/yy-hh:mm:ss-<Alarm Name>-< Alarm Description >-ON or OFF

*Analog Alarm:* < SMS Text > dd/mm/yy-hh:mm:ss-< Alarm Name>-

< Alarm Description >-Low or High threshold

- **Format A:** <TAG SMS> A<TAG SMS>< Alarm Name ><X Alarm Description >dd/mm/yyhh:mm

- **Formato B.** < Testo SMS> Generic Alarm

**Debug:** If checked, the file *debugsms.txt* will be generated in CPU's WEB directory where every sending operation's result will be reported.

For each one of the 8 alarms groups, the user has to set the following parameters:

**Sms:** If checked, the user enables the sms sending for the correspondent alarms 'group.

**Description:** Description of the event associated to the correspondent alarms' group.

**Sms Numbers:** Phone numbers to which sms will be sented ( separated by a comma ).

### 7.1.5 Call on Alarm

Z-NET 2 allows to enable the *Call on Alarm* functionality through which, the CPU ( Z-TWS or Z-RTU ) calls the center in case of alarms events. Obviously both CPU and controlling PC must be linked to a modem device.

Figura 7.3: Call on Alarm Window

**Server Tel Number:** Controlling PC's phone number.

**User:** User Name for PC's access.

**Password:** Password for PC's access.

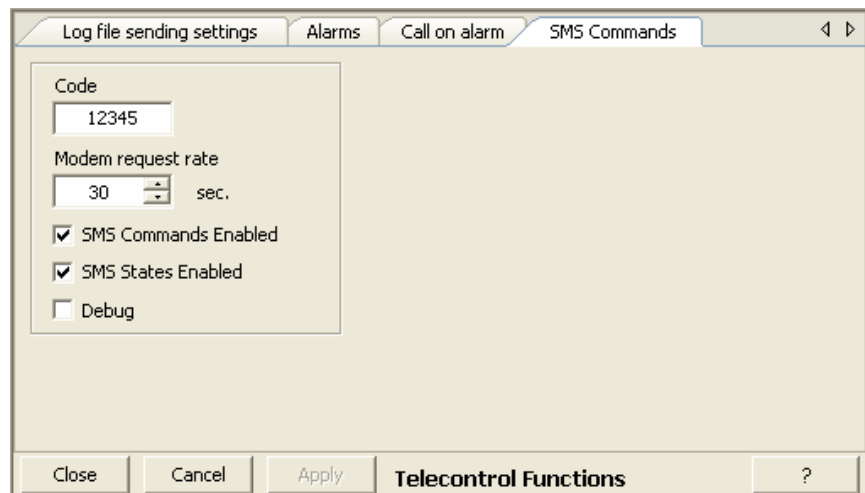
**Modem String:** String for the initialization of the modem linked to the CPU

**Caller TCP Address:** IP/TCP address of the CPU

**Call Timeout (sec):** Maximum Time on which the CPU will try to call the center. ( allowed values: from 30 sec to 360 sec )

### 7.1.6 SMS Commands

Through this section, it is possible to define all the general settings for **SMS Commands and States** functionality. **SMS Commands** functionality allows the CPU to receive SMS which can pilot the state of some specified digital variables. In other words, if the CPU receives, from a specified phone number, a SMS with a particular text, a digital variable can be set or reset. Instead **SMS States** functionality the CPU will send some informations about the state and value of some variables.



The parameters have the following meaning:

**Code:** Numeric Code which will be contained as first string of the SMS text.

**Modem Request rate:** Time (in seconds) after which the requests to receive SMS will be sent to the linked modem.

**SMS Commands Enabled:** If checked, the **SMS Commands** functionality is enabled.

**SMS States Enabled:** If checked, the **SMS States** functionality is enabled.

## 7.2 SMS COMMANDS

Through Seneca Z-NET3, the user may enable the *SMS COMMANDS* functionality, allowing the CPU to receive SMS which can pilot the state of some specified digital variables. In other words, if the CPU receives, from a specified phone number, a SMS with a particular text, a digital variable can be set or reset.

The window for *SMS COMMANDS* functionality's configuration is the following one:

Phone Numbers	Command	Variables	SMS Command
1. 0000000000	1. event1	OUT_D_1	12345 event1 ON
2. 0000000001	2. event2	OUT_D_2	12345 event2 ON
3. 0000000002	3. event3	OUT_D_3	12345 event3 ON
4. 0000000003	4. event4	OUT_D_4	12345 event4 ON
5. 0000000004	5. event5	OUT_D_5	12345 event5 ON
	6.		
	7.		
	8.		
	9.		
	10.		
	11.		
	12.		
	13.		
	14.		
	15.		
	16.		

The parameters of the window have the following meaning:

**Phone Numbers:** Phone numbers which will be allowed to execute SMS commands on CPU's variables.

**Command:** Command 's name.

**Variables:** Digital variable which will be changed by the correspondent SMS command.

**SMS Command:** Example of the text that has to be written on the sms in order to execute the correspondent command. If the user desires to reset OUT\_1\_DIG which is associated to *Pump1* command, he will be send a SMS with the following text: *12345 Pump1 OFF* .

## 7.3 SMS STATES

Through Seneca Z-NET3, it is possible to enable the **SMS States** functionality: the CPU will send some informations about the state and value of some variables. If Z-TWS2 or Z-RTU receives a sms with a particular text, it will send to the sender phone number, a sms with the values of a set of digital and analog variables.

The SMS states window is the following one:

The parameters have the following meaning:

**Plant name:** Name of the plant.

**SMS Provider Number:** SMS Provider's Number which will be used to send SMS.

**Call Timeout (sec):** Time ( in seconds ) after which SMS must be sent.

**Min interval between two SMS ( min ):** Minimum Time between two successive SMS sendings. In the meantime alarm's events will be joined in the same SMS.

**Timeout SMS max (sec):** Maximum time during which the CPU will try to send the SMS (included all the attempts).

**SMS Retry (sec):** Time in seconds, during which the CPU will try a single attempt of SMS sending.

At this point, it is possible to define up to 3 sets of variables. Each set is composed of 4 variables: two analog, two digital. For every group of variables, the setting is the following:

**Analog 1:** First analog variable whose value will be written on the sms.

**Analog 2:** Second analog variable whose value will be written on the sms.

**Digital 1:** First digital variable whose state will be written on the sms.

**Digital 2:** Second digital variable whose state will be written on the sms.

**SET Command:** Text of the SMS that once received from the CPU, will determine the SMS sending of the state correspondent variables set. To send the state SMS of the first variable set, the CPU will receive a SMS with the text: 12345 SET1. Where 12345 code is set on **SMS Commands and States** window of **Common Settings** .

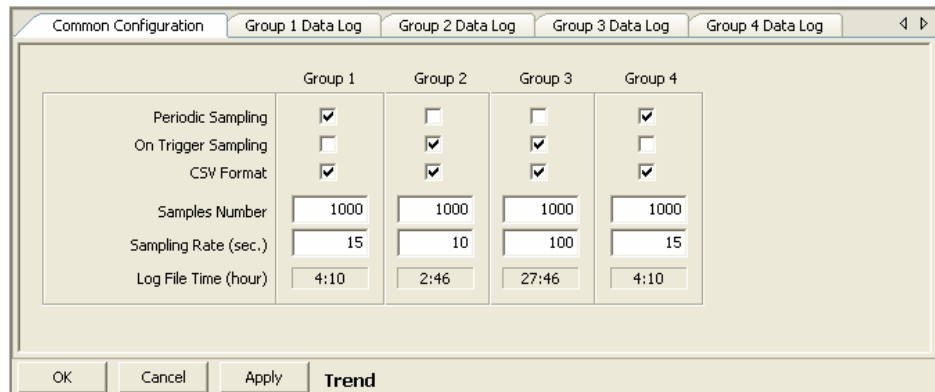
## 7.4 DATA LOG

On *Trend* directory of the CPU, there are four log files where data log are stored. The files are called:

GROUP1.DAT  
GROUP2.DAT  
GROUP3.DAT  
GROUP4.DAT

### 7.4.1 Common Configuration

The window for the general configuration of the Log files is the following one:



**Figure 7.4: Common Configuration Window**

Through this window, the user may define the general features of Data Log's groups.

