InteliCommunication Guide for ComAp Controllers

InteliLite-NT
InteliCompact-NT
InteliATS-NT

December 2009

ComAp
LEADER IN GEN–SET
COMMUNICATION SOLUTION

COMMUNICATION GUIDE

ComAp, spol. s r.o.
Kundratka 2359/17, 180 00 Praha 8, Czech Republic
Tel: +420 246 012 111 , Fax: +420 266 316 647
E-mail: info@comap.cz, www.comap.cz

AMPS

ISO 9001 CERTIFIED Quality Management ComAp, spol. s r.o.

Table of Contents

Document information

DOCUMENT HISTORY

REVISION NUMBER	DATE
1.0	20.2.2009
1.1	11.12.2009

Clarification of notation

NOTE:
This type of paragraph calls readers attention to a notice or related theme.

CAUTION!
This type of paragraph highlights a procedure, adjustment etc., which can cause a damage or unproper function of the equipment if not performed correctly and may not be clear at first sight.

WARNING!
This type of paragraph indicates things, procedures, adjustments etc. which need high level of attention, otherwise can cause personal injury or death.

Conformity Declaration

CE

The following described machine complies with the appropriate basic safety and health requirement of the EC Low Voltage Directive No: 73/23 / EEC and EC Electromagnetic Compatibility Directive 89/336 / EEC based on its design and type, as brought into circulation by us.

Introduction

This guide introduces the way how to connect to ComAp InteliLite-NT, InteliATS-NT and InteliCompact-NT controllers in order to access controller data and configure the controller.

Communication between controller(s) and superior, service or monitoring system (usually PC) is described within the manual providing essential information about the key components and methods of their usage.

This guide is not dedicated to the communication among controllers, communication with peripheral modules or ECU. Refer to the corresponding Reference Guides or Comap Electronic Engines Support guide to acquire necessary information in this field.

Available communication modules

Following communication plug-in modules are dedicated to be used with IL-NT, IC-NT and IA-NT controllers and are delivered separately from the controller. For more information about particular module go to chapter Communication modules.

SUPPORTED CONNECTIONS

COMMUNICATION MODULE (ORDER CODE)	DIRECT CONNECTION	MODEM CONNECTION	INTERNET CONNECTION
IL-NT-RS232	✓	✓	✗
IL-NT-RS232-485	✓	✓	✗
IL-NT-S-USB	✓	✗	✗
IB-Lite	✓	✗	✓

Available monitoring/configuration PC tools

MONITORING

PC TOOL	DIRECT CABLE	MODEM	INTERNET
LiteEdit	✓	✓	✓
InteliMonitor	✓	✓	✓
WinScope	✓	✗	✓
Modbus terminal	✓	✓	✓

PROGRAMMING AND CONFIGURATION

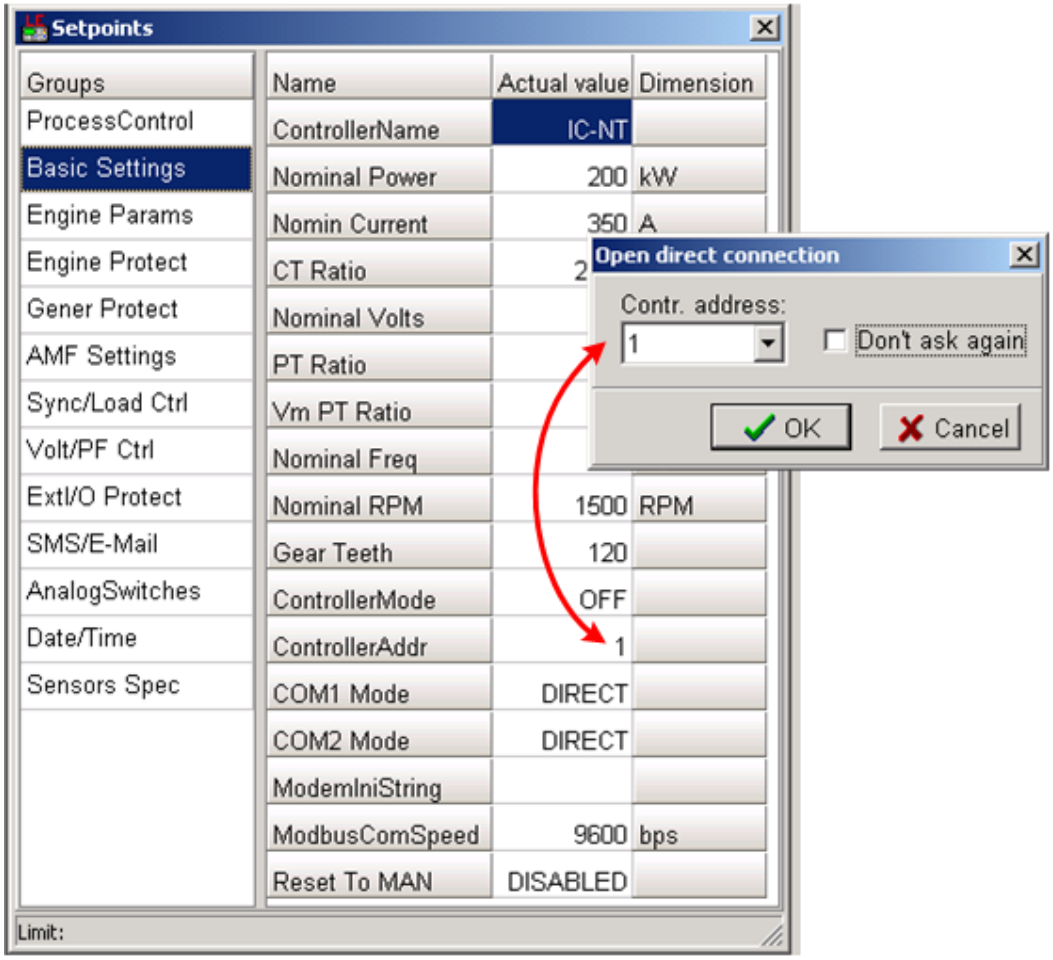
PC TOOL	DIRECT CABLE	MODEM	INTERNET
LiteEdit	✓	✓	✓
InteliMonitor	✗	✗	✗
WinScope	✗	✗	✗
Modbus terminal	✗	✗	✗

NOTE:
Details about ComAp PC tools can be found in the particular reference guides available on www.comap.cz for all ComAp Club members.

How to open connection

It is possible to connect to the controller using ComAp PC tools (LiteEdit, InteliMonitor and WinScope) or from web browser. Description how to open the connection follows.

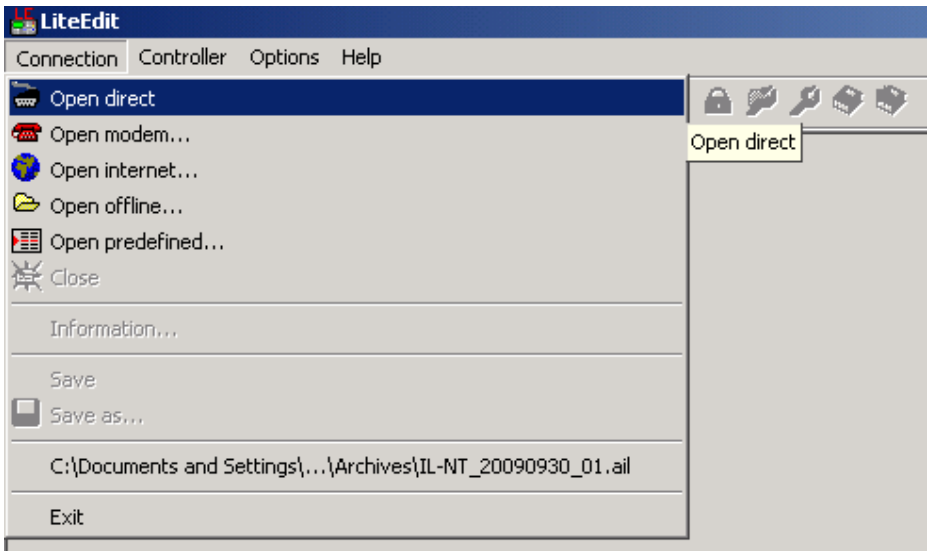
CAUTION!
When opening the connection to the controller it's address (ControllerAddr) has to correspond with PC SW communication setup setting.



THE CONTROLLER ADDRESS HAS TO MATCH WITH SETUP IN LITEEDIT/INTELIMONITOR

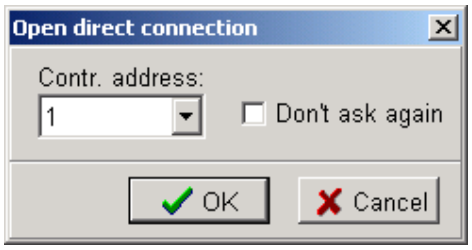
Open connection from LiteEdit

1. Go to menu ***Connection*** and select the type of connection you desire.

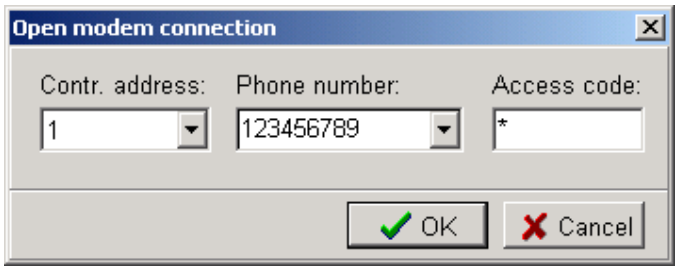


LITEEDIT - CONNECTION MENU

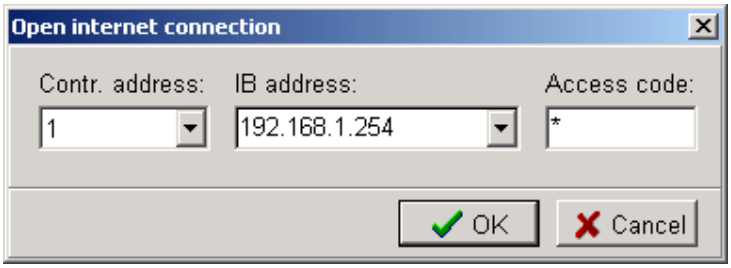
2. Enter controller address and further information depending on the selected connection type.



FOR DIRECT CONNECTION ENTER CONTROLLER ADDRESS

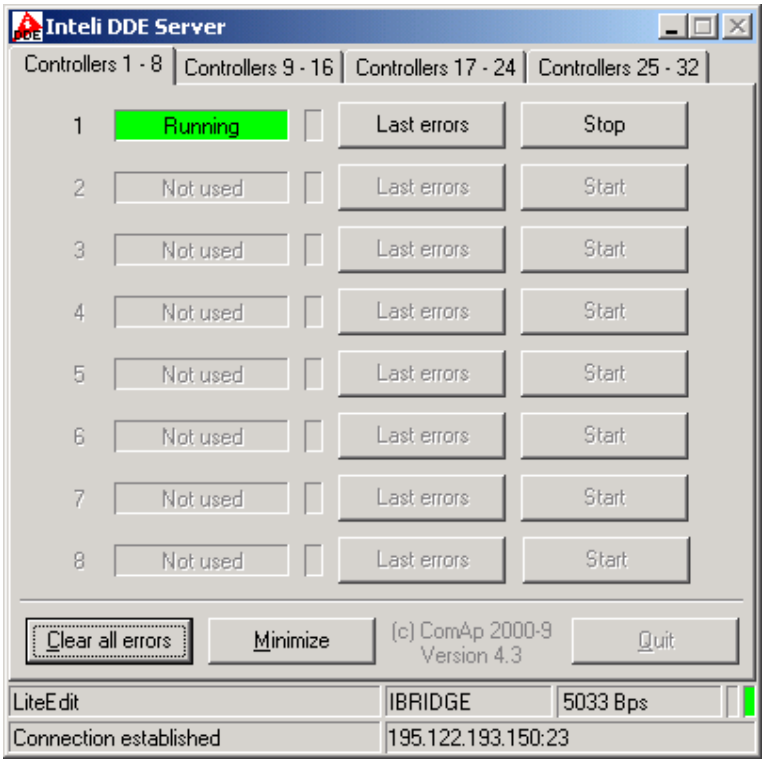


FOR MODEM CONNECTION ENTER CONTROLLER ADDRESS, PHONE NUMBER OF THE MODEM SITUATED AT THE CONTROLLER YOU WANT TO REACH, ACCESS CODE



FOR INTERNET CONNECTION ENTER CONTROLLER ADDRESS, IP ADDRESS OF THE IB-LITE MODULE FITTED IN THE CONTROLLER YOU WANT TO REACH, ACCESS CODE

3. You can check the status of communication in InteliDDE Server

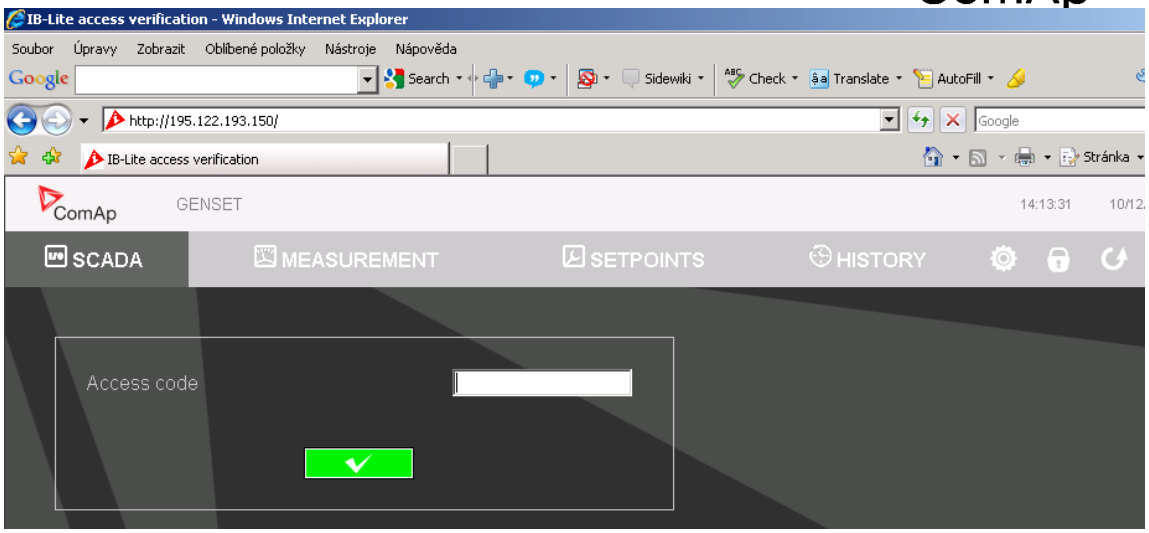


COMMUNICATION WITH THE CONTROLLER WITH ADDRESS 1 IS RUNNING THROUGH IB-LITE ON IP ADDRESS 195.122.193.150, STANDARD PORT 23

Open connection from web browser

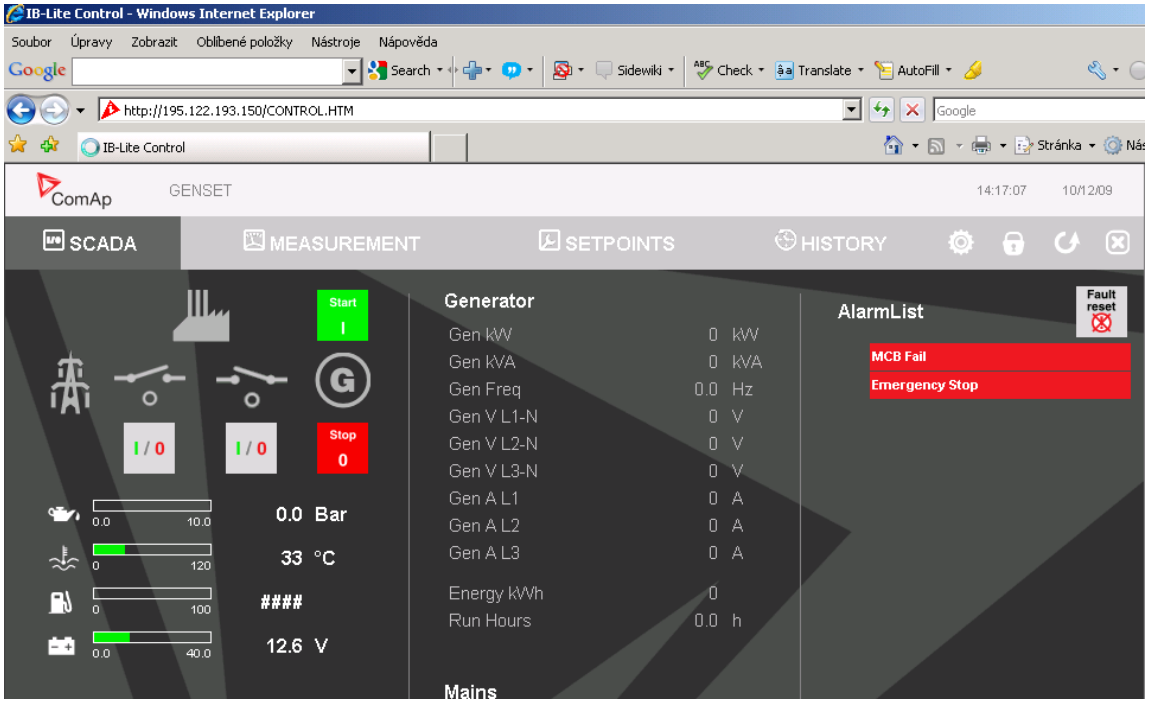
It is possible to connect from a web browser to any controller mounted with IB-Lite module and connected to internet (for more details about internet connection see related chapter or IB-Lite reference guide).

1. Open web browser
2. Enter IP address
3. IB-Lite access verification page appears



ACCESS VERIFICATION PAGE

4. Enter access code and Scada page appears

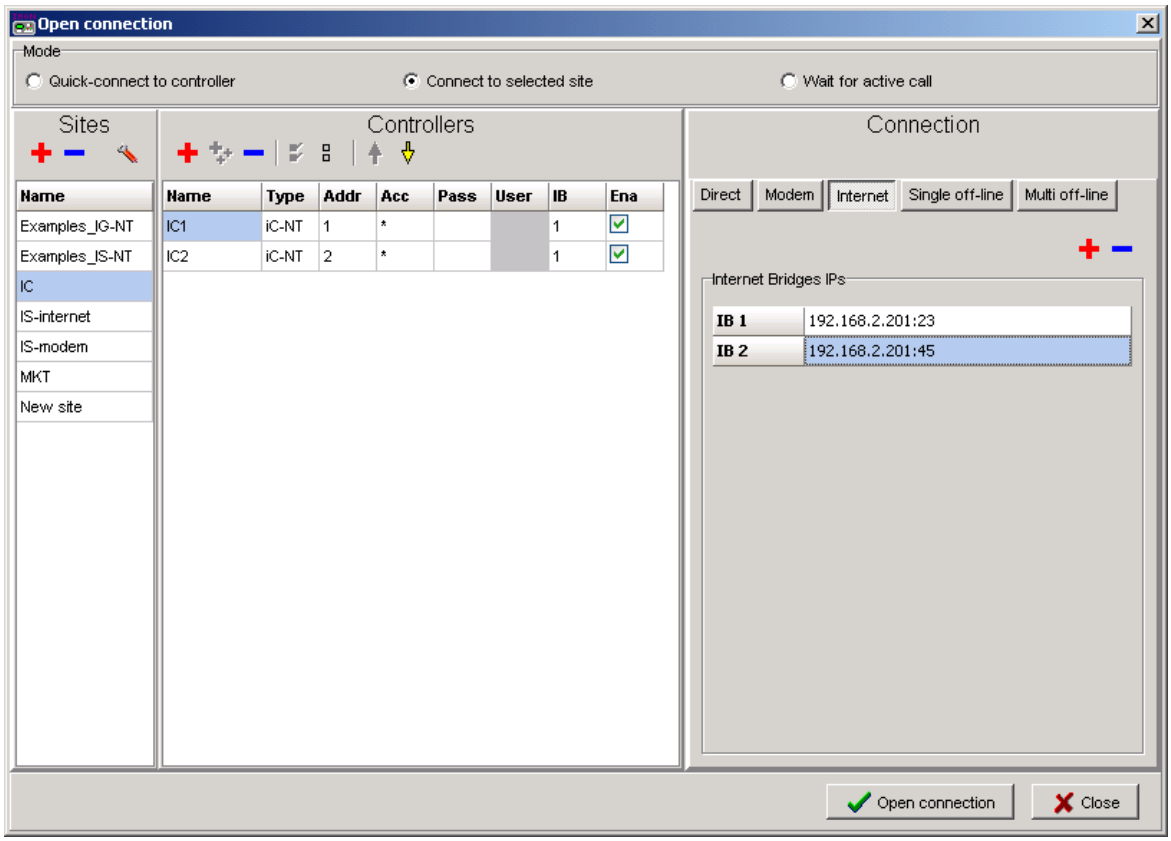


SCADA PAGE

NOTE:
You can try the connection to IB-Lite installed in ComAp on IP address 195.122.193.146 (IC-NT controller) and 195.122.193.150 (IL-NT controller). The access code is 0.

Open connection from InteliMonitor

1. Go to menu ***Connection*** -> ***Open connection...*** and select the type of connection you desire and site where you would like to connect.

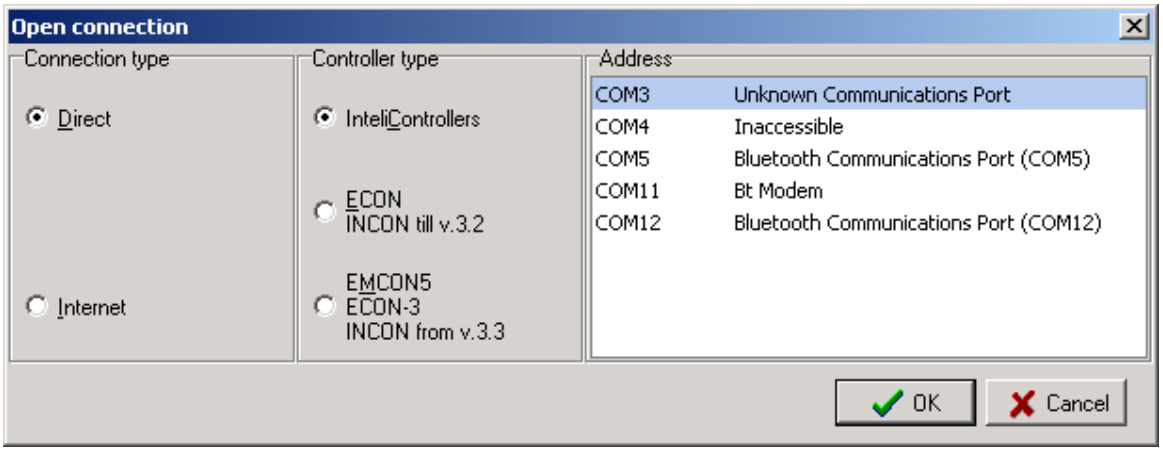


INTELIMONITOR - OPEN CONNECTION WINDOW

2. Enter necessary information depending on the selected connection type.
3. Press Open connection button

Open connection from WinScope

1. Go to menu ***Connection*** -> ***Open connection...*** and select the type of connection you desire in Open connection window (Inteli controllers).



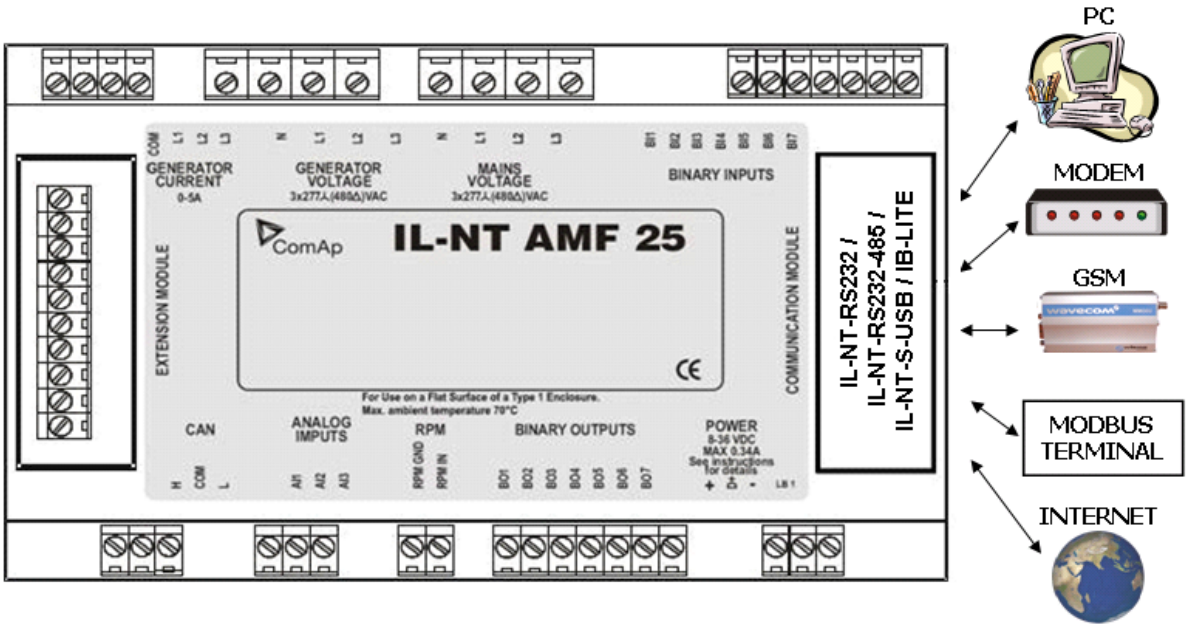
WINSCOPE - OPEN CONNECTION WINDOW

2. Proceed with selection of channels etc. according to WinScope reference guide

Controllers Communication Capabilities

IL-NT and IA-NT

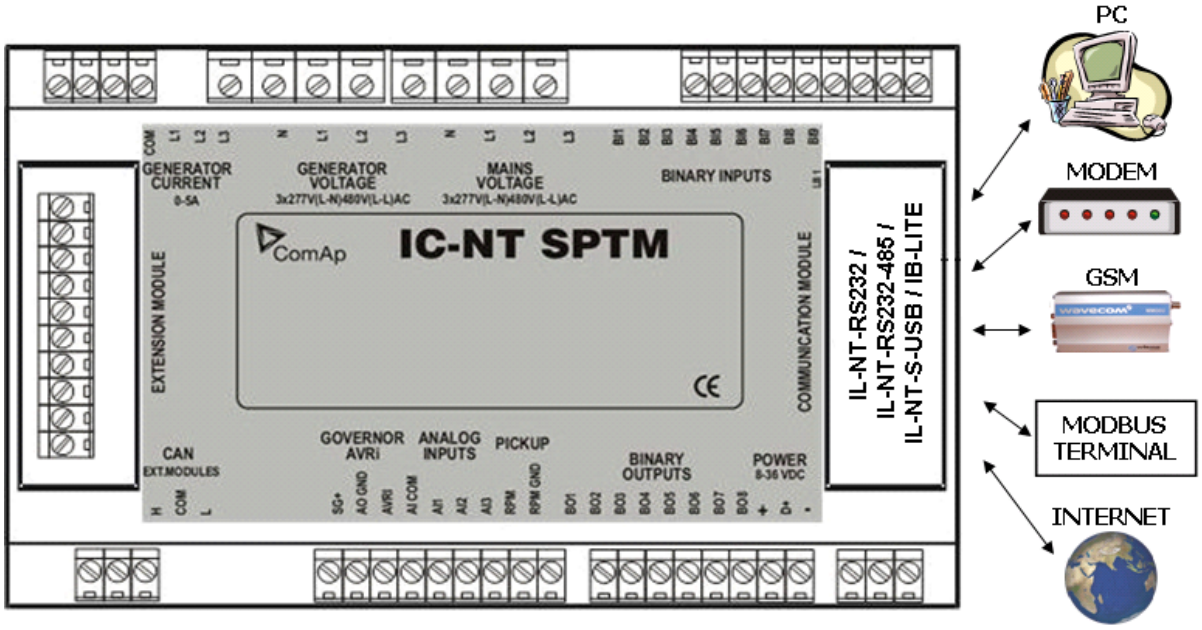
The following diagram shows the communication capabilities of IL-NT and IA-NT controller in full configuration.



