InteliGen/InteliSys-NT

ComAp
LEADER IN GEN–SET
COMMUNICATION SOLUTION

December 2009

COMMUNICATION GUIDE

Written by Ladislav Kadaník, Martin Málek
Copyright © 2008 ComAp s.r.o.

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

General Information

Communication is a very wide topic. From the ComAp control systems user point of view we have to distinguish at least following levels of communication:

Communication between Controller and Peripherals, between Controllers

Communication between the controller and its peripherals (such as IGS-PTM, IS-AIN8…) is described in the controller User guides.
Communication between controllers for multiple applications is also described in the User guide.
Communication between controllers and ECUs is described in Engines with ECU&Comap controllers.pdf.

For IG/IS-NT:
IGS-NT-MINT-x.y.pdf
IGS-NT-Application guide-x.y.pdf

x.y stands for controller firmware version

Communication features of the classic line controllers – InteliSys and InteliGen can be found in the InteliCommunicationGuide-February05.pdf document.

Communication between Controller(s) and Superior, Service or Monitoring System (Usually PC)

Communication between controller(s) and superior, service or monitoring system (usually PC) is described in this Communication guide.

There are following types of communication between controller(s) and superior system:
- Local Connection
- Remote Connection
- Modem Connection
- Internet Ethernet Connection
- Internet Dial-up Connection
- Active Call
- Active SMS
- Active e-mail
- Terminal connection

There is following communication option available for the user SW designers and third party software:
- Modbus Connection

How to Use Different Character Sets

Available character sets

Character set	IG-EE IG-NT	IG-EE GC IG-NT GC	IG-DISPLAY	IG-DISPLAY GC	IS-DISPLAY	IV-DISPLAY*
Standard character set + West European languages + East European languages (incl. Turkish, Russian)	✓	✓	✓	✓	✓	✓
Support of graphical languages e.g. Chinese	X	✓	X	✓	✓	✓

Only Chinese available

*IV-DISPLAY (InteliVision) is the new generation display unit for ComAp InteliGen NT / InteliSys NT (IGS-NT) controllers. More information about IV-DISPLAY you can find in InteliVision Reference Guide.

How to change character sets in IG/IS-NT

The controllers without Graphical Characters option support all mentioned character sets without any software or hardware modifications. It is only necessary to create an appropriate dictionary and download it into the controller. Then it is possible to change language from the controller front panel. See GenConfig and controller User manuals for details.

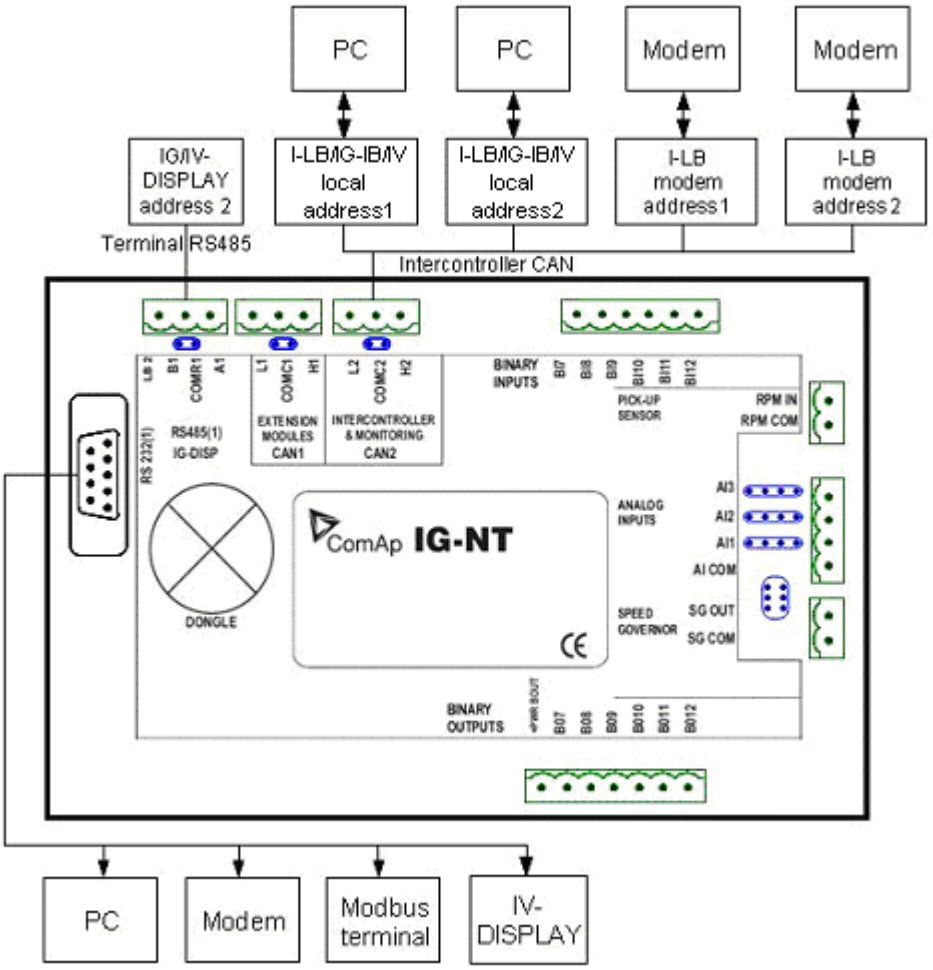
For support of graphical languages (e.g. Chinese, Japanese, Korean) it is necessary to order a “GC” modification of the controller. The way of adding the language is then the same as for standard controllers. IS-Display originally supports graphical character sets so there is no “GC” modification available.

Controllers Communication Capabilities

IG-NT, IG-EE

The following diagram shows communication capabilities of IG-NT/IG-EE controller in full configuration.

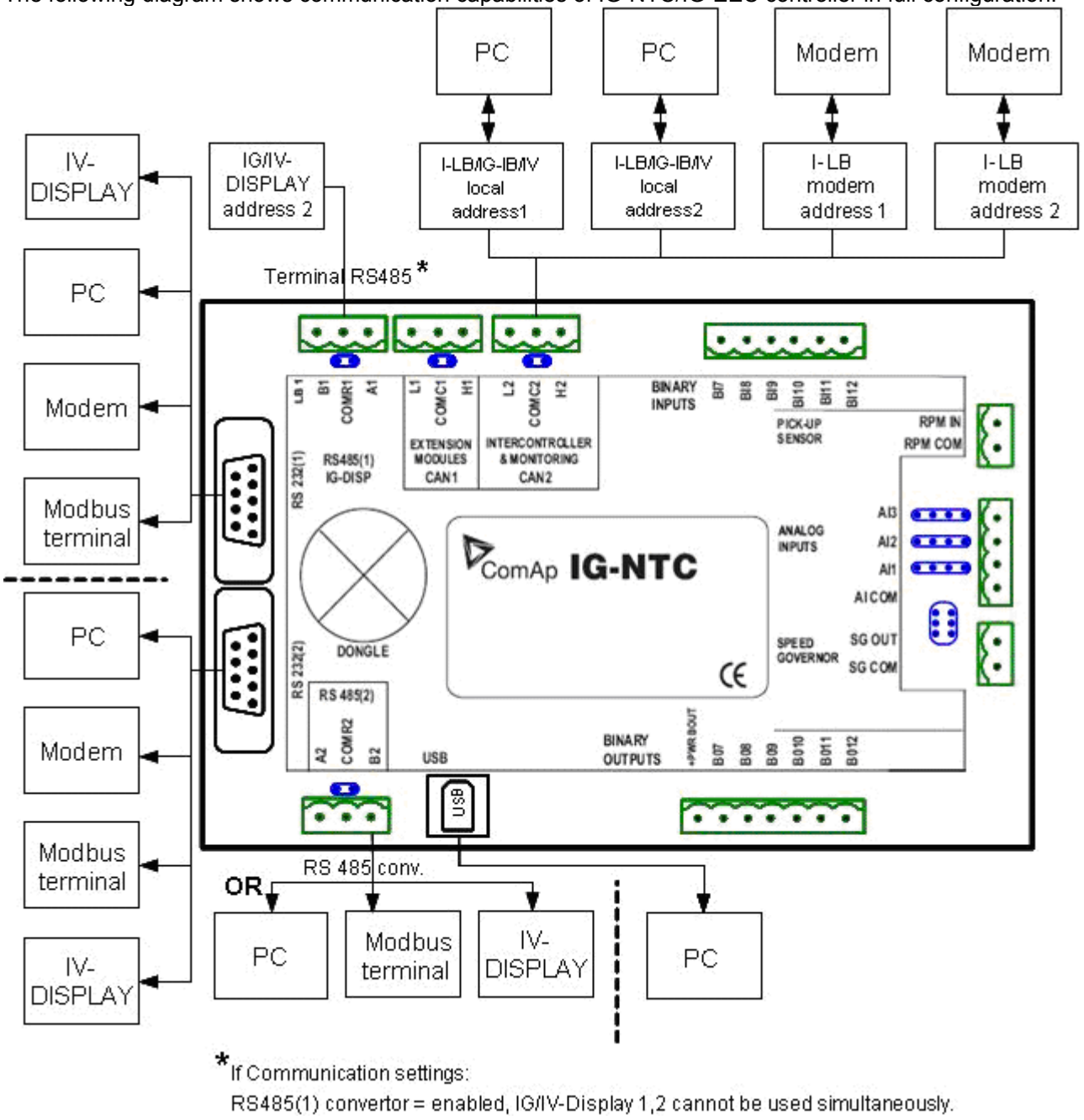
IV-DISPLAY (InteliVision) is the new generation display unit for ComAp InteliGen NT / InteliSys NT (IGS-NT) or InteliDrive (ID) controllers. More information about IV-DISPLAY you can find in InteliVision Reference Guide. See pictures below:



Hint:
Used CAN addresses for modules connected to CAN2 you can see in Addresses of Modules on CAN2 (IG/IS-NT, ID) chapter.

IG-NTC, IG-EEC

The following diagram shows communication capabilities of IG-NTC/IG-EEC controller in full configuration.

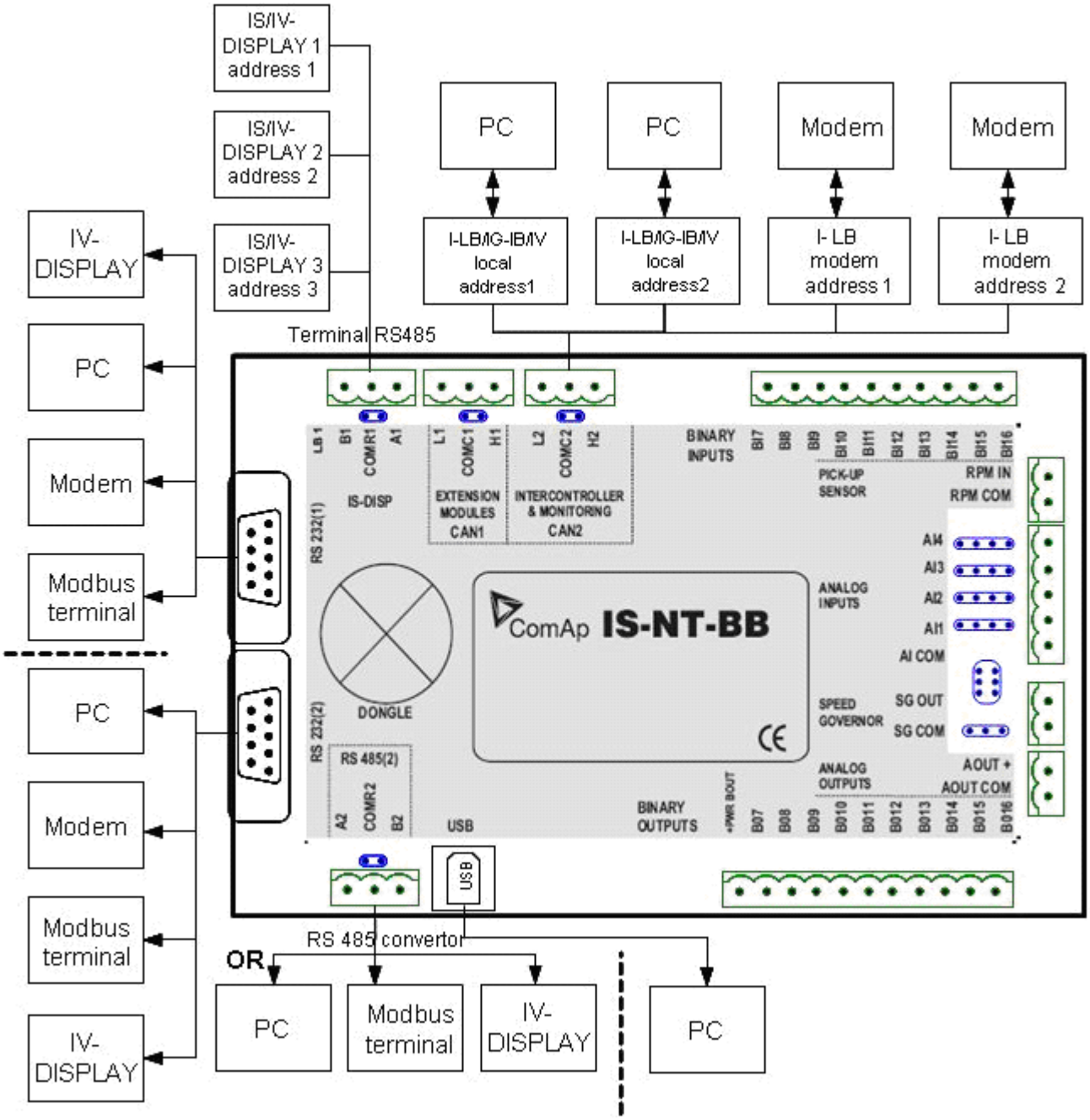


Hint:

Used CAN addresses for modules connected to CAN2 you can see in Addresses of Modules on CAN2 (IG/IS-NT, ID) chapter.

IS-NT

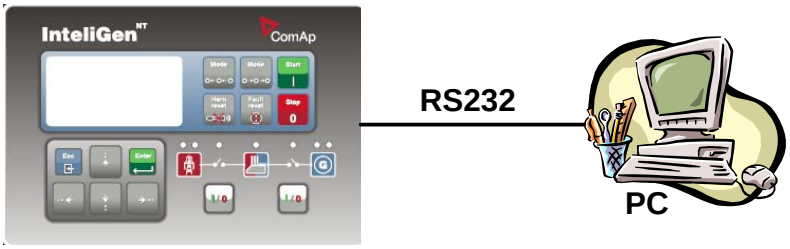
The following diagram shows communication capabilities of IS-NT controller in full configuration.



Hint:

Used CAN addresses for modules connected to CAN2 you can see in Addresses of Modules on CAN2 (IG/IS-NT, ID) chapter.

Local Connection – RS232



Controllers

Controllers	IG-NT/ IG-EE	IG-NTC/ IG-EEC	IS-NT
Connection applicable	YES	YES	YES
Available ports	RS232(1)	RS232(1), RS232(2)	RS232(1), RS232(2)

Controller setup

(**Basic settings/Comms settings** group)

Controllers	IG-NT/IG-EE/ IG-NTC/IG-EEC/IS-NT
ComAp protocol	RS232(1,2) mode = DIRECT
Modbus protocol	RS232(1,2) mode = MODBUS-DIRECT

Equipment

	Equipment needed
Controller side	-
Connection	RS232 cable up to 10m
PC side	PC with RS232

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	WinScope	Third party SW
Applicable	YES	YES	YES	YES (MODBUS)

For software descriptions see appropriate manuals – GenConfig-x.y.pdf, InteliMonitor-x.y.pdf.

PC Software features (GenConfig, InteliMonitor)

	On-line Gen-set control	Controller configuration	Read Gen-set values	Read history	Firmware programming	Archive files upload/download	Values time trends
GenConfig	-	X	-	-	X	X	-
InteliMonitor	X	-	X	X	-	-	-

Connection speed

	IG/IS-NT
Connection speed	Up to 57600 bps

RS232 cable

It is recommended to use standard Null-modem cable for local connection between controller and PC, although the three wires (TxD, RxD, GND) RS 232 connection is enough for direct controller to PC communication:

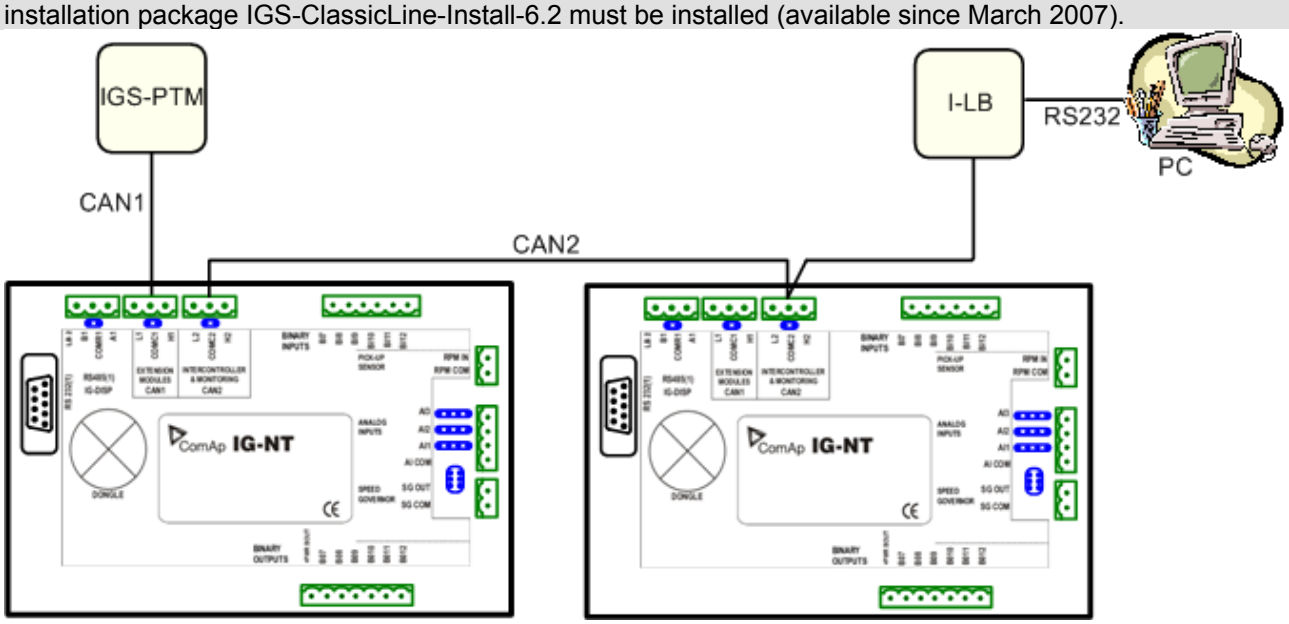
Controller connector D-SUB9 female	PC RS 232 connector D-SUB9 female
2 ----------	3 TxD
3 ----------	2 RxD
5 ----------	5 GND