For every group it will be possible to set the following parameters:

**Periodic Sampling:** If checked, the user enables periodic sampling

**On Trigger Sampling:** If checked the sampling will be commanded by a Trigger variable

**CSV Format:** If checked, the correspondent log file will be on CSV format ( values separated by semicolons ). Otherwise the file will be saved in a directly readable format with data arranged in explicit way.

**Samples Numbers:** Maximum number of samples which could be saved on the correspondent log file. Once the set number has been reached, the file is automatically renamed with **.OLD** extension and the registration will continue on the void file.

**Sampling Rate:** Sampling rate of group's variables ( allowed values: from 10 sec to 7200 sec ).

**Log File Time ( hour ):** Maximum duration ( in hours ) of the correspondent Log File. This time is automatically calculated through the set values of *Samples Numbers* and *Sampling Rate*.

## 7.4.2 Data Log Configuration

In order to define the Data Log of each group, the user has to open the *Group Data Log* window of the correspondent group.

All the three windows have the following features:

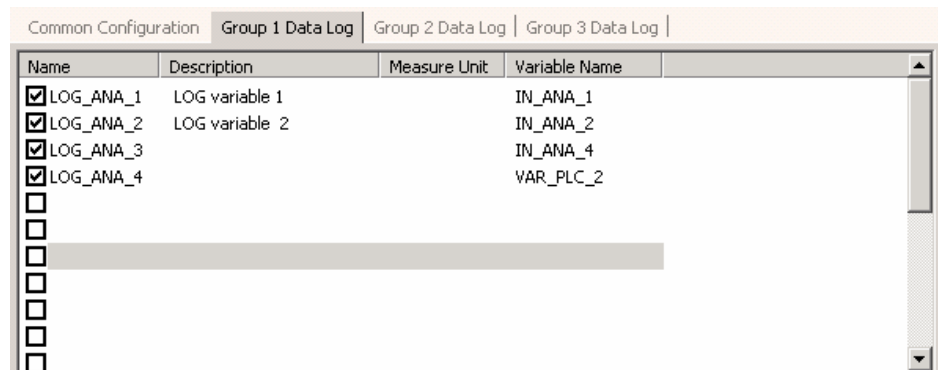


Figure 7.5: Group 1 Data Log Window

For each group, the user may define up till twenty Log variables.

By a double click on a free row, the user may insert a Log variable by setting the following configuration mask:

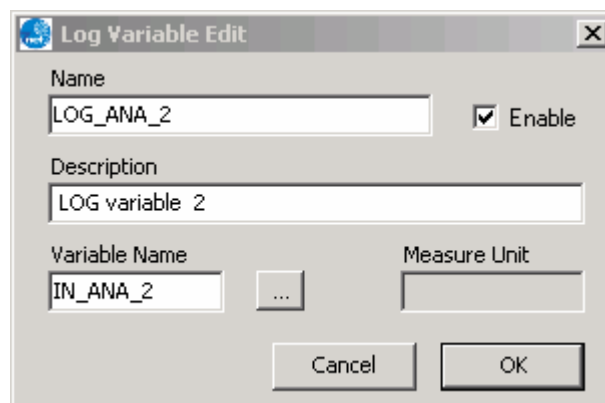


Figure 7.6: Log Variable Edit

The meaning of the parameters is the following one:

**Name:** Symbolic Name of the log variable.

**Enable:** If checked, the user enables variable's storing on the correspondent log file.

**Description:** Brief description of the defined log variable

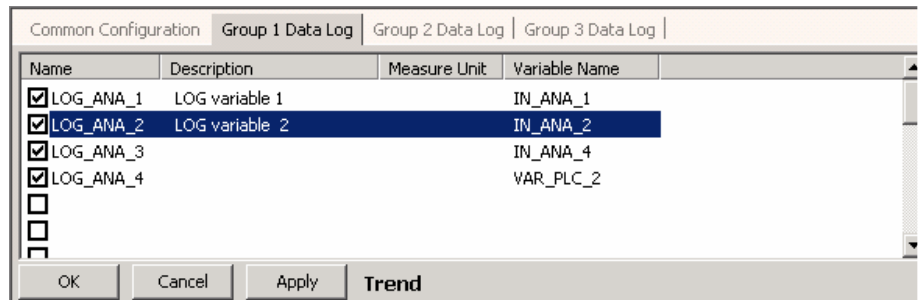
**Variable Name.:** Name of the analog variable to sample. The user may choose it through the selection button.

**Measure Unit:** Measure Unit of the selected variable.

Once the button OK has been pressed the log variable will be inserted as a row of the

principle table.

**A clarification is important: if a new scale has been defined for the variable on I/O Variables section, the values stored on the log file are in the new Scale( Technical scale )**



By a double click on the correspondent row, the user may modify a log variable.

## 7.5 SMS ALARMS AND EVENTS

The user may define eight groups of alarms. Sixteen analog and digital alarms could be associated to each group.

### 7.5.1 Digital Alarms

For each group, the user may define sixteen digital alarms. The configuration window is the following one:

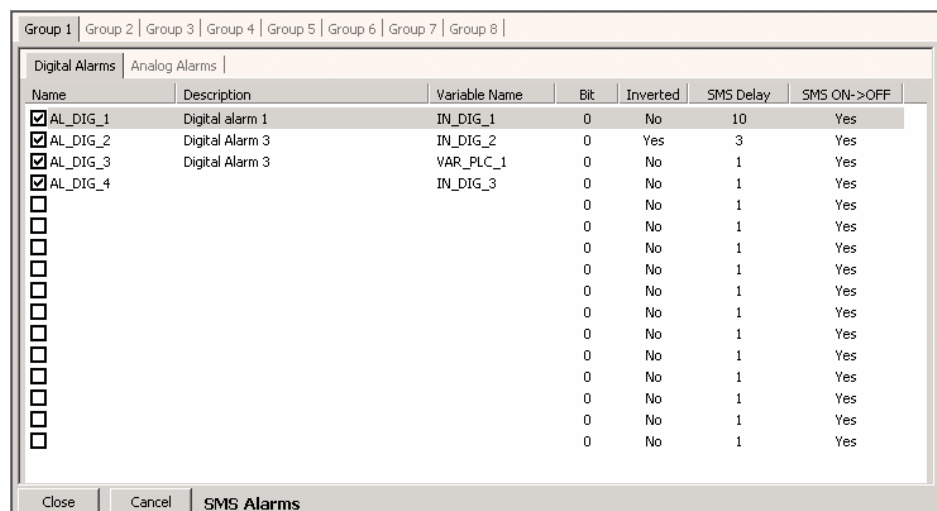


Figure 7.7: Group 1 Digital Alarm Window

By a double click on a free row, the user may insert a new alarm by setting the

following configuration mask:

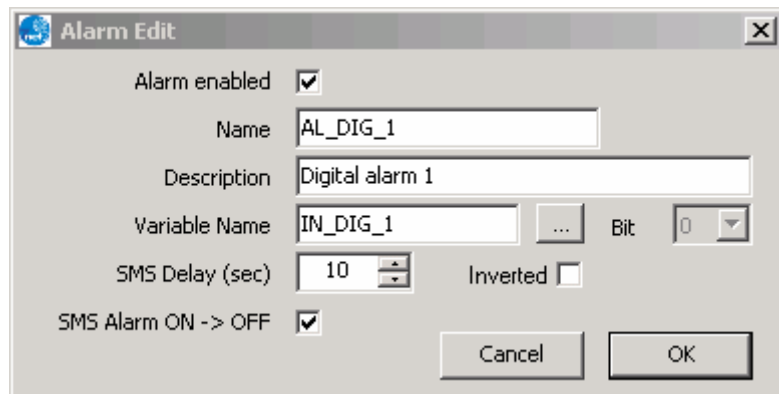


Figure 7.8: Alarm Edit Window

Where the parameters have the following meaning:

**Alarm Enabled:** If checked the alarm is enabled.

**Name:** Alarm's name.

**Description:** Description of the alarm.

**Variable Name:** Variable which will be associated to the alarm. By pressing the selection button, all the project's variables will be visualized.