POSSIBLE CONNECTIONS TO IL-NT/IA-NT CONTROLLER - DIRECT FROM PC, THROUGH MODEM, INTERNET AND FROM MODBUS TERMINAL.

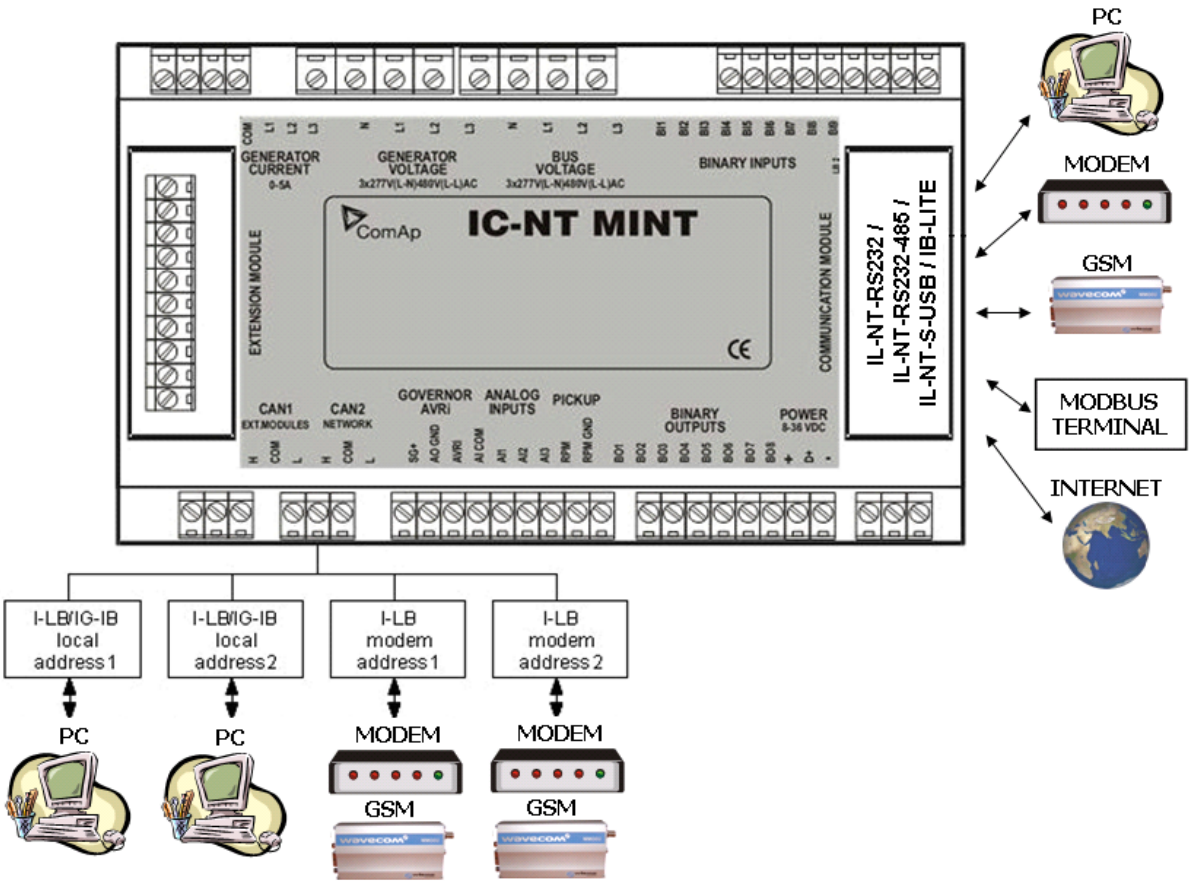
IC-NT SPTM

The following diagram shows the communication capabilities of IC-NT controller in SPTM.



IC-NT MINT

The following diagram shows the communication capabilities of IC-NT controller in MINT.

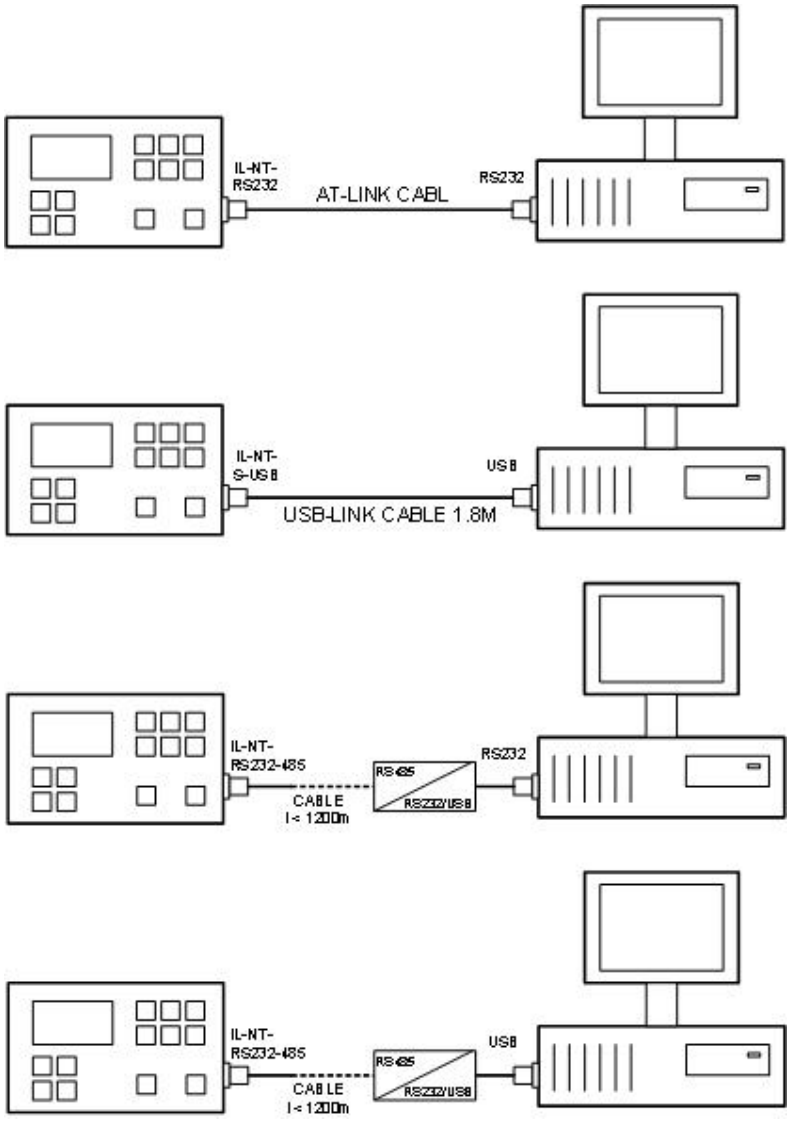


POSSIBLE CONNECTIONS TO IC-NT MINT CONTROLLER - DIRECT FROM PC, THROUGH MODEM, INTERNET AND FROM MODBUS TERMINAL.

Direct cable connection

An external communication module is necessary to enable direct cable connection to a PC. The module is plugged-in into the slot located on the rear side of the controller.

RS232, USB or RS485 interface can be used for direct cable connection to a PC. The setpoint COM1 Mode or COM2 Mode (according to the interface used) must be set to DIRECT position for this kind of connection.



DIRECT CABLE CONNECTION TYPES

Following modules are available for direct connection to a PC:

1. IL-NT-RS232
2. IL-NT-RS232-485
3. IL-NT-S-USB (USB easy removable service module)

The RS232 or USB interface uses COM1 port of the controller. The RS485 uses COM2.

The communication speed of direct connection is up to 38400 bps, via USB it is up to 115200 bps.

The RS485 communication line has to be terminated by 120 Ohm resistors on both ends. Follow RS485 converter user manual. More information about RS232/485 converter see in chapter Recommended converters.

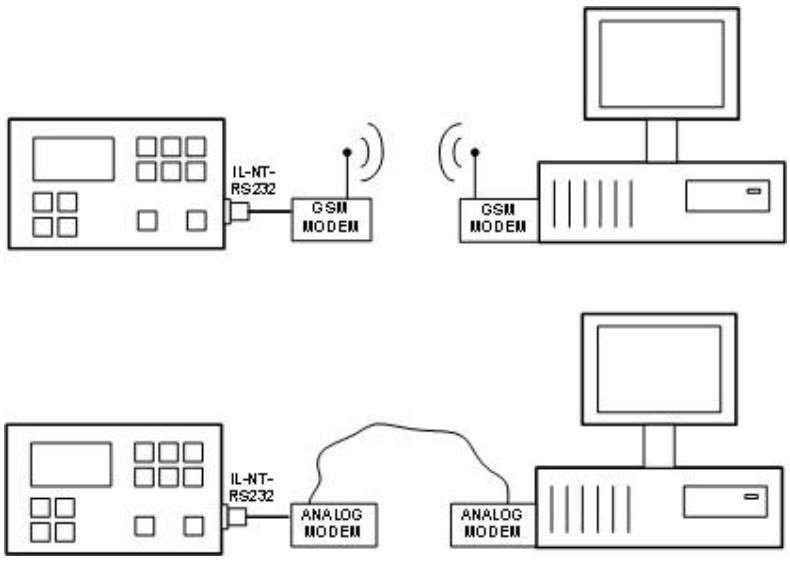
NOTE:
Use cross-wired serial communication cable with DB9 female connectors and signals Rx, Tx, GND for RS232 connection.

NOTE:
For connection to multiple controllers refer to separate chapter Connection to multiple controllers.

Modem connection

A PC can be connected to the controller also remotely via modems. Either an analog or GSM or ISDN modem must be connected to the RS232 interface and the setpoint COM1 Mode must be set to MODEM.

CAUTION!
For connection using GSM modems the CSD protocol must work in the network.



MODEM CONNECTION TYPES

Following modules can be used for modem connection to a PC:

1. IL-NT-RS232
2. IL-NT-RS232-485

The RS232 interface uses COM1 port of the controller.

In case of troubles with the modem communication an additional initialization string may be required. The reason can be for example some national telephone network specific feature. Use the setpoint ModemIniString to add some necessary AT commands which will be sent to the modem during the initialization. See the documentation of the modem for details.

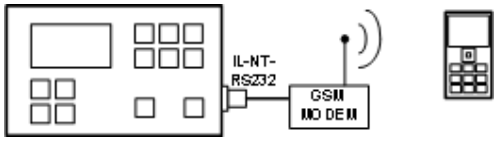
NOTE:
Use the same kind of modem (e.g. analog, GSM or ISDN) as used on the controller also at PC side.

NOTE:
The communication speed is up to 38400 bps (limited by modem used).

NOTE:
For connection to multiple controllers refer to separate chapter Connection to multiple controllers.

Active SMS

If SMS active calls are activated for alarms on site (yellow/red alarms) the controller sends SMS message to the predefined GSM number.



ACTIVE SMS

The controller first attempts to send SMS using modem connected to RS232 and then using modem connected to I-LB module.

Example of SMS sent by the controller in case that the water temperature exceeded the warning limit and Emergency stop input has been : #Gen-set name:AL=(Wrn Water temp; !Emergency stop)

Modem setup procedure

Analog modems obviously do not require any setup. The only case it could be needed is if the modem has been bought in other country with different telephony system than the target country where the modem will be used.

GSM modems need to be set-up prior to using with the controller. Use the *gm_setup* program (installed together with the LiteEdit) to make the initial setup of the modem. The setup must be done while a SIM card is inserted.

General conditions
Following conditions must be fulfilled:

1. There must be GSM modem on PC and controller side (not different modem types e.g. analog and GSM modem).
2. Data communication capability must be enabled for the SIM cards (CSD (Circuit Switch Data) must be supported). Ask your operator for this service. If it is not enabled, Gm_setup program shows "Command failure" message at the end of the log. To check SIM card data setting move SIM card from the GSM modem connected to the controller to mobile phone, call from LiteEdit to this mobile phone and check (on mobile phone) DATA call indication of incoming call. If phone does not indicate DATA - solve this with your GSM operator.

Modem configuration

1. Connect to the modem from your PC.
2. Run Gm_setup.exe (the program is in ../Tools/Gm_setup directory)
3. Select communication port (COM 1 - 32) and press Setup.
4. Enter SIM card PIN after you will be asked.
5. Enter SMS center address after you will be asked (ask your operator about this number).
6. If the Gm_setup writes "Setup terminated successfully" the SIM card is configured for the communication with the controller.

NOTE:
All SMS on SIM card will be erased during GSM modem initialization.

Controller configuration

1. In SMS/E-mail group of setpoints:

   To enable sending of SMS from the controller in case of alarms, you should select with setpoints Yel Alarm Msg and Red Alarm Msg, which levels of alarms shall be announced (red/yellow/both) and

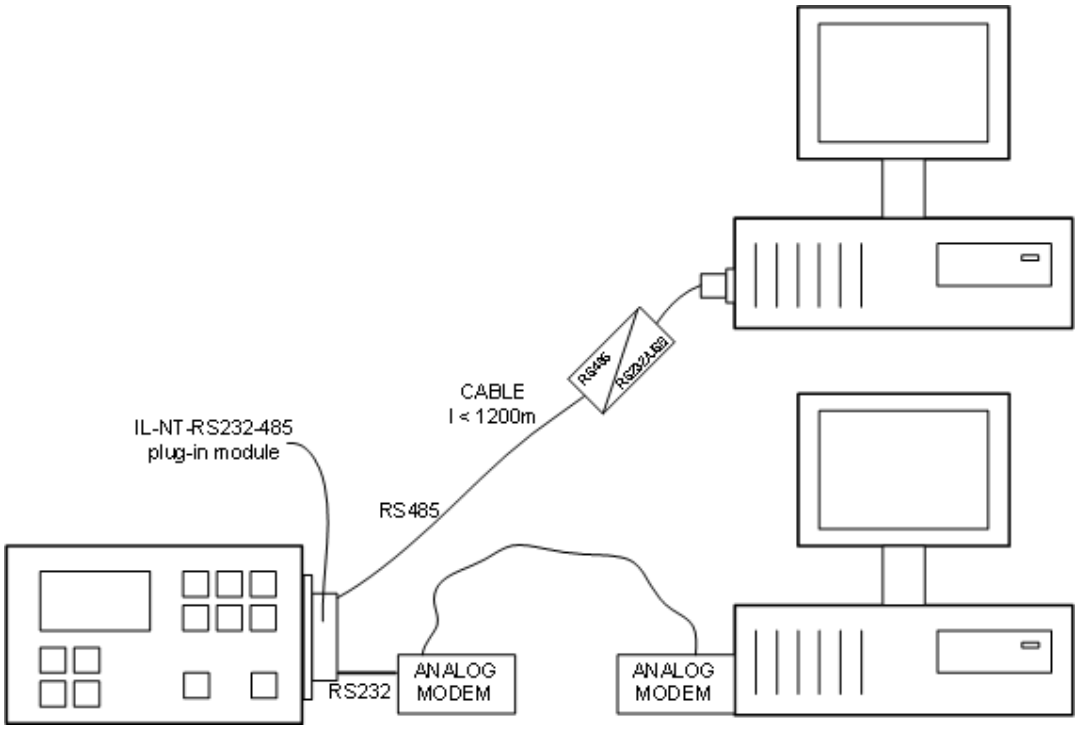
also enter valid GSM phone number and/or e-mail address to the setpoints TelNo/Addr Ch1 and TelNo/Addr Ch2.

2. Connect the controller to the modem or I-LB unit that has the modem connected.

Combination of direct cable and modem connections

Connection to single controller

The controller can be accessed through direct and modem connection simultaneously. One PC is connected via direct cable connection using RS485 (COM2 port) and second PC is connected via modem (COM1 port).



COMBINED CONNECTION TO THE SINGLE CONTROLLER

NOTE:
For connection to multiple controllers refer to separate chapter Connection to multiple controllers.

Internet connection

Internet connection

A PC can be connected to the controller also remotely via Ethernet (Internet, Intranet). An appropriate ethernet communication module must be used.

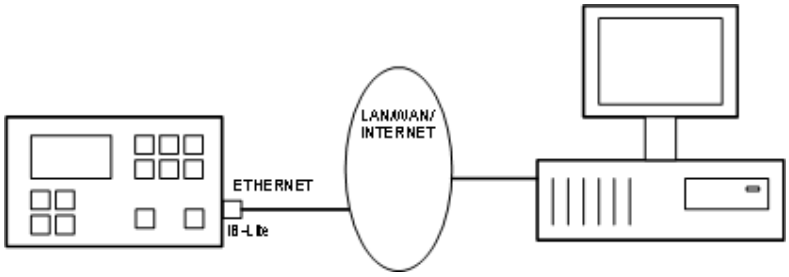
CAUTION!
It is necessary to fulfil one of the following conditions:

1. Provide static and public IP address
2. Provide static IP address within VPN

IL-NT, IA-NT-STD and IC-NT-SPtM

For connecting the IL-NT, IA-NT-STD or IC-NT SPtM controller to the internet use a plug-in communication module IB-Lite.

The setpoint COM1 Mode must be set to the DIRECT position.

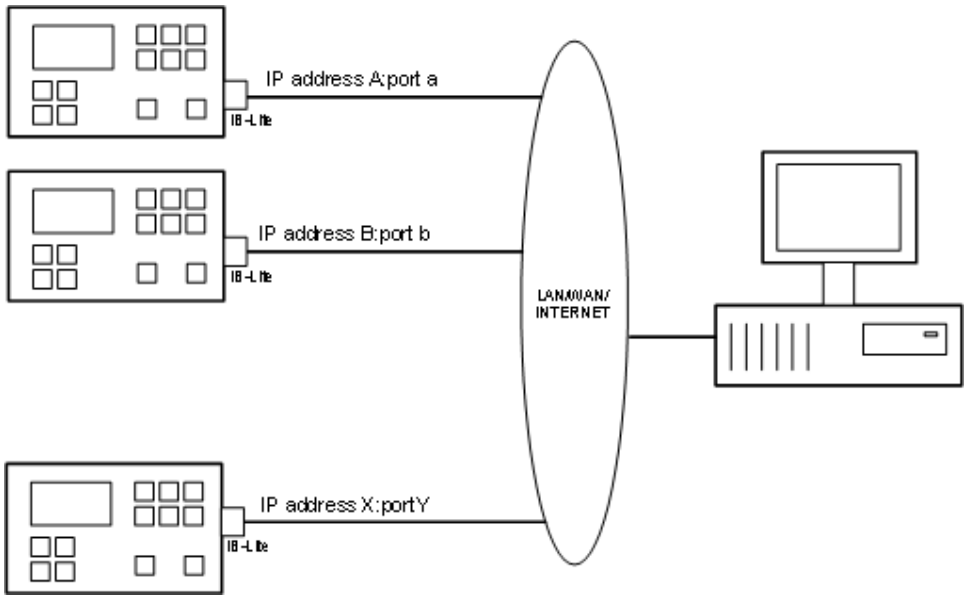


INTERNET CONNECTION USING IB-LITE

IL-NT, IA-NT-PWR and IC-NT-MINT

For connecting the IL-NT, IA-NT-PWR or IC-NT MINT controllers to the internet use a plug-in communication module IB-Lite on each controller. The setpoint COM1 Mode must be set to the DIRECT position.

CAUTION!
This type of connection is possible only for the IL-NT controllers which have the ControllerAddr setpoint available (IL-NT-MRS10, 11, 15, 16, AMF 20, 25).



INTERNET CONNECTION TO MULPTIPLE GEN-SETS USING IB-LITE

Each IB-Lite can have different IP address and different port number. There cannot be two IB-Lites with the same combination of IP address and port number on one site.

Default port number is 23. Default IP address is 192.168.2.254.

It is possible to connect simultaneously

- 2 clients with LiteEdit/InteliMonitor (Comap/TCP protocol)
- 1 client Modbus/TCP
- 2 clients with web interface

In case of connection from web browser there is 5 minutes timeout after closing the browser window. After that the client is automatically logged out.

Using a web browser

The IB-Lite module with firmware version 1.1 and above makes possible using any web browser for basic monitoring and adjustment of the controller. Simply put the IP address of the module into the address line in your web browser like `http://192.168.1.254` and then enter access code.

IB-Lite setup procedure

NOTE:
Setup of the module requires certain level of knowledge of networks administration. Ask your IT specialist for assistance.

Default setting of the module is IP = 192.168.1.254, Netmask = 255.255.255.0, Gateway = 192.168.1.1, mode 100Mbit. Default user name for service web pages is "comap", password "comap".

To restore default setting close the "restore default setting" jumper located on the module before switching the controller on and remove it few seconds after the controller was switched on.

Configuration

1. Plug the module into the controller and power the controller on.
2. Connect the module into your ethernet network. If the default address does not match local network parameters (i.e. the network segment does not use IP range 192.168.1.xxx or the IP 192.168.1.254

is occupied), connect the module directly to your PC using cross-wired cable. See details in the Installation chapter.

3. If you are connected direcly, you have to change temporarily IP address and subnet mask of your PC Ethernet connection. Use following setting: DHCP disabled, IP from the range 192.168.1.1 - 192.168.1.253 and subnet mask 255.255.255.0. After the IB-Lite setup is finished, restore your PC setting back to original values.
4. Start web browser and put `http://192.168.1.254/sp_config.htm` into the address line.
5. After successful login the configuration page will be displayed.
6. It is recommended to change the user name and password and keep the new values confidential.
7. Consult proper IP settings with your IT specialist.
8. Consult proper e-mail settings with your e-mail provider. Please note, that also most of public SMTP servers require authentification and e-mails must be sent from an existing addresses.
9. If you want to enable access only for clients with specified IP addresses, tick the checkbox "Trusted clients" and fill-in the allowed IP addresses.

NOTE:
For connection from PC see the manual of the PC program (InteliMonitor, LiteEdit). Open the Open connection window and set:

- Internet type of connection
- Controller address
- Access code
- IB-Lite IP address

Note that IP address you set can be different from the IP address of IB-Lite (when the IB-Lite IP address is not public). It depends on gateway setting, for more information see IB-Lite manual accessible on www.comap.cz.

Firmware upgrade

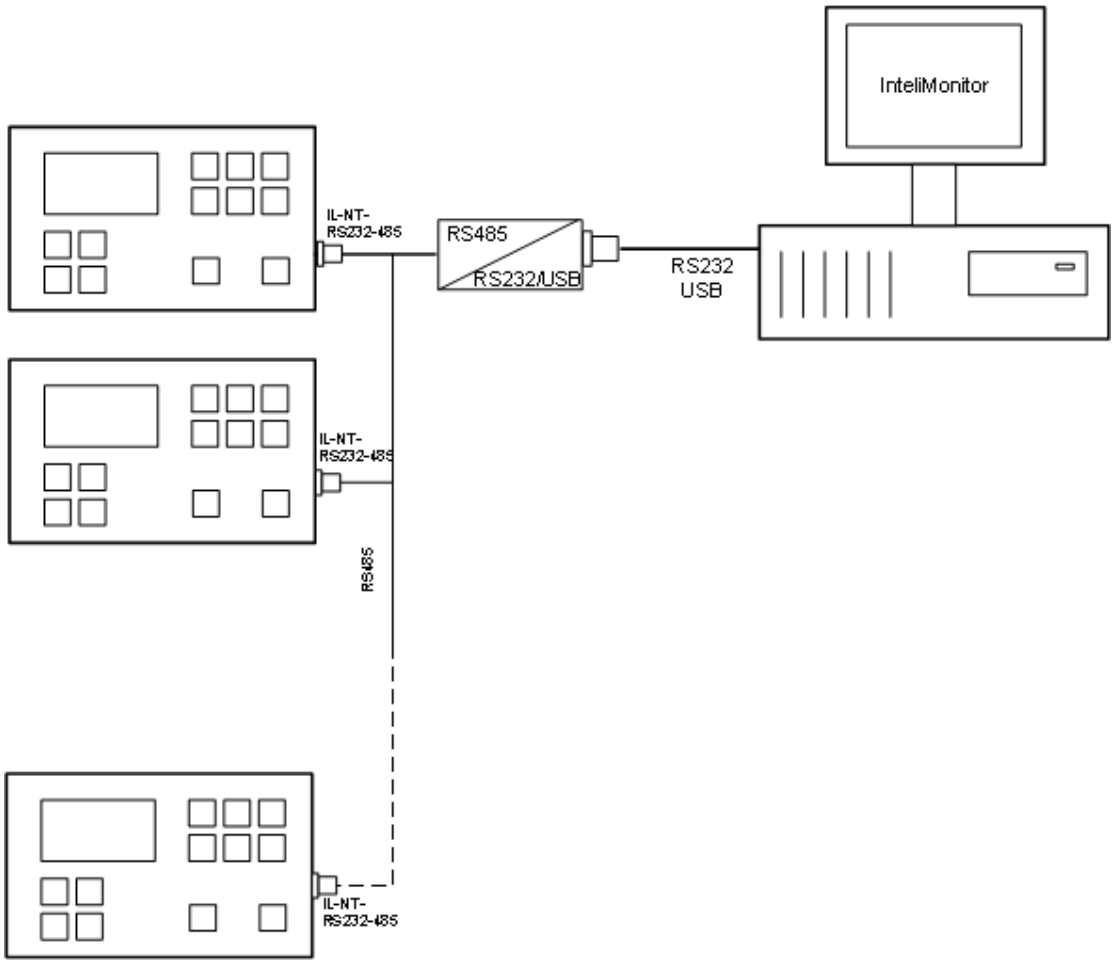
1. Follow steps 1-3 of the configuration procedure above.
2. Start web browser and put `http://192.168.1.254/sp_fw_upld.htm` into the address line.
3. After successful login the configuration page will be displayed.
4. Press the button "Browse" and select the appropriate firmware file.
5. Press "Upload new firmware" button. After the firmware upload is finished, the module will restart.

NOTE:
Interrupting the upload will NOT cause any damage. Just repeat the upload again.