Local Connection for Multiple Applications

IG/IS-NT CAN bus connection

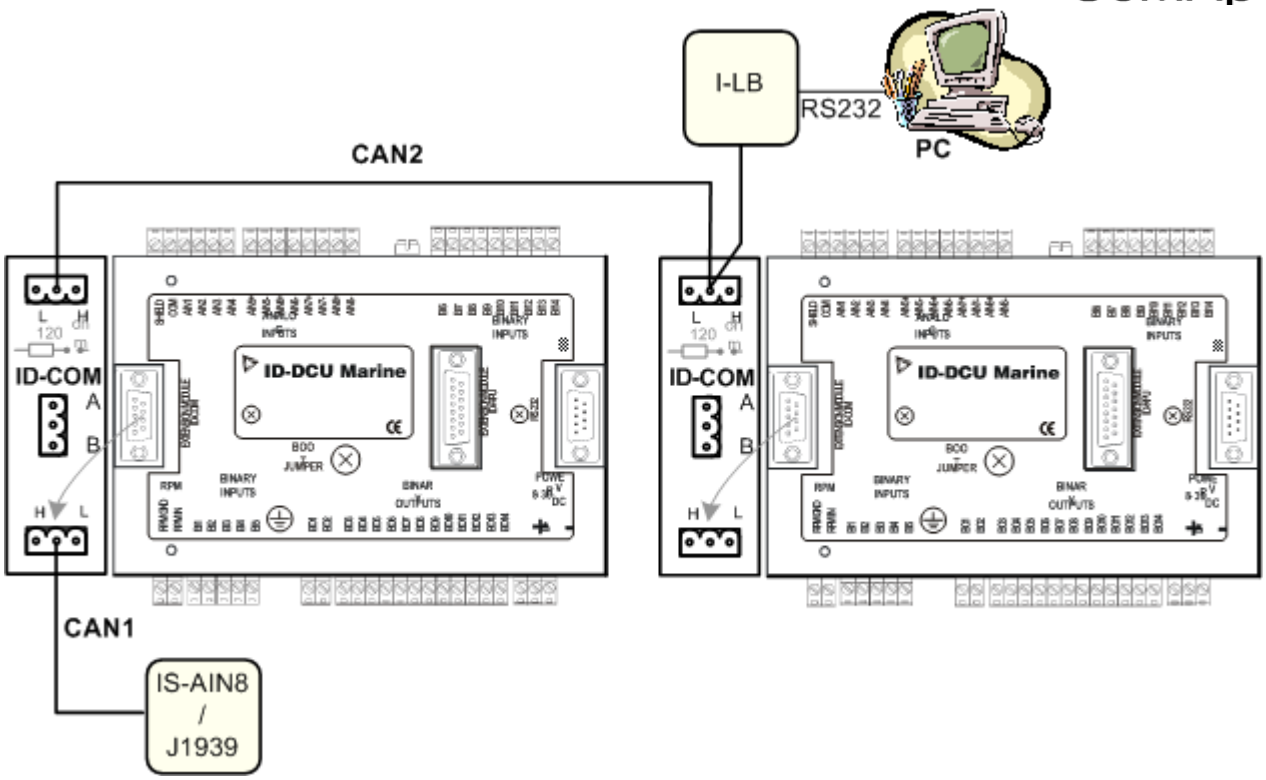
- For communication with a group of IG/IS-NT controllers connect the controllers via CAN bus (CAN2) and connect I-LB unit to this CAN bus.
- CAN bus terminals on IG/IS-NT are active all the time, not depend on Dongle.

Hint:
The IG-MU module can be used to cover most of the I-LB functionality but with slower performance. Otherwise the I-LB module is also compatible with the classic line controllers InteliSys and InteliGen but the installation package IGS-ClassicLine-Install-6.2 must be installed (available since March 2007).



There are two separate CAN bus lines on the controller. One for connection between controllers (CAN2), the second one (CAN1) is for external IGS-PTM, IS-AIN8, IS-BIN8/16 or IGL-RA15 modules or ECU connection.

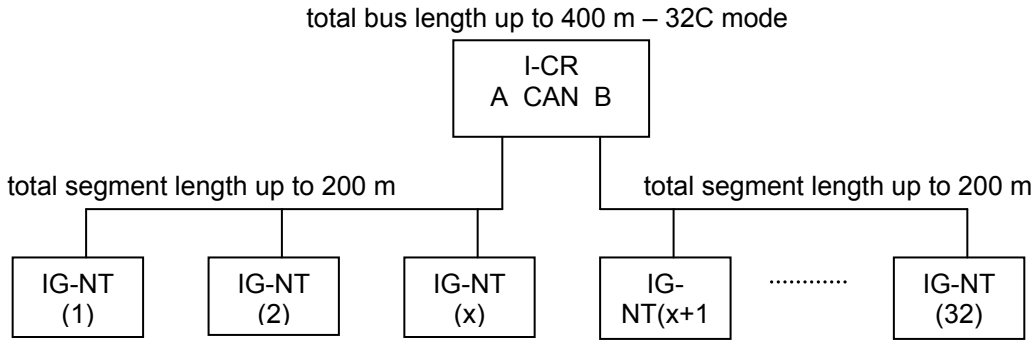
CAN bus length is limited to 200 meters for **Comms settings**:*CAN bus mode* = 32C (communication speed is 250kbps) or to 900 meters for **Comms settings**:*CAN bus mode* = 8C (communication speed is 50kbps).



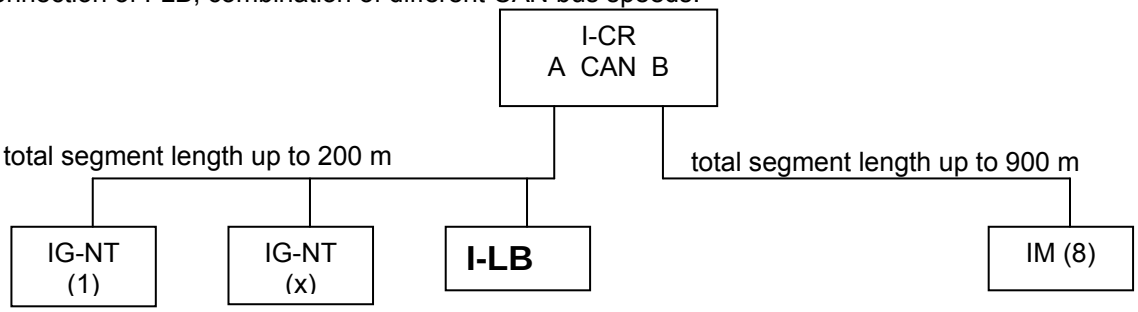
I-CR Module for CAN Bus Extension

If the distance between units is too high to fit into the 200 m limit (or 900 m for 8 controllers), CAN repeater module (I-CR) can be used to extend it.

Typical case – in line extension:



Connection of I-LB, combination of different CAN bus speeds:



This connection allows PC communication to all controllers in the system (e.g. via InteliMonitor), including a distant InteliMains unit.

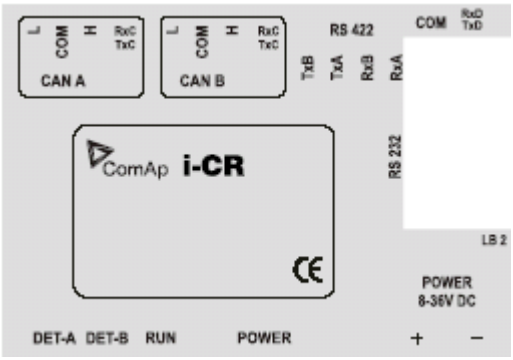
I-CR module functions:

- Intercontroller CAN bus extension (one or more I-CR modules can be used).
- Intercontroller CAN bus bus-tie bridging – makes groups of controllers in segments A and B "invisible" one for another depending on bus-tie breaker state, keeping the PC communication (I-LB, IG-IB) in function for all controllers.
- Peripheral CAN bus extension

I-CR configuration jumpers:

P2 – Forces 250 kbps mode (32C) on CAN A, otherwise speed autodetection is used.
P3 – Forces 250 kbps mode (32C) on CAN B, otherwise speed autodetection is used.
P4 – Activates Filter mode (bus-tie bridging).
P5 – Forces alternate controller address 3 for bus-tie status reading (default controller address is 4).
P10 – If "H" network configuration used (two I-CR units), it must be switched to RS-422 mode.

I-CR indication and diagnostic LEDs:



LED	State	Function
RUN	Lights	Firmware is OK and running
	Flashes	Firmware corrupted (periodic Watchdog reset)
DET-A	Lights	CAN A communication is running
	Flashes	CAN A communication speed autodetection is in progress (I-CR automatically enters this mode if no CAN message received for 2 s)
DET-B	Lights	CAN B communication is running
	Flashes	CAN B communication speed autodetection is in progress (I-CR automatically enters this mode if no CAN message received for 2 s)
CAN-A (TxC, RxC)	Flashes	CAN A interface activity
CAN-B (TxC, RxC)	Flashes	CAN B interface activity
COM (TxD, RxD)	Flashes	RS232/422 interface activity

For more detailed information about I-CR, see the Application sheet "**Extending the CAN bus**" or IGS-NT-x.y-Installation guide.pdf..

CAN bus terminating 120 ohms resistors availability

Following communication or extension modules connected to the CAN bus have internal 120 ohm terminating resistor that can be connected by jumper setting.

Module	120 ohm resistor jumper
I-LB	on CAN, RS485
IG-IB	No
ID-COM	on CAN2
IS-AIN8	No
IS-BIN16/8	No
IGS-PTM	Yes
IGL-RA15	No

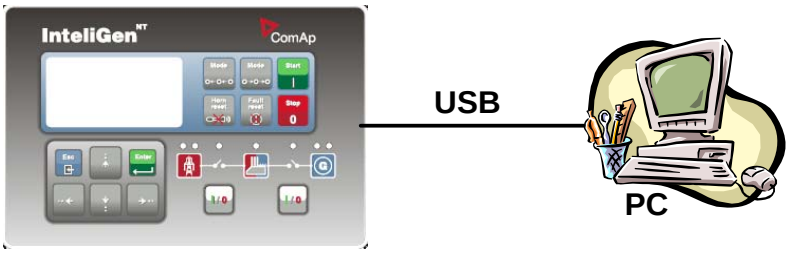
I-CB	Yes
I-CR	Yes

Hint:

CAN bus has to be terminated at both ends.

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-2.4-Installation guide.pdf.

Local Connection – USB



Controllers

Controllers	IG-NT/ IG-EE	IG-NTC/ IG-EEC	IS-NT
Connection applicable	-	YES	YES
Available ports	-	USB	USB

* Available as option (IL-NT-S-USB module)

Equipment

	Equipment needed
Controller side	-
Connection	USB cable A-B
PC side	USB port

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	WinScope	Third party SW
Applicable	YES	YES	YES	YES

PC Software features (GenConfig, InteliMonitor)

	On-line Genset control	Controller configuration	Read Genset values	Read history	Firmware programming	Archive files upload/download	Values time trends
GenConfig	-	X	-	-	X	X	-
InteliMonitor	X	-	X	X	-	-	-
WinScope	-	-	X	-	-	-	X

Connection speed

Up to 115 200 bps

USB cable

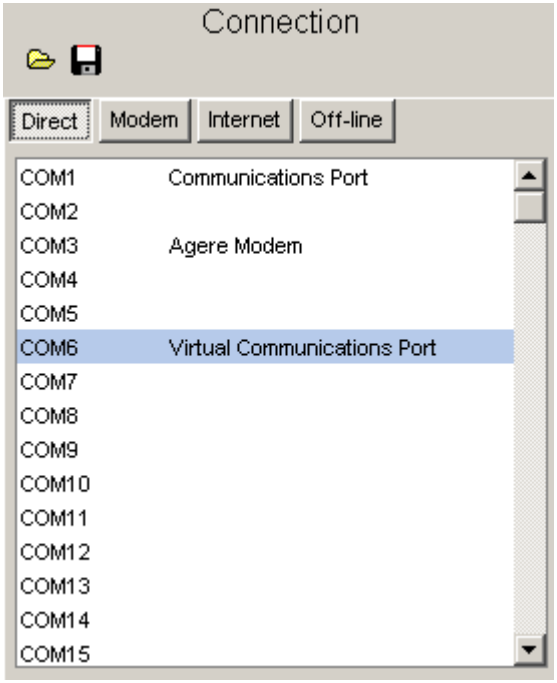
Use standard USB A-B cable (distance up to 5 meters).

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

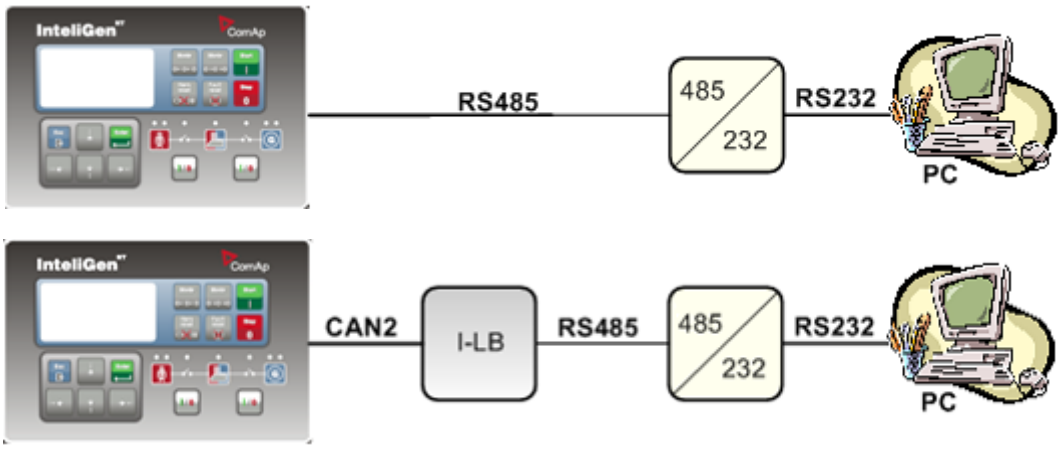
! IMPORTANT !

To use USB connection it is necessary to install drivers on your PC.
The drivers can be downloaded from the website http://www.ftdichip.com/Drivers/VCP.htm.
Download the driver for your operating system and follow the enclosed instructions.

After successful installation of the driver and connection of the controller or I-LB+ to the PC the new Virtual Communications Port appears in ComAp PC tools and it is possible to open connection via USB.



Remote Connection



RS485 communication line has to be terminated by 120 ohms resistors on both ends. Follow RS485 converter user manual.
I-LB has integrated this resistor – use "485- 120Ω" jumper.

Multiple controllers wiring with RS485, see Modbus connection chapter.

I-LB unit must be switched to RS485 mode for remote direct connection by "DIRECT/MODEM" and "RS485" jumper.

Hint:
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, see table.
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.

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-2.4-Installation guide.pdf.

Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES
Available ports	RS485(1)	RS485(1) RS485(2)	RS232(1)* RS485(2)

* with external converter only

Equipment

	Equipment needed
Controller side	- or Converter to RS485 or I-LB unit
Connection	Twisted pair, length up to 1 km
PC side	Converter to RS485

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	WinScope	Third party SW
Applicable	YES	YES	YES	YES (MODBUS)

PC Software features (GenConfig, InteliMonitor)

	On-line Genset control	Controller configuration	Read Genset values	Read history	Firmware programming	Archive files upload/download	Values time trends
GenConfig	-	X	-	-	X	X	-
InteliMonitor	X	-	X	X	-	-	-
WinScope	-	-	X	-	-	-	X

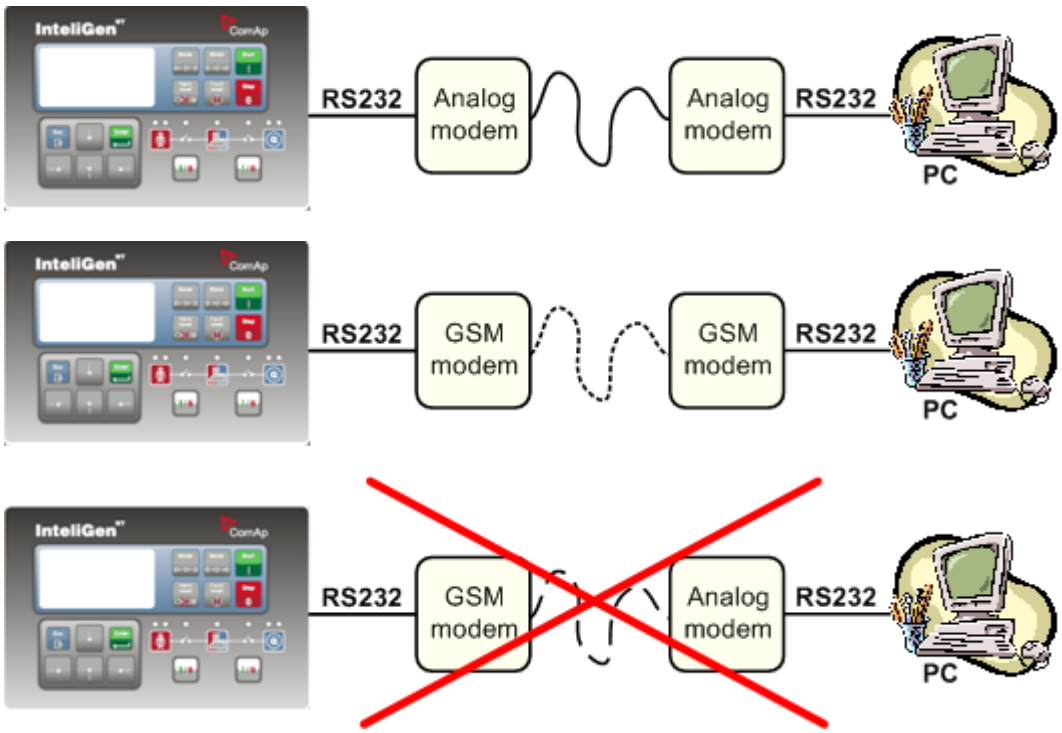
Hint:

It is possible to use WinScope with I-LB local connection, i.e. I-LB connected to multiple controllers via CAN bus and via RS232/RS485/USB to PC.