**Bit:** If an analog variable is chosen, it indicates the bit index into the variable.

**SMS Delay ( sec ):** Delay between alarm's activation and SMS sending.

**Inverted:** If checked, the alarm will be activated on falling edges of the variable. Otherwise it will be activated on rising edges

**SMS Alarm On->OFF:** If checked, a SMS will be sent also when alarm's condition ends.

By pressing **OK**, the alarm will be visualized as a row in the principle table.

Besides, by a double click on the correspondent row, the user may modify an alarm

## 7.5.2 Analog Alarms

Analog alarms are activated when the associated variables overshoot the set thresholds.

The correspondent configuration window is the following one:

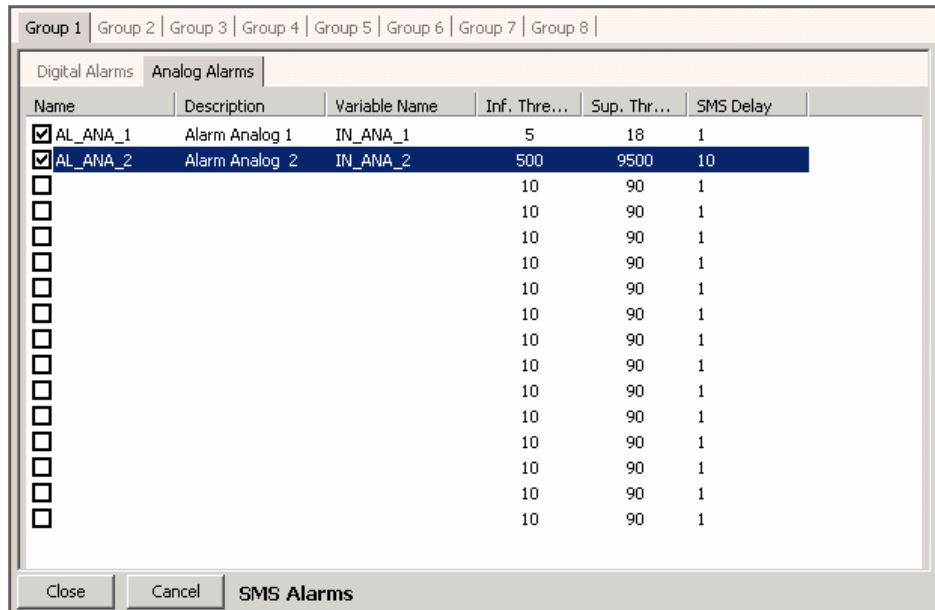


Figure 7.9: Analog Alarms Window

By a double click on a free row, the user may insert a new alarm by setting the following configuration mask:

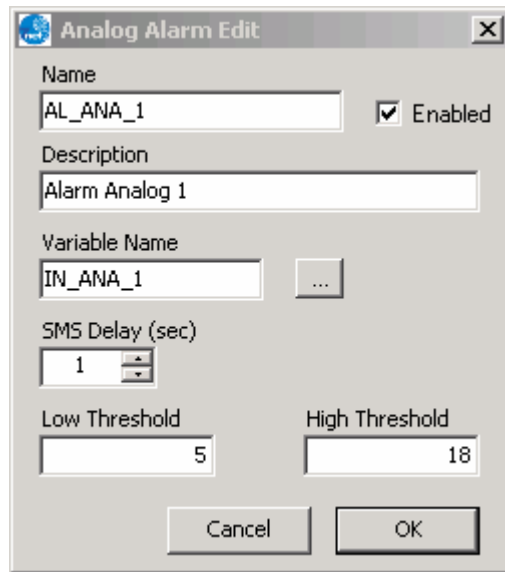


Figure 7.10: Analog Alarm Edit

Where the parameters have the following meaning:

**Name:** Alarm's name.

**Alarm Enabled:** If checked the alarm is enabled.

**Description:** Description of the alarm.

**Variable Name:** Variable which will be associated to the alarm. By pressing the selection button, all the project's variables will be visualized.

**SMS Delay ( sec ):** Delay between alarm's activation and SMS sending.

**Low Threshold:** Minimum value ( in Technical scale ) the variable may assume without activating the alarm.

**High Threshold:** Maximum value ( in Technical Scale ) the variable may assume without activating the alarm

**A clarification is important: if a new scale has been defined for the variable on I/O Variables section, the values of Low and High Thresholds are in the new Scale( Technical scale ).**

By pressing **OK**, the alarm will be visualized as a row in the principle table. Besides, by a double click on the correspondent row, the user may modify an alarm

## 7.6 ALARMS AND EVENTS

This section allows to define analog and digital alarms which will be memorized in a log file. Alarms events are stored in CPU's Trend directory, on ALLARMI.DAT file.

### 7.6.1 Digital Alarms

The configuration window of the digital alarms is the following one:

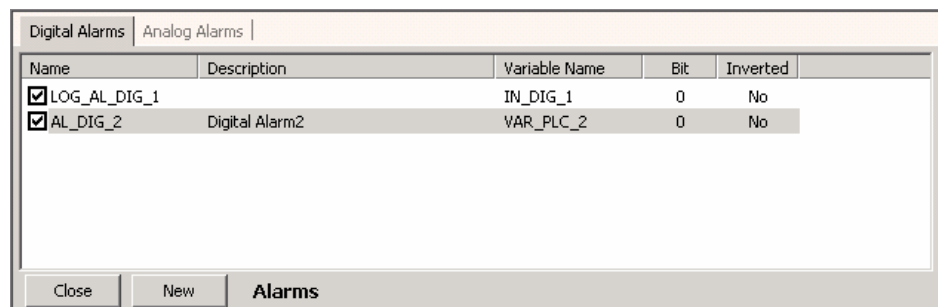


Figure 7.11: Alarms and Events window

By clicking on **New**, the user may insert a new alarm by setting the following configuration mask:

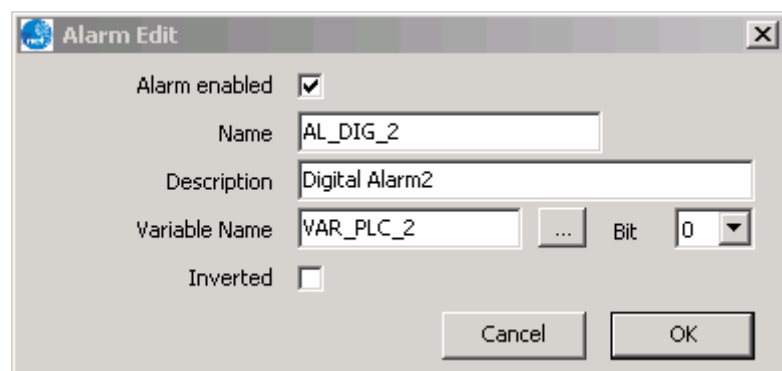


Figure 7.12: Alarm Edit Window

Where the parameters have the following meaning:

**Alarm Enabled:** If checked the alarm is enabled.

**Name:** Alarm's name.

**Description:** Description of the alarm.

**Variable Name:** Variable which will be associated to the alarm. By pressing the selection button, all the project's variables will be visualized.