Connection to mupltiple controllers

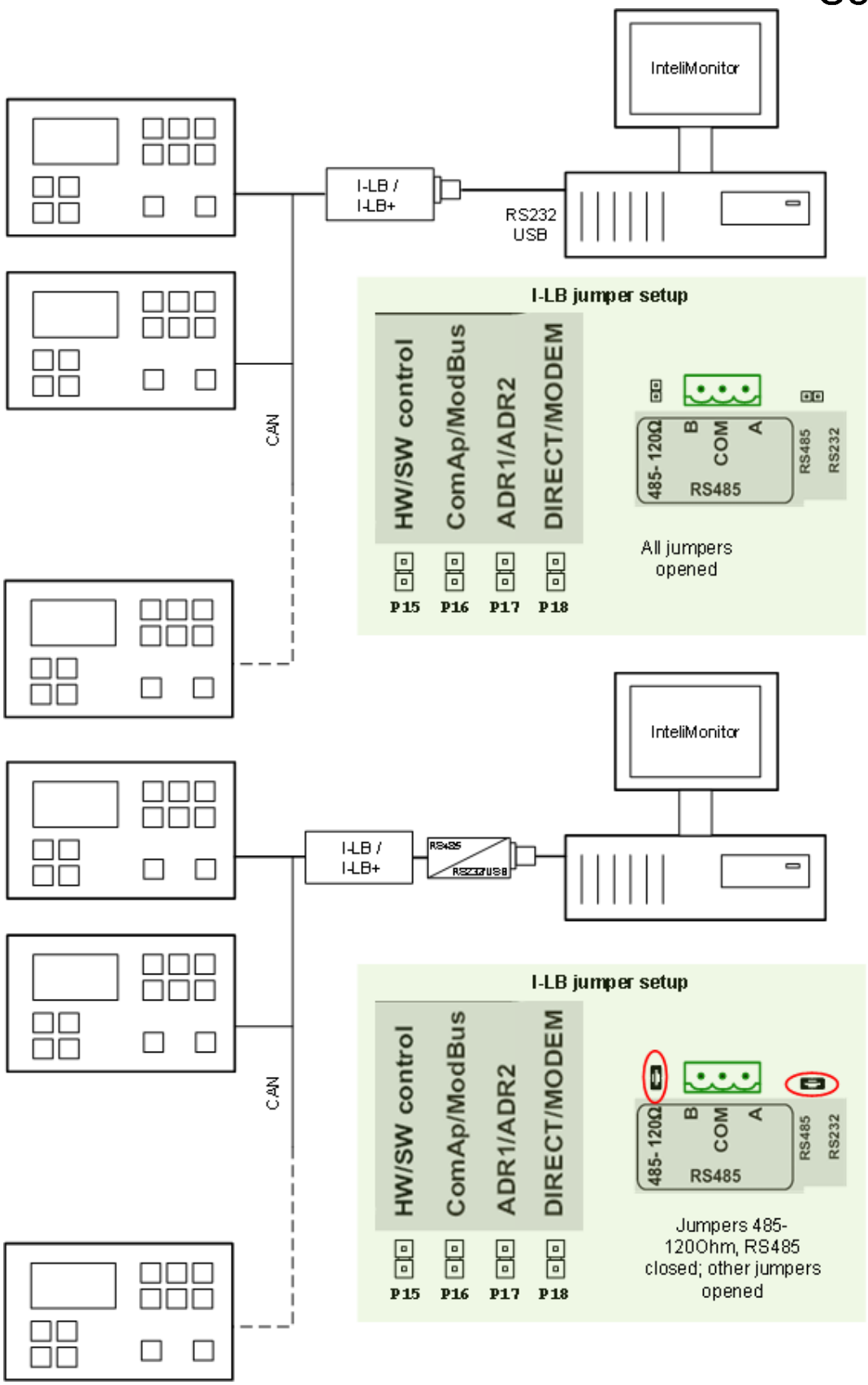
Direct cable connection to multiple controllers

It is possible to connect to multiple controllers on the site using RS485 network (IC-NT, IL-NT and IA-NT controllers).



DIRECT CABLE CONNECTION TO MULTIPLE CONTROLLERS

Using I-LB module connection to multiple IC-NT-MINT only controllers is also possible. The controllers are connected by the intercontroller CAN bus (marked as CAN2 on the rear sticker). There is another CAN bus port (marked as CAN1) which is used for connection of peripheral modules like IGS-PTM, IS-AIN8 etc. or ECU. The CAN bus length is limited to 200 meters if setpoint CAN Bus Mode is set to 32C (communication speed is 250kbps) or to 900 meters if it is set to 8C (communication speed is 50kbps). This setpoint defines the maximum number of controllers connected to the CAN bus.

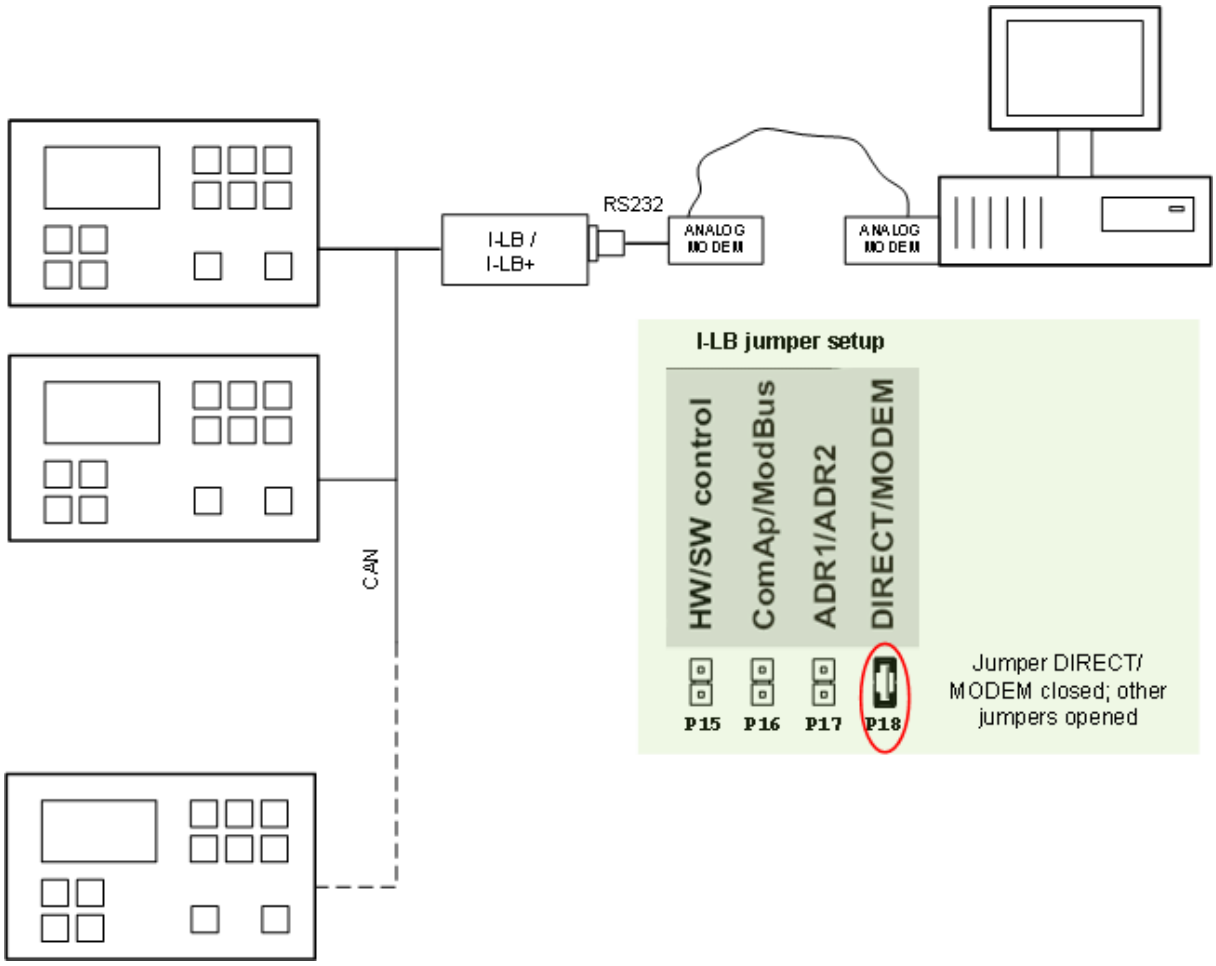


DIRECT CABLE CONNECTION TO MULTIPLE IC-NT CONTROLLERS THROUGH I-LB MODULE

NOTE:
I-LB unit must be switched to RS485 mode for remote direct connection by "DIRECT/MODEM" and "RS485" jumper. The terminating resistor is intergrated - use "RS485-120Ohm" jumper.

Modem connection to multiple controllers (IC-NT-MINT only)

Modem connection to multiple IC-NT controllers on the site is possible via I-LB module. Recommended modem types see in chapter Modem Recommendations.



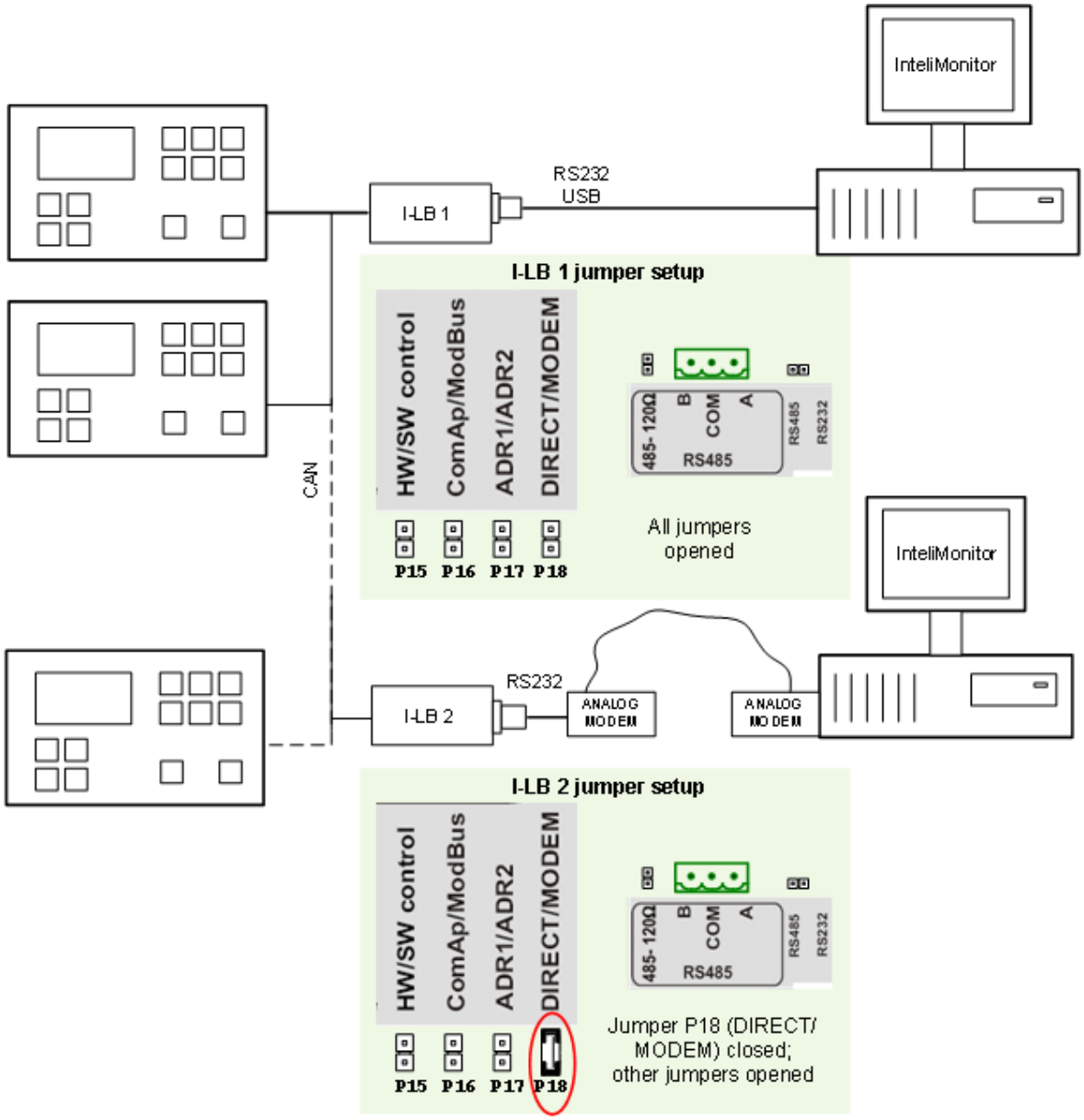
MODEM CONNECTION TO MULTIPLE CONTROLLERS VIA I-LB

I-LB has to be connected to modem via null modem cable with full handshaking where the DSR (Data Set Ready) signal detects modem presence.

NOTE:
The controller address has to be set correctly - each gen-set in the group must have it's own unique number in the range 1 to 32 (ControllerAddr).

Combined direct and modem connection to multiple IC-NT controllers

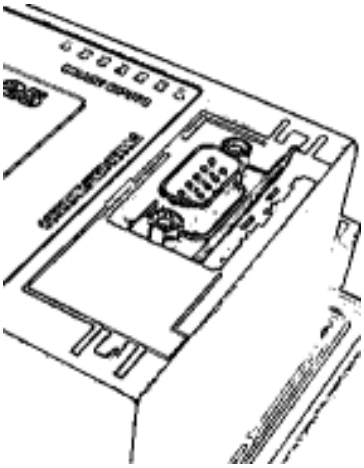
The controllers can be accessed through direct and modem connection simultaneously. One PC is connected via direct cable connection using RS232 and second PC is connected via modem.



COMBINED CONNECTION TO MULTIPLE CONTROLLERS

Communication modules

Communication module enables connection of a remote computer or other remote device such as PLC to the controller. The module is to be plugged-in into the slot in the rear side of the controller. The slot is accessible after slot cover is removed.

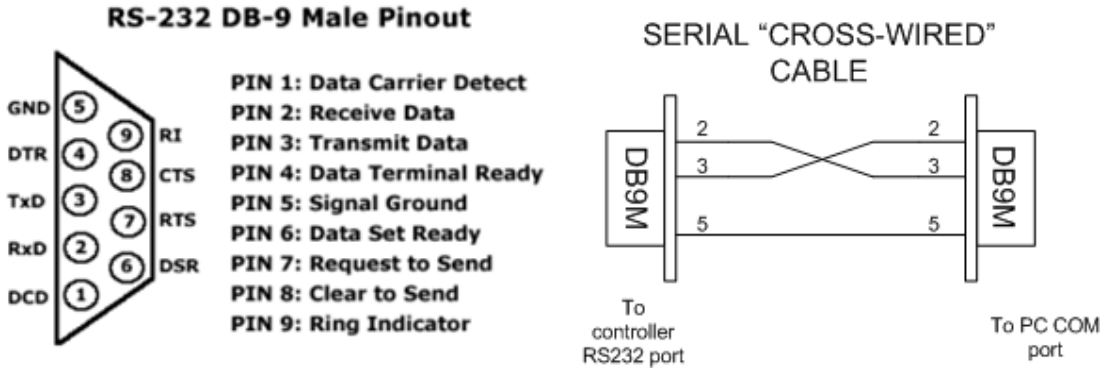


SLOT FOR COMMUNICATION MODULES

NOTE:
The modules are compatible with the IL-NT, IC-NT, IA-NT controllers. More information about how to install modules can be found in the controllers' manuals.

IL-NT-RS232

This module contains a RS232 port with all modem signals connected internally to the COM1 of the controller. DB9M connector is used on the RS232 side.



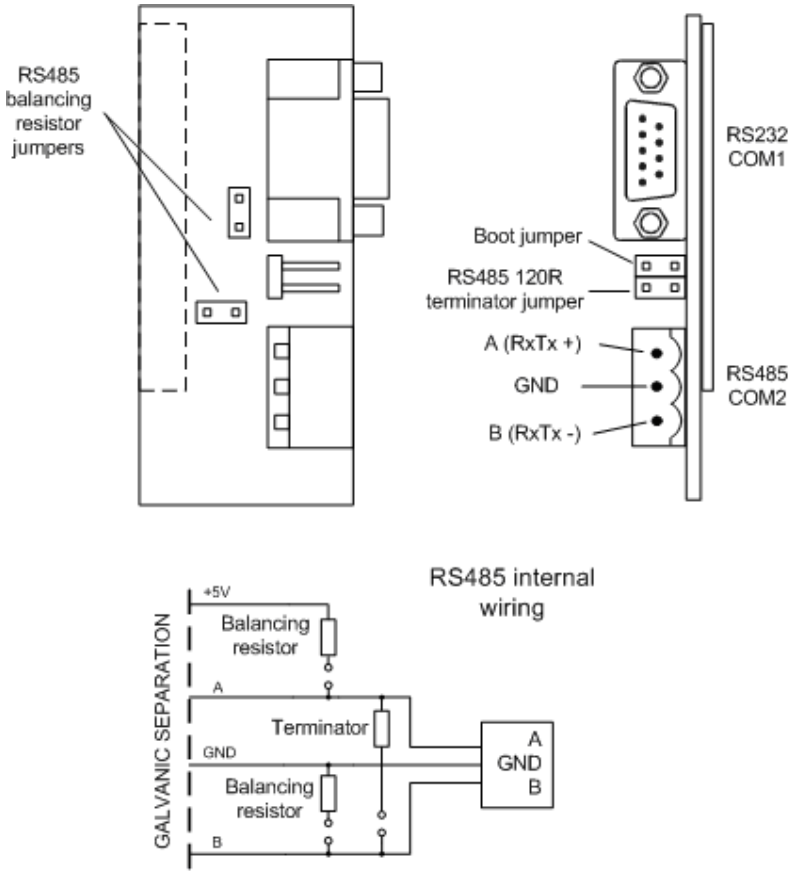
RS232 PINOUT AND CABLE WIRING

NOTE:
The Comap order code is AT-LINK CABL.

IL-NT-RS232-485

The IL-NT-RS232-485 is a dual port module with RS232 and RS485 interfaces at independent COM channels. The RS232 is connected to COM1 and RS485 to COM2.

For RS485 connection use twisted pair. The length is up to 1 km.



IL-NT-RS232-485 MODULE

IL-NT-S-USB

This module contains USB slave port connected internally to the COM1 of the controller and is designed as an easy removable service module.

This module requires a FTDI USB Serial converter driver installed in the PC. The driver creates a virtual serial port (COM) in the PC, which must be used in LiteEdit as communication port when a connection is beeing opened.

NOTE:
The FTDI driver is installed together with LiteEdit.

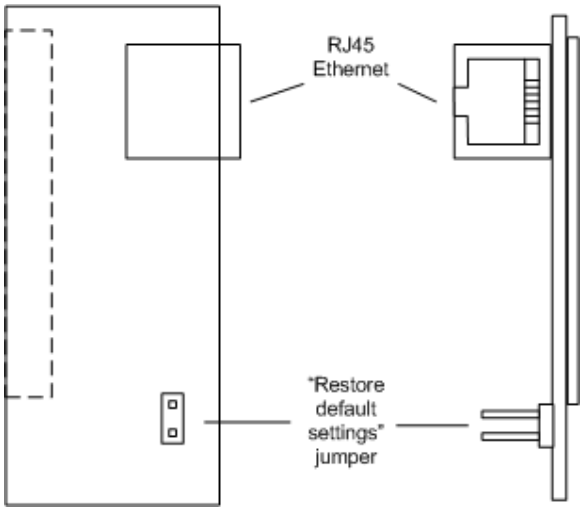
NOTE:
When the USB cable from the controller is plugged-in first time into different USB ports on the PC including USB hubs, it can be recognized as new hardware and the drivers are installed again with different number of the virtual serial port.

CAUTION!
Use shielded USB cable only! (ComAp order code: USB-LINK CABLE 1.8m)

USB A-B CABLE

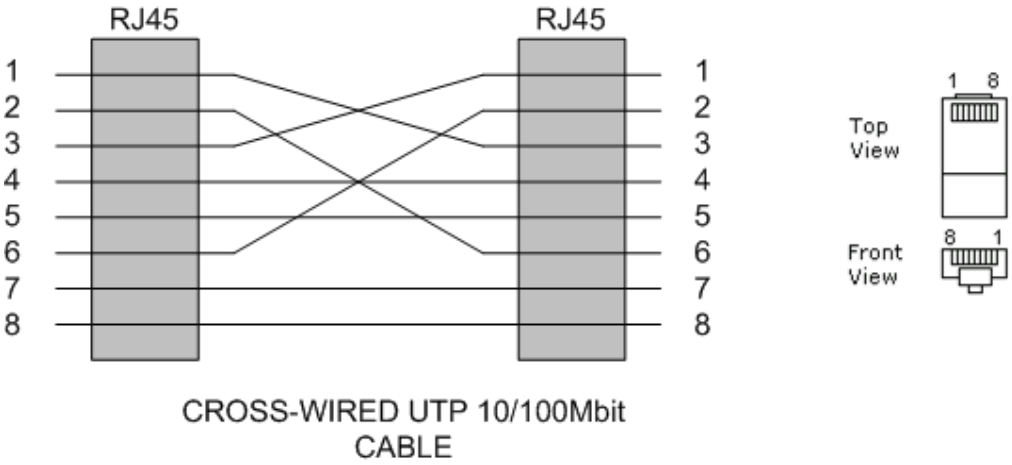
IB-Lite

IB-Lite is a plug-in module with Ethernet 10/100 Mbit interface in RJ45 connector. The module is internally connected to both COM1 and COM2 serial channels and provides an interface for connecting a PC with LiteEdit or InteliMonitor through ethernet/internet network, for sending active e-mails and for integration of the controller into a building management (Modbus/TCP protocol).



IB-LITE MODULE

Use Ethernet UTP cable with RJ45 connector for connection of the module into your ethernet network. The module can be also connected directly to a PC using cross-wired UTP cable.



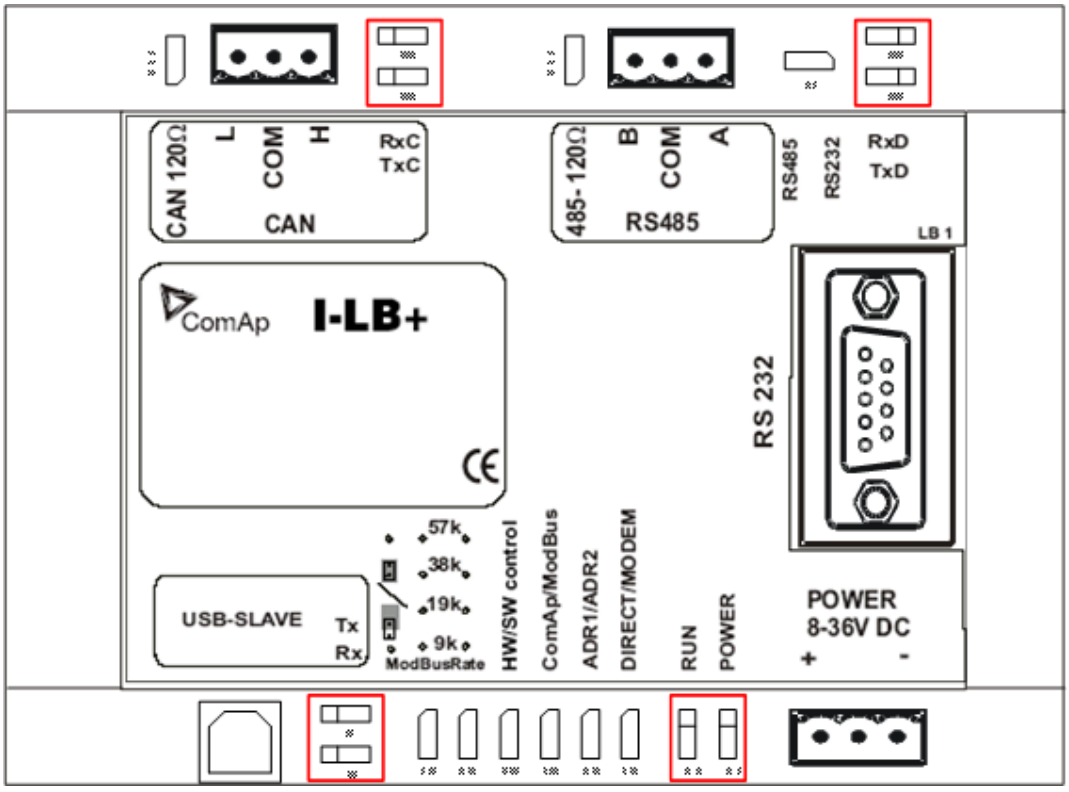
CROSS-WIRED UTP CABLE

NOTE:
The module requires some settings before initial usage. See chapter IB-Lite setup procedure

.

I-LB

Local bridge I-LB provides connection to up to 32 controllers via direct cable connection or analog/ISDN/GSM modem connection. It supports Modbus protocol and sending of active SMS using GSM modem.