Connection speed

	IG/IS-NT
Connection speed	Up to 57600 bps

Modem Connection



Hint:

Usage of Analog and GSM modems (generally different modem types) is not recommended due to possible incompatibility.

Controllers

Controllers	IG-NT/ IG-EE	IG-NTC/ IG-EEC	IS-NT
Connection applicable	YES	YES	YES
Available ports	RS232(1)	RS232(1), RS232(2)	RS232(1), RS232(2)

Controller setup

(**Basic settings/Comms settings** group)

Controllers	IG-NT/IG-EE/IG-NTC/IG-EEC/IS-NT
ComAp protocol	RS232(1,2) mode = MODEM (HW) / MODEM (SW)
Modbus protocol	RS232(1,2) mode = MODBUS-MDM(HW)

Equipment

	Equipment needed
Controller side	Analog, ISDN or GSM modem
Connection	Phone line or GSM
PC side	Analog, ISDN or GSM modem

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	Third party SW
Applicable	YES	YES	YES (MODBUS)

PC Software features (GenConfig, InteliMonitor)

	On-line Genset control	Controller configuration	Read Genset values	Read history	Firmware programming	Archive files upload/downl oad
GenConfig	-	X	-	-	X*	X
InteliMonitor	X	-	X	X	-	-

*Possible, but not recommended due to risk of crash of programming if the communication is lost.

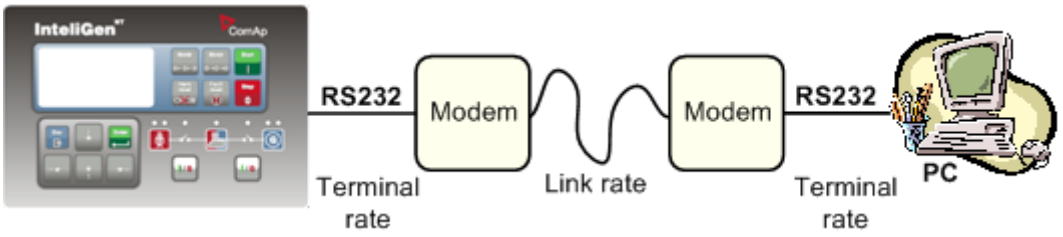
Connection speed

(limited by modem used)

	IG/IS-NT
Connection speed	Up to 57600 bps

Data Transfer Rates

Real data transfer rate is limited by the lowest rate in connection, mostly it is limited by Link rate.



Following table gives examples of data transfer rates. These values can differ case to case and depend on real link conditions and modem type. Data transfer rate was increased from DDE server version 2.9.

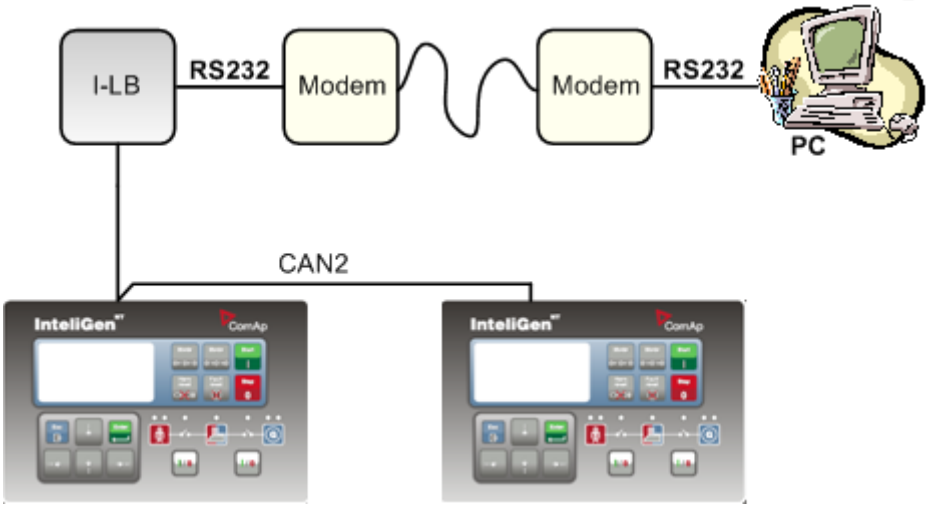
	Controller Terminal rate Bits per second	**Real link rate*** Bytes per second	PC Terminal rate Bits per second
Analog modem	19200	300 up to 1900	Depends on PC configuration
GSM modem	9600	80 up to 900	

Hint:

* Real link rate is visible in DDE server window.

Using Local Bridge I-LB

I-LB has to be connected to modem via standard modem cable (full RS232) where the DSR (Data Set Ready) signal detects modem presence (when MODEM (HW) type selected). Three-wire RS232 cable (TxD, RxD, GND) can be used (e.g. for GSM modems) when MODEM (SW) type selected.



Hint:
The IG-MU module can be used to cover most of the I-LB functionality but with slower performance. Otherwise the I-LB module is also compatible with the classic line controllers InteliSys and InteliGen but the installation package IGS-ClassicLine-Install-6.2 must be installed (available since March 2007).

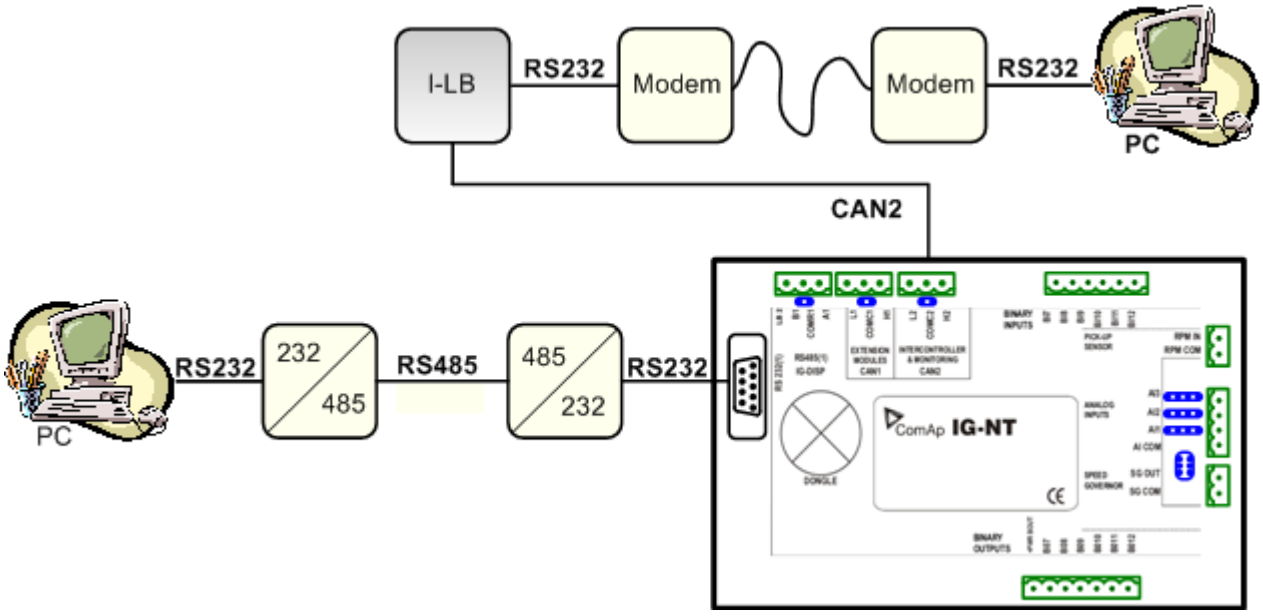
Important setpoint in the controller

Comms settings: *Contr addr* [1 to 32]
Controller identification number. Each gen-set in the group has to have its own unique number. Default value is 1.

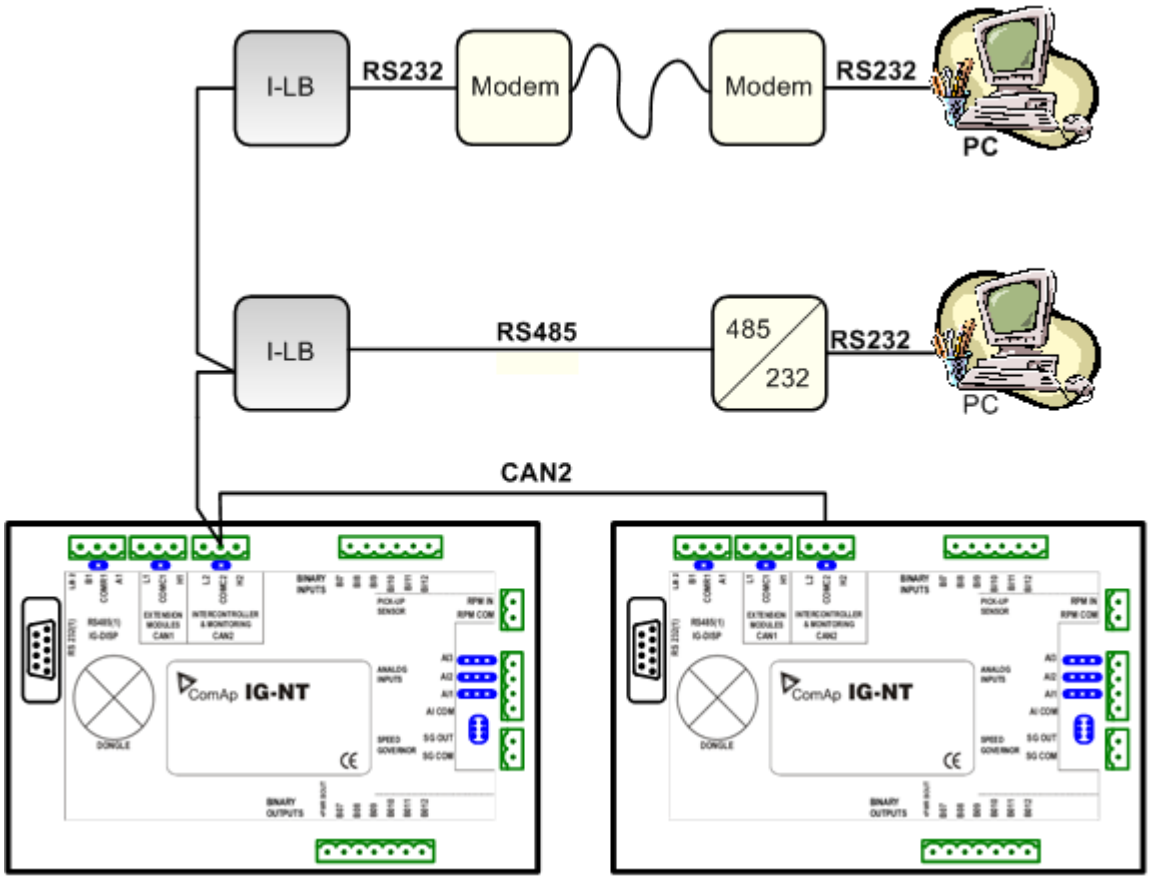
Hint:
When opening Direct or Modem connection to single controller , the *Controller address* has to correspond to PC SW communication setup setting.

Combined Communication - Remote and Modem

Option1:



Option2:

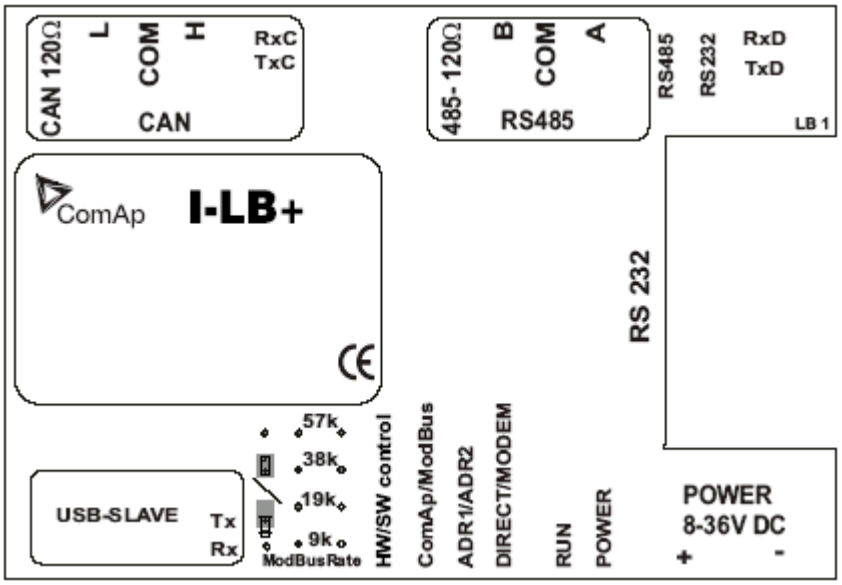


Combined Communication – I-LB with Maximum Configuration

There are more separate internal channels for NT controller connection which can operate at once (all via CAN2):

- Local connection 1 (I-LB local address 1)
- Local connection 2 (I-LB local address 2)
- Modem connection 1 (I-LB modem address 1)
- Modem connection 2 (I-LB modem address 2)

It is necessary to select which connection type(s) a particular I-LB is using. In I-LB+ version, there is additional USB slave port, which can be used as a local connection **only**, together with RS232-modem connection. See Available combinations of I-LB modules table below.



Jumper selection tree

ComAp / ModBus – selects between ComAp PC tools (InteliMonitor, WinScope, ...) and third party PC SW for monitoring:

- **ComAp**
  - **Direct / Modem** – selects between direct connection (via RS232 or RS485) and modem connection type
    - **DIRECT**
      - **RS232 / RS485** – selection of serial communication type
      - **ADR1 / ADR2** – selection between two available local communication channels; if I-LB+ is used, the USB communication automatically occupies the other channel
    - **MODEM**
      - **HW / SW control** – selection between modems with full interface
      - **ADR1 / ADR2** – selection between two available modem communication channels; IG/IS-NT controllers only, in ID the secondary modem channel not available
      - Setting **RS232 / RS485** jumper to RS232 position is obligatory

- **ModBus (not available at USB port of I-LB+, USB port always works in ComAp mode)**
  - **Direct / Modem** – selects between direct connection (via RS232 or RS485) and modem connection type
    - **DIRECT**
      - **RS232 / RS485** – selection of serial communication type
      - **ADR1 / ADR2** – selection between two available local communication channels; if I-LB+ is used, the USB communication automatically occupies the other channel
    - **MODEM**
      - **ADR1 / ADR2** – selection between two available modem communication channels; IG/IS-NT controllers only, in ID the secondary modem channel not available
      - Setting **HW / SW control** has no influence; a modem with HW control is always expected in this mode
  - **ModBus Rate** (9600 / 19200 / 38400 / 57600 bps) – selects the communication speed when ModBus protocol is selected, no matter if in Direct or Modem mode

Default jumper setting:

P1	CAN 120R	Opened
P2	RS485 120R	Opened
P3	1-2 RS232 / 2-3 RS485	1-2 (active RS232)
P13	Modbus rate	Opened
P14	Modbus rate	Opened (=9600 bps)
P15	HW/SW control	Opened (HW control)
P16	Comap/Modbus	Opened (Comap protocol)
P17	ADR1/ADR2	Opened (ADR1)
P18	Direct/Modem	Opened (Direct)

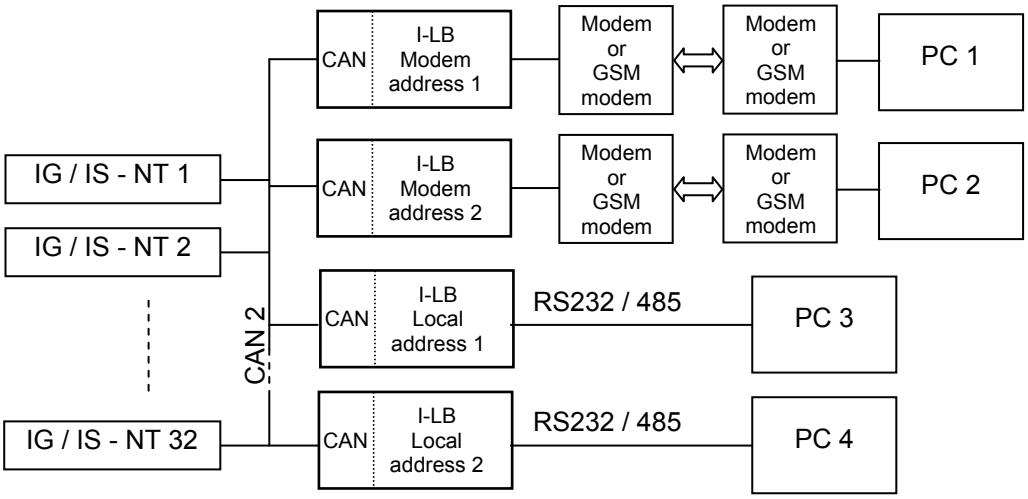
Available combinations of I-LB modules

There are four of communication channels available on the CAN2 link at the same time. See the connection possibilities in the table below:

I-LB/I-LB+ module		**DIRECT/ MODEM jumper**	**ADR1/ ADR2 jumper**	**channel 1** (local con. 1)	**channel 2** (local con. 2)	**channel 3** (modem con. 1)	**channel 4 (*)** (modem con. 2)
Connection possibilities of only I-LB or I-LB+ in following eight examples							
1.	I-LB	DIRECT	ADR1	RS232/485	-	-	-
2.	I-LB	DIRECT	ADR2	-	RS232/485	-	-
3.	I-LB	MODEM	ADR1	-	-	RS232-modem	-
4.	I-LB	MODEM	ADR2	-	-	-	RS232-modem
5.	I-LB+	DIRECT	ADR1	RS232/485	USB	-	-
6.	I-LB+	DIRECT	ADR2	USB	RS232/485	-	-
7.	I-LB+	MODEM	ADR1	-	USB	RS232-modem	-
8.	I-LB+	MODEM	ADR2	USB	-	-	RS232-modem
Connection possibilities of both I-LB+ (**) or I-LB and I-LB+ couple in following four examples							
9.	I-LB	DIRECT	ADR1	RS232/485			-
	I-LB+	MODEM	ADR1		USB	RS232-modem	-
10.	I-LB	MODEM	ADR1			RS232-modem	-
	I-LB+	DIRECT	ADR1	RS232/485	USB		-
11.	I-LB	MODEM	ADR1	-	-	RS232-modem	-
	I-LB+	MODEM	ADR2	USB	-	-	RS232-modem
12.	I-LB+	MODEM	ADR1	-	USB	RS232-modem	-
	I-LB+	MODEM	ADR2	USB	-	-	RS232-modem

(*) – available for IG/IS-NT and IC-NT controllers only, not for ID, IG, IS, IL-NT
(**) – there can be max. two of I-LB+ on the CAN2 link.