**Bit:** If an analog variable is chosen, it indicates the bit index into the variable.

**Inverted:** If checked, the alarm will be activated on falling edges of the variable. Otherwise it will be activated on rising edges

By pressing **OK**, the alarm will be visualized as a row in the principle table.

Besides, by a double click on the correspondent row, the user may modify an alarm.

## 7.6.2 Analog Alarms

Analog alarms are activated when the associated variables overshoot the set thresholds.

The correspondent configuration window is the following one::

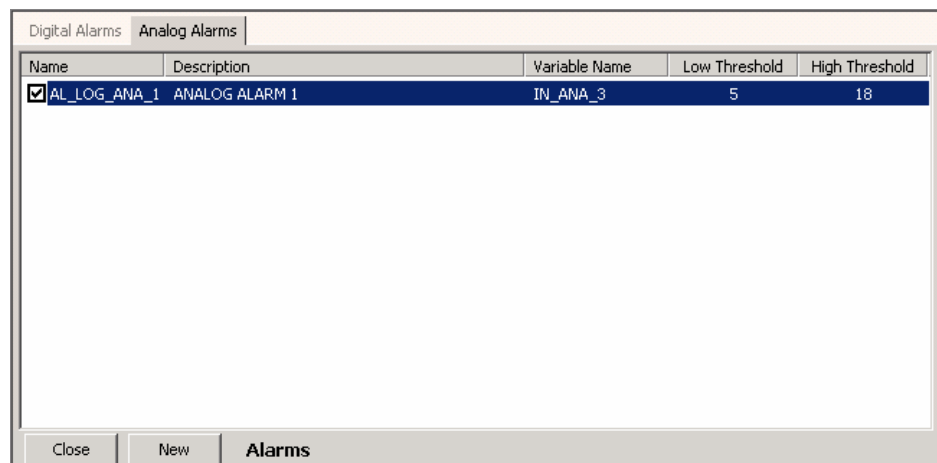


Figura 7.13: Log Analog Alarms Window

By clicking on **New**, the user may insert a new alarm by setting the following configuration mask:

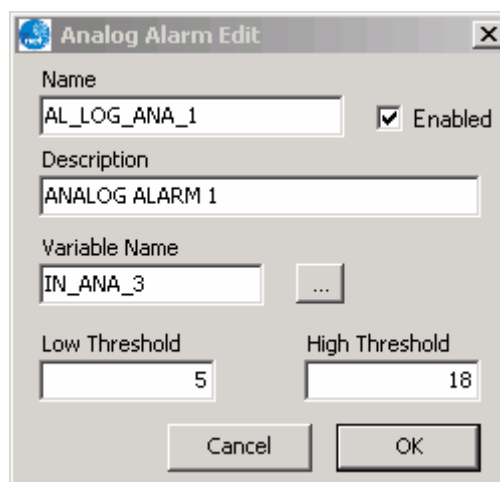


Figura 7.14: Analog Alarm Edit Window

Where the parameters have the following meaning:

**Name:** Alarm's name.

**Alarm Enabled:** If checked the alarm is enabled.

**Description:** Description of the alarm.

**Variable Name:** Variable which will be associated to the alarm. By pressing the selection button, all the project's variables will be visualized.

**Low Threshold:** Minimum value ( in Technical scale ) the variable may assume without activating the alarm.

**High Threshold:** Maximum value ( in Technical Scale ) the variable may assume without activating the alarm

**A clarification is important: if a new scale has been defined for the variable on I/O Variables section, the values of Low and High Thresholds are in the new Scale( Technical scale )**

By pressing **OK**, the alarm will be visualized as a row in the principle table.

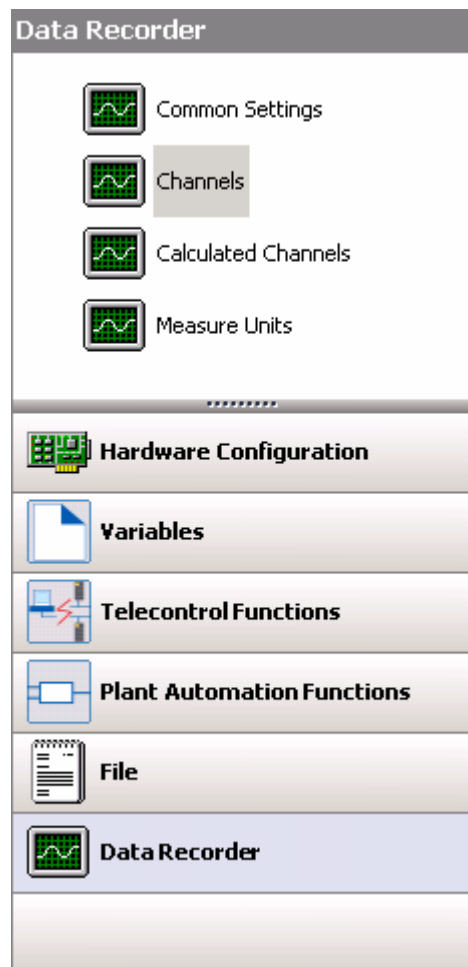
Besides, by a double click on the correspondent row, the user may modify an alarm.

## 8 DATA RECORDER PROJECT SETTINGS

Through Z-NET3 the user may configure the main settings of a DATA RECORDER project as regards the following arguments:

- Common Settings of serial or ethernet communication.
- Channels definition basing on Z-NET3 project variables.
- Calculated channels definition, obtained as mathematical functions of the channels previously defined.
- Measure Units Definition

By selecting from the menu **Project** , the voice **Data Recorder**, the following sections will be available in the project navigator

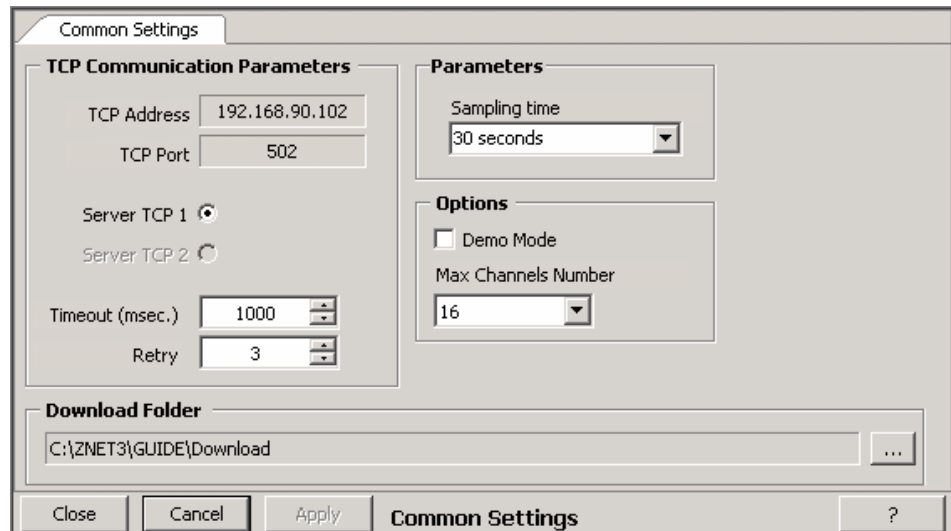


### 8.1 COMMON SETTINGS

Through this section it is possible to set the main parameters of communication and data memorization for the correspondent Data Recorder project. They obviously change very much if Z-TWS02 is used or not.