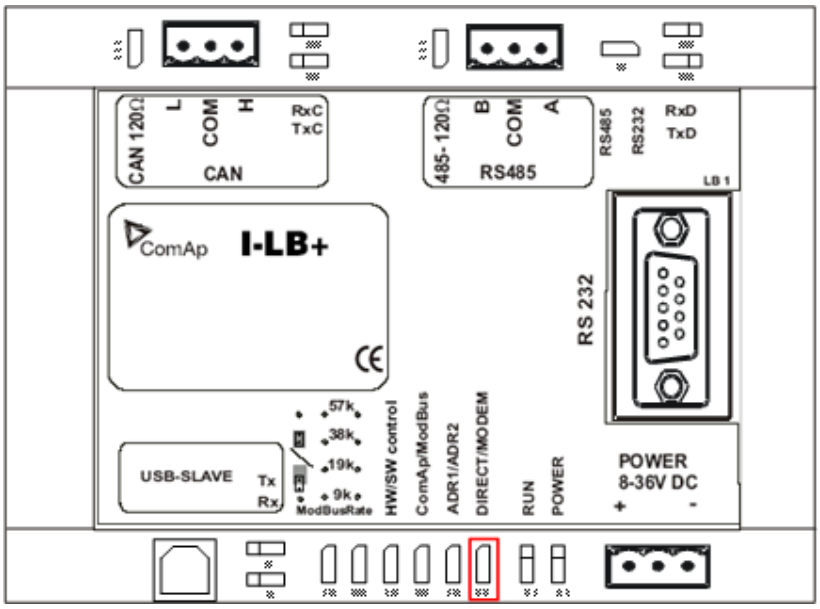


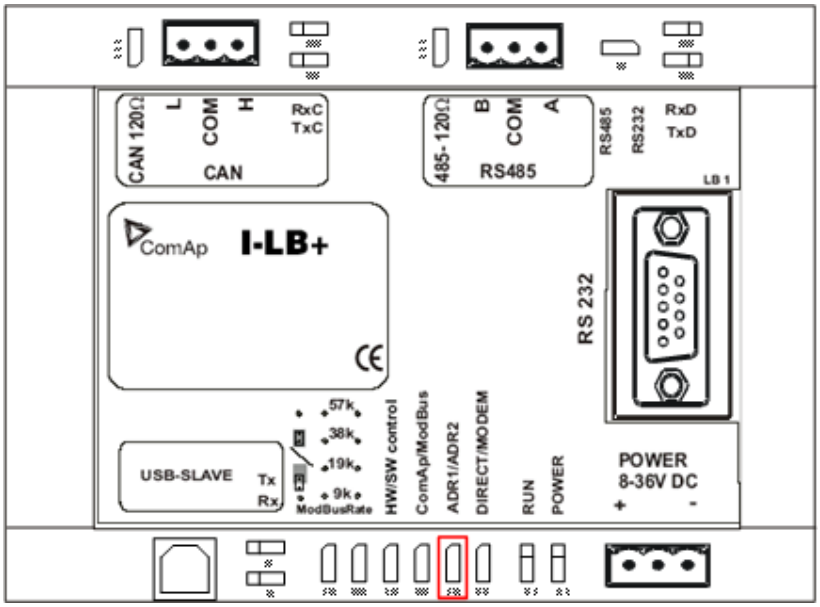
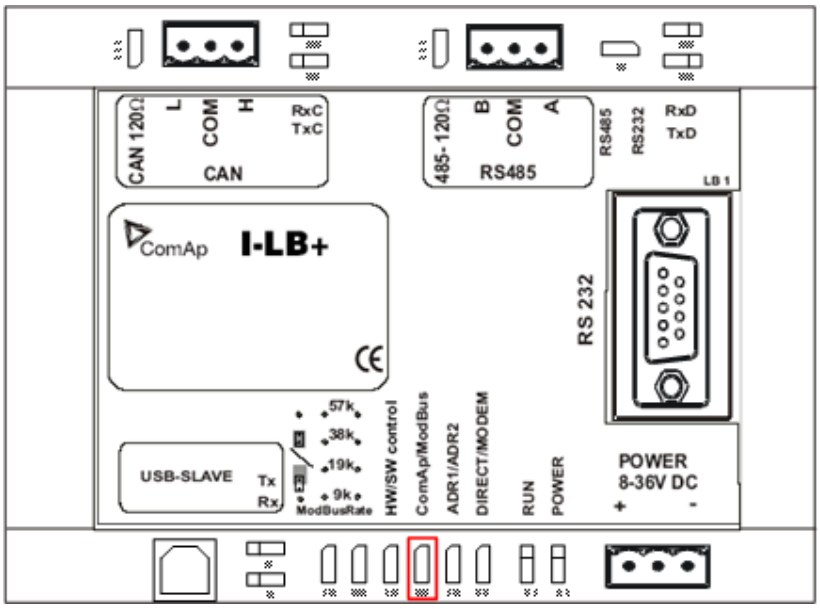
I-LB LEDS PLACEMENT

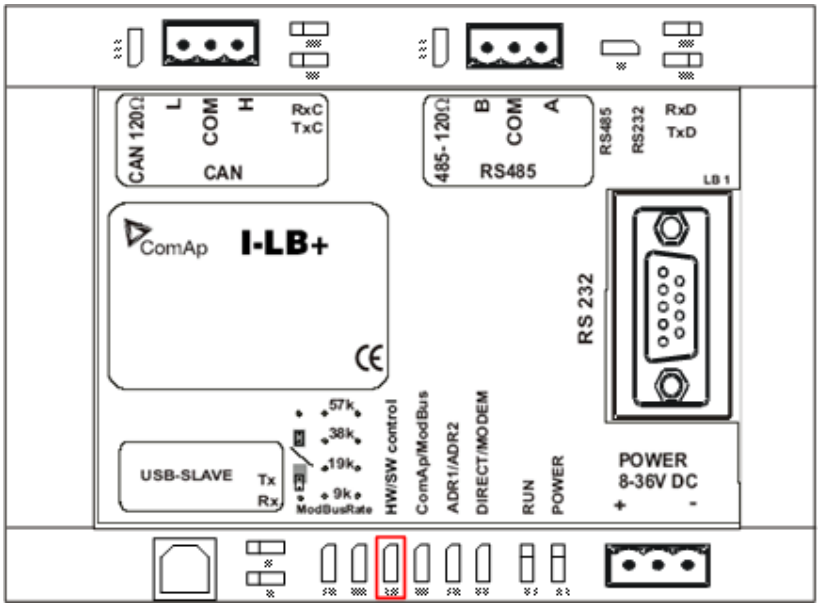
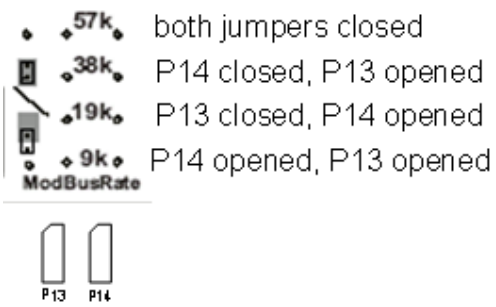
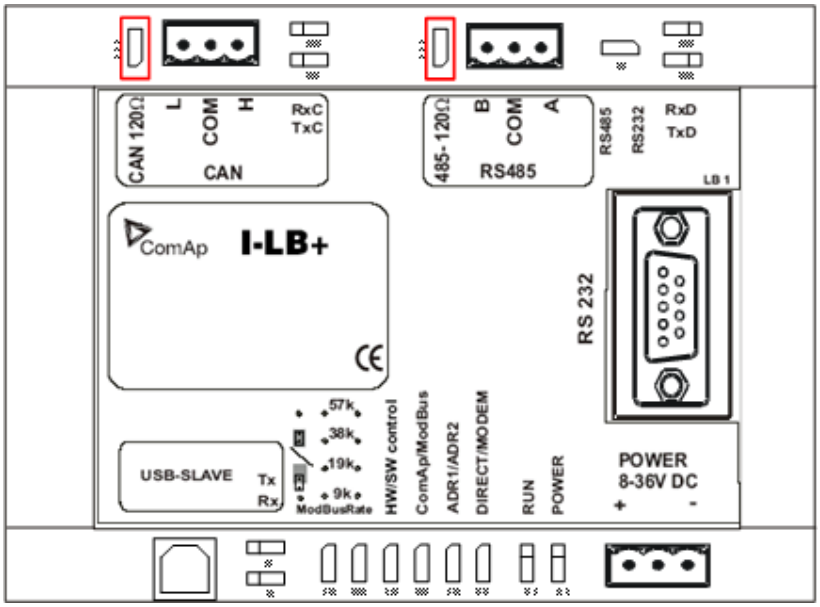
I-LB LEDs MEANING

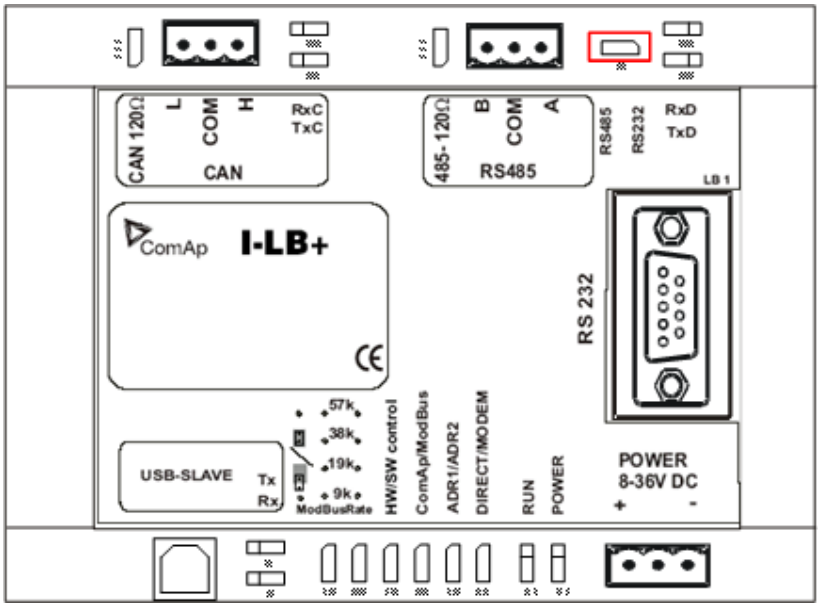
LED	MEANING
TxC, RxC	Indicates data transfer on the CAN line
TxD, RxD	Indicates data transfer on the RS232 line
RUN	Lights when at least one other unit is active on the CAN bus. Blinks when no unit is communicating on the CAN bus (during detection).
PWR	Lights all the time when power supply is switched on.

I-LB JUMPERS

JUMPER POSITION	JUMPER NAME	NOTE
	DIRECT/MODEM	Place jumper when I-LB is connected for modem communication

	ADR1/ADR2	Place jumper to choose address 2 (for local or modem connection)
	ComAp/Modbus	Place jumper for Modbus communication

	HW/SW control	Place jumper if the modem doesn't provide active DSR signal (IL-NT, IC-NT, IA-NT contollers can work with modems with HW control only so the jumper should be always opened)
	Modbus rate	Place jumpers to select Modbus communication speed
	CAN/RS485 120Ohm	Place jumper to connect 120Ohm terminating resistor for CAN bus/RS485

	RS485/232	Place jumper to select RS485 or RS232 connection

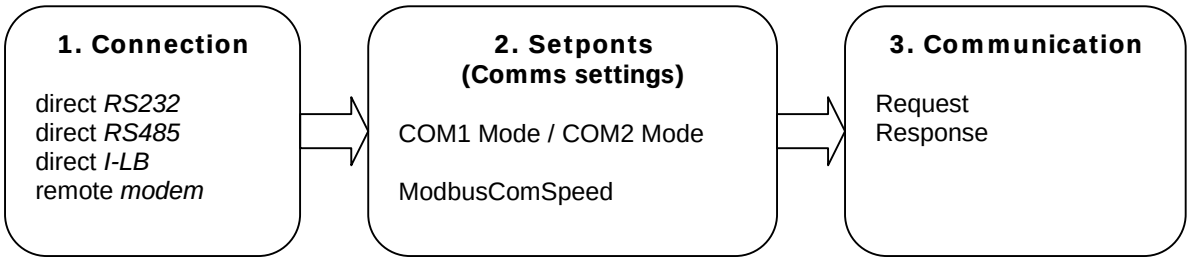
NOTE:
Other I-LB jumpers are designed exclusively for factory tests.

Modbus Connection

Modbus protocol was implemented into the controllers to enable the customer to design its own supervision software.

To learn more about Modbus interface see the training videos on
http://www.comap.cz/support/training/training-videos/

Modbus Step by Step



Important setpoints in the controller

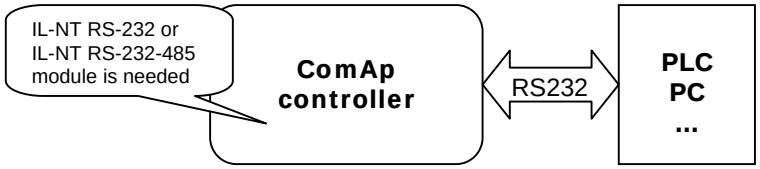
There are a lot of possibilities of Modbus connection to single or multiple controllers : direct via RS232, RS485 or via Modem.

Controller configuration:
Basic Settings:*COM1 Mode* = [DIRECT, MODEM, MODBUS, ECU LINK]
Basic Settings:*COM2 Mode* = [DIRECT, MODEM, MODBUS, ECU LINK]

Selection of Modbus communication speed:
Basic Settings:*ModbusComSpeed* = [9600 , 19200 , 38400 , 57600] bps

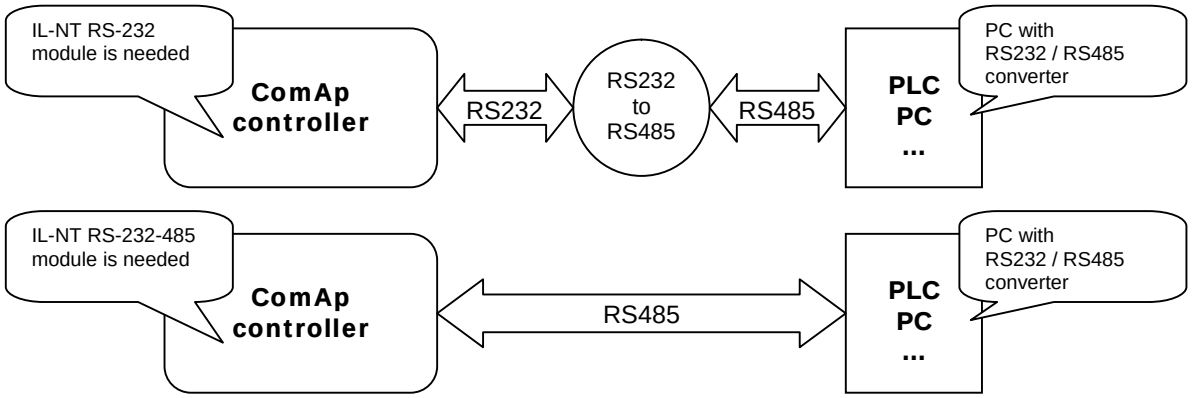
Modbus communication via RS232 – single controller

Controller configuration:
Basic Settings:*COM1 Mode, (COM2 Mode)* = **MODBUS**
Basic Settings:*ModbusComSpeed* = **select of Modbus communication speed**



Modbus communication via RS485

Controller configuration:
Basic Settings:*COM1 Mode, (COM2 Mode)* = **MODBUS**
Basic Settings:*ModbusComSpeed* = **select of Modbus communication speed**

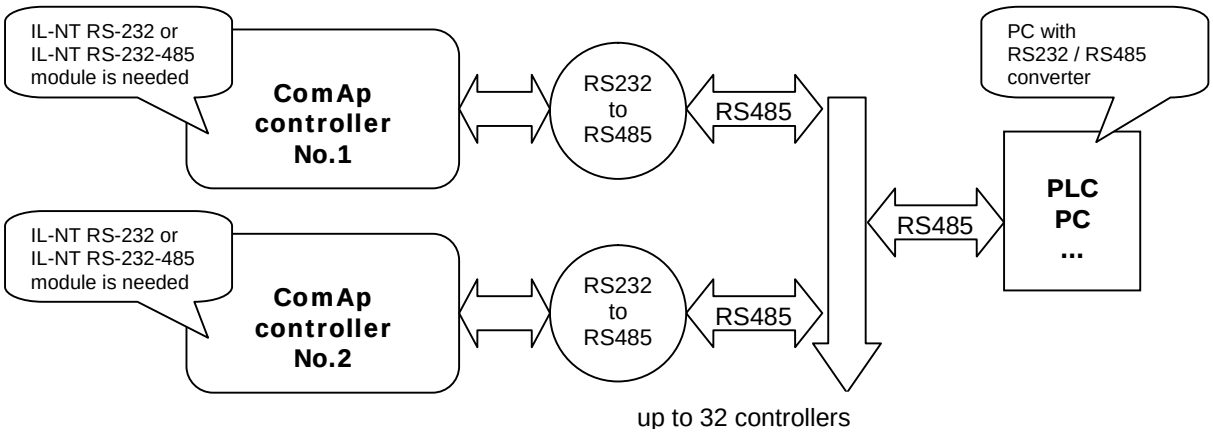


Hint:
The RS232/RS485 converter is not included in the IL-NT RS-232 accessory module for IL-NT and IC-NT controllers (external RS232/RS485 converter is needed).
The RS232/RS485 converter is included in the IL-NT RS-232-485 accessory module for IL-NT and IC-NT controllers (no external RS232/RS485 converter is needed).

Hint:
RS485 communication line has to be terminated by 120 ohms resistors at both ends – follow converter user manual. RS485 communication can be used for more controller monitoring and controlling via InteliMonitor.

Modbus communication via RS485 – multiple controllers

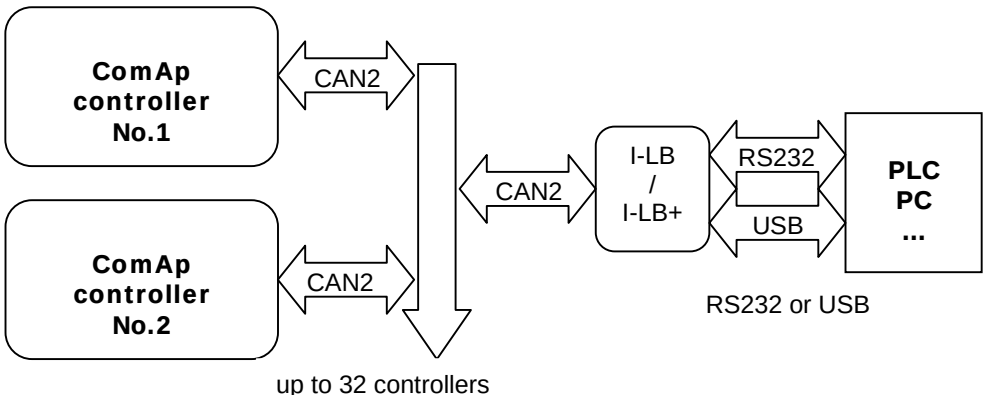
Controller configuration:
Basic Settings:*COM1 Mode, (COM2 Mode)* = **MODBUS**
Basic Settings:*ModbusComSpeed* = **select of Modbus communication speed**



Hint:
External RS232/RS485 converter is not needed, when IL-NT RS232-485 accessory module is used.

Modbus communication via I-LB

I-LB / I-LB+ configuration:
Jumpers P13, P14 = **select of Modbus communication speed**
Jumper P16 = **Modbus**
Jumper P17 = **Address 1 or Address 2**
Jumper P18 = **Direct**

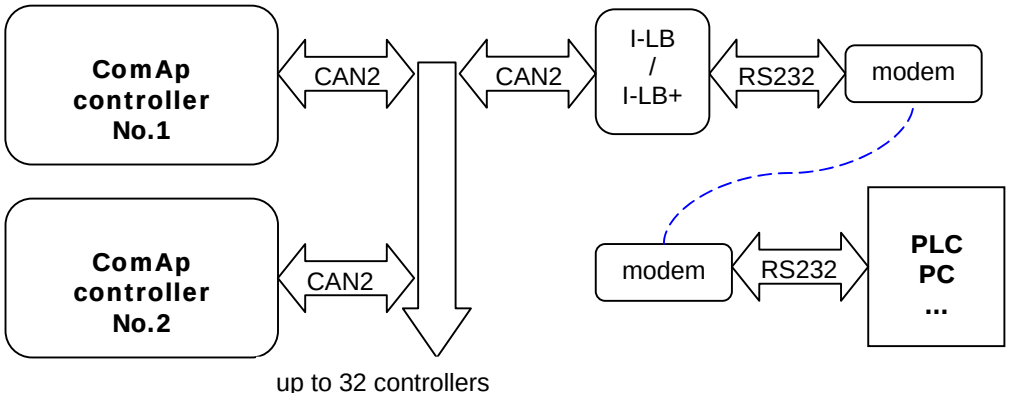


Hint:
To use I-LB Modbus communication connect Modbus jumper in I-LB unit (P16). Additionally, you can choose the communication speed using the speed selection jumpers (P13, P14). Their combination allows the speed settings of 9600 / 19200 / 38400 / 57600 bps.

Modbus communication via modem

I-LB / I-LB+ configuration:
Jumpers P13, P14 = **select of Modbus communication speed**
Jumper P16 = **Modbus**
Jumper P17 = **Address 1 or Address 2**
Jumper P18 = **Modem**

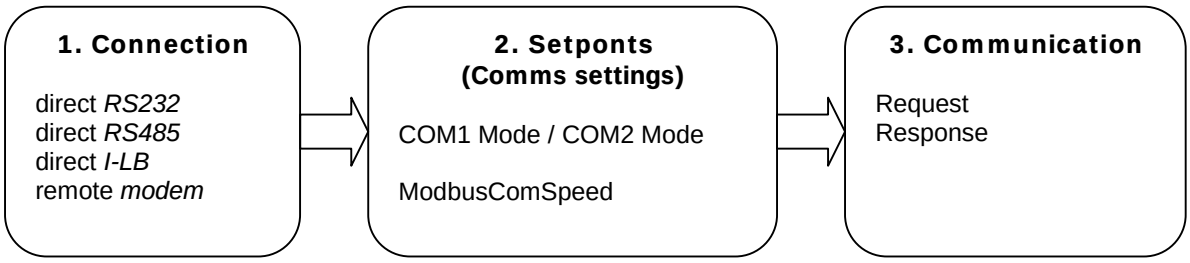
and correct modems settings – see modem data sheet.



Modbus Connection

Modbus protocol was implemented into the controllers to enable the customer to design its own supervision software.

Modbus Step by Step



Important setpoints in the controller

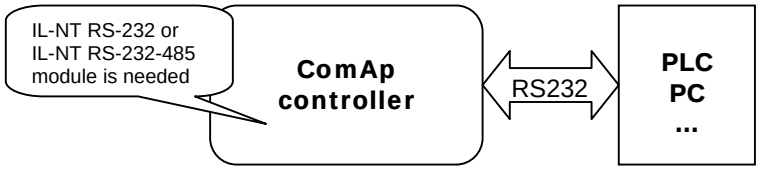
There are a lot of possibilities of Modbus connection to single or multiple controllers : direct via RS232, RS485 or via Modem.

Controller configuration:
Basic Settings:*COM1 Mode* = [DIRECT, MODEM, MODBUS, ECU LINK]
Basic Settings:*COM2 Mode* = [DIRECT, MODEM, MODBUS, ECU LINK]

Selection of Modbus communication speed:
Basic Settings:*ModbusComSpeed* = [9600 , 19200 , 38400 , 57600] bps

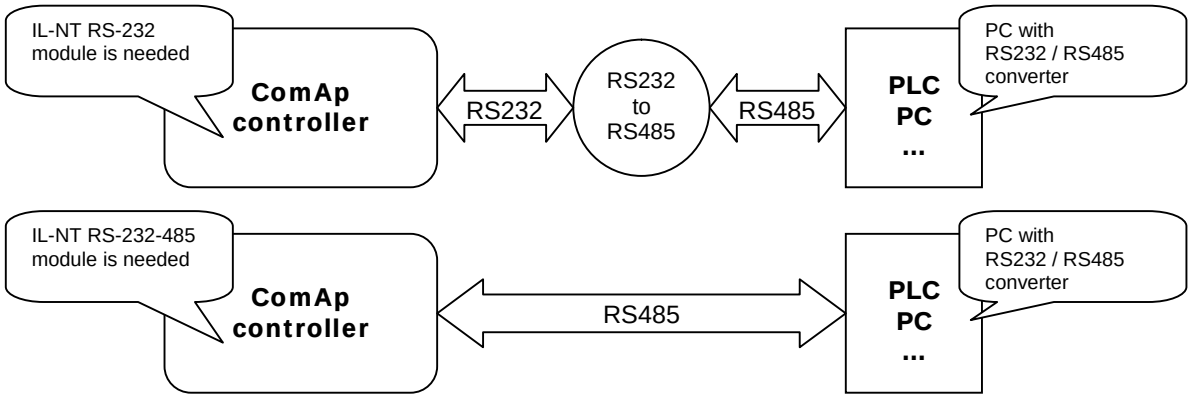
Modbus communication via RS232 – single controller

Controller configuration:
Basic Settings:*COM1 Mode, (COM2 Mode)* = **MODBUS**
Basic Settings:*ModbusComSpeed* = **select of Modbus communication speed**



Modbus communication via RS485

Controller configuration:
Basic Settings:*COM1 Mode, (COM2 Mode)* = **MODBUS**
Basic Settings:*ModbusComSpeed* = **select of Modbus communication speed**

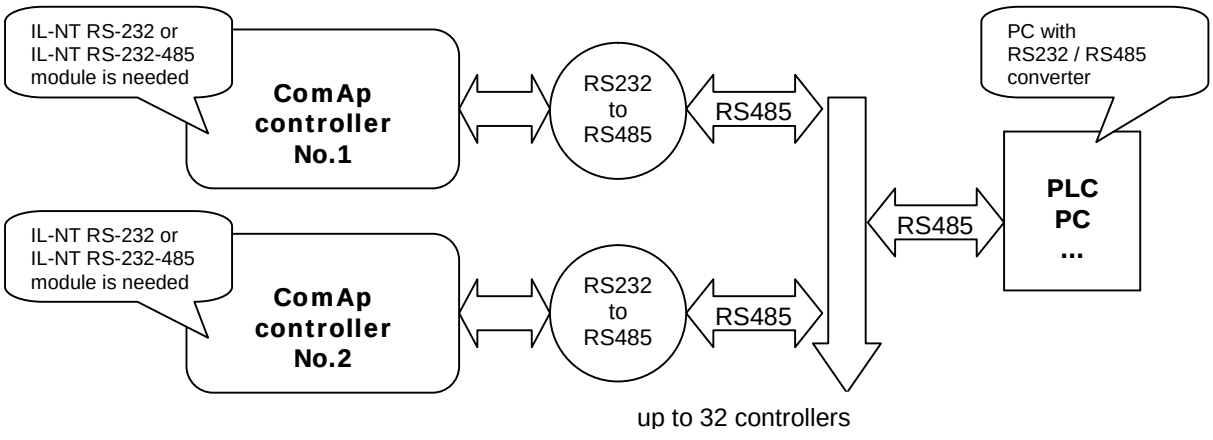


Hint:
The RS232/RS485 converter is not included in the IL-NT RS-232 accessory module for IL-NT and IC-NT controllers (external RS232/RS485 converter is needed).
The RS232/RS485 converter is included in the IL-NT RS-232-485 accessory module for IL-NT and IC-NT controllers (no external RS232/RS485 converter is needed).

Hint:
RS485 communication line has to be terminated by 120 ohms resistors at both ends – follow converter user manual. RS485 communication can be used for more controller monitoring and controlling via InteliMonitor.

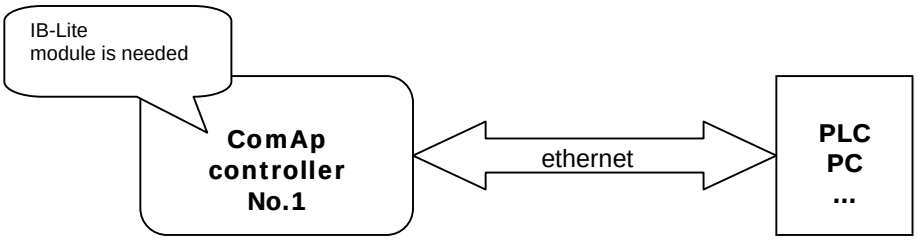
Modbus communication via RS485 – multiple controllers

Controller configuration:
Basic Settings:*COM1 Mode, (COM2 Mode)* = **MODBUS**
Basic Settings:*ModbusComSpeed* = **select of Modbus communication speed**



Hint:
External RS232/RS485 converter is not needed, when IL-NT RS232-485 accessory module is used.