Hint:
The IG-MU module can be used to cover most of the I-LB functionality but with slower performance. Otherwise the I-LB module is also compatible with the classic line controllers InteliSys and InteliGen but the installation package IGS-ClassicLine-Install-6.2 must be installed (available since March 2007).



Addresses of Modules on CAN2 (IG/IS-NT, ID, IC-NT)

Following CAN addresses are used for modules connected to CAN2 (intercontroller CAN bus). There cannot be more modules using the same address connected at the same time.
Addresses can be changed using jumpers, configuration program or from the display - refer to the corresponding chapter or reference guide for detailed description.

Real CAN2 Address	IG-MU	I-LB (local) (RS232/485)	I-LB (modem)	I-LB+ (USB)	IG-IB (IBConfig ≤ 1.5)	IG-IB (IBConfig ≥ 1.6)	IV-DISPLAY*	I-RD-CAN
122			addr. 2					
123	addr. 2	addr. 2		addr. 1	addr. 1	addr. 2	addr. 2	addr. 2
124	addr. 1	addr.1		addr. 2	addr. 2	addr. 1	addr. 1	addr. 1
125	modem		addr. 1					

*IV-DISPLAY (InteliVision) is the new generation display unit for ComAp InteliGen NT / InteliSys NT (IGS-NT) controllers. More information about IV-DISPLAY you can find in InteliVision Reference Guide.

Hint:
Please note that USB port is using its CAN address only if an external device is connected to the USB port of I-LB+. Make sure that other device (e.g. IG-IB) is not using the same CAN address as USB port of an I-LB+, because using USB port could interrupt CAN communication.

Hint:
Please note that adresses 1 and 2 (123, 124) are exchanged in versions IBConfig ≤ 1.5 and IBConfig ≥ 1.6 (see the table above).

IG-MU
Address 1 => CAN-address 124
Address 2 => CAN-address 123
When switched to modem mode it uses CAN-address 125.

I-LB Direct connection
Address 1 => CAN-address 124
Address 2 => CAN-address 123

I-LB Modem connection
Address 1 => CAN-address 125
Address 2 => CAN-address 122

I-LB+ when USB terminal is connected
Address 1 => CAN-address 123
Address 2 => CAN-address 124

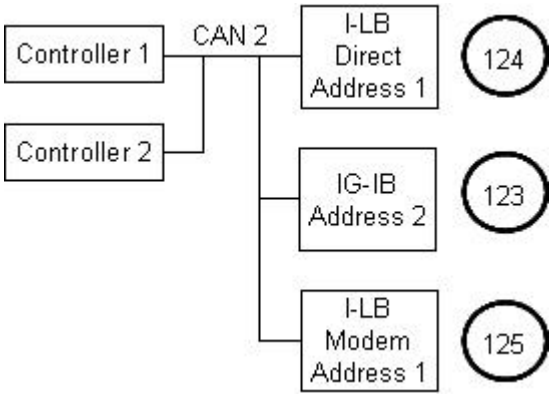
IG-IB
Address 1 => CAN-address 124
Address 2 => CAN-address 123

Hint:
These addresses are valid for IBConfig ≥ 1.6.

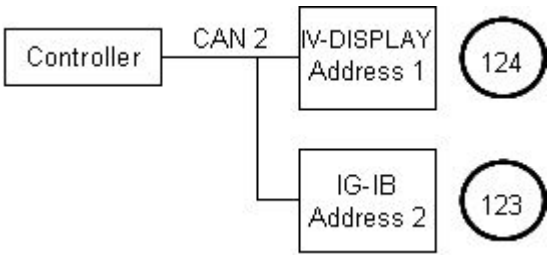
IV-DISPLAY
Address 1 => CAN-address 124
Address 2 => CAN-address 123

I-RD-CAN
Address 1 => CAN-address 124
Address 2 => CAN-address 123

Example 1:

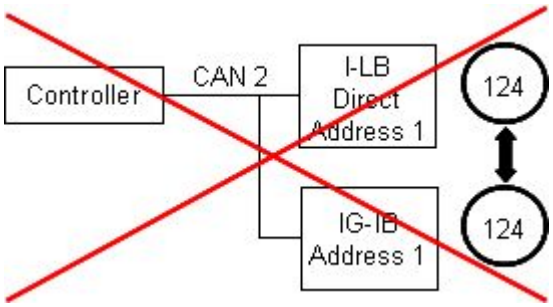


Example 2:



Hint:
In these examples IBConfig ≥ 1.6 is used for programming IG-IB.

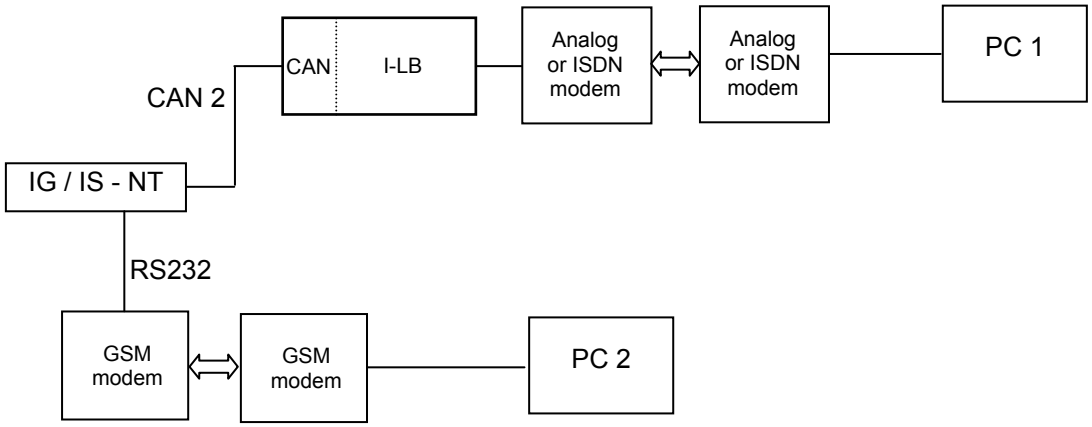
Example 3: Wrong connection – conflict of addresses:



Hint:
In this example IBConfig ≥ 1.6 is used for programming IG-IB.

Two Modem Types Connection

It is possible to connect two modems (e.g. Analog and GSM) to one controller at once to have two independent modem channels for DATA and GSM messages.
One modem is connected via I-LB – CAN bus, the second is connected directly to RS 232 controller interface.
When an active call is activated in the controller, the modem connected directly to the controller tries to call/send an SMS. If this is not successful, the Active call fail alarm appears. The second modem doesn't then try to substitute the first one.



GSM Modem Configuration

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. Ask your operator for this service. If it is not enabled, Gm_setup program which is part of controller installation package shows "Command failure" message at the end of the log.

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.

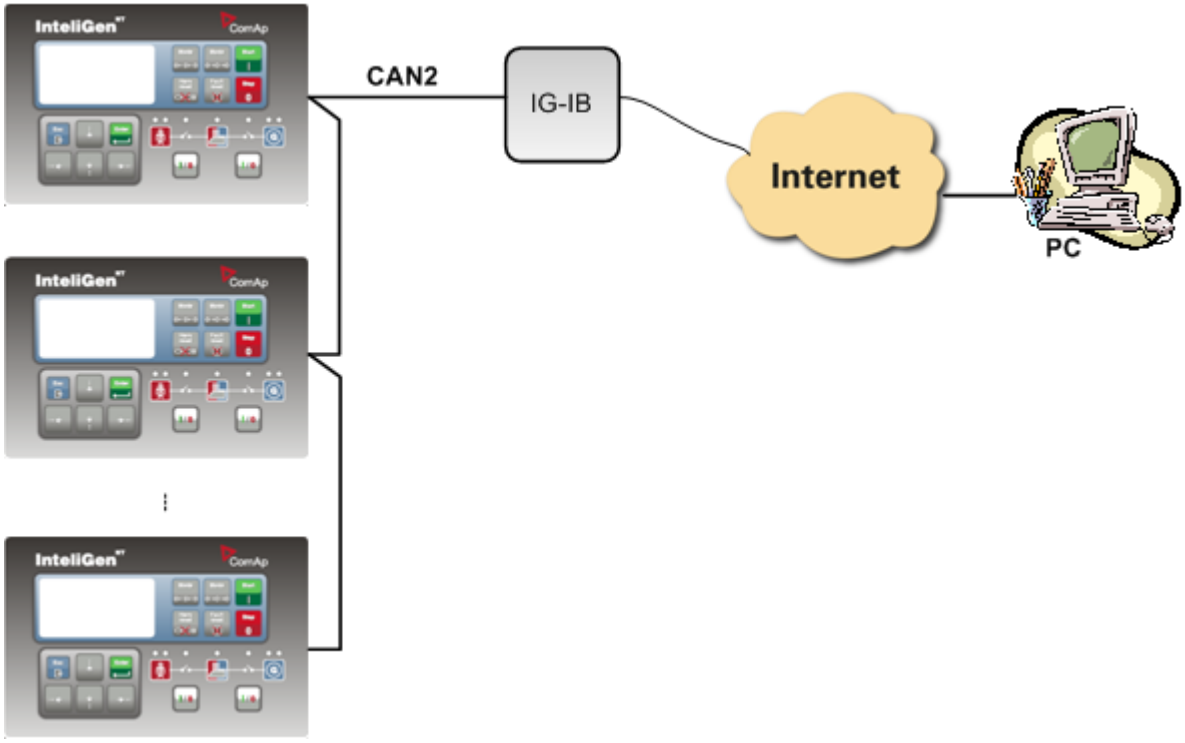
4. Enter the SIM card PIN after you are be asked.
5. Enter SMS center address after you will be asked (ask your operator for this number).
6. If the Gm_setup writes “Setup terminated successfully” the SIM card is configured for the communication with the controller.

Controller configuration

1. In Act.calls/SMS group of setpoints set:
   ENABLED to desired type of active call depending on what level of alarm should activate the call/SMS sending (this setting is not important if you don't want to use InteliMonitor/DriveMonitor in Active call mode (waiting for the active call from the controller) or sending SMS in case of an alarm)
   Acall-CHX-Type – choose type of active call (see a controller manual for more information)
   Acall-CHX-Addr - phone number of the SIM card in the modem connected to your PC (where the controller should call) or your email address
2. Connect the controller to the modem or I-LB that has the modem connected.

Internet Ethernet Connection

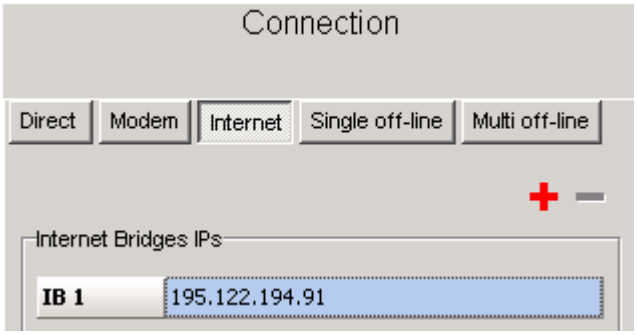
One IG-IB connection using CAN2



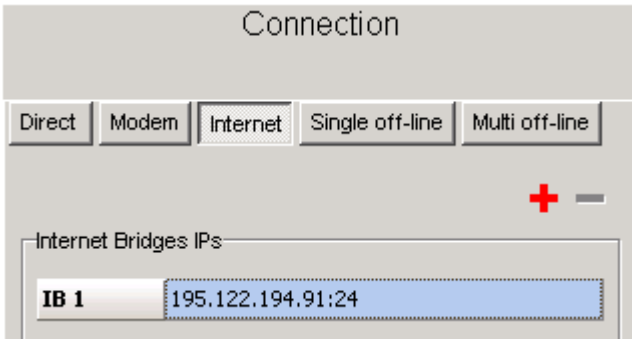
Up to 32 controllers can be monitored via one IG-IB. Response time of a system with this type of connection depends on number of controllers, higher number of controllers means slower system response time.

Use the next type of connection (Several IG-IB's connection using RS232) for faster communication with more than 10 controllers.

IP address of IG-IB needs to be set in InteliMonitor in this way if default port number 23 is used:

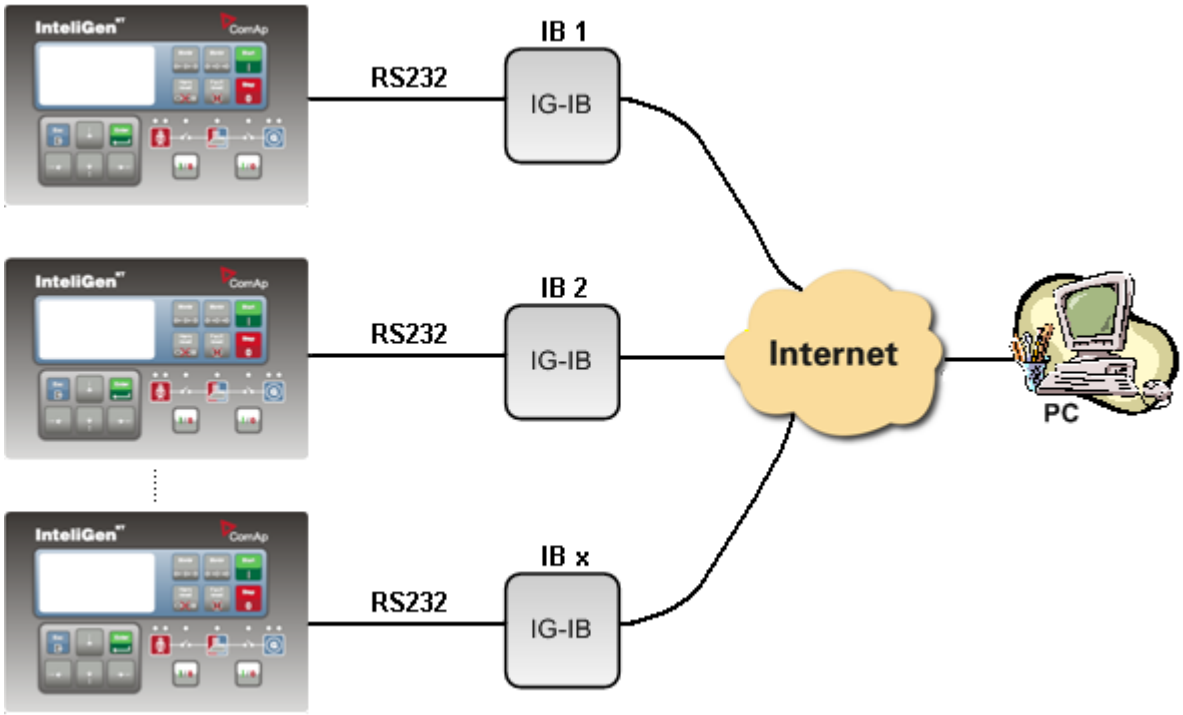


IP address of IG-IB needs to be set in InteliMonitor in this way if port number other than 23 is used:



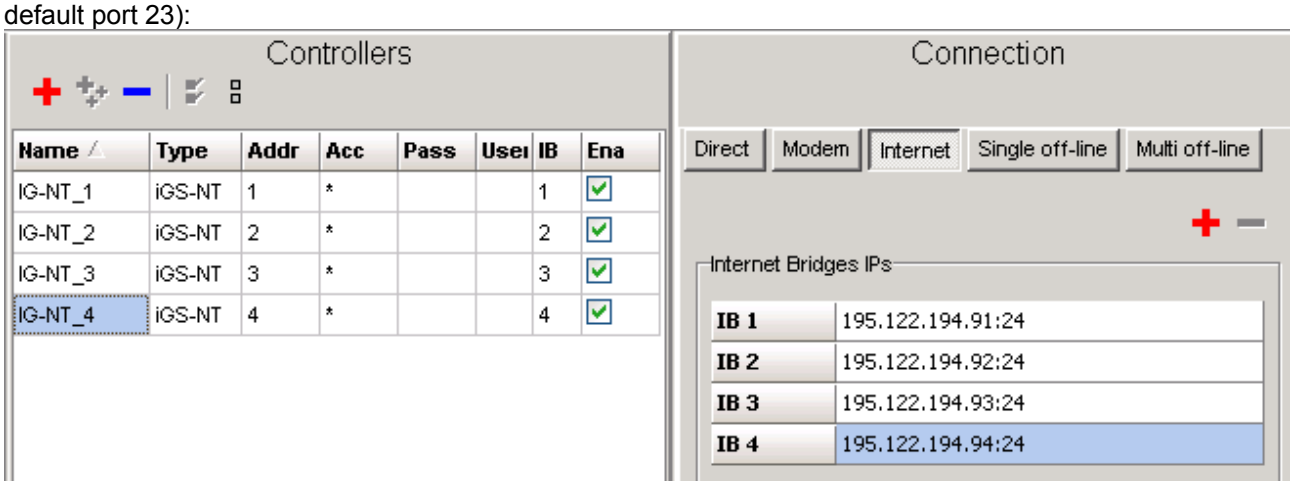
Internet Bridges IP's definition consist of two parts:
195.122.194.91 : 24
IP address Port number

Several IG-IB's connection using RS232



One IG-IB is connected to each controller via RS232 for faster communication. It means that up to 32 IG-IB's can be used on one site. Use this connection for communication with 10 or more controllers. Communication speed gain achieved with this connection is not significant for less than 10 controllers.

IP addresses of IG-IB's needs to be set in InteliMonitor in this way (if port number 24 is used instead of the default port 23):



Internet Bridges IP's definition consist of two parts:
195.122.194.91 : 24
IP address Port number

Controllers

Controllers	IG-NT/ IG-EE	IG-NTC/ IG-EEC	IS-NT
Connection applicable	YES	YES	YES
Available ports	CAN2*	CAN2*	CAN2*

* To connect IG-IB.

Equipment

	Equipment needed
Controller side	InternetBridge (IG-IB) unit configured for Ethernet connection (Ethernet connection firmware); Internet connection with “visible” IG-IB
Connection	Ethernet
PC side	Internet connection

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	InteliSupervisor	Third party SW
Applicable	YES	YES	YES*	NO

* on request

PC Software features (GenConfig, InteliMonitor)

	On-line Genset control	Controller configuration	Read Genset values	Read history	Firmware programming	Archive files upload/download
GenConfig	-	X	-	-	X*	X
InteliMonitor	X	-	X	X	-	-

* Possible, but not recommended due to risk of crash of programming if the communication is lost.