### 8.1.1 Project with Z-TWS2

If a Z-TWS02 is present in the current Z-NET3 project, the communication is performed through the Ethernet port and ModBus TCP/IP protocol. The **Common Settings** window is the following one:



The parameters have the following meaning:

#### TCP COMMUNICATION PARAMETERS

**TCP Address:** TCP/IP Address of the Z-TWS02.

**TCP Port:** TCP Port used for the communication with the CPU (Modbus windows of the CPU).

**Server TCP 1:** If in the *Modbus* window of the CPU, the Server TCP 1 is enabled, then it can be used from Data Recorder for the communication.

**Server TCP 2:** If in the *Modbus* window of the CPU, the Server TCP 2 is enabled, then it can be used from Data Recorder for the communication.

**Timeout (msec):** represents the timeout of the TCP/IP communication between PC and Z-TWS02.

**Retry:** Number of attempts in case of communication failure.

#### PARAMETERS

**Sampling Time:** sampling time of memorization of the defined channels on the database.

#### OPTIONS

**Demo Mode:** if checked, set the demo mode functioning for Data Recorder. In this mode it is possible to visualize simulated channels even if no hardware system is really

connected.

**Max Channels number:** the user may select the maximum number of channels that the software will allow to define. Obviously the user has to insert the number of channels allowed from its licence.

#### DOWNLOAD FOLDER

It indicates the path of the directory where the software saves the databases containing the memorized value. The default directory is the directory **Download** inside the folder of the correspondent Z-NET3 project. The user may modify it, by clicking on the selection button.

### 8.1.2 Project with only I/O modules

If the current project is a Z-NET3 I/O project, the communication between the I/O modules and the PC is performed through serial port and Modbus RTU protocol.

The **Common Settings** window is the following one:

The parameters have the following meaning:

#### SERIAL PORT PARAMETERS

These parameters are inherited from the correspondent Z-NET3 I/O project.

**COM Port** : Standard serial port which will be used for the communication with I/O modules.

**Baudrate**: Communication speed on the serial port.

**Parity:** Parity control on the serial port.

**Stop Bits:** Number of stop bits .

**Data Bits:** data length.

**Timeout (msec):** represents the timeout of the Modbus RTU communication between PC and I/O modules.

**Retry:** Number of attempts in case of communication failure.

#### PARAMETERS

**Sampling Time:** **Sampling Time:** sampling time of memorization of the defined channels on the database.

#### OPTIONS

**Demo Mode:** if checked, set the demo mode functioning for Data Recorder. In this mode it is possible to visualize simulated channels even if no hardware system is really connected.

**Max Channels number:** the user may select the maximum number of channels that the software will allow to define. Obviously the user has to insert the number of channels allowed from its licence.

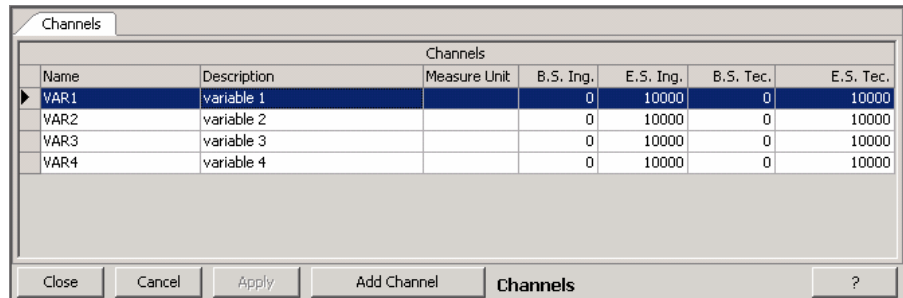
#### DOWNLOAD FOLDER

It indicates the path of the directory where the software save the databases containing the memorized value. The default directory is the directory **Download** inside the folder of the correspondent Z-NET3 project. The user may modify it, by clicking on the selection button.

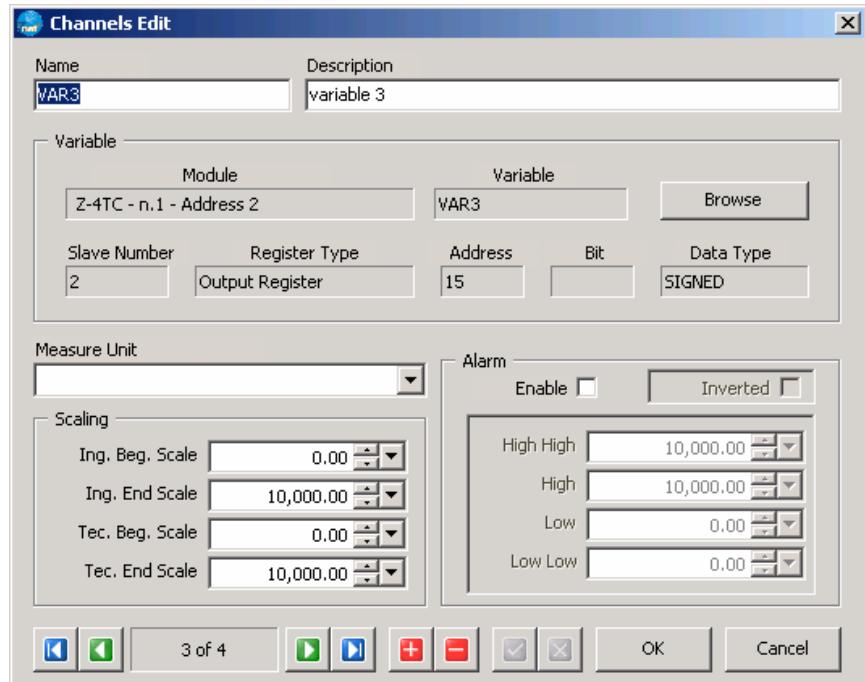
## 8.2 CHANNELS

On this section the user may add the channels to visualize and memorize through Data Recorder software.

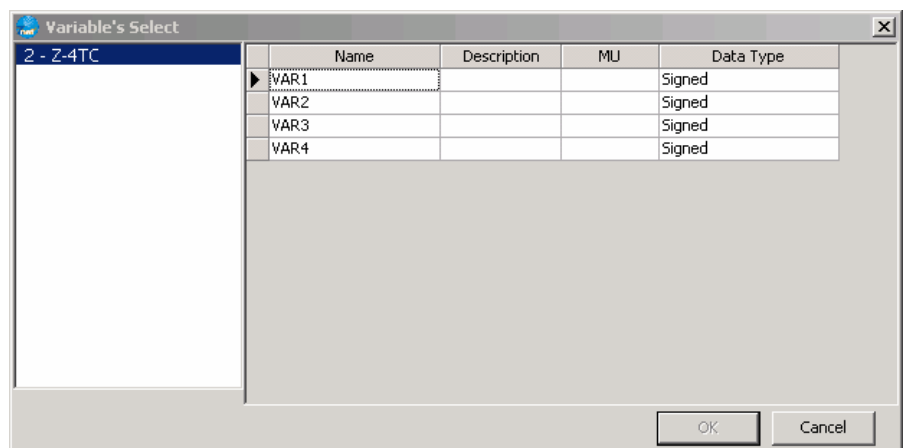
The **Channels** window is the following:



To add a new channel, press on **Add channel** and the following mask will be opened:



To choose which variable will be associated to the channel press on **browse**. The following window will be opened:



At this point it will be possible to select one of the variables previously defined in the sections **Variables** and **Digital Variables**.