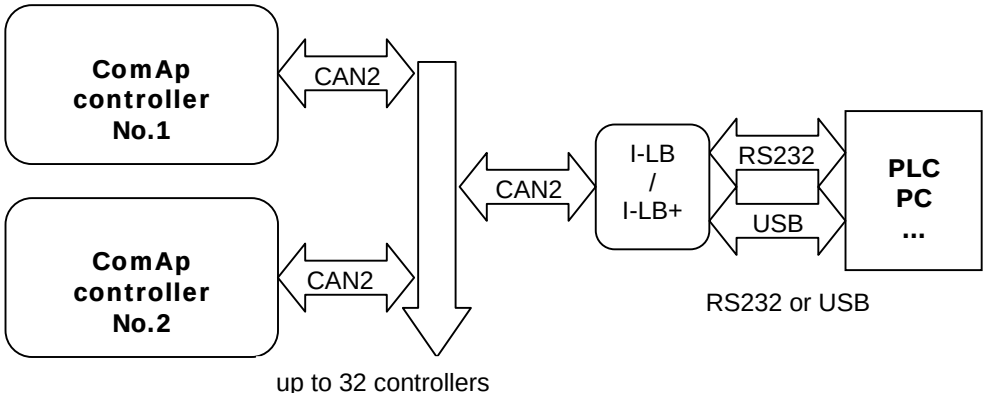
Modbus communication via IB-Lite



Hint:
The IB-Lite uses the Modbus/TCP protocol. IP address is the same as uses ComAp PC tools (WinEdit or InteliMonitor). The IP address is configurable by IB-Lite web-config (in default settings `http://192.168.1.254/sp_config.htm`). Service port is **502**, without possibility to configure it.
For more information about IB-Lite configuration see the *IB-Lite-x.y-Reference Guide*

Modbus communication via I-LB

I-LB / I-LB+ configuration:
Jumpers P13, P14 = **select of Modbus communication speed**
Jumper P16 = **Modbus**
Jumper P17 = **Address 1 or Address 2**
Jumper P18 = **Direct**

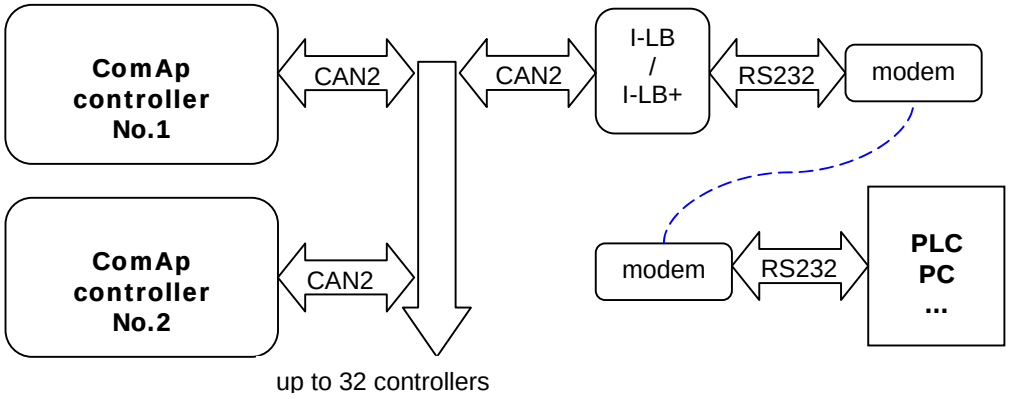


Hint:
To use I-LB Modbus communication connect Modbus jumper in I-LB unit (P16). Additionally, you can choose the communication speed using the speed selection jumpers (P13, P14). Their combination allows the speed settings of 9600 / 19200 / 38400 / 57600 bps.

Modbus communication via modem

I-LB / I-LB+ configuration:
Jumpers P13, P14 = **select of Modbus communication speed**
Jumper P16 = **Modbus**
Jumper P17 = **Address 1 or Address 2**
Jumper P18 = **Modem**

and correct modems settings – see modem data sheet.



Modbus communication

Hint:
In the firs time, you have to correctly configure the controller connection.

Data reading

The function Read Multiple Registers has to be used for data reading. The terminal sends a query and from the controller receives either the normal response containing the requested data or the exceptional response indicating a read error.

- It is possible to use function 3 for reading (*Read Multiple Registers*).
- It is not possible to read from the middle. The register number must correspond with the beginning of the data object. The only exception are the objects of „multipacket values“ (registers 46367 – 46491) and „data part of the history record“ (registers 46543 – 46667).
- All read registers must be implemented. If an unimplemented register appears among the read registers, the controller returns an error message.
- Even unnamed values can be included among read registers (See Cfg image - column Name = (N/A)). The read value must be treated as meaningless.
- The length of a block is 127 registers.

Data writing

All data can be written by the function Write Multiple Registers. Data up to 2 bytes can be written by the function Write Single Register, too. The terminal sends a query containing a written data and the controller either confirms it (normal response) or refuses it (exceptional response).

- For writing it is possible to use function 6 (*Write Single Register*) or function 16 (*Write Multiple Registers*).
- Using function 16 it is possible to write maximum 16 registers at once.
- Data cannot be written from the middle. Register number must correspond with the beginning of the data object. Written data must be complete to perform writing of all requested data objects.
- Writing to EEPROM is executed using a queue. The queue is common for writing from all terminals. The request for next writing is accepted in case that there is empty space in the queue. Otherwise the controller returns an error message and the terminal must repeat the request.
- All written registers must be implemented. If an unimplemented register appears among the read registers, the controller returns an error message.
- It is possible to include also unnamed registers in the written sequence (See Cfg image - column Name = (N/A)). The controller confirms this writing but writing of unnamed registers is not performed.

Request :

- controller address (1 - 32), you can set or check your controller's address in the controller setpoints. *Setpoints -> Comms settings -> Contr.address*
- Modbus function code, you can use the 3, 6, 16 Modbus function code,

  Function 3 (Read Multiple Registers)
  Function 6 (Write Single Register)
  Command 10
  Function 16 (Write Multiple Registers)

- Register address (40001 - 47168), it means Modbus address of controller communication object (setpoint, value, et al.). You can crate list of Modbus registers, if you can't find the register address in this list, see the table of dedicated communication objects.
- Number of registers (1 - 127). It means, how many registers you want read.
- CRC (no range)

After sent your request, you receive the response. The response has also five parts:

- Controller address (1 - 32), the same as the address in the request
- Modbus function code (3,6,16, …), mostly the same as in the request
- Length of data (1 - 127), here is specified the length of the received data
- Data (0 - FF), data are in the HEX form, length is defined above
- CRC (no range)

Examples of Modbus communication

In this chapter are some examples, how does communicate controller via Modbus.

Battery voltage – reading (read multiple registers)

Request: **01 03 00 32 00 01** 25 5C

01 = Controller address
– *see your controller settings*

03 = Modbus function code (Read Multiple Registers)

00 32 = Register address: Register number (Ubat => 40051 for *IL-NT*) (Ubat => 40058 for *IC-NT*)
– 40051 - 40001 = 50 DEC => 0032 HEX
– *see your Cfg Image or list of dedicated communication objects*

A part of Cfg Image (Modbus Register …)

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40051	8213	Battery Volts	V	Integer	2	1	-	-	Controller
40052	10124	(N/A)							
40053	10603	D+	V	Integer	2	1	-	-	Controller
40054	8227	Oil Pressure	Bar	Integer	2	1	-	-	Controller
40055	8228	Engine Temp	°C	Integer	2	0	-	-	Controller
40056	8229	Fuel Level	%	Integer	2	0	-	-	Controller
40057	8978	IOM AI1	U4	Integer	2	0	-	-	Extension

00 01 = Number of registers
– 40013, it is one register = 01 DEC => 0001 HEX
– *you have to calculate number of register which you want read*

5C 25 = CRC
–*CRC has to be written LSB then MSB ! See how to calculate CRC. or implementation in C language – page 57.*

Response: **01 03 02 00 DC** B9 DD

01 = Controller address
– *see your controller settings*

03 = Modbus function code (Read Multiple Registers)

02 = Length of read data in Bytes (in HEX)
– 02 HEX => 2 DEC
– *define the length of data*

00 DC = Value of battery voltage
– DC HEX => 220 DEC => Batt. voltage is represented with 1 decimal => 22,0 VDC
– *convert the data from hex to dec. Use the multiplication factor (In this case 0.1) !*

DD B9 = CRC
– *check with your CRC, because of data validity*

A part of Cfg Image (Modbus Register …)

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40051	8213	Battery Volts	V	Integer	2	1	-	-	Controller
40052	10124	(N/A)							
40053	10603	D+	V	Integer	2	1	-	-	Controller
40054	8227	Oil Pressure	Bar	Integer	2	1	-	-	Controller
40055	8228	Engine Temp	°C	Integer	2	0	-	-	Controller
40056	8229	Fuel Level	%	Integer	2	0	-	-	Controller
40057	8978	IOM AI1	U4	Integer	2	0	-	-	Extension

Values (Oil press, Engine temp, Fuel level) – reading

Request: **01 03 00 35 00 03 15 C5**

01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 35 = Register address: Register number (40054) – 40001 = 53 DEC => 35 HEX *IL-NT*
Register address: Register number (40061) – 40001 = 60 DEC => 3C HEX *IC-NT*
00 03 = Number of registers (40054 – Oil press, 40055 – Engine temp, 40056 – Fuel level) = 3 DEC = > 03 HEX
C5 15 = CRC (write LSB MSB !)

A part of Cfg Image (Modbus Register …)

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40051	8213	Battery Volts	V	Integer	2	1	-	-	Controller
40052	10124	(N/A)							
40053	10603	D+	V	Integer	2	1	-	-	Controller
40054	8227	Oil Pressure	Bar	Integer	2	1	-	-	Controller
40055	8228	Engine Temp	°C	Integer	2	0	-	-	Controller
40056	8229	Fuel Level	%	Integer	2	0	-	-	Controller
40057	8978	IOM AI1	U4	Integer	2	0	-	-	Extension

Response: **01 03 06 00 27 00 2E 00 2B 35 64**

01 = Controller address
03 = Modbus function code (Read Multiple Registers)
06 = Length of read data in Bytes (in HEX)
00 27 = 27 HEX => 39 DEC => 3,9 Bar (Oil pressure is represented with 1 decimal in Bars)
00 2E = 2E HEX => 46 DEC => 46°C (Engine temperature is represented with 0 decimals in °C)
00 2B = 2B HEX => 43 DEC => 43% (Fuel level is represented with 0 decimals in %)
64 35 = CRC

Binary input - reading

Request: **01 03 00 3D 00 01 15 C6**

01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 3D = Register address: Register number (40062) – 40001 = 61 DEC => 3D HEX *IL-NT*
Register address: Register number (40069) – 40001 = 68 DEC => 44 HEX *IC-NT*
00 01 = Number of registers (40001) = 01 DEC => 01 HEX
C6 15 = CRC (write LSB MSB !)

Response: **01 03 02 18 01 73 84**

01 = Controller address
03 = Modbus function code (Read Multiple Registers)
02 = Length of read data in Bytes (in HEX)
18 01 = Object data value (Binary input = 0001100000000001 i.e. B1, B12 and BI3 are set)*
84 73 = CRC

* Table of binary inputs (BI)

BI16	BI15	BI14	BI13	BI12	BI11	BI10	BI9	BI8	BI7	BI6	BI5	BI4	BI3	BI2	BI1
1				**8**				**0**				**1**			
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1

Password decode - reading

Request: **01 03 00 71 00 02 94 10**

01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 71 = Register address: Register number (40114) – 40001 = 113 DEC => 71 HEX *IL-NT*
Register address: Register number (40143) – 40001 = 142 DEC => 8E HEX *IC-NT*
00 02 = Number of registers (40112 and 40113) = 02 DEC => 02 HEX
10 94 = CRC (write LSB MSB !)

Response: **01 03 04 68 73 90 00 7B 88**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
04 = Length of read data in Bytes (in HEX)
68 73 90 00 = 68739000 HEX => 1752403968 DEC = > password decode is **1752403968**
88 7B = CRC

Gen-set name - reading

Request: **01 03 0B C5 00 08 56 15**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
0B C5 = Register address: Register number (43014) – 40001 = 3013 DEC => BC5 HEX *IL-NT*
Register address: Register number (43017) – 40001 = 3016 DEC => BC8 HEX *IC-NT*
00 08 = Number of registers (43001 - 43008) = 08 DEC => 08 HEX
15 56 = CRC (write LSB MSB !)

Response: **01 03 10 49 4C 2D 4E 54 2D 41 4D 46 32 35 00 14 00 00 00 96 04**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
10 = Length of read data in Bytes (in HEX)
49 4C = Object data value (IL)
2D 4E = Object data value (-N)
54 2D = Object data value (T-)
41 4D = Object data value (AM)
46 32 = Object data value (F2)
35 00 = Object data value (5 _)
14 00 = Object data value (_ _)
00 00 = Object data value (_ _) =.> gen-set name is **IL-NT-AMF25**
04 96 = CRC

Controller Mode - reading

Request: **01 03 00 46 00 01 65 DF**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 46 = Register address: Register number (40071) – 40001 = 070 DEC => 46 HEX *IL-NT*
Register address: Register number (40080) – 40001 = 079 DEC => 4F HEX *IC-NT*
00 01 = Number of registers (40163)
DF 65 = CRC (write LSB MSB !)

Response: **01 03 02 00 00 B8 44 84**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
02 = Length of read data in Bytes (in HEX)
00 00 = Object data value – see the List#10 in the Cfg Image => (**OFF**)
84 44 = CRC

A part of Cfg Image (Modbus Register …)

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
43131	8315	ControllerMode		List#10	1	-	60	63	Basic Settings
43132	8482	Yel Alarm Msg		List#11	1	-	188	189	SMS/E-Mail
43133	8484	Red Alarm Msg		List#11	1	-	188	189	SMS/E-Mail

List#10

Value	Name
0	OFF
1	MAN
2	AUT
3	TEST

Gear teeth – writing

Request: **01 06 0B D7 00 7D FB F7**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D7 = Register address: Register number (43032) – 40001 = 3031 DEC => BD7 HEX *IL-NT*
Register address: Register number (43035) – 40001 = 3034 DEC => BDA HEX *IC-NT*
00 7D = Gear teeth > 125 DEC => 7D HEX
F7 FB = CRC (write LSB MSB !)

Response: **01 06 0B D7 00 7D FB F7**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D7 = Register addres
00 7D = Set the setpoint gear teeth to > 7D HEC => 125 DEC = **125**
F7 FB = CRC

Nominal RPM – writing

Request: **01 06 0B D5 01 F4 9A 01**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D5 = Register address: Register number (43030) – 40001 = 3029 DEC => BD5 HEX *IL-NT*
Register address: Register number (43033) – 40001 = 3032 DEC => BD8 HEX *IC-NT*
01 F4 = Nominal power > 500 DEC => 1F4 HEC
01 9A = CRC (write LSB MSB !)

Response: **01 06 0B D5 01 F4 9A 01**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D5 = Register addres
01 F4 = Set the setpoint nominal power to > 1F4 HEC => 500 DEC = **500**
01 9A = CRC

Mode – writing

Request: **01 06 0C 3A 00 00 AA 97**
01 = Controller address
06 = Modbus function code (Write Single Register)
0C 3A = Register address: Register number (43131) – 40001 = 3130 DEC => C3A HEX *IL-NT*
Register address: Register number (43157) – 40001 = 3156 DEC => C54 HEX *IC-NT*
00 00 = Set the controller mode to > OFF **=>** 00 **–** see the List#10 in the Cfg Image
97 AA = CRC (write LSB MSB !)

A part of Cfg Image (Modbus Register …)

List#10

Value	Name
0	OFF
1	MAN
2	AUT
3	TEST

Response: **01 06 0C 3A 00 00 AA 97**

01	= Controller address
06	= Modbus function code (Write Single Register)
0C 3A	= Register addres
00 00	= Object data value > **OFF**
97 AA	= CRC (write LSB MSB !)

History – reading

See more information about History reading on page *57.*

Hint:
If you use the ModScan32 PC tool, use the script for this issue.

1 of 3 - first the index of history record must be entered:
Request: **01 06 18 D4 00 00 CF 52**
01 = Controller address
06 = Modbus function code (Write Single Register)
18 D4 = Register address of the history index (46357) – 40001 = 6356 DEC => 18D4 HEX
00 00 = First history record (index = 0)
52 CF = CRC (write LSB MSB !)

Response: **01 06 18 D4 00 00 CF 52**

A part of dedicated communication objects table

Registers (*)	Register addresses (*)	Number of registers	Access	Data type	Meaning
46354	6353	1	read	Unsigned8	Number of records in the alarm list
46356	6355	1			Reserved (register not implemented)
46357	6356	1	read/write	Integer16	Index of requested history record (# 5)
46493 – 46541	6492 – 6540	50	read	String	Header of the particular history record (# 1)
46543 – 46667	6542 – 6666	125	read	Domain	Data part of the particular history record (# 2)
46668	6667	1			Reserved (register not implemented)
46669 – 46693	6668 – 6692	25	read	String	1. record in alarm list (# 1)
46694 – 46718	6693 – 6717	25	read	String	2. record in alarm list (# 1)
46719 – 46743	6718 – 6742	25	read	String	3. record in alarm list (# 1)

2 of 3 - reading of history record header:
Request: **01 03 19 5C 00 32 03 51**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
19 5C = Register address of history record header (46493) *2 – 40001 = 6492 DEC => 195C HEX
00 32 = Number of registers > 46493 – 46541 => 50 DEC => 32 HEX
51 03 = CRC (write LSB MSB !)

Response: **01 03 64 4D 43 42 20 63 6C 6F 73 65 64 20 20 20 20 20 20 20 20 20 20 30 33 2F 30 39 2F 32 30 30 38 20 20 31 35 3A 34 34 3A 35 37 2E 39 00 … 00 00 0E E0**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
64 = Length of read data in Bytes (in HEX)
4D .. 39 … = Object data value > 1.record in alarmlist is **MCB closed 03.09.2008 15:44:57.9**
E0 0E = CRC

3 of 3 - reading of the data part of history record:
Request: **01 03 19 8E 00 7D E2 9C**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
19 8E = Register address of history record header (46543) – 40001 = 6542 DEC => 198E HEX
00 7D = Number of registers > 46542 – 46667 => 125 DEC => 7D HEX
9C E2 = CRC (write LSB MSB !)

Response: **01 03 FA 00 00 00 00 00 00 00 20 00 … 00 00 F4 01 FD 00 FD 00 FD 00 00 00 00 00 00 00 00 64 20 00 00 00 00 64 00 D8 00 55 01 00 00 A1 00 7A 00 64 00 0A 00 18 00 00 00 00 … 00 00 20 3B**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
FA = Length of read data in Bytes (in HEX)
00 .. 00 = Object data value > for reading this data see table 7 *History Record* in **Communication object description** (in PC tool -> File –> Generate Cfg Image –> Generate Cfg Image (Comm. Objects …))
3B 20 = CRC

AlarmList – reading

See more information about AlarmList reading on page *57.*

Request: **01 03 1A 0C 00 19 43 B1**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
1A 0C = Register address: Register number (46669) – 40001 = 6668 DEC => 1A0C HEX
00 19 = Number of registers > 46669 – 46693 => 25 DEC => 19 HEX
B1 43 = CRC (write LSB MSB !)

A part of dedicated communication objects table

Registers (*)	Register addresses (*)	Number of registers	Access	Data type	Meaning
46354	6353	1	read	Unsigned8	Number of records in the alarm list
46357	6356	1	read/write	Integer16	Index of requested history record (# 5)
46364	6363	1	write	Unsigned16	Entering of password for writing (# 4)
46542	6541	1			Reserved (register not implemented)
46668	6667	1			Reserved (register not implemented)
46669 – 46693	6668 – 6692	25	read	String	1. record in alarm list (# 1)
46694 – 46718	6693 – 6717	25	read	String	2. record in alarm list (# 1)
46719 – 46743	6718 – 6742	25	read	String	3. record in alarm list (# 1)

Response: **01 03 32 2A 53 64 20 53 44 20 31 32 … 00 00 18 F5**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
32 = Length of read data in Bytes (in HEX)
2A 53 = Object data value (* S)
64 20 = Object data value (d _)
53 44 = Object data value (S D)
20 31 = Object data value (_ 1)
32 00 … = Object data value (2) = > 1.record in alarmlist is ***Sd SD 12** (inactive, not accepted)
F5 18 = CRC

Response: **01 03 32 21 2A 53 64 20 53 44 20 31 32 00 00 … 00 00 89 38**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
32 = Length of read data in Bytes (in HEX)
21 2A = Object data value (! *)
53 64 = Object data value (S d)
20 53 = Object data value (_ S)
44 20 = Object data value (D _)
31 32 … = Object data value (1 2) = > 1.record in alarmlist is **!*Sd SD 12** (active, not accepted)
38 89 = CRC

Change the communication language (only String type data)

Write to the communication object 6350 the index of language to be used.

A part of dedicated communication objects table

Registers (*)	Register addresses (*)	Number of registers	Access	Data type	Meaning
46349 – 46350	6348 – 6349	2	read/write	Date	Actual date
46351	6350	1	read/write	Unsigned8	Language index selected for displaying of texts specified by data type String (# 7)
46352 – 4653	6351 – 6352	2	read	Domain	Code of the last communication fault See Error list
46354	6353	1	read	Unsigned8	Number of records in the alarm list
46349 – 46350	6348 – 6349	2	read/write	Date	Actual date
46351	6350	1	read/write	Unsigned8	Language index selected for displaying of texts specified by data type String (# 7)

Request: **01 06 18 CE 00 01 2F 55**
01 = Controller address
06 = Modbus function code (Write Single Register)
18 CE = Register address: Register number (46351) – 40001 = 6350 DEC => 18CE HEX
00 01 = Set the language index to > 1
55 2F = CRC (write LSB MSB !)

Response: **01 06 18 CE 00 00 EE 95**
01 = Controller address
06 = Modbus function code (Read Multiple Registers)
18 CE = Register address
00 01 = Language index set to > **1**
55 2F = CRC

Reset / Confirm Alarm

Request: **01 10 18 D6 00 03 06 08 F7 00 00 00 01 49 CB**
01 = controller address
10 = Modbus command
18 D6 = Register address: Object for engine commands (46359) – 40001 = 6358 DEC
=> 18D6 HEX
00 03 = number of Modbus registers
06 = data length in bytes (08F70000+0001)
08F70000 = argument for **Fault reset** (page 63)
0001 = command number (page 63)
CB 49 = CRC (write LSB MSB !)

A part of dedicated communication objects table

Registers (*)	Register addresses (*)	Number of registers	Access	Data type	Meaning
46359 – 46360	6358 – 6359	2	read/write	Unsigned32	For writing:command argument For reading: command release value (# 3)
46361	6360	1	write	Unsigned16	Command (# 3)

A part of list of commands

Command	Meaning	Argument (*)	Return value (*)	
1	Engine start	01FE0000	000001FF	OK
			2	Argument has not been written
	Engine stop	02FD0000	000002FE	OK
			2	Argument has not been written
	Horn reset	04FB0000	000004FC	OK
	Fault reset	08F70000	000008F8	OK
	ECU Fault reset	10EF0000	000010F0	OK
		other	1	Wrong argument

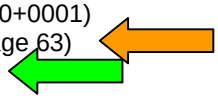
Response: **01,10,18,D6,00,03,67,50**
01 = Controller address
10 = Modbus command
18 D6 = Register addres
00 03 = Release value, number of written Modbus registers
50 67 = CRC

Start the engine – in one step

To start the engine it is necessary to enter an appropriate user and his password first to enable commands, if these are protected by level 1-7.

Request: **01 10 18 D6 00 03 06 01 FE 00 00 00 01 95 53**

01	= controller address
10	= Modbus command (Write Multiple Register)
18D6	= 6538 object for engine commands (46359) – 40001 = 6358 DEC => 18D6 HEX
0003	= number of Modbus registers
06	= data length in bytes (01FE0000+0001)
01FE0000	= argument for **Engine start** (page 63)
0001	= command number (page 63)
53 95	= CRC (write LSB MSB !)



Response: **01 10 18 D6 00 03 67 50**

01	= Controller address
10	= Modbus function code (Read Multiple Registers)
18 D6	= Register addres
00 03	= Release value, number of written Modbus registers
50 67	= CRC

Start the engine – in two steps

Request 1: **01 10 18 D6 00 02 04 01 FE 00 00 B4 D5**

01	= Controller address
10	= Modbus function code (Write Multiple Register)
18 D6	= Register address for command argument (46359) – 40001 = 6358 DEC => 18D6 HEX
00 02	= Number of registers
04	= Number of bytes that will be written (01FE0000)
01 FE 00 00	= command number (page 63)
D5 B4	= CRC (write LSB MSB !)

Request 2: **01 06 18 D8 00 01 CE 91**

01	= Controller address
06	= Modbus function code (Write Single Register)
18 D8	= Register address for command (46361) – 40001 = 6360 DEC => 18D8 HEX
00 01	= command number (page 63)
91 CE	= CRC

Modbus Protocol Description

- Direct connection:
  - RS232 only with IL-NT RS-232 or IL-NT RS-232-485, RS485 only with IL-NT RS-232-485, (I-LB)
  - 8 data bits
  - 1 stop bit
  - no parity
- Modem connection
  - 8 data bits
  - 1 stop bit
  - no parity
- Communication speed:
  - 9600 / 19200 / 38400 / 57600 bps
- Transfer mode RTU
- Function codes
  - 3 (Read Multiple Registers)
  - 6 (Write Single Register)
  - 10 (Command)
  - 16 (Write Multiple Registers)
- The response to an incoming message depends on the communication speed. The delay is not shorter than the time needed to send/receive 3 and ½ characters.

The complete description of Modbus communication protocol can be found in
http://modbus.org/docs/PI_MBUS_300.pdf
and
http://www.rtaautomation.com/modbustcp/files/Open_ModbusTCP_Standard.pdf.

Read Multiple Registers

`Query`

Byte	Meaning	Note
0	Controller address	1 to 32
1	3	Modbus function code
2 3	Communication object number - upper byte (MSB) - lower byte (LSB)	See List of communication objects
4 5	Communication object length expressed by the number of registers - upper byte (MSB) - lower byte (LSB)	Greater than 0
6 7	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

`Standard response`

Byte	Meaning	Note
0	Controller address	Same as in the query
1	3	Same as in the query
	Length of read data in bytes (L)	Number of registers * 2
3 4	Data of the 1st register - upper byte (MSB) - lower byte (LSB)	
5 6	Data of the 2nd register - upper byte (MSB) - lower byte (LSB)	
…		
L + 1 L + 2	Data of the last register - upper byte (MSB) - lower byte (LSB)	
L + 3 L + 4	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

`Exceptional response`

Byte	Meaning	Note
0	Controller address	Same as in the query
1	131	Modbus fun.number + 128
2	2	See Error list
3 4	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Write Single Register

Query

Byte	Meaning	Note
0	Controller address	1 to 32
1	6	Modbus function code
2 3	Communication object number - upper byte (MSB) - lower byte (LSB)	See List of communication objects
4 5	Data - upper byte (MSB) - lower byte (LSB)	
6 7	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Standard response

Byte	Meaning	Note
0	Controller address	Same as in the query
1	6	Same as in the query
2 3	Communication object number - upper byte (MSB) - lower byte (LSB)	Same as in the query
4 5	Data - upper byte (MSB) - lower byte (LSB)	Same as in the query
6 7	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Exceptional response

Byte	Meaning	Note
0	Controller address	Same as in the query
1	134	Modbus fun.number + 128
2	2	See Error list
3 4	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Write Multiple Registers

Query

Byte	Meaning	Note
0	Controller address	1 to 32
1	16	Modbus function code
2 3	Communication object number - upper byte (MSB) - lower byte (LSB)	See List of communication objects
4 5	Communication object length expressed by the number of registers - upper byte (MSB) - lower byte (LSB)	Greater than 0
6	Length of written data in bytes (L)	Number of registers * 2
7 8	Data of the 1st register - upper byte (MSB) - lower byte (LSB)	
9 10	Data of the 2nd register - upper byte (MSB) - lower byte (LSB)	
...		
L + 5 L + 6	Data of the last register - upper byte (MSB) - lower byte (LSB)	
L + 7 L + 8	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Standard response

Byte	Meaning	Note
0	Controller address	Same as in the query
1	16	Same as in the query
2 3	Communication object number - upper byte (MSB) - lower byte (LSB)	Same as in the query
4 5	Communication object length expressed by the number of registers - upper byte (MSB) - lower byte (LSB)	Same as in the query
6 7	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Exceptional response

Byte	Meaning	Note
0	Controller address	Same as in the query
1	144	Function code + 128
2	2	See Error list
3 4	Check field CRC - lower byte (LSB) - upper byte (MSB)	See Check field calculation

Alarm list reading

It is not possible to read alarm list simultaneously from more terminals. If the terminal starts reading, the reading is locked for other terminals. It is unlocked 5 seconds after last reading of alarm list. The locked terminal indicates to another terminal an error message.
The whole alarm list is stored in the cache memory at the moment of locking and the following reading of records is performed from this memory. Locking is done only while reading the first record. So the successive reading from the first to the last record is supposed.