Connection speed

Up to 3000 Bps

Hint:

IG-IB communication is typically faster than modem:

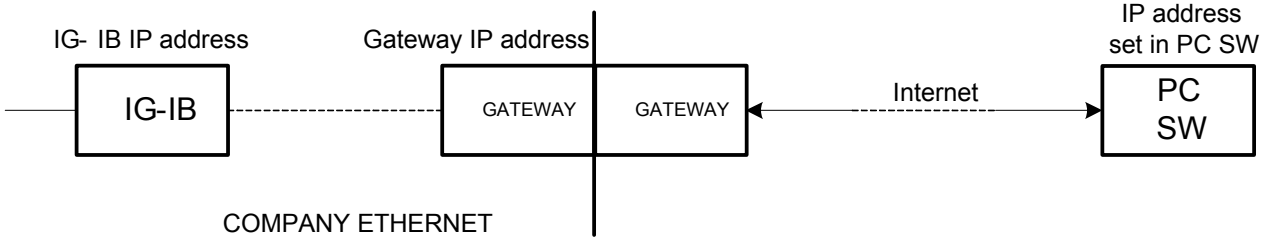
- when IG-IB is connected to controller via CAN and LAN - 3000 Bps
- when IG-IB is connected to controller via RS232 and LAN - 1200 Bps
- when IG-IB is connected to controller via CAN and Internet - ~500 Bps
- when IG-IB is connected to controller via RS232 and Internet - ~500 Bps (depends on the internet line throughput).

Connection from PC

For connection from PC see the manual of the PC program (InteliMonitor, etc.).
Open the Open/Select connection window and set:

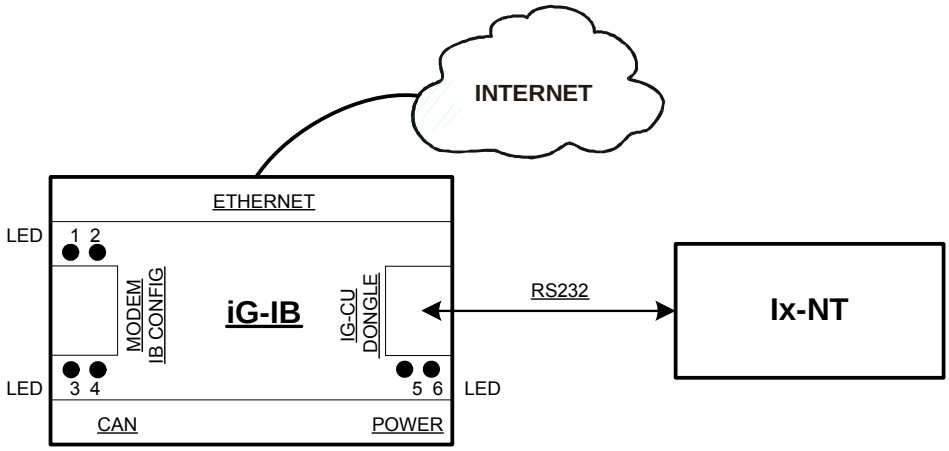
- Internet type of connection
- Controller address
- Access code
- IG-IB IP address

Note that IP address you set can be different from the IP address of iG-IB (when the iG-IB IP address is not public). It depends on gateway setting:



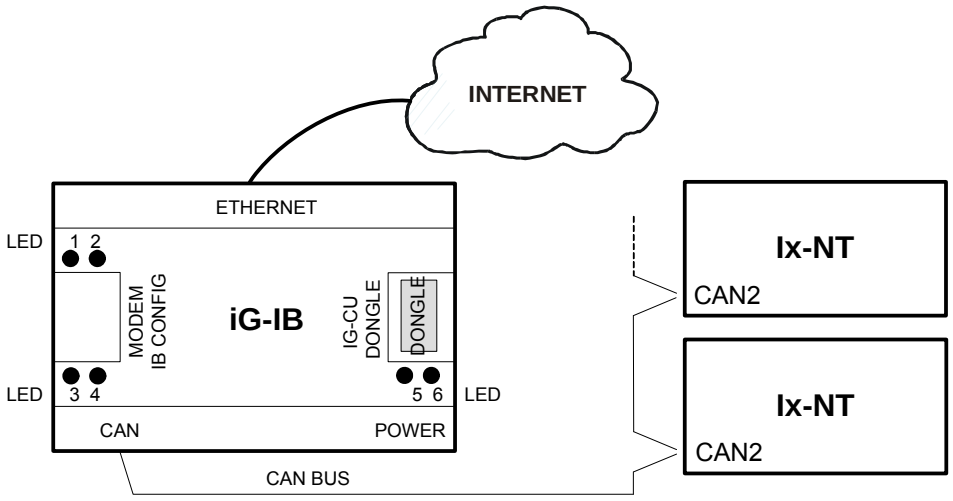
One controller - RS232 connection

In this case no dongle is necessary.
One unit could be also connected via CAN interface.

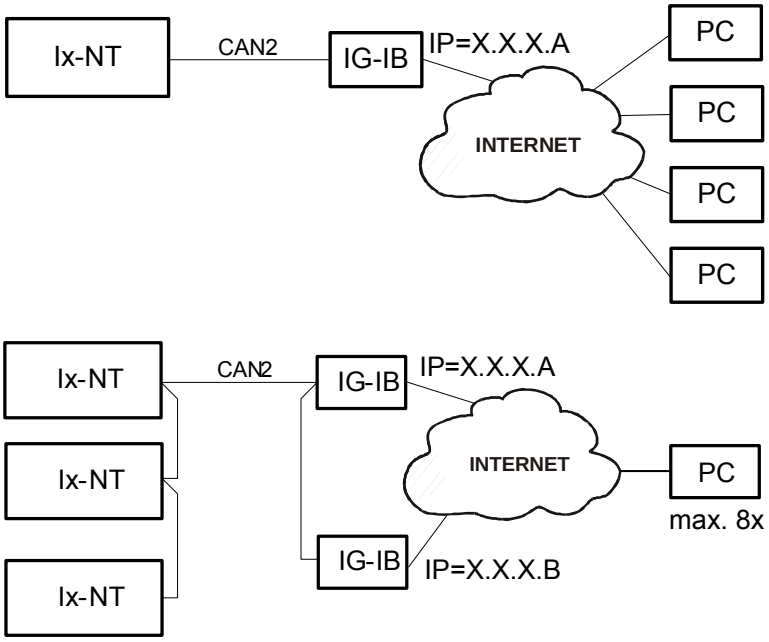


Two controllers – CAN / Ethernet connection

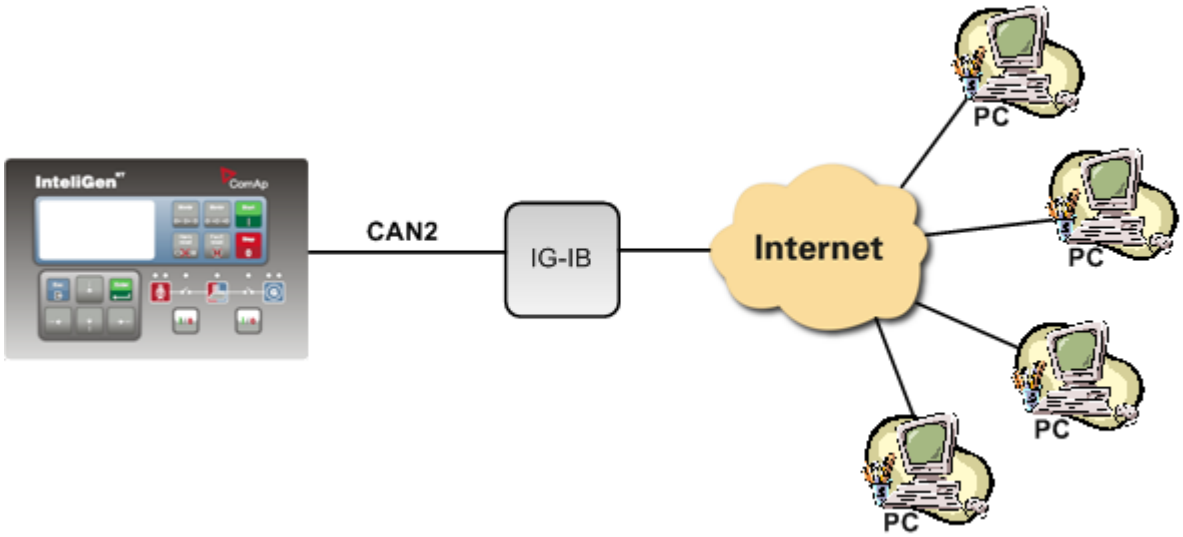
The dongle must be used. The type of IG-IB dongle limits number of accessible controllers (up to 32).
Dongles for up to 3, 7, 15 and 32 controllers are available. See also IG-IB Dongle



Max four PCs can be connected to one iG-IB at once



Internet Dialup Connection



Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES
Available ports	CAN2*	CAN2*	CAN2*

* To connect IG-IB.

Equipment

	Equipment needed
Controller side	InternetBridge (IG-IB) unit configured for Dialup connection (Dialup firmware); Dialup internet connection
Connection	Ethernet
PC side	Internet connection

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	InteliSupervisor	Third party SW
Applicable	YES	YES	YES*	NO

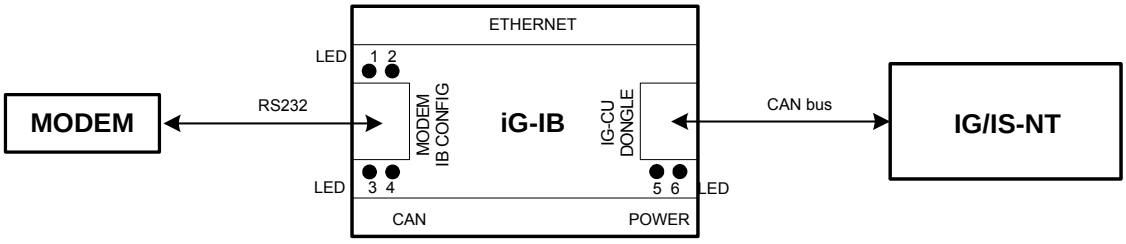
* on request

PC Software features (GenConfig, InteliMonitor)

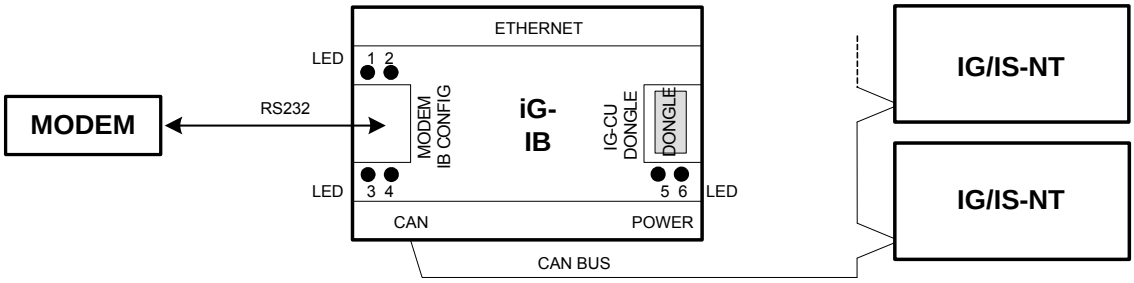
	On-line Genset control	Controller configuration	Read Genset values	Read history	Firmware programming	Archive files upload/download
GenConfig	-	X	-	-	X*	X
InteliMonitor	X	-	X	X	-	-

* Possible, but not recommended due to risk of crash of programming if the communication is lost.

One controller - RS232



More controllers - CAN



Active Call

Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES

Equipment

Same as for Modem connection.

Available software for IG/IS-NT

Software	GenConfig	InteliMonitor	Third party SW
Applicable	NO	YES	YES

Function

When active calls are activated for alarms on site (warning, shut-down…) the controller calls to the preselected telephone number and sends the ANT archive file.
Software (e.g. InteliMonitor) on the PC side must be running and waiting for active call.

Controller setting

For this type of active call set **Act. calls/SMS**: *AcallCH1(-3)-Type* = DATA, **Act. calls/SMS**: *AcallCH1(-3)-Addr* = telephone number.

Terminal attempt order for active call in IG/IS-NT

1) Analog/GSM/ISDN/CDMA modem connected to RS232(1)
2) Analog/GSM/ISDN/CDMA modem connected to RS232(2)*
3) Analog/GSM/ISDN/CDMA modem connected to I-LB (modem address 1)
4) Analog/GSM/ISDN/CDMA modem connected to I-LB (modem address 2)

* Available only in IG-NTC/EEC and IS-NT.

Active SMS

Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES

Equipment

	Equipment needed
side	GSM Modem or I-LB + GSM Modem
Connection	GSM
PC side	GSM Mobile Phone

Function

When SMS active calls are activated for alarms on site (warning, shut-down…) the controller sends SMS message to the predefined GSM number.

Controller setting

For this type of active call set:
Act. Calls/SMS: *AcallCH1(-3)-Type* = SMS
Act. calls/SMS: *AcallCH1(-3)-Addr* = mobil phone number

Terminal attempt order for SMS active call – IG/IS-NT

1) GSM / CDMA modem connected to RS232(1)
2) GSM / CDMA modem connected to RS232(2)*
3) GSM / CDMA modem connected to I-LB (modem address 1)
4) GSM / CDMA modem connected to I-LB (modem address 2)

* Available only in IG-NTC/EEC and IS-NT.

Example

SMS in format

```
#Gen-set name:AL=(Wrn PrimWater temp, !Emergency stop)
```

is sent in case that the primary water temperature exceeded the warning limit and Emergency stop input has been deactivated.

Hint:
For IG/IS-NT FW version 1.1 and lower and for ID controllers, character ASCII codes lower than $20 and higher than $7F are prohibited to appear in an SMS body. If they appear, the SMS sending may be blocked by GSM modem. I.e. those character codes are not allowed in gen-set name and all messages that can appear in an active SMS.
In IG/IS-NT FW version 2.0 and higher, these characters are replaced by character "_".

Active E-mail (SMS E-mail)

Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES

Equipment

	Equipment needed
Controller side	InternetBridge (IG-IB) unit configured for Ethernet connection or Dialup connection
Connection	Internet
PC side	e-mail message box (GSM mobile phone for SMS e-mail)

Function

When active e-mails are activated for alarms on site (warning, shut-down…) the controller sends e-mail message to the predefined e-mail address.

Controller setting

For this type of active call set:
Act. calls/SMS: *AcallCH1(-3)-Type* = E-MAIL (maximum length of email address is 31 characters)
Act. calls/SMS: *AcallCH1(-3)-Addr* = email address

Terminal attempt order for SMS e-mail active call – IG/IS-NT

1) IG-IB connected to RS232(1) and internet connection currently active[1]
2) IG-IB connected to CAN (address 2) and internet connection currently active[1]
3) IG-IB connected to CAN (adderss 1) and internet connection currently active[1]
4) IG-IB connected to RS232(2) and internet connection currently active[1]
5) IG-IB connected to RS232(1) and internet connection currently inactive[2]
6) IG-IB connected to CAN (address 2) and internet connection currently inactive[2]
7) IG-IB connected to CAN (adderss 1) and internet connection currently inactive[2]
8) IG-IB connected to RS232(2) and internet connection currently inactive[2]

[1] – For IG-IB connected to Internet via leased line – always active;
For IG-IB connected to Internet using dial-up connection – active only when IG-IB is connected to Internet provider.
[2] – Only for dial-up connection – when IG-IB is not connected to Internet provider.

Example of active e-mail (from controller)

Example of active E-mail for **Ethernet - LAN** connection

```
iG-IB
-----------------------------------------------
IP address:     192.168.1.10
Connection:     Ethernet LAN

Controller
-----------------------------------------------
Name:           InteliSys
Serial number:  06F20093
SW branch:      Standard
SW version:     2.7
Application:    SPtM
Appl. version:  2.7
Time:           15:21:06
Date:           13/06/03

Alarm list
-----------------------------------------------
!Sd SD 11

History events
-----------------------------------------------
  0  13/06/03  15:21:05  NotReady
 -1  13/06/03  15:21:02  Sd SD 11
 -2  13/06/03  15:20:56  Running
 -3  13/06/03  15:20:46  Started
 -4  13/06/03  15:20:44  Start
 -5  13/06/03  15:20:40  Ready
 -6  13/06/03  15:17:18  Passw3 entered
 -7  13/06/03  14:58:37  NotReady
 -8  13/06/03  14:58:37  PickupFault
 -9  13/06/03  14:58:37  Ready
-10  13/06/03  14:58:33  Stop
-11  13/06/03  14:58:03  Cooling
-12  13/06/03  14:58:03  RemControlUART
-13  13/06/03  14:52:11  Running
-14  13/06/03  14:52:01  Started
-15  13/06/03  14:51:59  Start
-16  13/06/03  14:51:59  RemControlUART
-17  13/06/03  14:37:27  Ready
-18  13/06/03  14:37:21  Switched On
-19  11/06/03  12:29:47  Ready
```

Example of active E-mail for **Dialup** connection

```
iG-IB
-----------------------------------------------
IP address:     192.168.1.10
Connection:     Dialup (until 10:52:05)
```

… the rest of message is the same as in Ethernet LAN connection example above.

Hint:

IG-IB does not respond e-mails which are sent to iG-IB mailbox.

Active call – EML-SMS

You should set Setpoints **Act.cals/SMS**:*AcallCHx-Type* = EML-SMS and specify the e-mail address of a mobile phone if you want to send active messages from controller using SMS e-mail.

The Active call – EML-SMS service informs the user of mobile phone about current items in the Alarm list.

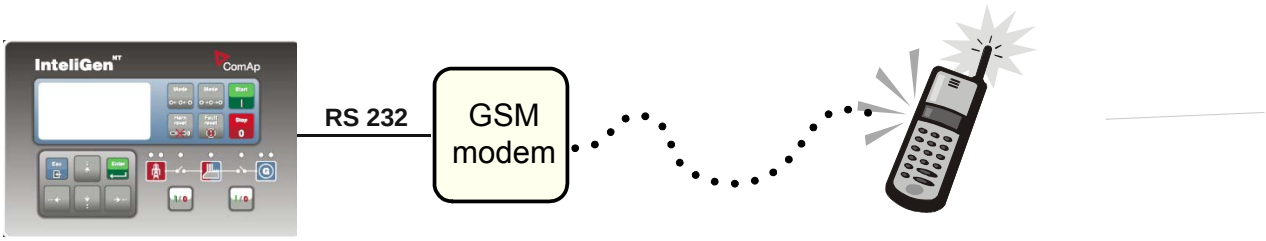
Example of active EML-SMS

```
IS_ABC: B:AL=(Sd Water Temp,Wrn Water Temp,!Emerg Stop,ActCallCH1Fail)
```

Hint:

EML-SMS channel setting depends on local GSM provider.