If the **Demo Mode** has been selected, the **Description** field is of great importance: the user has to write the function to give to the channel. It is possible to insert all the Functions admitted for the Calculated Channels.

If an analog variable is selected, a new scale of visualization and memorization may be defined.

Therefore in **Ing Beg Scale** and in **Ing End Scale** there are the values of beginning and end scale of the associated variable while the **Tec. Beg Scale** and **Tec End Scale** fields set the values of beginning and end scale of the new visualization technical scale.

Besides it is possible to define an alarm:

If the selected variable is a digital one, the user has to set the following parameters to define the alarm:

**Enable:** if checked, the alarm is enabled.

**Inverted:** If checked, the alarm will be activated on falling edges (True-->False) of the variable. Otherwise it will activated on rising edges (False-->True).

if the selected variable is an analog one, to define the alarm the user has to set the following parameters:

**Enable:** if checked, the alarm is enabled.

**High High:** sets the highest threshold (in technical scale) of the alarm.

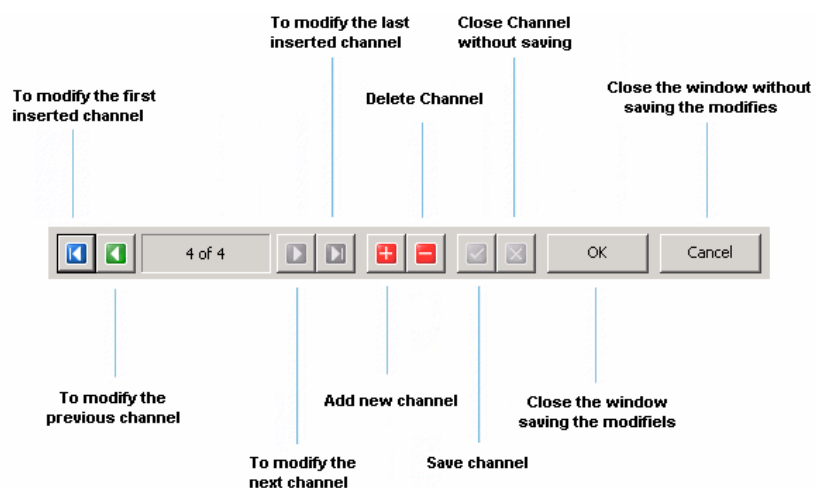
**High:** sets the high threshold (in technical scale) of the alarm.

**Low:** sets the low threshold (in technical scale) of the alarm.

**Low Low:** sets the lowest threshold (in technical scale) of the alarm.

Once the adding has been completed, click on the correspondent icon of the below menu bar.

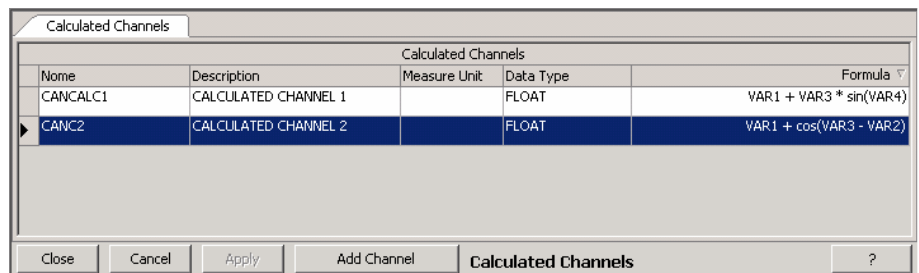
The icons of the menu bar have the following meaning:



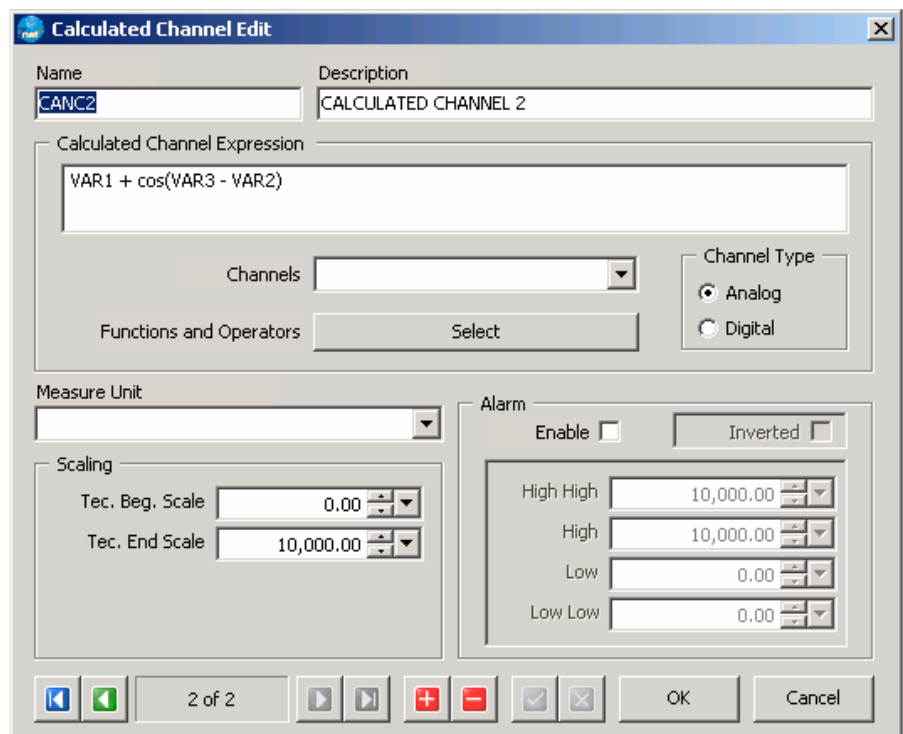
### 8.3 CALCULATED CHANNELS

Through this section the user may define some calculated channels which are functions of the channels previously defined in **Channels** section.

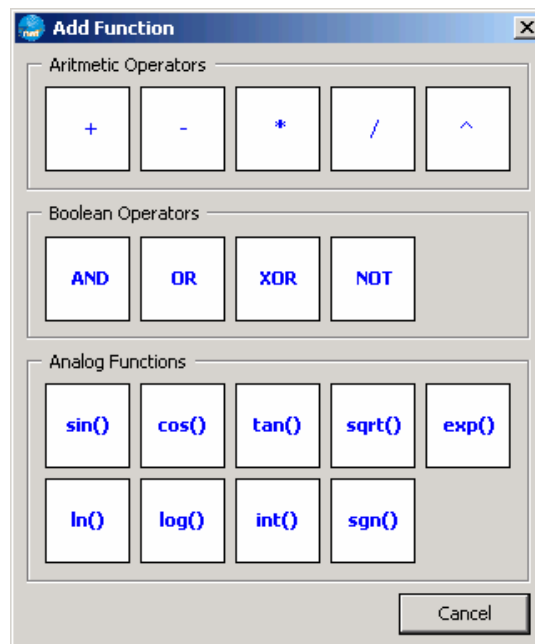
**Calculated Channels** window is the following one:



To add a new Calculated Channel, press on **Add Channel** and the following inserting mask will be opened:



To select a function to insert, press on **Functions and Operators**. The following Functions mask will be opened:



At this point it will be possible to choose one of the functions or operators in the table above.

Once a function has been inserted, the user has to add the argument, by writing directly the name of a channel previously defined (Channels section) or by selecting the name from the menu **Channels**.

The user may select through **Type Channel** field, if the channel will be analog or digital

Through **Measure Unit** menu, the user may select a measure unit to associate among the ones previously defined in the apposite section.