History reading

It is not possible to read history from more terminals simultaneously. Reading must be started by writing of an index of requested history record. If the index is not written it is not possible to read neither history header nor data part of the record. In this case the controller returns an error message. If the terminal writes the index of requested record, history reading is locked for other terminals (i.e. reading and writing of an index of requested record, reading of header and data part of the record). It is unlocked 5 seconds after the last history reading. Locked history is indicated to other terminals by an error message.

Requested history record is stored at the moment of locking in the cache memory and following reading is performed from this memory.

Check field calculation

The check field allows the receiver to check the validity of the message. The check field value is the Cyclical Redundancy Check (CRC) based on the polynomial $x^{16}+x^{15}+x^{2}+1$. CRC is counted from all message bytes preceding the check field. The algorithm of CRC calculation is introduced below on an example of a C language function.

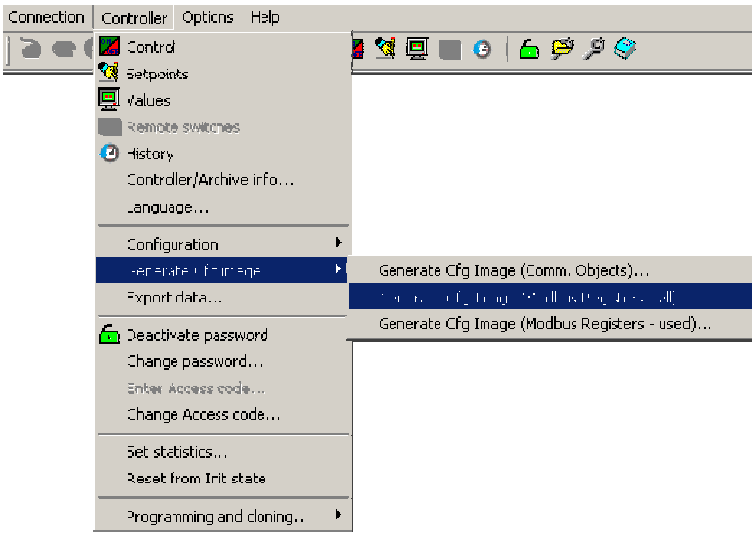
```
unsigned short count_CRC(unsigned char *addr, int num)
{
  unsigned short CRC = 0xFFFF;
  int i;

  while (num--)
  {
    CRC ^= *addr++;
    for (i = 0; i < 8; i++)
    {
      if (CRC & 1)
      {
        CRC >>= 1;
        CRC ^= 0xA001;
      }
      else
      {
        CRC >>= 1;
      }
    }
  }
  return CRC;
}
```

"0103000C0001" (hex)	
1 byte checksum	17
CRC-16	0x1244
CRC-16 (Modbus)	0x0944
CRC-16 (Sick)	0x2110
CRC-CCITT (XModem)	0xCE32
CRC-CCITT (0xFFFF)	0xC022
CRC-CCITT (0x1D0F)	0xFF0C
CRC-CCITT (Kermit)	0xCDAD
CRC-DNP	0x6CB2
CRC-32	0x4323C124

Online CRC calculator: http://www.lammertbies.nl/comm/info/crc-calculation.html
Controllers use the CRC-16 (Modbus). Data in examples in this manual are in HEX format.

Cfg Image Modbus registers and Communication object list

Communication objects can be spitted into two groups:
1. Communication objects dependent on the application type.
2. Communication objects independent on the application type.



Use LiteEdit menu command Controller
--> Generate Cfg Image
-> Generate Cfg Image (Comm. Objects …)

and

-> Generate Cfg Image (Modbus Registers all/used).

A part of Cfg Image (Modbus Register …) file for IL-NT controller

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40001	8192	Gen V L1-N	V	Unsigned	2	0	-	-	Generator
40002	8193	Gen V L2-N	V	Unsigned	2	0	-	-	Generator
40003	8194	Gen V L3-N	V	Unsigned	2	0	-	-	Generator
40051	8213	Battery Volts	V	Integer	2	1	-	-	Controller
40052	10124	(N/A)							
40053	10603	D+	V	Integer	2	1	-	-	Controller
40054	8227	Oil Pressure	Bar	Integer	2	1	-	-	Controller
40055	8228	Engine Temp	°C	Integer	2	0	-	-	Controller
40056	8229	Fuel Level	%	Integer	2	0	-	-	Controller
40057	8978	IOM AI1	U4	Integer	2	0	-	-	Extension
40058	8759	IOM AI2	U5	Integer	2	0	-	-	Extension
40059	8760	IOM AI3	U6	Integer	2	0	-	-	Extension
40060	8761	IOM AI4	U7	Integer	2	0	-	-	Extension
40061	8237	(N/A)							
40062	8235	Bin Inputs		Binary#1	2	-	-	-	Controller
40063	8239	Bin Outputs		Binary#2	2	-	-	-	Controller
40064	8602	IOM Bin Inp		Binary#3	2	-	-	-	Extension
40065	8316	Led GCB Green		Unsigned	1	0	-	-	Invisible
40066	8318	Led MCB Green		Unsigned	1	0	-	-	Invisible
40067	8320	Led GEN Green		Unsigned	1	0	-	-	Invisible
40068	8321	Led GEN Red		Unsigned	1	0	-	-	Invisible
40069	8322	Led MAINSGreen		Unsigned	1	0	-	-	Invisible
40070	8323	Led MAINS Red		Unsigned	1	0	-	-	Invisible
40071	8330	Engine State		Unsigned	2	0	-	-	IL Info
40072	8455	Breaker State		Unsigned	2	0	-	-	IL Info
40073	8954	Timer Text		Unsigned	2	0	-	-	IL Info
40074	8955	Timer Value	s	Unsigned	2	0	-	-	IL Info
40075	8707	FW Branch		Unsigned	1	0	-	-	IL Info
40076	8393	FW Version		Unsigned	1	1	-	-	IL Info
40077	8480	Application		Unsigned	1	0	-	-	IL Info
40078	8449	(N/A)							
40079	9651	ControllerMode		Unsigned	1	0	-	-	Invisible
40080	9574	ControllerMode		Unsigned	1	0	-	-	Invisible
40081	11134	(N/A)							
40082	9978	Ig Nom		Unsigned	2	0	-	-	Invisible
40083	8450	IgMax		Unsigned	2	0	-	-	Invisible
40084	8451	ST		Unsigned	2	0	-	-	Invisible

A part of Cfg Image (Modbus Register …) file for IC-NT controller

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40001	9143	LogBout0		Binary	2	-	-	-	Invisible
40002	9144	LogBout1		Binary	2	-	-	-	Invisible
40003	9145	LogBout2		Binary	2	-	-	-	Invisible
40004	9146	LogBout3		Binary	2	-	-	-	Invisible
40005	9147	LogBout4		Binary	2	-	-	-	Invisible
40006	9148	LogBout5		Binary	2	-	-	-	Invisible
40007	9149	LogBout6		Binary	2	-	-	-	Invisible
40008	8192	Gen V L1-N	V	Unsigned	2	0	-	-	Generator
40009	8193	Gen V L2-N	V	Unsigned	2	0	-	-	Generator
40010	8194	Gen V L3-N	V	Unsigned	2	0	-	-	Generator
40058	8213	Battery Volts	V	Integer	2	1	-	-	Controller
40059	10124	CPU Temp	°C	Integer	2	1	-	-	Controller
40060	10603	D+	V	Integer	2	1	-	-	Controller
40061	8227	Oil Press	Bar	Integer	2	1	-	-	Controller
40062	8228	Water Temp	°C	Integer	2	0	-	-	Controller
40063	8229	Fuel Level	%	Integer	2	0	-	-	Controller
40064	8978	IOM AI1	U4	Integer	2	0	-	-	Extension
40065	8759	IOM AI2	U5	Integer	2	0	-	-	Extension
40066	8760	IOM AI3	U6	Integer	2	0	-	-	Extension
40067	8761	IOM AI4	U7	Integer	2	0	-	-	Extension
40068	8237	(N/A)							
40069	8235	Bin Inputs		Binary#1	2	-	-	-	Controller
40070	8239	Bin Outputs		Binary#2	2	-	-	-	Controller
40071	8602	IOM Bin Inp		Binary#3	2	-	-	-	Extension
40072	8316	Led GCB gr		Unsigned	1	0	-	-	Invisible
40073	8318	Led MCB gr		Unsigned	1	0	-	-	Invisible
40074	8320	Led GEN gr		Unsigned	1	0	-	-	Invisible
40075	8321	Led GEN red		Unsigned	1	0	-	-	Invisible
40076	8322	Led MAINS gr		Unsigned	1	0	-	-	Invisible
40077	8323	Led MAINS red		Unsigned	1	0	-	-	Invisible
40078	11420	Led BUS		Unsigned	1	0	-	-	Invisible
40079	9647	Led MCB fdb		Unsigned	1	0	-	-	Invisible
40080	8330	Engine State		Unsigned	2	0	-	-	Info
40081	8455	Breaker State		Unsigned	2	0	-	-	Info
40082	8954	Timer Text		Unsigned	2	0	-	-	Info
40083	8955	Timer Value	s	Unsigned	2	0	-	-	Info
40084	8707	FW Branch		Unsigned	1	0	-	-	Info
40085	8393	FW Version		Unsigned	1	1	-	-	Info
40086	8480	(N/A)							
40087	8449	ST		Binary	2	-	-	-	Invisible
40088-40089	11388	(N/A)							
40090	9651	ControllerMode		Unsigned	1	0	-	-	Invisible
40091	8450	ST		Unsigned	2	0	-	-	Invisible

Description of Cfg Image

Header	Descritpion
Registers(s)	Register number; register address = register number – 1
Com.Obj.	Corresponding communication object number
Name	Communication object name
Dim	Value dimension
Type	Value data type (see Data types)
Len	Data length in Bytes (max. 64)
Dec	Number of decimals
Min	Value low limit
Max	Value high limit
Group	Group of setpoints/values

Dedicated communication objects

These objects are always available regardless of the controller software modification:

Registers (*)	Register addresses (*)	Number of registers	Access	Data type	Meaning
46347 – 46348	6346 – 6347	2	read/write	Time	Actual time
46349 – 46350	6348 – 6349	2	read/write	Date	Actual date
46351	6350	1	read/write	Unsigned8	Language index selected for displaying of texts specified by data type String (# 7)
46352 – 4653	6351 – 6352	2	read	Domain	Code of the last communication fault See Error list
46354	6353	1	read	Unsigned8	Number of records in the alarm list
46355	6354	1	read	Unsigned16	Number of records in history (# 6)
46356	6355	1			Reserved (register not implemented)
46357	6356	1	read/write	Integer16	Index of requested history record (# 5)
46358	6357	1	write	Unsigned16	Remote key
46359 – 46360	6358 – 6359	2	read/write	Unsigned32	For writing:command argument For reading: command release value (# 3)
46361	6360	1	write	Unsigned16	Command (# 3)
46362	6361	1			Reserved (register not implemented)
46363	6362	1	read/write	Unsigned8	User identification number (# 4)
46364	6363	1	write	Unsigned16	Entering of password for writing (# 4)
46365	6364	1			Reserved (register not implemented)
46366 – 46490	6365 – 6489	125	read	Domain	Values multipacket(#8)
46491	6490	1			Reserved (register not implemented)
46493 – 46541	6492 – 6540	50	read	String	Header of the particular history record (# 1)
46542	6541	1			Reserved (register not implemented)
46543 – 46667	6542 – 6666	125	read	Domain	Data part of the particular history record (# 2)
46668	6667	1			Reserved (register not implemented)
46669 – 46693	6668 – 6692	25	read	String	1. record in alarm list (# 1)
46694 – 46718	6693 – 6717	25	read	String	2. record in alarm list (# 1)
46719 – 46743	6718 – 6742	25	read	String	3. record in alarm list (# 1)
46744 – 46768	6743 – 6767	25	read	String	4. record in alarm list (# 1)
46769 – 46793	6768 – 6792	25	read	String	5. record in alarm list (# 1)
46794 – 46818	6793 – 6817	25	read	String	6. record in alarm list (# 1)
46819 – 46843	6818 – 6842	25	read	String	7. record in alarm list (# 1)
46844 – 46868	6843 – 6867	25	read	String	8. record in alarm list (# 1)
46869 – 46893	6868 – 6892	25	read	String	9. record in alarm list (# 1)
46894 – 46918	6893 – 6917	25	read	String	10. record in alarm list (# 1)
46919 – 46943	6918 – 6942	25	read	String	11. record in alarm list (# 1)
46944 – 46968	6943 – 6967	25	read	String	12. record in alarm list (# 1)
46969 – 46993	6968 – 6992	25	read	String	13. record in alarm list (# 1)
46994 – 47018	6993 – 7017	25	read	String	14. record in alarm list (# 1)
47019 – 47043	7018 – 7042	25	read	String	15. record in alarm list (# 1)
47044 – 47068	7043 – 7067	25	read	String	16. record in alarm list (# 1)
47069 – 47168	7068 – 7167	100			Reserved (registers not implemented)

(*) in DEC

 # 1
The result of reading of an unused record is an empty string.
 # 2
The result of reading of an unused record is a domain with zero value.
 # 3
An argument must be written before writing of a command code, because immediately after the command code has been written, the command is executed. It is recommended to write an argument and command simultaneously, in a multiple registers write. As the argument has lower register address than command, the required sequence is maintained. See List of commands and modbus communication examples.
 # 4
Before entering the password for writing it is necessary to define user identification number. It is recommended to enter user identification number and password simultaneously. Entered password stays valid 5 minutes after the last successful writing.
 # 5
The latest record has index 0, older record has index -1, next record has index -2, …

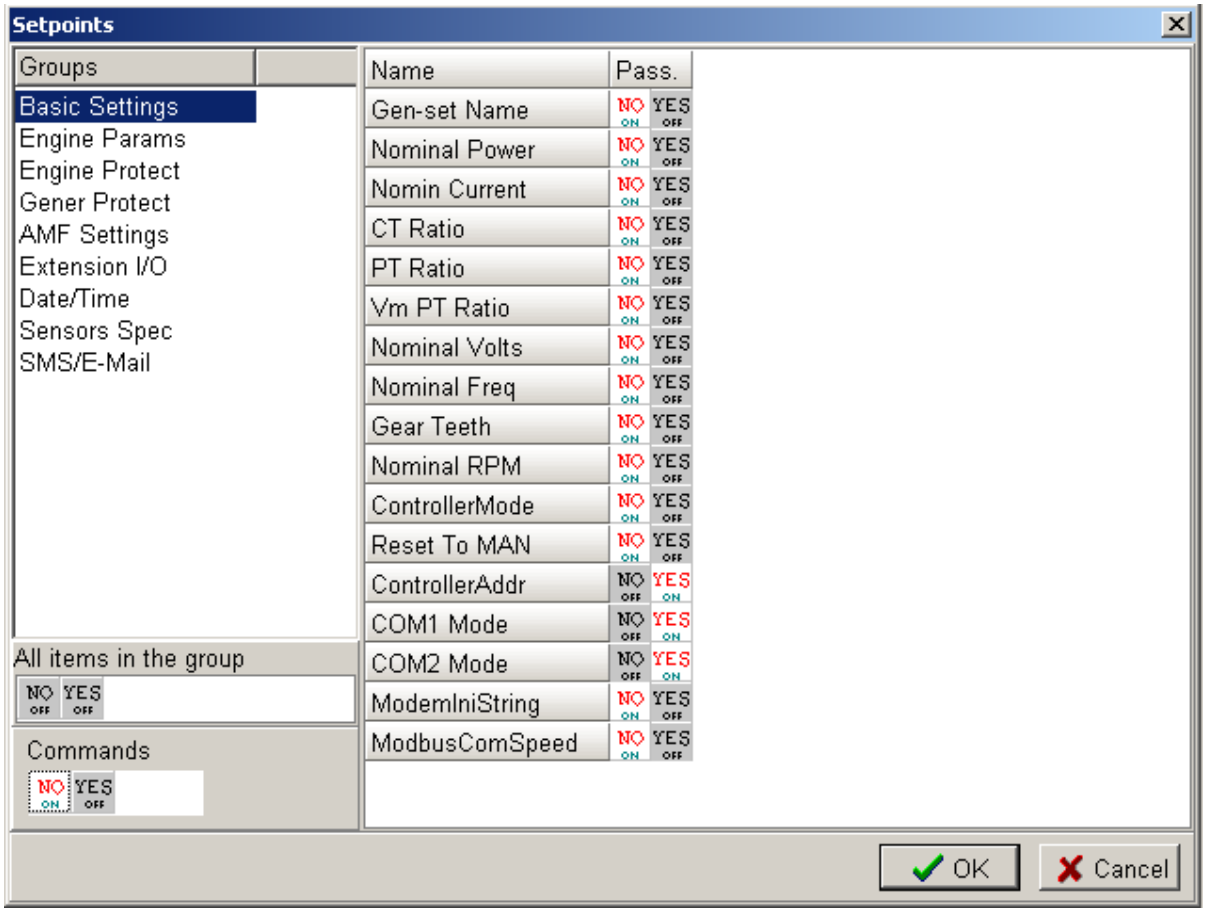
 # 6
It is possible to read and write only in case that history reading is not locked by another terminal. Second necessary condition is to previously write the index.

 # 7
Implicitly = 0.
#8
„Values multipacket“ contains values that are currently configured in the history record.

Access to dedicated communication objects of the controller

Dedicated communication objects are setpoints and commands that are protected by a password against writing. The set of protected objects is given in the controller configuration and is fixed for a particular controller.

In IL-NT controllers it is possible to define password for setpoints. If is the setpoint protected, the user can change this setpoint only after unlocking setpoint with this password. After writing the password will be unlocked all protected setpoints. For example setpoints in the Basic Settings group can be configured in LiteEdit on Setpoints card:



Commands

Command (*)	Meaning	Argument (*)	Return value (*)	
1	Engine start	01FE0000	000001FF	OK
			2	Argument has not been written
	Engine stop	02FD0000	000002FE	OK
			2	Argument has not been written
	Horn reset	04FB0000	000004FC	OK
	Fault reset	08F70000	000008F8	OK
	ECU Fault reset	10EF0000	000010F0	OK
		other	1	Wrong argument
2	Close generator circuit breaker	11EF0000	000011F0	OK
			2	Argument has not been written
	Open generator circuit breaker	11F00000	000011F1	OK
			2	Argument has not been written
	Close/open mains circuit breaker	12ED0000	000012EE	OK
			2	Argument has not been written
	Close mains circuit breaker	12EE0000	000012EF	OK
			2	Argument has not been written
	Open mains circuit breaker	12EF0000	000012F0	OK
			2	Argument has not been written
		other	1	Wrong argument
5	Reset from Init state (#1)	44440000	00004445	OK
			1	Not possible to perform
7	Statistics reset	007C0000	0000007D	OK
			1	Not possible to perform
8	Set kWh counter	New value	N/A	
C	Set kVAhr counter	New value	N/A	
E	Set counter of engine starts	New value	N/A	
D	Set runhours counter	New value	N/A	
19	Set counter of unsuccessful engine starts	New value	N/A	

(*) in HEX

 # 8
If the controller setpoints are not valid after it is switched on, the controller goes to a blocked state. In this state it is necessary to modify the setpoints from the controller keypad and switch off and on the controller or from the external terminal and unblock the controller by **Reset from Init state** command. Another condition necessary to unblock the application function of the controller is valid configuration.

Modbus Appendix

Error list

If the controller encounters an error when processing the query, it returns the exceptional response instead of the normal one to the terminal. An exception has always the value 2 (*Illegal Data Address*). After receiving the exceptional response, the terminal should read the communication object 24523 containing the last error specification. The meaning of an error can be found out from the following table.

MSB1 (*)	LSB1 (*)	MSB2 (*)	LSB2 (*)	Meaning
0	0	0	0	No error.
0	0	2	6	Comm. Object nonexistent.
0	0	1	6	Illegal access: Read (write) of the communication object. Object intended only for write (read).
255	0	0	8	Controller application isn't active
254	0	0	8	Inexpectant message
253	0	0	8	No more unread records in event history.
252	0	0	8	Setpoint not defined in controller configuration.
251	0	0	8	Bad write data length.
250	0	0	8	Invalid password.
249	0	0	8	No more free space in front for EEPROM
248	0	0	8	Too long parameter
247	0	0	8	Invalid controller configuration.
246	0	0	8	Undefined command
245	0	0	8	Command can't be done
244	0	0	8	Too long data answer form peripheries (max. 4B)
243	0	0	8	Too long data for peripheries (max. 4B)
242	0	0	8	Unavailable peripheries
241	0	0	8	Required operation isn't available in peripheries
240	0	0	8	Operation cannot be performed now, the terminal has to repeat the request. This error can occur when an operation with EEPROM memory (setpoint write, history record read) is required at the same time while an internal EEPROM write cycle takes place.
239	0	0	8	Controller programming can't be carry out
238	0	0	8	Write cannot be performed – power supply failure detected.
237	0	0	8	Another active call request is present. This error code can be returned by the controller as the response to the communication object. Object 24540 write – active call termination.
236	0	0	8	Programming error
235	0	0	8	This error is reported by iG-MU module (*Modem Bridge*) in the case of a connection failure between the module and the addressed controller. The terminal can evaluate this error as a communication *timeout* with the controller.
234	0	0	8	Write cannot be performed – periphery not responding.
233	0	0	8	Write cannot be performed – setpoint nonexistent in any periphery.
232	0	0	8	Bad access code for communication from a remote terminal.
231	0	0	8	Invalid controller address: value out of range 1 to 32 or already used. This error is a reaction on communication object. Object 24537 write.
230	0	0	8	Error in definition for communication oscilloscope objects
229	0	0	8	Undefined action. A reaction on communication object. Object 24521 write.
228	0	0	8	Action (although defined) cannot be performed. A reaction on communication object. Object 24521 write.
227	0	0	8	Written object value is not acceptable.
226	0	0	8	No more free slots

MSB1 (*)	LSB1 (*)	MSB2 (*)	LSB2 (*)	Meaning
225	0	0	8	No connection
224	0	0	8	Locked, block reading is active
223	0	0	8	Locked, commanding is active
222	0	0	8	Locked, the history reading is active
221	0	0	8	Locked, the programming is active
220	0	0	8	Communication error
219	0	0	8	Request for – data
218	0	0	8	Request for – SMS
217	0	0	8	Request for – email
216	0	0	8	Request for – mobile email
215	0	0	8	Request for - fax
214	0	0	8	Wrong access code, the connection must be terminated.
213	0	0	8	Reserved for HW key
212	0	0	8	Reserved for DENOX
211	0	0	8	Unsufficient access rights.
210	0	0	8	The request can be submitted only by the administrator (User 0).
209	0	0	8	The administrator has entered a wrong user identification number.
208	0	0	8	Not possible to write, the communication object has forced value.
207	0	0	8	The administrator requests an unsupported operation.
206	0	0	8	Selected communication mode doesn't allow required interface
205	0	0	8	Selected interface doesn't allow required communication mode
204	0	0	8	HW data flow control for modem communication. Sending as answer to request to read 24437 communication object
203	0	0	8	SW data flow control for modem communication. Sending as answer to request to read 24437 communication object
202	0	0	8	Access denied from actual IP address
201	0	0	8	Unknown fault.
200	0	0	8	Invalid register.
199	0	0	8	Reading of alarm list is locked.
198	0	0	8	Reading of history is locked.
197	0	0	8	Reading of alarm list has to be started by reading the first record.
196	0	0	8	The history record is not defined for reading of history.
195	0	0	8	It is not possible to request such number of registers.
201	0	0	8	Unknown fault.
200	0	0	8	Invalid register.
199	0	0	8	Reading of alarm list is locked.
198	0	0	8	Reading of history is locked.

(*) in DEC dedicated for Internet Bridge

Error list

If the controller encounters an error when processing the query, it returns the exceptional response instead of the normal one to the terminal. An exception has always the value 2 (*Illegal Data Address*). After receiving the exceptional response, the terminal should read the communication object 24523 containing the last error specification. The meaning of an error can be found out from the following table.

MSB1 (*)	LSB1 (*)	MSB2 (*)	LSB2 (*)	Meaning
0	0	0	0	No error.
0	0	2	6	Comm. Object nonexistent.
0	0	1	6	Illegal access: Read (write) of the communication object. Object intended only for write (read).
253	0	0	8	No more unread records in event history.
252	0	0	8	Setpoint not defined in controller configuration.
251	0	0	8	Bad write data length.

MSB1 (*)	LSB1 (*)	MSB2 (*)	LSB2 (*)	Meaning
250	0	0	8	Invalid password.
247	0	0	8	Invalid controller configuration.
240	0	0	8	Operation cannot be performed now, the terminal has to repeat the request. This error can occur when an operation with EEPROM memory (setpoint write, history record read) is required at the same time while an internal EEPROM write cycle takes place.
238	0	0	8	Write cannot be performed – power supply failure detected.
237	0	0	8	Another active call request is present. This error code can be returned by the controller as the response to the communication object. Object 24540 write – active call termination.
235	0	0	8	This error is reported by iG-MU module (*Modem Bridge*) in the case of a connection failure between the module and the addressed controller. The terminal can evaluate this error as a communication *timeout* with the controller.
234	0	0	8	Write cannot be performed – periphery not responding.
233	0	0	8	Write cannot be performed – setpoint nonexistent in any periphery.
232	0	0	8	Bad access code for communication from a remote terminal.
231	0	0	8	Invalid controller address: value out of range 1 to 32 or already used. This error is a reaction on communication object. Object 24537 write.
229	0	0	8	Undefined action. A reaction on communication object. Object 24521 write.
228	0	0	8	Action (although defined) cannot be performed. A reaction on communication object. Object 24521 write.
227	0	0	8	Written object value is not acceptable.
214	0	0	8	Wrong access code, the connection must be terminated.
211	0	0	8	Unsufficient access rights.
210	0	0	8	The request can be submitted only by the administrator (User 0).
209	0	0	8	The administrator has entered a wrong user identification number.
208	0	0	8	Not possible to write, the communication object has forced value.
207	0	0	8	The administrator requests an unsupported operation.
201	0	0	8	Unknown fault.
200	0	0	8	Invalid register.
199	0	0	8	Reading of alarm list is locked.
198	0	0	8	Reading of history is locked.
197	0	0	8	Reading of alarm list has to be started by reading the first record.
196	0	0	8	The history record is not defined for reading of history.
195	0	0	8	It is not possible to request such number of registers.