SMS Message Control



You can control and setup the Genset using SMS messages from your mobile phone.

Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES

Equipment

	Equipment needed
Controller side	GSM Modem or I-LB + GSM Modem
Connection	GSM
PC side	GSM Mobile Phone

Features

	On-line Genset control	Controller configuration	Read Genset values	Read history	Firmware programming	Archive files upload/download
Available	YES	YES	YES	NO	NO	NO

SMS message format

SMS message format:

- Start with # character, followed *controller address*, followed colon character and *access code,*
- Commands are comma separated,
- Commands are not case sensitive,
- Maximum message length is limited up to 160 characters,
- Controller or I-LB answers only message with valid Access code,
- Answer exceeds 160 characters is separated to more messages.

SMS message header

Every SMS must start with header in format:

```
#address:access command1, command2
```

where `address` is controller address 1 to 32
`access` is valid access code set-up by PC SW (up to 15 characters length),
`#` character indicates beginning of message,
`:` character separates *controller address* and *access code*

Hint:
For direct communication to one controller is possible skip address setting.

SMS message commands

1. *Controller address*
   Controller address is unique controller identification number located in setpoint group **Basic setting** **:** *Contr addr [1 to 32]*.

   Syntax: **#XX**
   XX … controller address [1 to 32]
   Example: #5
   Message is addressed to controller with address 5.

2. *Access code*
   InteliGen / InteliSys Access code is 15 characters length string. Access code is separated from controller address by column.

   Syntax: #5**:X**
   X … controller access code up to 15 characters length
   Example: #5:accesscode
   Message is addressed to controller with address 5 and its access code is set to value 'accesscode'.

3. *Read value or setpoint*
   Command for reading of selected value or setpoint. Return value is in appropriate numerical or string format.

   Syntax: **r XXXX** (or rXXXX)
   r … command
   XXXX… value or setpoint code
   Example: #5:accesscode **r 8252**
   Reading of setpoint 8252 (8252 = Gear teeth)

Hint:
Access code can't contain space character. Access code can be changed in InteliMonitor only.

4. *Adjust setpoint*
   Command for adjusting of selected setpoint. Answer message contains only confirmation of successful adjusting or appropriate error.

   Syntax: **w XXXX YYYY** (or wXXXX YYYY)
   w … command
   XXXX… setpoint code
   YYYY… value of setpoint in appropriate format
   Example: #5:accesscode **w 8252 144**
   Adjusting of setpoint 8252 to value 144 (8252 = Gear teeth).

   Return code:
   ok … adjusting setpoint was correct
   w_err … adjusting setpoint was not successful
   er_pass … adjusting setpoint required that valid password was entered
   er_old … command for adjusting was read out from SMS during GSM modem initialization – in this case command will not be served.

5. *Enter password*
   Password setting command. Password has to be set before adjusting of protected setpoint or calling protected gen-set control command. Setting password command is not necessary before every adjusting. Password is a number in range 0 to 65535 and is valid for all rest of SMS.

   Syntax: **p PPPP** (or pPPPP)

p … command
PPPP… password

Example: `#5:accesscode` **`p 1234`**`, w 8252 144`
Setting password before adjusting protected setpoint.

Return code: `ok` … setting password was successful
`er_pass` … setting password is not valid

6. *Gen-set control*

SMS command for invoking gen-set control command as Start, Stop, Fault reset etc.

Syntax: **`c Y`** `(or cY)`
c … command
Y … type of operation

Y	Type of operation		Y	Type of operation
1	Start		7	MCB ON
2	Stop		8	MCB OFF
3	Horn Reset		9	GCB ON/OFF
4	Fault Reset		10	MCB ON/OFF
5	GCB ON		11	Next Mode
6	GCB OFF		12	Previous Mode

Example: `#5:accesscode p 1234`**`, c1`**
This SMS command invokes genset Start. Password setting is needed in case of password protection was configured for genset commands.

Return code:
`ok` … genset command was accepted
`er_pass` … valid password was not set before executing the command
`c?` … unknown genset command
`c_er` … gen-set command execution is not allowed in actual state (e.g. attempt to start the genset in OFF mode).
`er_old` … command was read out from SMS during GSM modem initialization – in this case command will not be served.

7. *Read Alarm list*

Read actual Alarm list.

Syntax: **`a`**
a … command

Example: `#5:accesscode` **`a`**
Request of actual Alarm list.

Return code: `AL=(items of alarm list)` … comma separated items of Alarm list. Exclamation mark in front of Alarm list item indicates inverse record (still active alarm).

Note:
1. Answer message contains at most eight items of Alarm list.
2. Alarm list is not separated to more messages.

8. *Time delay*

Insert time delay before serving next part of SMS command.

Syntax: **`d T`**
d … command
T … time delay in sec (in range 1 to 600)

Example: `#5:accesscode` **`d 10`**
Request 10 sec delay before serving next SMS command.

Return code: `d_ok` … time delay was successful performed
`d_over` … requested time delay is out of range (1 to 600 sec)

Note: *Any other SMS messages are not served during time delay!*

9. *Remote switches (IG/IS-NT only)*
Set or reset RemoteControl1-8 output.

Syntax: **`s 1/0`**
s … command
1/0 … set/reset
Example: `#5:accesscode p0,` **`s1 1`**
Enters password p0 and sets *RemoteControl1* output.

Return code: p_OK,s_OK

10. *ExtValues (IG/IS-NT only)*
Enters value to ExtValue.

Syntax: **`e xxx`**
e … command
xxx … value
Example: `#5:accesscode p0,` **`e1 50`**
Enters password p0 and sets *ExtValue1* = 50.

Return code: p_OK,e_OK

11. *Help (IG/IS-NT only)*
Request for list of supported SMS command.

Syntax: **`?`**
Example: `#5:accesscode` **`?`**
Return code: ?=(p <user:>passwd,r comm_obj,w com_obj val,c cmd_num,d sec,a,sx y,ex y,?)…...
..... list of supported SMS commands
Note: *Return code is not separated to more message.*

12. *Answer message*
Answer message start with # character followed by Gen-set name. Colon separates this header form return codes of SMS commands. Answer message is generated during serving of received message and is sent in case that 160 characters or end of received message are achieved. Answer message is sent to the originator phone number. Tree dots at the end of message indicate separation and next following message.

Example: `#5:accesscode r8252,w8252 100,r8252`
answer message
`#Gen-setname: 144,ok,100`

13. *Examples of SMS commands*
Here is following several examples of SMS messages addresses to controller *IG/IS-NT* with address 5, named '*Gen-set name*'. Access code in this controller is set to '*accesscode*' and password is '*1234*'. In examples are used setpoints and values 8276 – Nomin.power, 10123 – RPM, 8315 – Controller Mode, 8235 – binary inputs, 8296 – Gen > f.

Example 1 – reading value

SMS:	#5:accesscode r8276	*read value 8276*
Answer:	#Gen-set name:100	

Example 2 – adjusting setpoint

SMS:	#5:accesscode p 1234, r8276,w8276 110,r8276	*read value 8276,* *write 110,* *read value 8276*

Answer:	#Gen-set name:ok,100,ok,110	*Password was accepted,* *read value of 8276 is 100,* *writing to 8276 was ok,* *new value of 8276 is 110*
	If wrong password sent: #Gen-set name:p_er,100, w_pass, 100	*Password was not accepted,* *read value of 8276 is 100* *writing to 8276 was not successful* *read value of 8276 is still 100*

Example 3 – Gen-set control and delay time

SMS:	#5:accesscode r8276,c1,d30,r10123	*read value 8276,* *invoke gen-set command START,* *delay 30 sec,* *read value 10123*
Answer:	#Gen-set name:110,ok,d_ok,1499	*read value of 8276 is 110,* *Gen-set command START was accepted,* *confirm delay command,* *read value of 10123 is 1499*

Example 4 – adjusting special setpoint

SMS:	#5:accesscode r8315,w8315 0,r8315	*read value 8315,* *write 0 (index of stringlist type),* *read value 8315*
Answer:	#Gen-set name:MAN,ok,OFF	*read value of 8315 as string,* *writing was ok,* *read new value of 8315 as string*

Hint:
Setpoints Stringlist type (e.g. Controller Mode) is read as string and adjusted as index of string item in string list. e.g. Controller Mode:

Read value [as string]	Write value [as index]
OFF	0
MAN	1
SEM	2
AUT	3
TEST	4

Example 5 – reading and writing other type

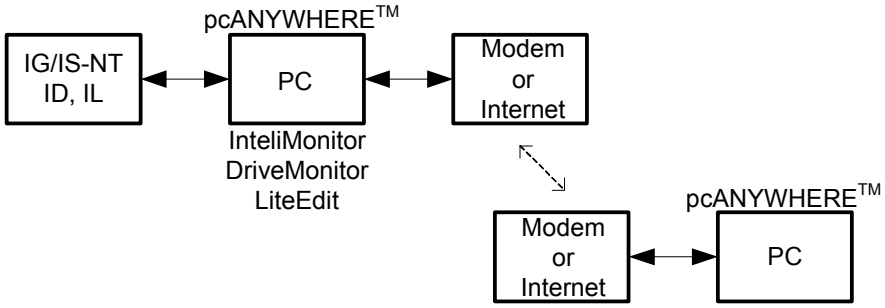
SMS:	#5:accesscode r8235,w8296 110.2	*read value 8235,* *write 110.2 with decimal point*
Answer:	#Gen-set name:OIIIOOIIO,ok	*read value of 8235 (binary value),* *writing was ok*

Note: *1. Writing of binary setpoint is not supported.*
2. Writing of setpoint with decimal point is automatically converted to appropriate number of decimal places.

Example 6 – reading actual Alarm list

SMS:	#5:accesscode a	*read actual Alarm list*
Answer:	#Gen-set name:AL=(!Wrn PrimWater temp, !Wrn SecWater temp, Batt volt)	*Actual Alarm list contains three items.*

Terminal Connection



Controllers

Controllers	IG-NT/IG-EE	IG-NTC/IG-EEC	IS-NT
Connection applicable	YES	YES	YES

Equipment

	Equipment needed
Controller side	Local connection + PC terminal software running (e.g. pcANYWHERE™) + InteliMonitor installed + modem
Connection	Phone line / internet
PC side	Modem + terminal SW

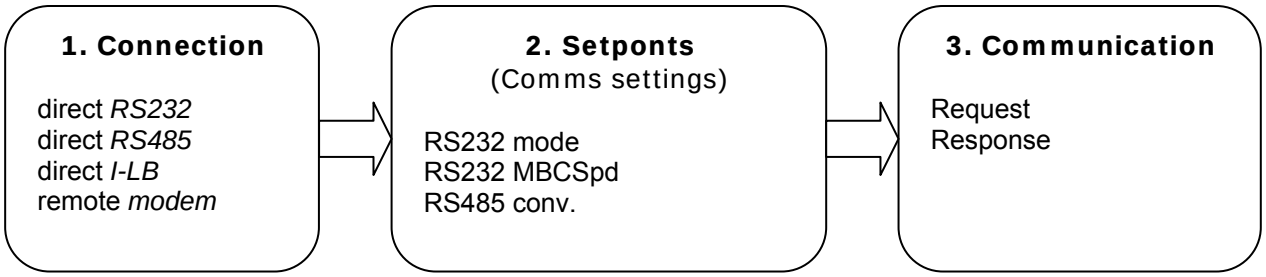
Function

This way you can remotely handle local PC and remotely start InteliMonitor or other software on local computer and make remote software upgrade or other procedures.
See website Symantec Enterprise Solutions

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 three possibilities of Modbus connection to single or multiple controllers : direct via RS232, RS485 or via Modem.

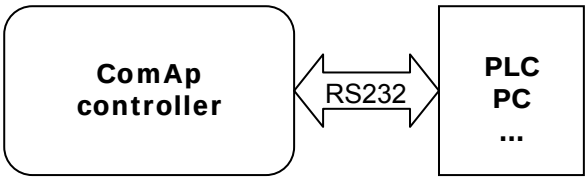
Controller's configuration:
Comms settings:*RS232(1) mode* = [MODBUS-MDM(HW), MODBUS-DIRECT]
Comms settings:*RS232(2) mode* = [MODBUS-MDM(HW), MODBUS-DIRECT] (when RS232(2) port available)

Selection of Modbus communication speed:
Comms settings:*RS232(1)MBCSpd* = [9600 , 19200 , 38400 , 57600] bps
Comms settings:*RS232(2)MBCSpd* = [9600 , 19200 , 38400 , 57600] bps (when (2) port available)

Selection of RS232/RS485 Converter:
Comms settings:*RS485(1)Conv.* = [DISABLED, ENABLED]
Comms settings:*RS485(2)Conv.* = [DISABLED, ENABLED] (when (2) port available)

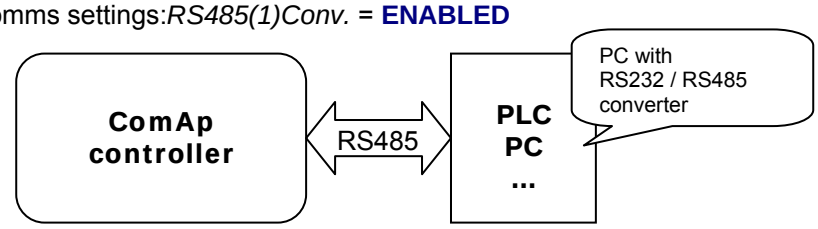
Modbus communication via RS232 – single controller

Controller's configuration:
Comms settings:*RS232(1) mode* = **MODBUS-DIRECT**
Comms settings:*RS232(1)MBCSpd* = **select of Modbus communication speed**
Comms settings:*RS485(1)Conv.* = **DISABLED**



Modbus communication via RS485

Controller's configuration:
Comms settings:*RS232(1) mode* = **MODBUS-DIRECT**
Comms settings:*RS232(1)MBCSpd* = **select of Modbus communication speed**
Comms settings:*RS485(1)Conv.* = **ENABLED**

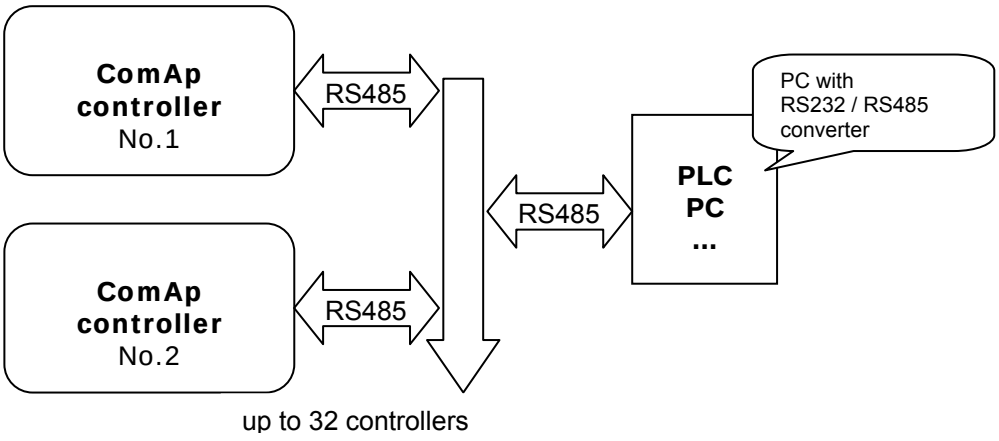


Hint:
The RS232/RS485 converter is included in the IG-NTC, IG-EEC and IS-NT controllers (no external RS232/RS485 converter is needed).

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.

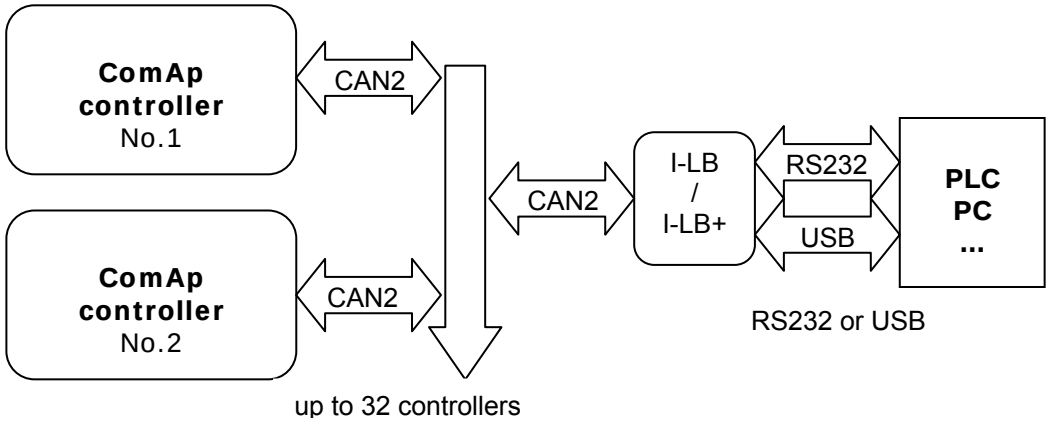
Modbus communication via RS485 – multiple controllers

Controller's configuration:
Comms settings:*RS232(1) mode* = **MODBUS-DIRECT**
Comms settings:*RS232(1)MBCSpd* = **select of Modbus communication speed**
Comms settings:*RS485(1)Conv.* = **ENABLED**



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**



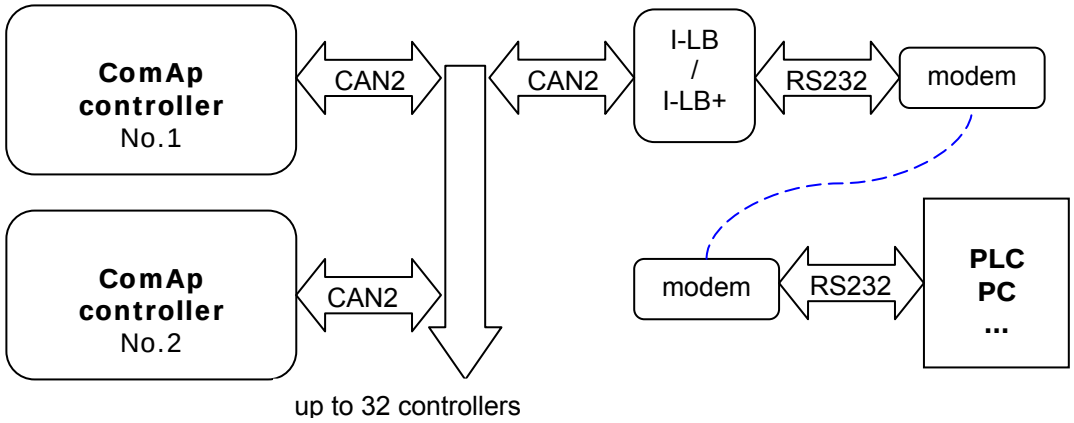
Hint:
To use I-LB Modbus communication connect Modbus jumper in I-LB unit (P16). Additionally, you can choose the communication speed using the speed selection jumpers (P13, P14). Their combination allows the speed selection of 9600 / 19200 / 38400 / 57600 bps.