In **Scaling** section it will be possible to select a new visualization scale ( **Tec Beg Scale** , **Tec End Scale** )

Besides it is possible to define an alarm:

If the selected variable is a digital one, the user has to set the following parameters to define the alarm:

**Enable:** if checked, the alarm is enabled.

**Inverted:** If checked, the alarm will be activated on falling edges (True-->False) of the variable. Otherwise it will activated on rising edges (False-->True).

if the selected variable is an analog one, to define the alarm the user has to set the following parameters:

**Enable:** if checked, the alarm is enabled.

**High High:** sets the highest threshold (in technical scale) of the alarm.

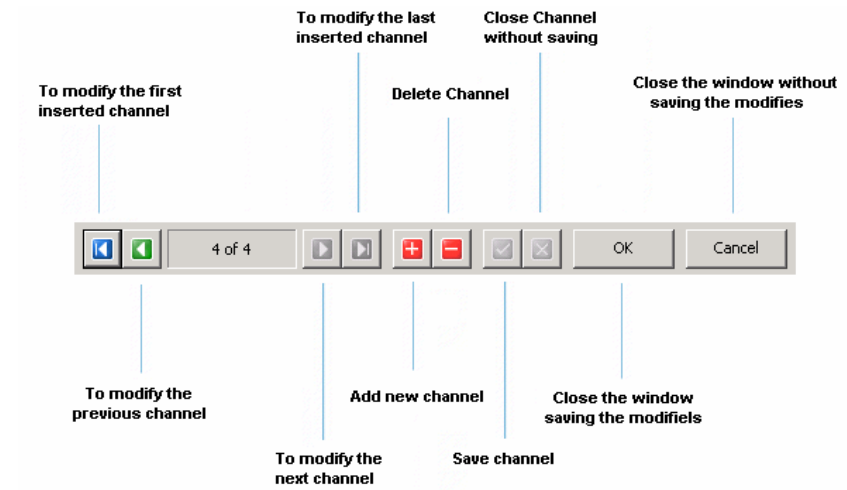
**High:** sets the high threshold (in technical scale) of the alarm.

**Low:** sets the low threshold (in technical scale) of the alarm.

**Low Low:** sets the lowest threshold (in technical scale) of the alarm.

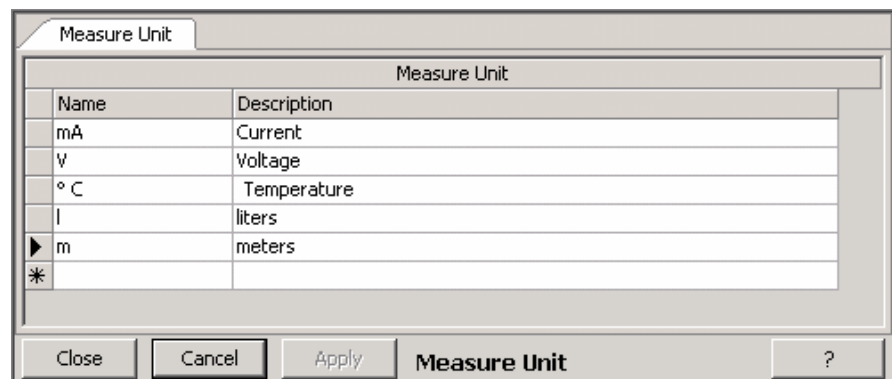
Once the adding has been completed, click on the correspondent icon of the below menu bar.

The icons of the menu bar have the following meaning:



## 8.4 MEASURE UNITS

Through this section it is possible to define some measure units to associate to the channels and calculated channels:



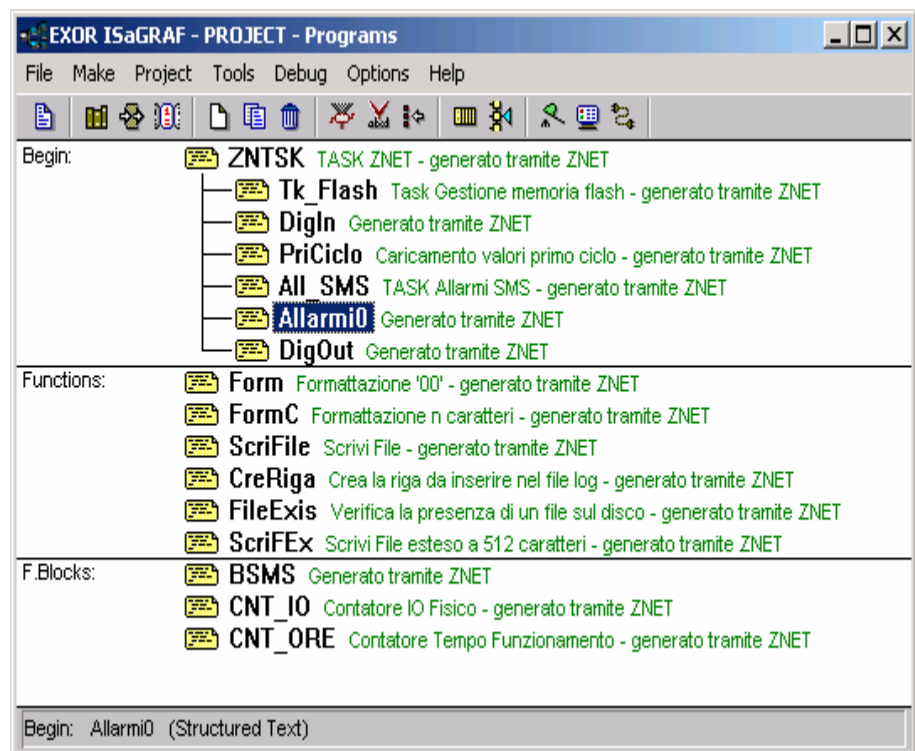
To define a new measure unit, it will be sufficient write the Name and the Description on the correspondent fields of the above table.

## 9 ISAGRAF PROJECT

Configuration Software Z-NET 2 allows the automatic generation of ISaGRAF variables and files which could be directly compiled and downloaded through Z-NET 2. Obviously if user desires to modify PLC's application, he has to open it with ISaGRAF environment.

### 9.1 ISAGRAF FILES GENERATION

The user may automatically generate ISaGRAF files, by clicking on **File | Make File ISaGRAF** or on the correspondent icon in the tools bar. Besides Z-NET 2 allows ISaGRAF project's direct compilation, by clicking on the icon in the tools bar. The user may also download PLC application by selecting **Online|Download ISaGRAF Project**. Once the ISaGRAF file has been generated, the user may start ISaGRAF by clicking the correspondent icon in the tools bar and open PLC's project.. For example Z-NET 2 creates the following ISaGRAF project:



The project is composed of three blocks.

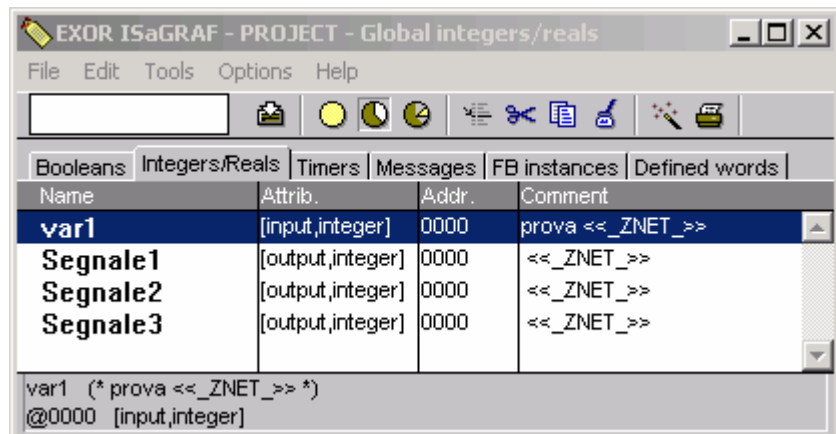
In the first block, the user may find all the programs which manage the operations on the flash memory, digital I/O, telecontrol's functionalities.