(*) in DEC

Data types

The following table contains the communication objects data types and their representation in the data part of the communication function.

Data type	Meaning	Number of registers	Data part of the communication function[1]
Integer8	Signed integer – 8 bits	1	MSB1 = sign extension LSB1 LSB1 = comm. object value
Unsigned8	Unsigned integer – 8 bits	1	MSB1 = 0 LSB1 = comm. object value
Integer16	Signed integer – 16 bits	1	MSB1 = comm. object value, bits 15-8 LSB1 = comm. object value, bits 7-0
Unsigned16	Unsigned integer – 16 bits	1	MSB1 = comm. object value, bits 15-8 LSB1 = comm. object value, bits 7-0

[1] MSBx = register x, bits 15-8
LSBx = register x, bits 7-0

Integer32	Signed integer – 32 bits	2	MSB1 = comm. object value, bits 31-24 LSB1 = comm. object value, bits 23-16 MSB2 = comm. object value, bits 15-8 LSB2 = comm. object value, bits 7-0
Unsigned32	Unsigned integer – 32 bits	2	MSB1 = comm. object value, bits 31-24 LSB1 = comm. object value, bits 23-16 MSB2 = comm. object value, bits 15-8 LSB2 = comm. object value, bits 7-0
Binary8	Binary number – 8 bits	1	MSB1 = 0 LSB1 = comm. object value
Binary16	Binary number – 16 bits	1	MSB1 = comm. object value, bits 15-8 LSB1 = comm. object value, bits 7-0
Binary32	Binary number – 32 bits	2	MSB1 = comm. object value, bits 31-24 LSB1 = comm. object value, bits 23-16 MSB2 = comm. object value, bits 15-8 LSB2 = comm. object value, bits 7-0
Char	ASCII character	1	MSB1 = 0 LSB1 = comm. object value
List	String list	1	MSB1 = 0 LSB1 = comm. object value
ShortStr	ASCII string of max. length of 15 characters (zero terminated string)	8	MSB1 = 1. character of the string LSB1 = 2. character of the string MSB2 = 3. character of the string LSB2 = 4. character of the string …
LongStr	ASCII string of max. length of 31 characters (zero terminated string)	16	MSB1 = 1. character of the string LSB1 = 2. character of the string MSB2 = 3. character of the string LSB2 = 4. character of the string …
Date	Date	2	MSB1 = BCD(day) LSB1 = BCD(month) MSB2 = BCD(year) LSB2 = 0 example: MSB1 = 18 (HEX) LSB1 = 04 (HEX) MSB2 = 01 (HEX) LSB2 = 0 ⇒ Date = 18.4.(20)01
Time	Time	2	MSB1 = BCD(hour) LSB1 = BCD(minute) MSB2 = BCD(second) LSB2 = 0 example: MSB1 = 20 (HEX) LSB1 = 24 (HEX) MSB2 = 02 (HEX) LSB2 = 0 ⇒ Time = 20:24:02
Domain	Field n bytes C-declaratione: unsigned char x[n]	n	MSB1 = x[0] LSB1 = x[1] MSB2 = x[2] LSB2 = x[3] … n is even number: MSBm-1 = x[n-2] LSBm = x[n-1] n is odd number: MSBm-1 = x[n-1] LSBm = 0

String	String (Zero terminated string)	depends on register number	string characters coding depends on chosen language (8bit coding, EUC)

Communication status

Communication object number:

24571

Operation:

Read only

Data type:

Binary32

Meaning:

Bit 0	Internal terminal in InteliSys does not work (0 for other controllers)
Bit 1	Invalid controller software (based on CRC).
Bit 2	Invalid controller configuration (based on CRC).
Bit 3	In the event history is present at least one unread record.
Bit 4	P type setpoints are invalid. P type setpoints are representing the controller setpoints. Values of these setpoints can be set from connected terminals. If these setpoints are invalid, the application functions are blocked. Setpoints recovery is needed.
Bit 5	R type setpoints are invalid. R type setpoints are representing the data, that is only initialized from connected terminals, but its updating is made by the controller itself (e.g. statistic or time and date). If these setpoints are invalid, their change from the controller is blocked. Setpoints recovery is needed.
Bit 6	The event history was cleared.
Bit 7	The event history was filled up at least once.
Bit 8	P type setpoint change occurred (reading resets this bit).
Bit 9	R type setpoint change occurred (reading resets this bit).
Bit 10	Controller type – see the table below.
Bit 11	Alarm list not empty.
Bit 12	Alarm list change (reading resets this bit).
Bit 13	New item added into alarm list (reading resets this bit).
Bit 14	Internal controller terminal is locked up for setpoint change.
Bit 15	Invalid configuration format.
Bit 16	Diagnostic codes change (reading resets this bit, only for IL-NT /ID controllers).
Bits 20	Controller type (*)
Bit 21-17	Reserve (= 0)
Bits 22-21	Password level for Setpoints and Commands write (only for IL-NT /ID controllers).
Bit 23	Controller was initiated.
Bits 28-24	Communication module version.
Bits 29	Remote terminal is connected.
Bits 30	Controller type – see the table below.
Bits 31	Reserve (= 0)

(*) Controller type

Bit 20	Bit 30	Bit 10	Controller
0	0	0	InteliSys
0	0	1	InteliGen
0	1	0	IL-NT
0	1	1	InteliDrive
1	0	0	IG/IS-NT
1	0	1	Reserve
1	1	0	Reserve
1	1	1	Reserve

Hint:

The MODE< and MODE> commands have not been implemented to the register oriented modbus commands.

Modem Recommendations

The controller has to be connected to modem via standard modem cable where the DSR (Data Send Ready) signal detects modem presence (ComAp order code AT-LINK CABL).

Hint:
It is recommended to use the same type of modem on the both sides of connection.
For GSM modem proper set-up use automatic ComAp GSM set-up software from the installation package. Setup software runs independently. In MS Windows select: Start - Program files – Comap PC Suite– Tools – Gm_setup.exe

Analog Modem with DC Supply

Devolo Microlink 56k I is designed for the industrial applications. Power supply range is 9 - 30 V AC and 9 - 42 V DC. See http://www.devolo.de/.
INSYS Modem 56k small INT 2.0, 10-32 VDC. See www.insys-tec.cz.

Recommended ISDN Modem

Askey TAS-200E (power supply 12 V DC)
ASUScom TA-220ST
Devolo Microlink ISDN i

Hint:
The ISDN modems must work in the X.75 or V.120 protocols. The internet connection (HDLC-PPP) does not work.

Recommended CDMA Modems

Maxon MM-5100, 800MHz, 1xRTT (tested in Australia)
AirLink Raven XT (tested in USA)

Hint:
The usage possibility depends on the network type.

Recommended GSM Modems

Siemens M20, TC35, TC35i, ES75, MC39 (baud rate 9600 bps), TC65.
Wavecom M1200/WMOD2 (baud rate 9600 bps).
Wavecom - Maestro 20, dual 900/1800MHz.
Wavecom – Fastrack M1306B (GSM/GPRS CI.10 Modem), dual 900/1800 MHz (Fastrack M1206B is **NOT** recommended)
FALCOM A2D, dual 900/1800MHz.
CEP GS64 Terminal
Wavecom Fastrack Supreme 10

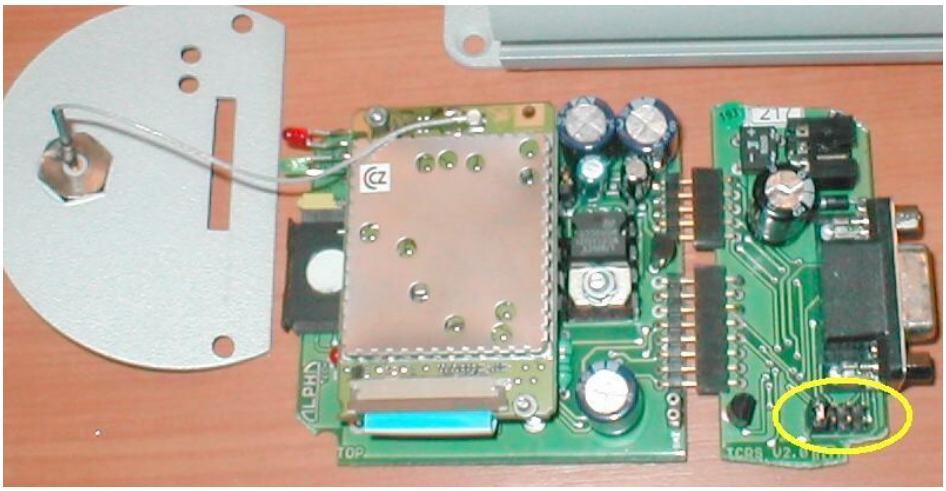
GSM modem wiring notes

IL-NT/IC-NT controllers automatically detect modem connection via RS232 - DSR (Data Set Ready) signal. Controllers detect modem when DSR is active and direct connection when passive or not connected.

Any connected modem has to be set to active DSR after switch on and has to be connected via standard modem cable.

Leave I-LB jumper “HW/SW control” opened for this connection.

Hint:
Make sure all signals are connected and activated in modem when it is not possible to open connection. Some types of GSM module have jumpers select table control and handshaking signals.



TC35 Alphatech GSM modem: the first jumper from the left is closed.

When modem **TC35i** does not respond for sending command SMSes, do the following:

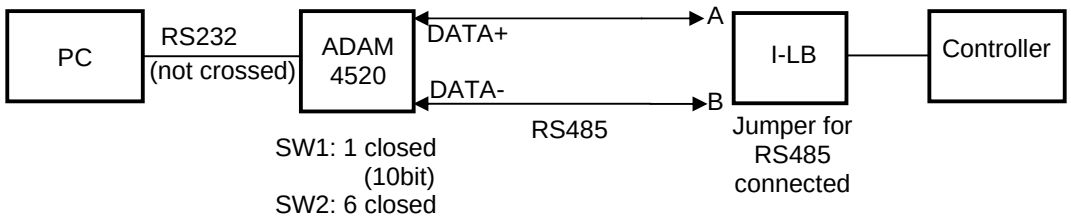
1. Send **AT+CPMS="MT","MT","MT"** command via hyperterminal or by means of ModemIniString parameter.
2. Send **AT+CPMS="SM","SM","SM"** command via hyperterminal or by means of ModemIniString parameter.
3. Restart the modem.

3G Modems

The functionality of 3G modems with Comap controllers depends on the operator and his network settings. Therefore it is recommended to first test the controller with such modem.

Converters

Converter RS232 « RS485



General properties of RS232 to RS485 converters:

- Has to be set to passive DSR signal (when DSR connected) after switch on.
- No external data flow control signals are allowed – automatic data flow control required.

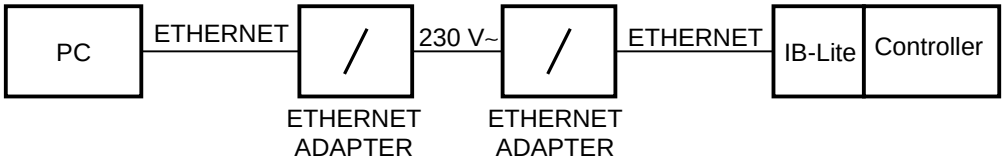
Recommended converters

- External:
  ADAM 4520,
  ADVANTECH, (http://www.advantech.com/)
  - DIN rail, automatic RS485 bus supervision, no external data flow control signals, galvanic isolated, baud rate 19200 bps for IL-NT v.1.2, 57600 bps for IL-NT from v.1.3 and IC-NT
  - When communication is working the LED on ADAM 4520 is going from full illumination to short darkness then again full illuminated
  - When communication of IG-MU is working, PWR and RUN LEDs full red illuminated; TxD and RxD flashing when transmitting
  - When ADAM module is used then connect Rx, Tx-A to DATA+ and Rx, Tx-B to DATA-. Shielding connect to ADAM GND on both sides and external 120ohm resistor between DATA+ and DATA- on ADAM side (in off state). Internal ADAM 4520 switches: set Data format 10 bit and baud rate. Cable must not be crossed (RxD-TxD) RS232 for connection between ADAM and PC SUB9 RS232 connector wiring: 2 – 2, 3 – 3, 5 – 5, 7 – 7.

- Internal for PC:
  PCL-745B or PCL745S,
  ADVANTECH, (http://www.advantech.com/)
  (Dual port RS422/485 Interface card, automatic RS485 bus supervision, no external data flow control signals, galvanic isolated, baud rate 19200 bps)

Hint:
In the case of surge hazard (connection out of building in case of storm etc.) see the "Recommended CAN/RS485 connection" chapter of the IGS-NT-2.2-Installation guide.pdf.

Converter 230 V AC « TCP/IP



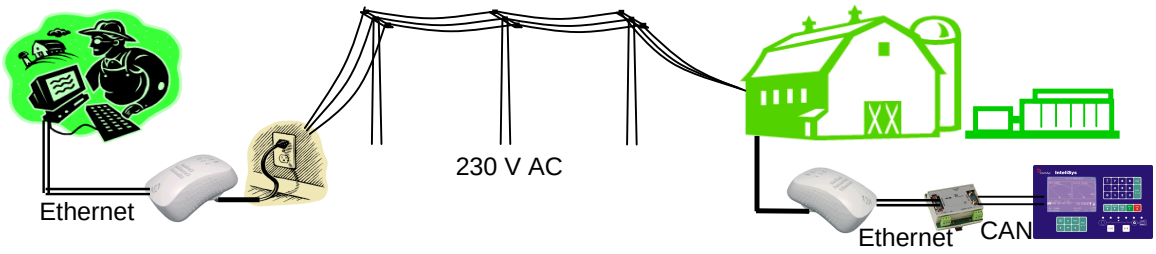
- For installations where IB-Lite is used but internet connection is not available
- The connection can be established using electric grid (230 V AC) using Ethernet/230VAC converters

- Can be used for distances up to 200 meters

Recommended converter

- Powerline Ethernet Wall Mount,
  Corinex Communications (http://www.corinex.com/)

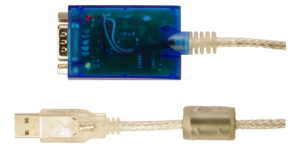
Example



Converter USB « RS232

- Useful for PC/laptops without serial port

Recommended converters



- **UCAB232 Full**,
  ASIX (http://www.asix.cz/)
  UCAB232 is designated for all standard RS232 devices (mouses, modems, data terminals, barcode readers, serial printers) and industrial applications. UCAB232 supports Baud rates from 300 Bd to 250 kBaud (guaranteed) / 500 kBaud (typ.).
- **VPI - USS-101/111**,
  VPI (http://www.vpi.us/usb-serial.html)
  Supports serial devices with speeds up to 230kb/sec (e.g. PDAs, modems, scanners, etc.).
- **C-232-MM,**
  (http://www.usbgear.com/item_288.html)
  The USB Serial Adapter provides instant connectivity with modems, ISDN TAs, PDS, handheld & pocket PCs, digital cameras, POS, serial printers, etc. It supports data rates up to 230 Kbps.

Converter USB « RS485

- Extends distance between PC and controller up to 1200 meters

Recommended converter

- SB485,
  PaPouch elektronika
  (http://www.papouch.com/)

Converter RS-422/485 « Ethernet

Recommended converter

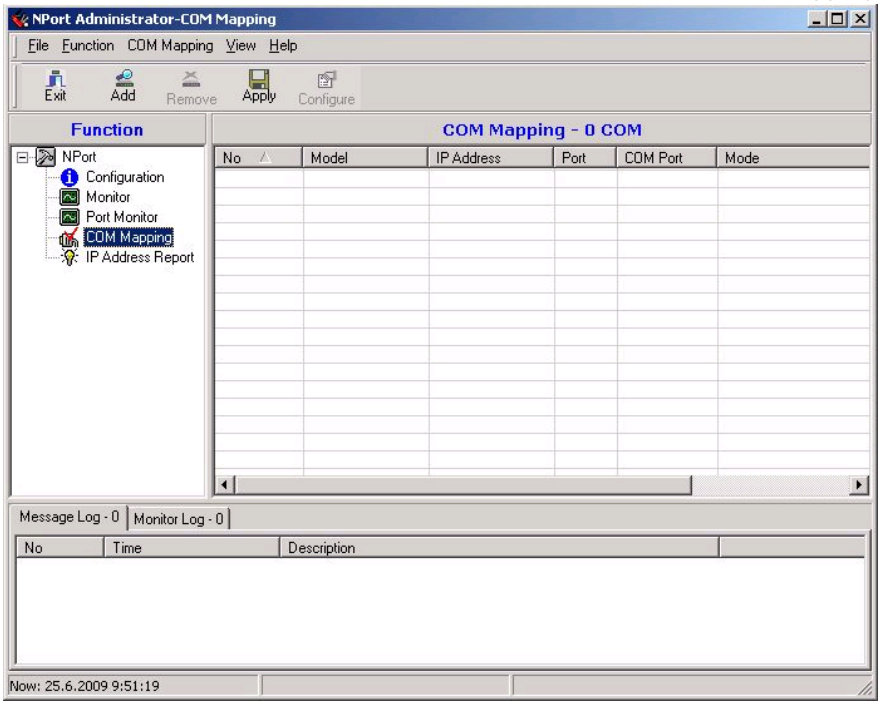
- Nport 6110, MOXA (www.moxa.com)
- NPort 5110
- NPort 5130

Recommended settings

NPort5110

Item	Value
Operation mode	Real COM mode
Network settings – IP address	Static IP

The IP address must be also set in NPort Administrator program. From MOXA webpage download the *NPort Administration Suite.* There is NPort Administrator with function COM mapping:



Here assign the COM port and the IP address.

Nport6110
(Settings in Modbus Gateway Configurator – download from http://web4.moxa.com/support/download.asp)

Card	Item	Value
Network settings	IP Configuration	Static IP
Modbus Settings	Attached Serial Device type	Modbus RTU slave
	Initial delay time	1000
Advanced Modbus Settings	Auto Slave Unit ID	Enable
	Character Timeout	10
	Message Timeout	100
	Modbus/TCP Exception	Yes
Modbus Serial Settings	Interface Mode	Select interface you are using
	Baud Rate	Select Baud rate you are using
	Parity	None
	Data Bits	8
	Stop Bit	1

Controller

IG/IS-NT
Comms settings: *RS232(1) mode / RS232(2) mode** = MODBUS-DIRECT
Comms settings: *RS232(1)MBCSpd / RS232(2)MBCSpd** = 9600 / 19200 / 38400

When using RS485 don't forget to set also
Comms settings: *RS485(1)conv. / RS485(2)conv.** = ENABLED

* Second RS232/485 port available only in IG-NTC/EEC and IS-NT.

IS-CU
Basic settings: *RS232 mode* = MODBUS

Only Baud rate 9600 bps available in IS-CU.

Isolator RS232

- For galvanic separation of the line between Inteli controllers and PC
- Useful when different ground potentials are present

Recommneded isolators

- UC232,
  PaPouch elektronika (http://www.papouch.com/)

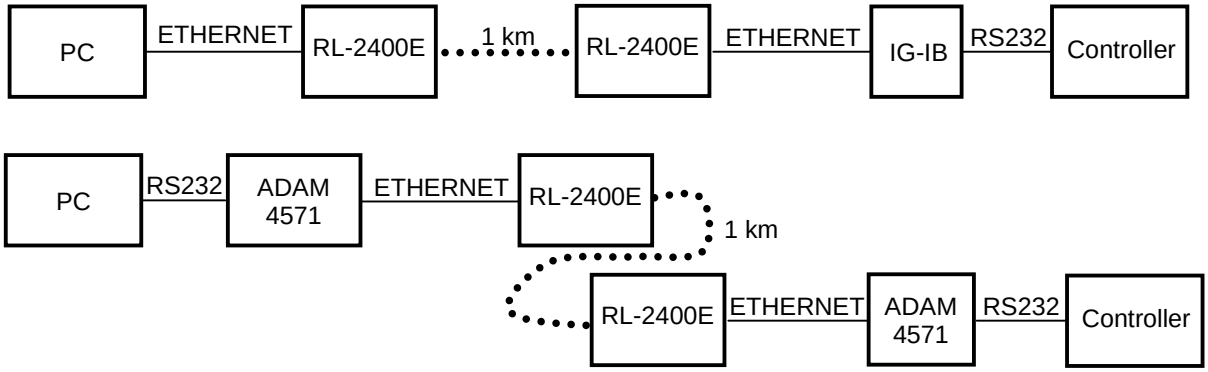
The isolator UC232 can be used instead of UC232-7. The only difference is that UC232 needs external power supply. It can be 5V stabilized or 7-17V unstabilized. The power supply voltage must be specified in the order. Suitable 5V power supply is also available from the Papouch company.

- UC UC232-7,
  PaPouch elektronika (http://www.papouch.com/)

Recommended optical USB extension cables

- Opticis M2-100-xx http://opticis.com
- USB Rover 200 http://www.icron.com

Radio Link

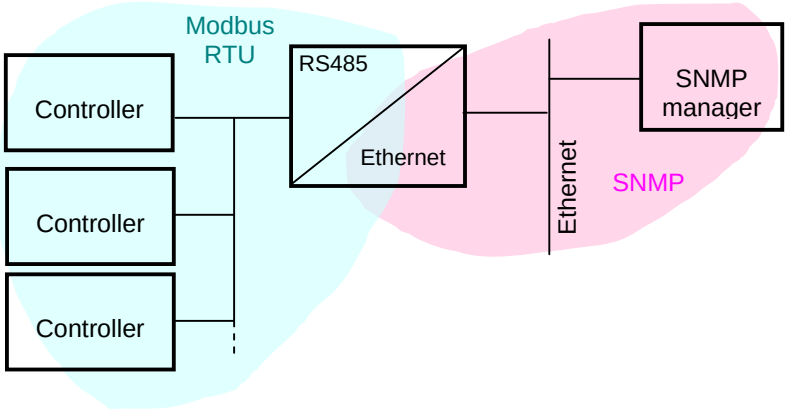


- Useful when the control room is distant from the site
- Can be more economical than to hard wire it

Recommended equipment

- RadioLinx RL-2400E wireless Ethernet switch, ProSoft Technology Inc. (www.prosoft-technology.com)
- ADAM-4571, ADVANTECH (www.advantech.com)
- MOXA DE311, MOXA (www.moxa.com)
- MOXA Nport 5230, MOXA

Converter Modbus RTU « SNMP



- For connection of 1-32 IG/IS-NT (stadard line) controllers to a SNMP supervision system
- Supports GET, SET, TRAP transactions

Hint:
For testing purposes there is IG-NT controller with this converter on address 195.122.193.153 (controller address = 1). Appropriate MIB table is available on www.comap.cz.

MIB Table

The MIB table contains following data objects
Read only:

Modbus Register(s)	Com.Obj.	Name	Dim	Type*	Decimals
40003	8253	Binary inputs		Binary16	
40012	8239	Binary outputs		Binary16	
40013	8213	Ubat	V	Integer16	1
40016	9155	Analog inp. 1 CU		Integer16	1
40017	9156	Analog inp. 2 CU		Integer16	0
40018	9157	Analog inp. 3 CU		Integer16	0
40168	9574	ControllerMode		Unsigned16	-
40249	8192	Gen V L1-N	V	Unsigned16	0
40250	8193	Gen V L2-N	V	Unsigned16	0
40251	8194	Gen V L3-N	V	Unsigned16	0
40256	8210	Gen freq	Hz	Unsigned16	1
40261	8204	Pwr factor		Integer16	2
40262	8395	Load char		Char	

Modbus Register(s)	Com.Obj.	Name	Dim	Type*	Decimals
40264	8202	Act power	kW	Integer16	0
40288	8195	Mains V L1-N	V	Unsigned16	0
40289	8196	Mains V L2-N	V	Unsigned16	0
40290	8197	Mains V L3-N	V	Unsigned16	0
40296	8211	Mains freq	Hz	Unsigned16	1
43589	8207	Num starts		Unsigned16	0
43587	8206	Run hours	h	Integer 32	0
46354		Num items alarmlist		Unsigned16	
46669		Item 1 alarmlist		String	
46694		Item 2 alarmlist		String	
46719		Item 3 alarmlist		String	
46744		Item 4 alarmlist		String	
46769		Item 5 alarmlist		String	
46794		Item 6 alarmlist		String	
46819		Item 7 alarmlist		String	
46844		Item 8 alarmlist		String	
46869		Item 9 alarmlist		String	
46894		Item 10 alarmlist		String	
46919		Item 11 alarmlist		String	
46944		Item 12 alarmlist		String	
46969		Item 13 alarmlist		String	
46994		Item 14 alarmlist		String	
47019		Item 15 alarmlist		String	
47044		Item 16 alarmlist		String	

Read / Write

43027	8315	ControllerMode		Unsigned16
46359		Action argument		Integer32

Write only

24470	24470	User identification number		Unsigned16
24524	24524	Password		Unsigned16
46361		Action command		Unsigned16

* SNMI data types are INTEGER32 for all numerical values except “Run hours”, which is GAUGE32. The column “Type” means how the data shall be interpreted.

Converter settings

Setup of the converter is done via TELNET at port **9999** instead of standard port 21. The default IP address is 192.168.1.254.

To enter setup mode:

1. Connect the converter to LAN
2. Put command “telnet 192.168.1.254 9999” to the windows command line on any computer connected to the same LAN

Setup procedure:

1. Press “0” key to change server configuration (converter IP address, netmask, gateway address etc..)
2. Press “1” key to change device configuration (read/write community, SNMP manager address*..)
3. Press “9” key to save parameters to the memory and exit setup mode

SNMP manager address is IP address of the device the TRAPs are addressed to.

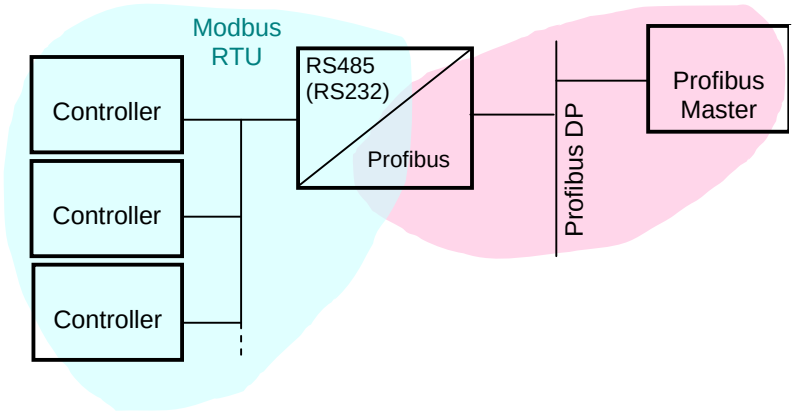
Controller settings

IG/IS-NT
Comms settings: *RS232(1) mode / RS232(2) mode** = MODBUS-DIRECT
Comms settings: *RS232(1)MBCSpd / RS232(2)MBCSpd** = 57600
Comms settings: *RS485(1)conv. / RS485(2)conv.** = ENABLED

* Second RS232/485 port available only in IG-NTC/EEC and IS-NT.