Hint:
IG-IB communication unit doesn't support Modbus/TCP protocol!

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.



Hint:
IG-IB communication unit doesn't support Modbus/TCP protocol!

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)

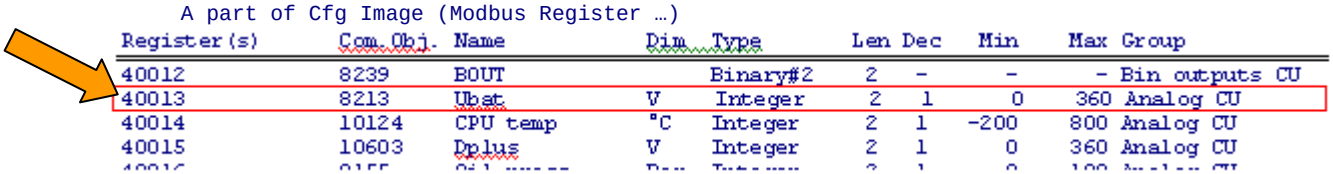
Request: **01 03 00 0C 00 01** 44 09

01 = Controller address
- *see your controller settings*

03 = Modbus function code (Read Multiple Registers)

00 0C = Register address: Register number (Ubat => 40013)
- 40013 - 40001 = 12 DEC => 000C 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
40012	8239	BOUT		Binary#2	2	-	-	-	Bin outputs CU
40013	8213	Ubat	V	Integer	2	1	0	360	Analog CU
40014	10124	CPU temp	°C	Integer	2	1	-200	800	Analog CU
40015	10603	Dplus	V	Integer	2	1	0	360	Analog CU

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*

09 44 = CRC
- *CRC has to be written LSB then MSB ! See how to calculate CRC. or implementation in C language – page 66.*

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

01 = Controller address
- *see your controller settings*

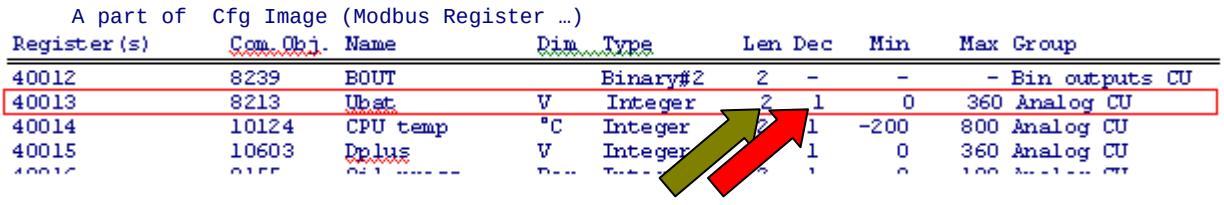
03 = Modbus function code (Read Multiple Registers)

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 …)

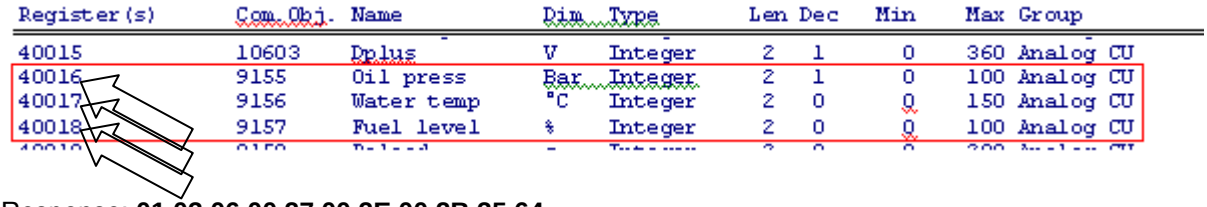


Register(s)	Com. Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40012	8239	BOUT		Binary#2	2	-	-	-	Bin outputs CU
40013	8213	Ubat	V	Integer	2	1	0	360	Analog CU
40014	10124	CPU temp	°C	Integer	2	1	-200	800	Analog CU
40015	10603	Dplus	V	Integer	2	1	0	360	Analog CU

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

Request: **01 03 00 0F 00 03 35 C8**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 0F = Register address: Register number (40016) – 40001 = 15 DEC => 0F HEX
00 03 = Number of registers (40016 – Oil press, 40017 – Water temp, 40018 – Fuel level) = 3 DEC = > 03 HEX
C8 35 = CRC (write LSB MSB !)

A part of Cfg Image (Modbus Register …)



Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40015	10603	Dplus	V	Integer	2	1	0	360	Analog CU
40016	9155	Oil press	Bar	Integer	2	1	0	100	Analog CU
40017	9156	Water temp	°C	Integer	2	0	0	150	Analog CU
40018	9157	Fuel level	%	Integer	2	0	0	100	Analog CU

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 (Water 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 02 00 01 25 CA**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 02 = Register address: Register number (40003) – 40001 = 02 DEC => 02 HEX
00 01 = Number of registers (40003) = 01 DEC => 01 HEX
CA 25 = CRC (write LSB MSB !)

Response: **01 03 02 00 0A 38 43**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
02 = Length of read data in Bytes (in HEX)
00 0A = Object data value (Binary input = 0000000000001010 i.e. BI2 and BI4 are set)*
43 38 = CRC

* Table of binary inputs (BI)

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

Password decode - reading

Request: **01 03 00 A0 00 02 C4 29**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 A0 = Register address: Register number (40161) – 40001 = 160 DEC => A0 HEX
00 02 = Number of registers (40161 and 40162) = 02 DEC => 02 HEX
29 C4 = 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 B8 00 08 C6 0D**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
0B B8 = Register address: Register number (43001) – 40001 = 3000 DEC => BB8 HEX
00 08 = Number of registers (43001 - 43008) = 08 DEC => 08 HEX
0D C6 = CRC (write LSB MSB !)

Response: **01 03 10 49 47 53 2D 4E 54 00 00 00 00 00 00 00 00 00 00 D7 6A**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
10 = Length of read data in Bytes (in HEX)
49 47 = Object data value (IG)
53 2D = Object data value (S-)
4E 54 = Object data value (NT)
00 00 = Object data value (_ _)
00 00 = Object data value (_ _)
00 00 = Object data value (_ _)
00 00 = Object data value (_ _)
00 00 = Object data value (_ _) =.> gen-set name is **IGS-NT**
6A D7 = CRC

Engine state - reading

Request: **01 03 00 A2 00 01 25 E8**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
00 A2 = Register address: Register number (40163) – 40001 = 162 DEC => A2 HEX
00 01 = Number of registers (40163)
E8 25 = CRC (write LSB MSB !)

Response: **01 03 02 00 02 39 85**
01 = Controller address
03 = Modbus function code (Read Multiple Registers)
02 = Length of read data in Bytes (in HEX)
00 02 = Object data value – see the List#1 in the Cfg Image => **(NotReady)**
85 39 = CRC

A part of Cfg Image (Modbus Register …)

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group
40001	8505	(N/A)							
40163	9244	Engine State		List#1	1	-	429	449	Info
43025	8252	Gear teeth		Unsigned	2	0	0	500	Basic settings
43026	8253	Nominal RPM	RPM	Unsigned	2	0	100	4000	Basic settings
43027	8315	ControllerMode		List#3	1	-	359	363	Basic settings

A part of Cfg Image (Modbus Register …)

List#1

Value	Name
0	Init
1	Ready
2	NotReady
3	Prestart
4	Cranking
5	Pause

Gear teeth – writing

Request: **01 06 0B D0 00 7D 4A 36**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D0 = Register address: Register number (43025) – 40001 = 3024 DEC => BD0 HEX
00 7D = Gear teeth > 125 DEC => 7D HEX
36 4A = CRC (write LSB MSB !)

Response: **01 06 0B D0 00 7D 4A 36**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D0 = Register addres
00 7D = Set the setpoint gear teeth to > 7D HEC => 125 DEC = **125**
36 A4 = CRC

Nominal Power – writing

Request: **01 06 0B C0 01 F4 8B C5**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B C0 = Register address: Register number (43009) – 40001 = 3008 DEC => BC0 HEX
01 F4 = Nominal power > 500 DEC => 1F4 HEC
C5 8B = CRC (write LSB MSB !)

Response: **01 06 0B C0 01 F4 8B C5**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B C0 = Register addres
01 F4 = Set the setpoint nominal power to > 1F4 HEC => 500 DEC = **500**
C5 8B = CRC

Mode – writing

Request: **01 06 0B D2 00 03 6B D6**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D2 = Register address: Register number (43027) – 40001 = 3026 DEC => BD2 HEX
00 03 = Set the controller mode to > TEST => 03 – see the List#3 in the Cfg Image
D6 6B = CRC (write LSB MSB !)

A part of Cfg Image (Modbus Register …)

List#3

Value	Name
0	OFF
1	MAN
2	AUT
3	TEST

Response: **01 06 0B D2 00 03 6B D6**
01 = Controller address
06 = Modbus function code (Write Single Register)
0B D2 = Register addres
00 03 = Object data value > **TEST**
D6 6B = 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 70)
0001 = command number (page 70)
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

Remote Switch 1 – Set (Remote Control 1)

Request: **01 10 18 D6 00 03 06 00 20 00 00 00 1A 7C 9A**
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+001A)
00200000 = argument for **RemoteSwitch1-8** (page 70)
001A = command number (page 70), command = 26 DEC = 001A HEX
9A 7C = CRC (write LSB MSB !)

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

Request: **01 10 18 D6 00 03 06 00 10 00 00 00 1A 3C E9 –** Remote Switch 1 – Off
Request: **01 10 18 D6 00 03 06 00 20 00 01 00 1A 8C 9A** – Remote Switch 2 – Set

RemoteSwitch	1	2	3	4	5	6	7	8
Set / On	00200000	00200001	00200002	00200003	00200004	00200005	00200006	00200007
Off	00100000	00100001	00100002	00100003	00100004	00100005	00100006	00100007

External Value1 – writing

Request: **01 10 18 D6 00 03 06 00 02 00 24 00 23 04 84**
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 (00020024+0023)
00020024 = argument for **Set ExtValue 1-4** (page 70) (value 36DEC = 0024HEX)
0023 = command number (page 70), command = 35 DEC = 0023 HEX
84 04 = CRC (write LSB MSB !)

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 70)
0001 = command number (page 70)
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 70)
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 70)
91 CE	= CRC

History – reading

See more information about History reading on page *66.*

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 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 *66.*

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

Modbus Protocol Description

- Direct connection:
  - RS232, RS485, (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	Check field CRC - lower byte (LSB)	See Check field calculation

Byte	Meaning	Note
7	- upper byte (MSB)	

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.

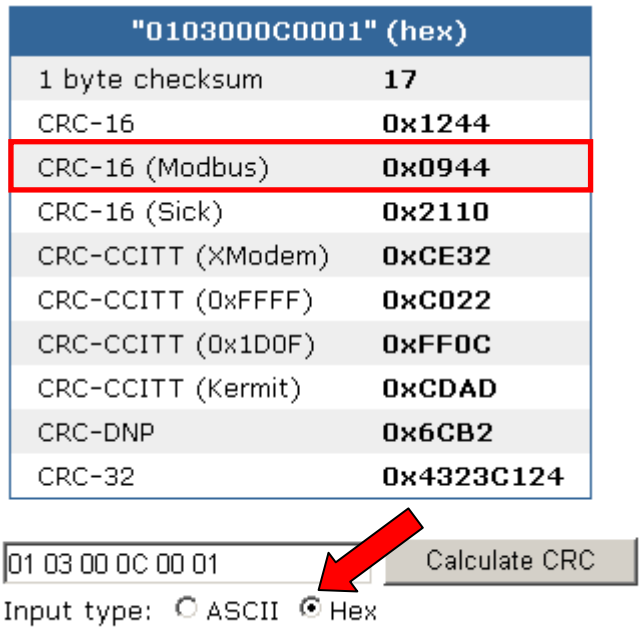
Requested history record is stored at the moment of locking in the cache memory and following reading is performed from this memory.

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.