The second section is composed of some functions which control characters' formatting, files' writing and lines' creation in log files.

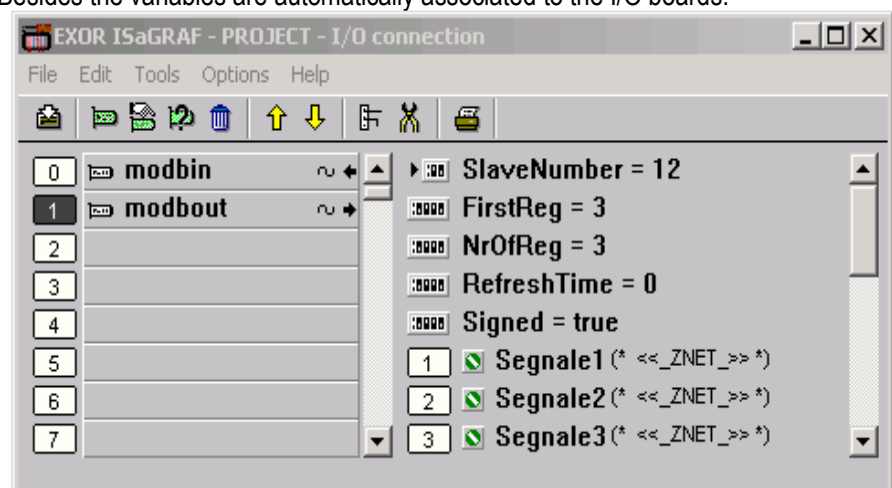
At the end there are all the used structural blocks.

## 9.2 VARIABLES DATABASE AND I/O BOARDS

Z-NET 2 creates automatically ISaGRAF variables database and the I/O boards for data acquisition from Z-PC I/O modules:



Besides the variables are automatically associated to the I/O boards:



## 9.3 LIBRARIES

SENECA developed some function libraries for ISAGRAF for the following purposes:

- Managing I/O modules through RS485 MODBUS connection
- Remote connection (PPP, sendMail,
- Miscellaneous (downloading/uploading files, reading writing variables from other TWSs, etc.)

These tools will help developing application with SENECA and other brand I/O modules.

To upload these libraries within ISAGRAF the following procedures must be followed:

- From main menu select Tools/Libraries.

- ii) Form menu on toolbar select the first row (I/O Configuration).
- iii) From menu Tool select Tools/Archive.
- iv) Select directory where Libraries have been stored (i.e. D:\Software\Isagraf Libraries).
- v) From window Archive all function regarding this typology (I/O Configuration) will be shown. Select all functions.
- vi) Press Restore.
- vii) Repeat step from ii) to vii) for all function types:
  - IO Complex Equipment
  - IO Boards
  - Functions
  - Function Blocks
  - C Functions
  - C Function Blocks
  - Conversion Blocks
- viii) Please note that some of the step for number vii) may not show any function.

In the following table the user may Nella tabella seguente viene fornito un elenco dei blocchi funzionali ISaGRAF implementati da Seneca con una breve descrizione della sua funzionalità. Per una descrizione più dettagliata si rimanda alla consultazione dell' APPENDICE 9 dove verrà spiegato in termini generali l'ambiente di programmazione PLC ISaGRAF.

Tipo Funzioni	Nome Funzione	Est.	Descrizione
<b>I/O CONFIGURATION</b>	SIM_ALL	ria	I/O simulation board
<b>I/O BOARDS</b>	MODBINADW	bia	Read double input registers
	MODBIN	bia	Read holding registers
	MODBINA	bia	Read input register
	MODBIND	bia	Read digital inputs
	MODBINDW	bia	Read double holding registers
	MODBOUTW	bia	Write double holding registers
	MODBOUT	bia	Write holding registers
	MODBOUTD	bia	Write digital outputs
	MODBOUTM	bia	Write holding registers( by using command 16)
<b>C FUNCTION</b>	ABS_A	uia	Absolute value ( integer ) soluto
	END_SER	uia	Close serial Port
	F_DELETE	uia	Delate file
	F_RENAME	uia	Rename file
	F_WOPENA	uia	Open a file to write it
	GET_SER	uia	Read characters from theserial port
	INIT_SER	uia	Initialize serial port
	PACKBOO	uia	Packs 16 digital variables into 1 analog
	PUT_SER	uia	Send characters through the serial port
	SCMFLRD	uia	Read variables from the flash memory
	SPRINTF	uia	Create a message variable withC
	TEST_SER	uia	Check the presence of a character in the serial buffer
<b>C FUNCTIONS BLOCKS</b>	MOD_PROT	fia	Implements Modbus Master functions
	PPPCONTACT	fia	Connection with PPP protocol
	SCMFLWR	fia	Write variables on flash memory
	SENDMAIL	fia	Sends Email Messages
	SENDSMS	fia	Sends SMS through GSM modem
	TWSREAD	fia	Read local or remote variable from Z-TWS

---

	TWSREADf	fia	Read a local or remote file fromTWS
	TWSWRITE	fia	Write a local or remote variable from TWS
	TWSWRITEf	fia	Wrrite a local or remote file from TWS
	UNPACKBO	fia	Unpacks 1 analog variable into 16 digitals

---

## 10 APPENDIX A: MODBUS

In this Appendix the main features of ModBus protocol will be described:

- Message's structure
- Functions,
- Communication's modalities
- Data exchange between master and slaves .

### 10.1 MODBUS MESSAGE STRUCTURE

Controllers can be setup to communicate on standard Modbus networks using either of two transmission modes:

- **ASCII mode:** each 8-bit byte in a message is sent as two ASCII Characters
- **RTU mode:** each 8-bit byte in a message contains two 4 bit hexadecimal characters.

A Modbus message has the following structure:

ADDRESS    FUNCTION    DATA    CHECKSUM

Where the fields' meaning is the following:

**ADDRESS:** If the message is a master's request, this field represents the address of the interrogated slave. Otherwise if the message is the slave's answer, this field will contain its address, to allow the identification by the master. The address has to be included between 0..247 ( decimal ).

**FUNCTION:** If the message is a master's request, this field contains the action to be done by the slave. Otherwise if the message is the slave's answer, the field gives an information about the operation's result.

**DATA:** Through this field some informations are given. It is not always present, dependig on the requested action.

### 10.2 MODBUS FUNCTIONS

Modbus protocol supports several functions which allows diverse operations on all data types. In Table 1 all Modbus functions and their identification code are listed.

Code	Name
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
07	Read Exception Status
08	Diagnostics
09	Program 484
10	Poll 484
11	Fetch Comm. Event Ctr.
12	Fetch Comm. Event Log
13	Program Controller
14	Poll Controller
15	Force Multiple Coils
16	Preset Multiple Registers
17	Report Slave ID
18	Program 884/M84
19	Reset Comm. Link
20	Read General Reference
21	Write General Reference

**Table 1**

In Table 2 the user may read Modbus Addresses of the different data types and the supported functions.

Data Type	Description	Modbus Address	Reading Functions	Writing Functions
IS	Digital Input	10001	02	
CS	Digital Coils	00001	01	15, 05
IR	Input Registers	30001	04	
HR	Holding Registers	40001	03	16, 06

**Table 2**