Hint:
The converter provides communication only with controllers (addresses) that are present on startup of the converter. It means any controller powered-up later than the converter is not recognized and supported. The converters are supposed to work with IG/IS-NT controllers of standard line (version 2.1 and higher), IC-NT and IL-NT controllers (standard branches).

Converter Modbus RTU « Profibus



- For connection of 1-32 controllers to a Profibus network
- RS485 or RS232 physical layer for connection to the controller(s)
- Full Profibus-DP slave functionality according IEC61158
- 244 bytes input data size (122 Modbus registers)
- 244 bytes output data size (122 Modbus registers)
- 416 bytes total
- See deatils on the web page of the manufacturer:
  http://www.geindustrial.com/cwc/Dispatcher?REQUEST=PRODUCTS&pnlid=6&id=p485

Converter settings

- Use EnerVista software to setup the converter. It can be downloaded from the web page http://pm.geindustrial.com/download/download.asp?id=p485&file=1.
- To configure the converter in the Profibus network, download the description file http://www.geindustrial.com/products/software/d485/P48509E5.zip and import it to the Profibus configuration tool.

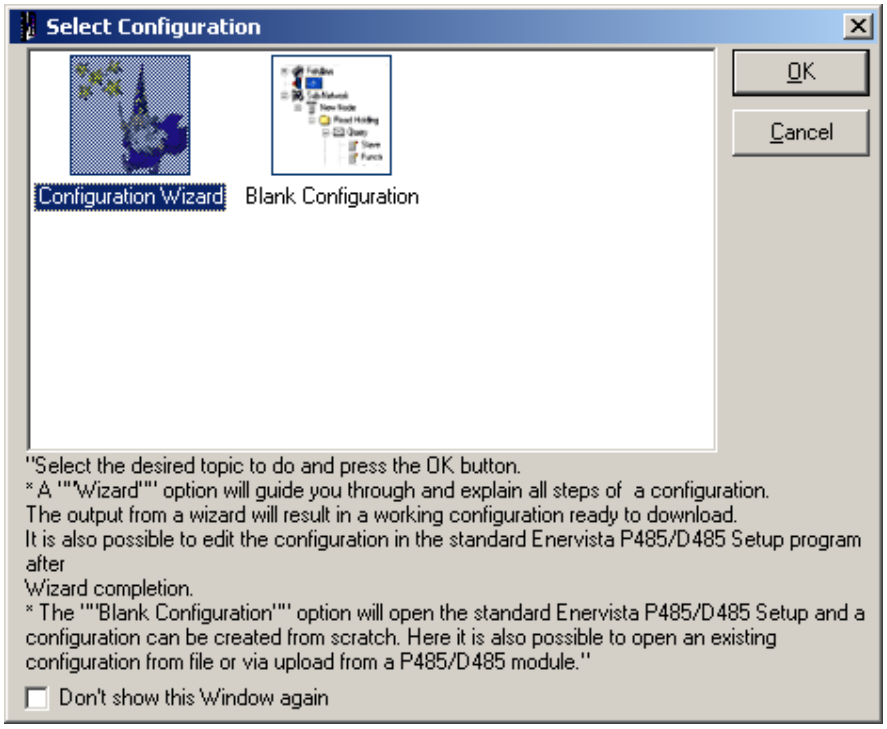
Follow instructions in the P485 manual while configuring the converter. The setup wizard incorporated in the Enervista software will make the setup process much easier, but finally some manual corrections are needed. Below are some notes specific to the connection with ComAp controllers.

1. The physical layer for Modbus communication is select table. The selected type (RS232/RS485) and speed must be same in the P485 and controller, see Controller settings.
2. Use RS485 in case more controllers are connected to the P485.

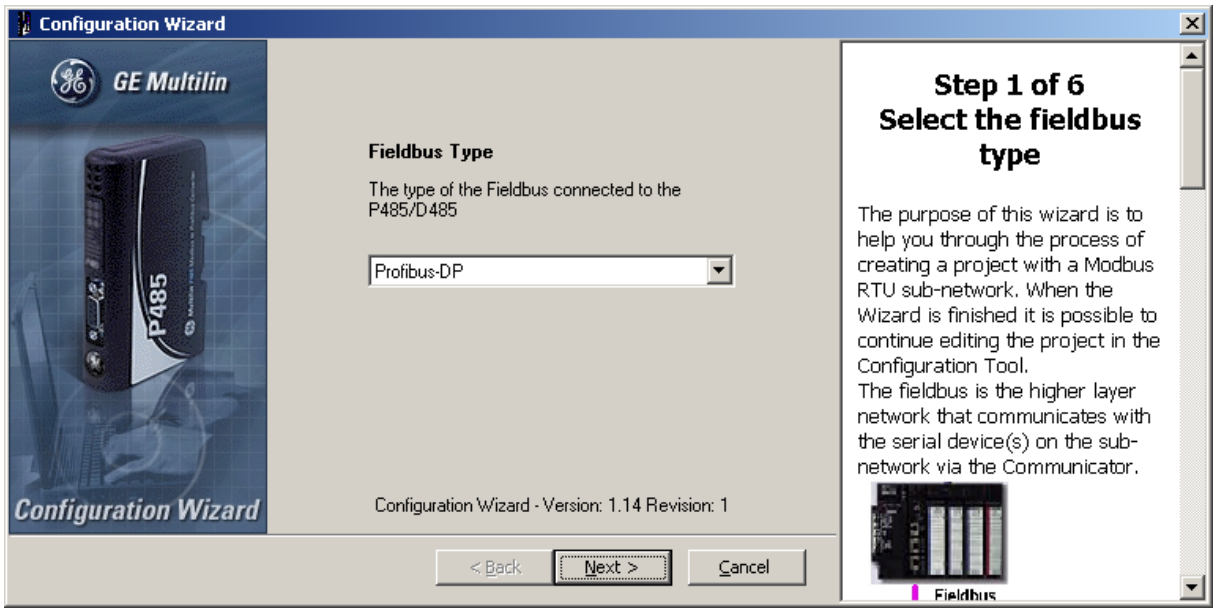
3. A *Device* mentioned in the wizard represents a controller type. Once a device is defined, more nodes of the same type (device) can be created easily.
4. A *modbus network node* represents a controller. The *slave address* must correspond to the Controller address setpoint of the related controller.
5. See Modbus Connection chapter in this document for details about Modbus, register numbers, sizes etc.
6. **Use triggered update mode for writing objects (registers) to the controller. Never use cyclic update mode!**

Setup example (using wizard):

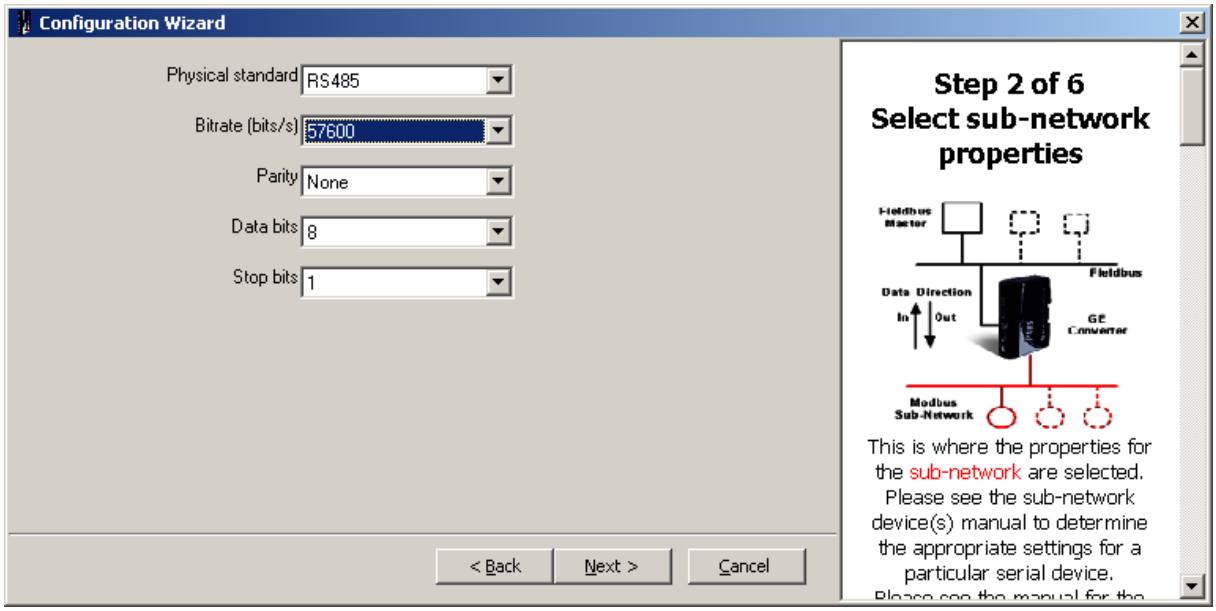
1. Select wizard.



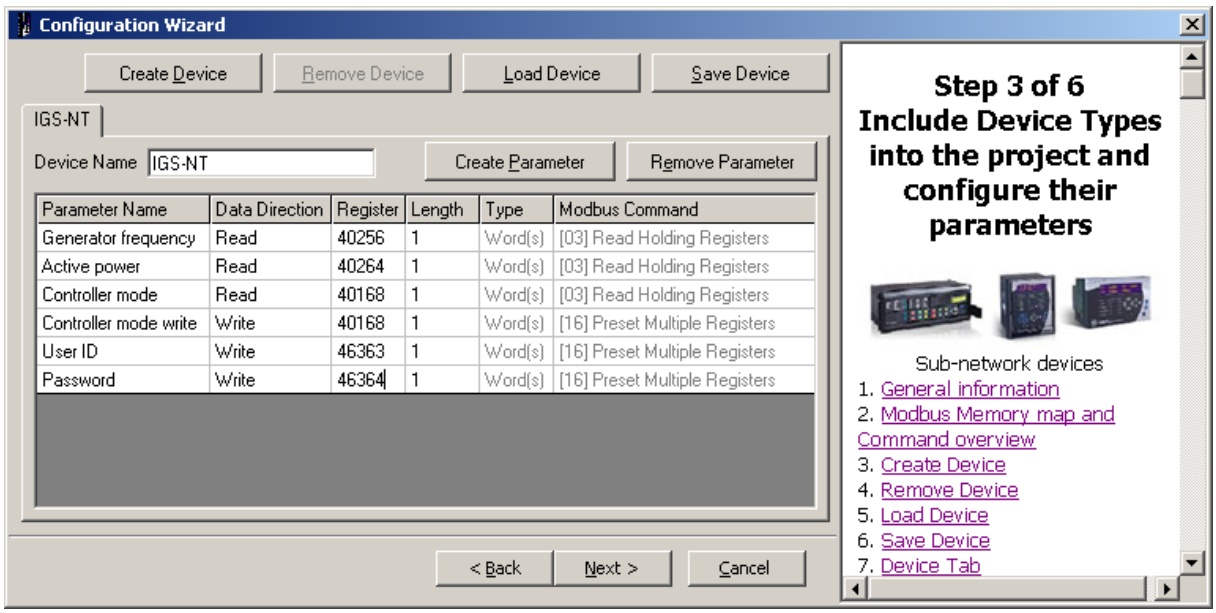
2. Select fieldbus type.



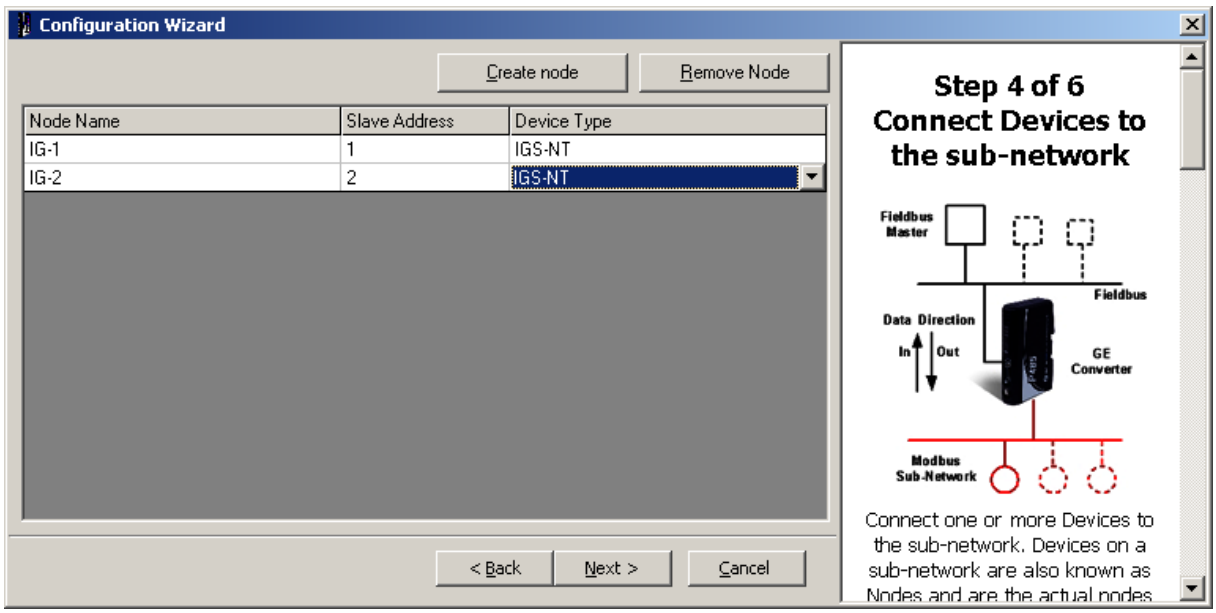
3. Select physical layer and communication parameters for Modbus.



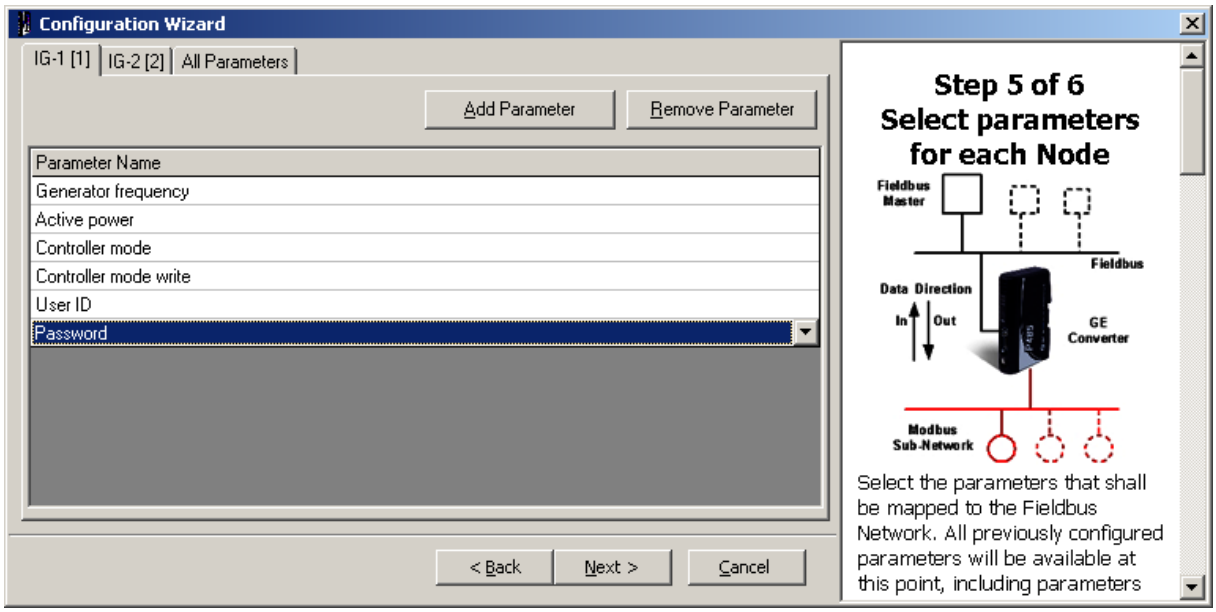
4. Define IGS-NT Device, it's Parameters and related Modbus registers.



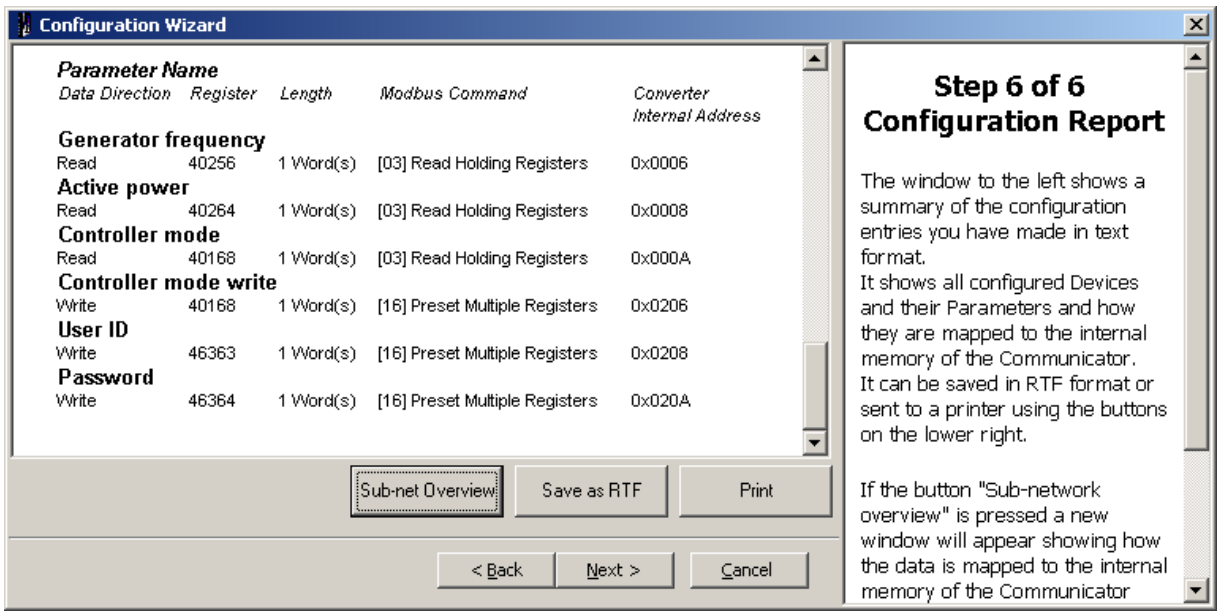
5. Define nodes connected to the Modbus network.



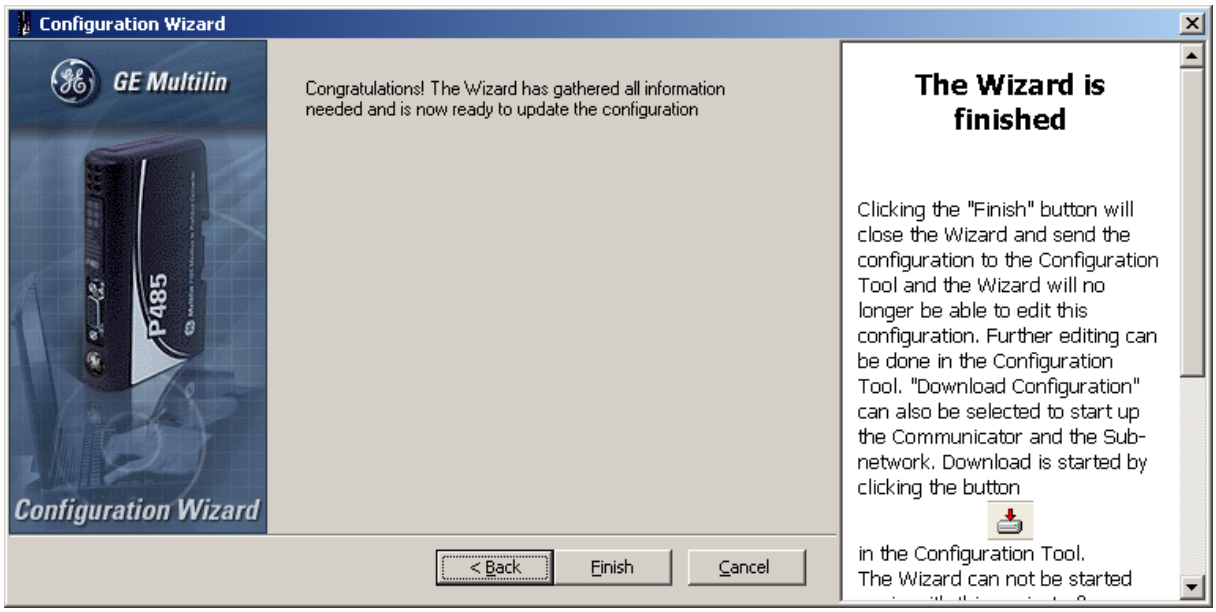
6. For each node select parameters that have to be mapped to the Profibus network.



7. Save the configuration overview (the picture below is only a general example, not a real one).

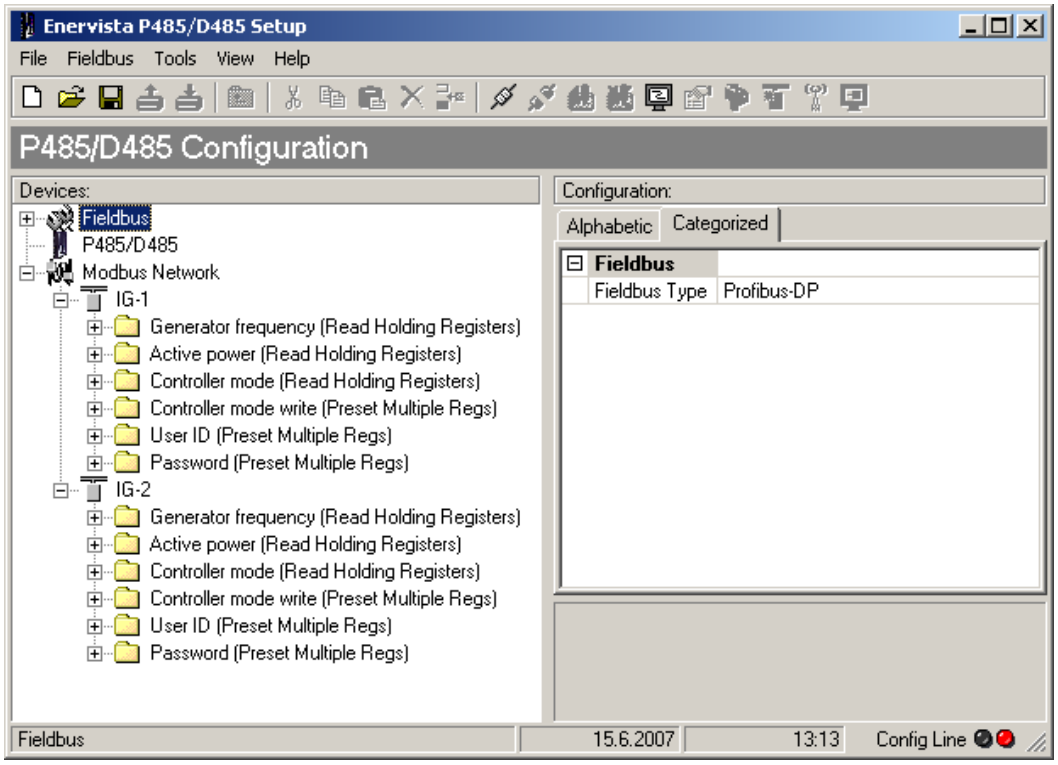


8. The configuration using wizard is finished.

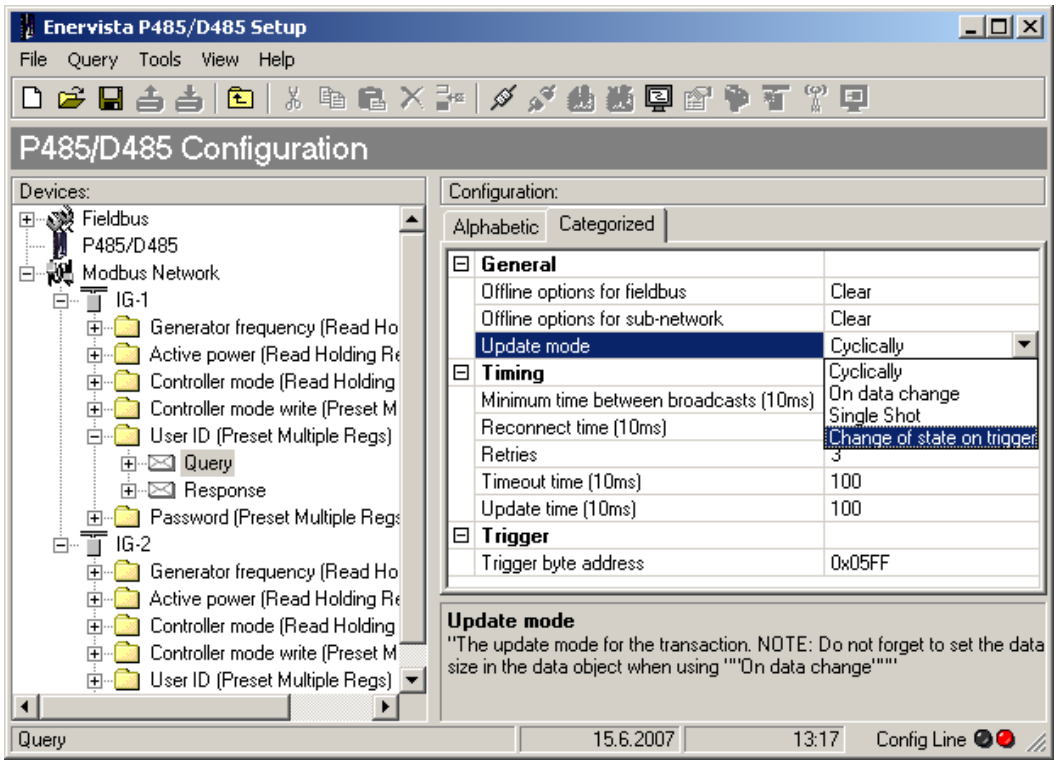


9. Save the configuration to a file.

10. Navigation window will appear:



11. For each write-type parameter modify the property *Update mode* to triggered mode and define *trigger byte address*. The Profibus master must update the parameter data field first and then increase the trigger byte value to write the parameter (register) to the controller.



12. Write the configuration to the P485 and save it also to disk as backup.

Controller settings

IG/IS-NT
Comms settings: *RS232(1) mode* / *RS232(2) mode** = MODBUS-DIRECT
Comms settings: *RS232(1)MBCSpd* / *RS232(2)MBCSpd** = according to converter setting
Comms settings: *RS485(1)conv.* / *RS485(2)conv.** = according the converter setting

* Second RS232/485 port available only in IG-NTC/EEC and IS-NT.