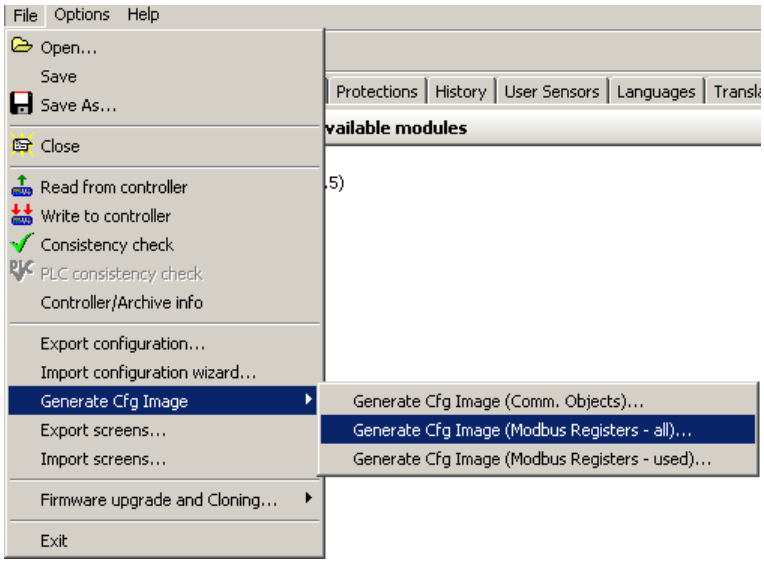
```
unsigned short count_CRC(unsigned char *addr, int num)
{
  unsigned short CRC = 0xFFFF;
  int i;

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



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 (SSB, SPM, SPtM, MEXT, MINT, …).
2. Communication objects independent on the application type.

Use GenConfig menu command File –> Generate Cfg Image –> Generate Cfg Image (Comm. Objects …) and Generate Cfg Image (Modbus Registers all/used).

A part of Cfg Image (Modbus Register …) file

Register(s)	Com.Obj.	Name	Dim	Type	Len	Dec	Min	Max	Group	
40001	8505	(N/A)								
40002	8506	(N/A)								
40003	8235	BIN		Binary#1	2	-	-	-	Bin inputs CU	
40004	9107	(N/A)								
40005	9108	(N/A)								
40006	9109	(N/A)								
40007	9110	(N/A)								
40008	9111	(N/A)								
40009	9112	(N/A)								
40010	9113	(N/A)								
40011	9114	(N/A)								
40012	8239	BOUT		Binary#2	2	-	-	-	Bin outputs CU	
40013	8213	Ubat	V	Integer	2	1	0	360	Analog CU	
40014	10124	CPU temp	°C	Integer	2	1	-200	800	Analog CU	
40015	10603	Dplus	V	Integer	2	1	0	360	Analog CU	
40016	9155	Oil press	Bar	Integer	2	1	0	100	Analog CU	
40017	9156	Water temp	°C	Integer	2	0	0	150	Analog CU	
40018	9157	Fuel level	%	Integer	2	0	0	100	Analog CU	
40019	9158	Reload	s	Integer	2	0	0	200	Analog CU	
43001-43008 (8)	8637	Gen-set name		String0	16	-	-	-	Basic settings	
43009	8276	Nomin power	kW	Unsigned	2	0	1	32000	Basic settings	
43010	8275	Nomin current	A	Unsigned	2	0	1	10000	Basic settings	
43011	8274	CT ratio prim	A	Unsigned	2	0	1	10000	Basic settings	
43012	10556	CT ratio sec		List#9	1	-	41	42	Basic settings	
43013	8566	Im3/ErFlCurCTp	A	Unsigned	2	0	1	10000	Basic settings	
43014	10557	Im3/ErFlCurCTs		List#9	1	-	41	42	Basic settings	
43015	9579	VT ratio	V/V	Unsigned	2	1	1	5000	Basic settings	
43016	10662	Vg InpRangeSel		List#10	1	-	43	44	Basic settings	
43017	9580	Vm VT ratio	V/V	Unsigned	2	1	1	5000	Basic settings	
43018	10663	Vm InpRangeSel		List#10	1	-	43	44	Basic settings	
43019	8277	GenNomV	V	Unsigned	2	0	80	30000	Basic settings	
43020	9673	GenNomVph-ph	V	Unsigned	2	0	130	60000	Basic settings	
43021	9888	MainsNomV	V	Unsigned	2	0	80	30000	Basic settings	
43022	9907	MainsNomVph-ph	V	Unsigned	2	0	130	60000	Basic settings	
43023	10647	VoltProtSelect		List#11	1	-	45	46	Basic settings	
40163	9244	Engine State		List#1	1	-	429	449	Info	
43025	8252	Gear teeth		Unsigned	2	0	0	500	Basic settings	

Application values (read only)

Application setpoints (read/write)

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.

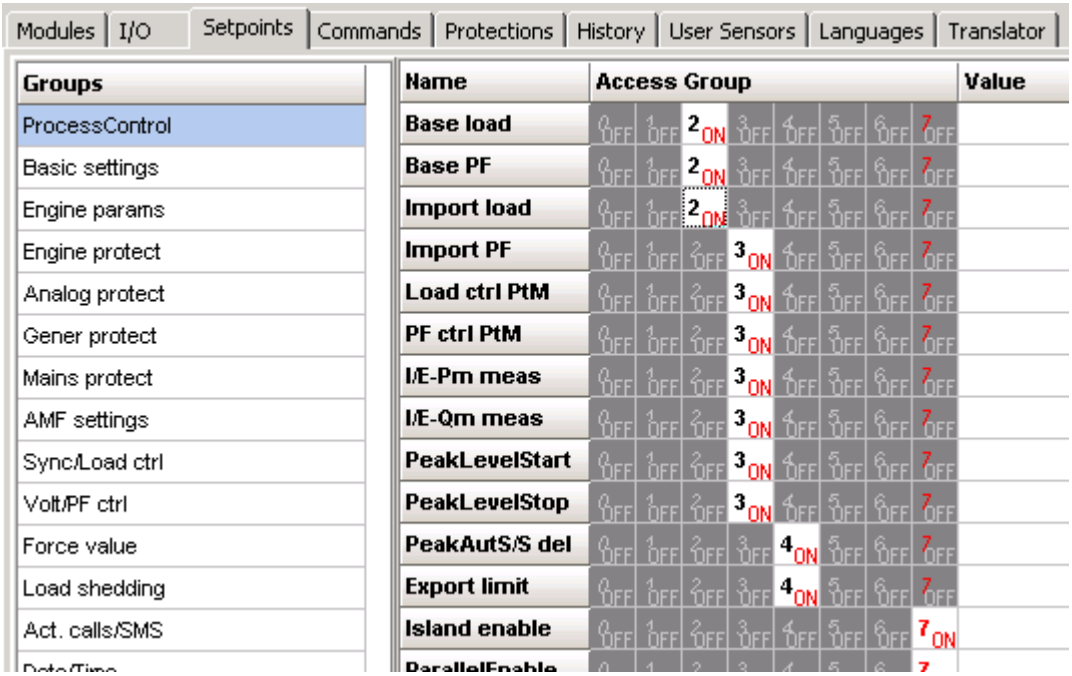
 # 3
An argument must be written before writing of a command code, because immediately after the command code has been written, the command is executed. It is recommended to write an argument and command simultaneously, in a multiple registers write. As the argument has lower register address than command, 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 IG/IS-NT controllers it is possible to specify access levels to protected objects for 8 different users. For each user a set of access attributes is defined and each of them has his password. The user can gain the right for writing to 8 groups of objects with different access levels by entering his password. The objects are assigned into groups in the controller configuration. For example setpoints in the ProcessControl group can be configured in GenConfig on Setpoints card:



Each user has his identification number (0 – 7). User with identification number 0 has an exceptional position. This user has access to all groups of protected objects (this access cannot be changed anyhow) and can define groups of access attributes to other users (1 – 7), reset their password and set their name (alias of an identification number). Entering of password must be foregone by writing of a user identification number.

Commands for IGS-NT

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

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/open generator circuit breaker (IGS-NT) Clutch ON/OFF (ID)	11EE0000	000011EF	OK
			2	Argument has not been written
	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	
D	Set counter of engine starts	New value	N/A	
E	Set runhours counter	New value	N/A	
19	Set counter of unsuccessful engine starts	New value	N/A	
1A	Set binary output RemoteSwitch1-8 (RemoteControl1-8)	00200000	N/A	
	Reset binary output RemoteSwitch1-8 (RemoteControl1-8)	00100000	N/A	
1F,20,21,22	Set pulse counters (IS-NT only)	XXXXYYYY (XXXX – Upper part of a new value; YYYY – Lower part of a new value)	Upper value + 1	OK
			1, 2	Not possible to perform
23,24,25,26	Set ExtValue1-4 (#2)	0002YYYY (YYYY - new value)	3	OK
			1, 2	Not possible to perform

(*) 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.
 # 2
Check if the setpoints ExtValueXLoLim and ExtValueXHiLim allow set the requested value to ExtValue.

Commands for IM-NT

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

Command	Meaning	Argument (*)	Return value (*)	
1	Start command	01FE0000	000001FF	OK
			2	Argument has not been written
	Stop command	02FD0000	000002FE	OK
			2	Argument has not been written
	Horn reset	04FB0000	000004FC	OK
	Fault reset	08F70000	000008F8	OK
		other	1	Wrong argument
2	Close/open MGCB	11EE0000	000011EF	OK
			2	Argument has not been written
	Close MGCB	11EF0000	000011F0	OK
			2	Argument has not been written
	Open MGCB	11F00000	000011F1	OK
			2	Argument has not been written
	Close/open MCB / BTB	12ED0000	000012EE	OK
			2	Argument has not been written
	Close MCB / BTB	12EE0000	000012EF	OK
			2	Argument has not been written
	Open MCB / BTB	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
8	Set kWh counter	New value	N/A	
C	Set kVAhr counter	New value	N/A	
1A	Set binary output RemoteSwitch1-8 (RemoteControl1-8)	00200000	N/A	
	Reset binary output RemoteSwitch1-8 (RemoteControl1-8)	00100000	N/A	
1F,20,21,22	Set ExtValue1-4 (#2)	0002YYYY (YYYY - new value)	3	OK
			1, 2	Not possible to perform

(*) in HEX

 # 9
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.
 # 2
Check if the setpoints ExtValueXLoLim and ExtValueXHiLim allow set the requested value to ExtValue.

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

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

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

Unsigned16	Unsigned integer – 16 bits	1	MSB1 = comm. object value, bits 15-8 LSB1 = comm. object value, 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.

Note:
ComAp believes that all information provided herein is correct and reliable and reserves the right to update at any time. ComAp does not assume any responsibility for its use unless otherwise expressly undertaken.

I-LB, IG-IB Communication Units

Local Bridge I-LB

Local Bridge I-LB provides connection of up to 32 ***InteliGen*** or ***InteliSys - NT*** controllers.
If I-LB not available, IG-MU can be used to cover most of the I-LB functionality.

Modem support

I-LB supports Hayes compatible analog modems with baud rate 9600bps, 19200bps and 38400bps.
I-LB supports GSM modems witch baud rate 9600bps or 19200bps.

See Recommended modems.

I-LB current SW version supports:

- Analog modem Hayes compatible,
- ISDN ASKEY modem,
- GSM modem for data communication
- Direct connection to PC via RS232, RS485
- I-LB+ version additionally supports USB connection
- Modbus protocol,
- Capability to connect up to four I-LB on one CAN bus (two as *local bridge*, two as *modem bridge*),
- Serve active call requested by controllers ***InteliGen*** / ***InteliSys – NT***
- Send active SMS, receive and serve SMS via GSM modem.
- I-LB reads Number of rings setpoint from controller of lowest address during switch on and each 20 minutes (Number of rings change is not accepted immediately).

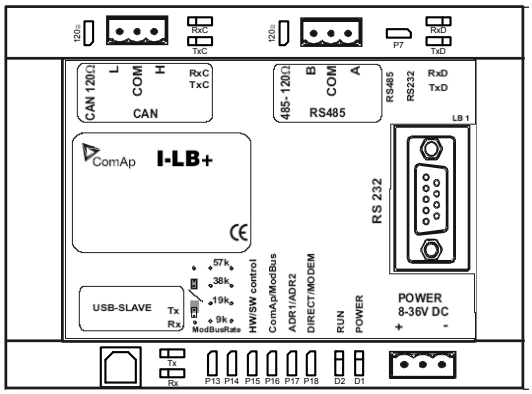
Hint:

Check I-LB sw version in InteliMonitor -> Monitor -> Controller/Archive info.

If I-LB SW version is displayed in Multiedit About window, it is shown as IG-MU v.2.2 due to backward compatibility reasons.

The latest I-LB and IG-IB sw version description see in document HW-SW versions.

I-LB Terminals and jumpers



Indication LED:

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 communicated on the CAN bus (during detection).
PWR	Lights All the time when power supply is switched on.

Jumper		Note	Default setting
P18	Direct/Modem	Place jumper when I-LB is connected for modem communication	Opened
P17	ADR1/ADR2	Place jumper to choose address 2 (for local or modem connection).	Opened
P16	ComAp/Modbus	Place jumper for Modbus communication.	Opened
P15	HW/SW control	Place jumper if the modem doesn't provide active DSR signal.	Opened
P13/P14	Modbus rate	Place jumpers to select Modbus communication speed.	Opened
P1/P2	CAN/RS485 120Ω	Place jumper to connect 120Ω terminating resistor for CAN bus/RS485	Opened
P7	RS485/RS232	Place jumper to select RS485 or RS232connection	Opened

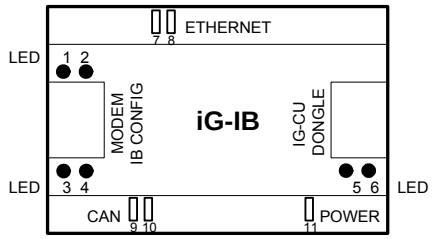
Hint:
Other I-LB jumpers are designed exclusively for factory tests.

IG-IB Internet Bridge

If more than 8 IGS-NT or other controllers are used with intercontroller CAN bus and monitored via IG-IB or I-LB units, the reaction on commands issued from InteliMonitor or other monitoring SW can be delayed up to several seconds. The same holds for values reading, i.e. refresh of measured power etc. in InteliMonitor or other PC SW.

Hint:
Contact local IT manager in any case before using iG-IB.

Indication and Diagnostic LED's



LED 1,2	Mode of iG-IB operation - see the table below
LED 3,4	Modem / Config line Rx, Tx activity
LED 5, 6	IG-CU data Rx, Tx activity
LED 7	Ethernet LINK (connection)
LED 8	Ethernet ACT (activity)
LED 9,10	CAN interface activity
LED 11	POWER supply indication

LED1	**LED2**	**IG-IB V2.0**
Light	Dark	After power switch on = iG-IB hardware fail. Unit has to be sent to repair.
Blinks together		Wrong configuration, iG-IB must be configured by IBConfig software tool 2.0
Lights both		Active IBConfig software. iG-IB does not communicate to ETHERNET or modem.
Blinks alternately		Invalid firmware or firmware downloading
Light	Dark	E-mail data reading from controller
Dark	Light	E-mail data sending to internet
Dark both		OK running state

Fast 0,1s blink	Dark	Modem initialization
Slow 0,3s blink	Dark	CAN bus rate detection

Hint:
Maximal length of UTP cable must be smaller than 100m to the nearest Switch/hub according to IEEE802.3 (100Base-T)

IG-IB Configuration

It is necessary to configure unit before its first use because new IG-IB from factory is not configured.

Internet Bridge can be connected to Internet
via LAN – Ethernet connector or
via dial up – MODEM connector.

Maximal length of UTP cable (between IG-IB and Switching hub) is 100m according to IEEE 802.3.

Corresponding firmware must be downloaded to IG-IB depends on required connection type.

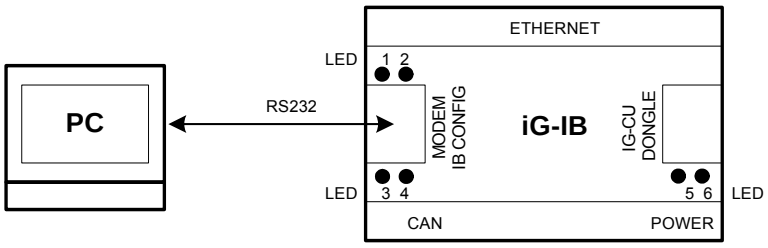
IG-IB interface	Firmvare V2.5
LAN – Ethernet	IG-IB-ethn_2.5.bin
Dial up – Modem	IG-IB-dial_2.5.bin

For IG-IB V2.5 firmware configuration use IG-IB configuration tool (IBConfig.exe) at least V1.2 or higher.
IBConfig of V1.2 can be used for IG-IB firmware 1.0 and 1.1 as well.
For IG-IB V2.1 firmware configuration use IBConfig at least V1.4 or higher.

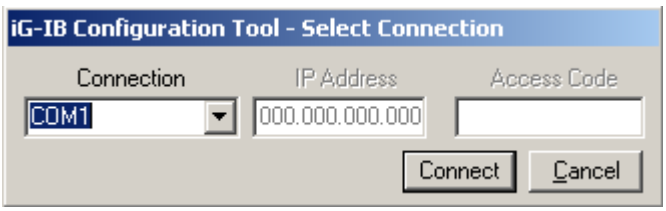
Hint:
You cannot update firmware in IS-Display terminal by means of IG-IB version 2.0 and lower. Available support for IGS-NT is in IG-IB 2.1 and higher. In case you use 2.0 version, the message appears (Access not allowed (object 24431)) because IG-IB is not able to work with IS-Display.

IG – IB jumpers:
The hardware jumpers have following meaning:
P1 – test mode – should be disconnected!
P2 – internal use – should be disconnected!
P3 – reset – should be disconnected!

Connect
iG-IB power supply,
"MODEM/IB CONFIG" RS232 interface cable to PC and
Run
… \ ComAp \ TOOLS \ IBConfig \ IBConfig.exe software.



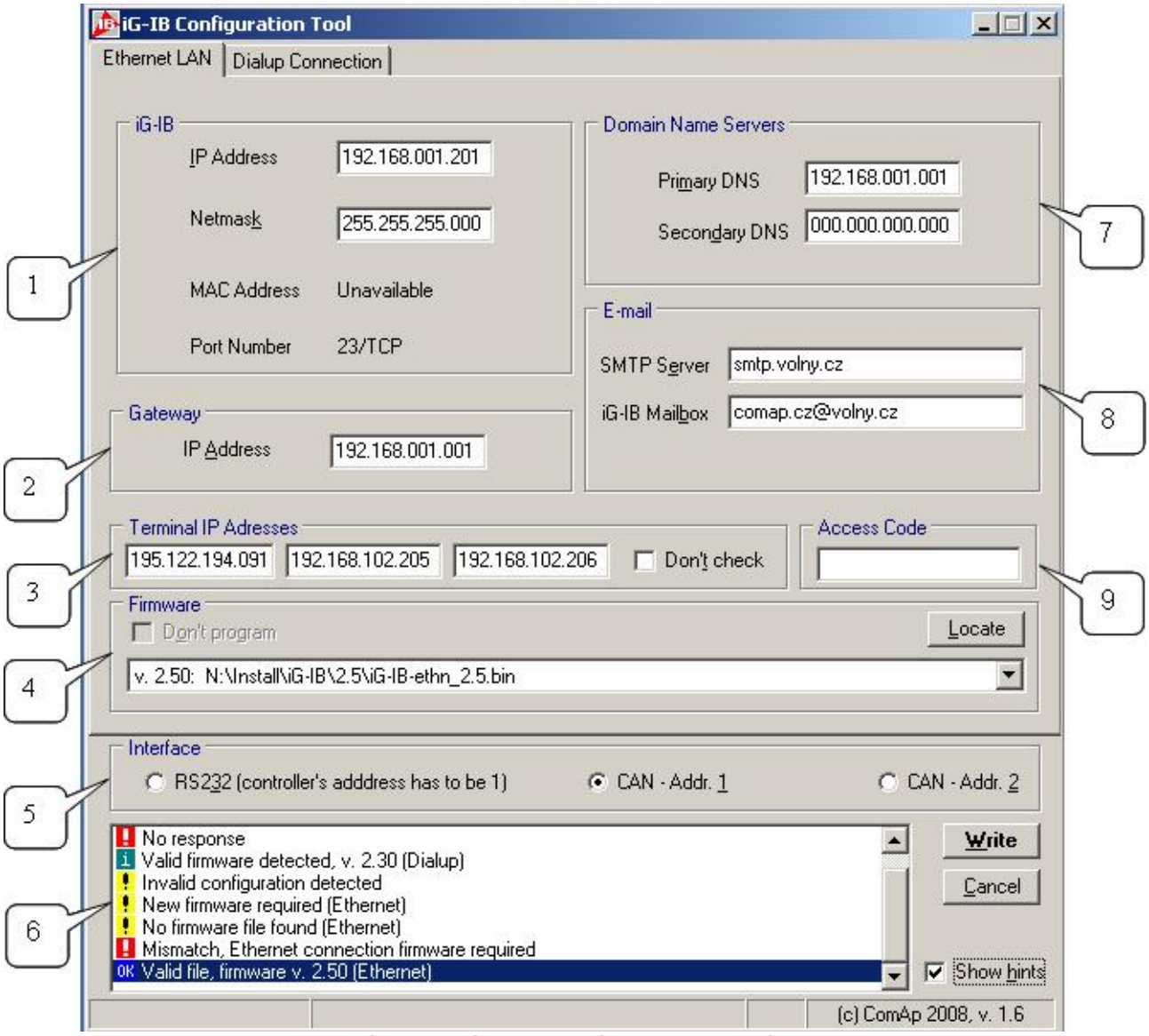
Set COM port:



Remote access (TCP/IP connection) to IG-IB may be denied if incorrect Access Code is used. Access denied warning is displayed by IBConfig then:

What you need to do in such case is to paste the string (it is saved automatically) into an e-mail and send it to your distributor to obtain a correct access code.

Ethernet LAN Configuration



These settings are only as an example!

Please notice:
Setting CAN-Addr.1 was interchanged with setting CAN-Addr.2 starting with IBConfig-1.6. It means that IG-IB's CAN address has to be set to CAN-Addr.2, if I-LB which is set to the address ADR1 is connected to the same CAN bus as IG-IB. Use option CAN-Addr.1 if I-LB is set to ADR2.
It is necessary to set IG-IB's CAN address using IBConfig-1.5 in this way: IG-IB's CAN address has to be set to CAN-Addr.1, if I-LB which is set to the address ADR1 is connected to the same CAN bus as IG-IB. Use option CAN-Addr.2 if I-LB is set to ADR2.

Set following items in IG-IB configuration window.

1	**IG-IB**	IP Address Netmask MAC Address Port Number	Ask your IT manager Ask your IT manager Ask your IT manager Ask your IT manager
2	**Gateway**	IP Address	Ask your IT manager
3	**Terminal IP Addresses**	Don't check	Connection will be restricted to these terminal IP addresses, i.e. IG-IB won't connect to any other address. Tick "***Don't check***" if you want to allow connection to all terminal IP addresses.
4	**Firmware**		Firmware file name (*.bin) is displayed in this window when old iG-IB firmware version is detected. It is possible to select any iG-IB "****.bin***" firmware file using **Locate** button. iG-IB firmware is located in WinEdit directory ***\Program files\Comap\WinEdit\Tools\IBConfig\Firmware\ *.bin***. If you check "***Don't program***" checkbox, the firmware is not downloaded (only settings).
5	**Interface**	RS232 CAN1 CAN2	Connection of IG-IB to controller via RS232 Connection of IG-IB to controller via CAN Two IG-IB's on CAN bus
6	**Message window**		Messages window contains messages regarding detected (actual) firmware version and configuration validity. Here you can also get information on configuration process. If you get the "***No response***" message, check once more COM setting and communication cable. If everything is OK and you still get the "***No response***" message the unit must be reprogrammed in the factory.
7	**Domain Name Servers**	Primary DNS Secondary DNS	Get from your IT manager
8	**E-mail**	SMTP Server IG-IB Mailbox	Necessary only when Active e-mails are enabled in Controller. It is used as a sender address when iG-IB sends active e-mail. See Setpoints ***Act.cals/SMS***: *AcallCHxType* = E-MAIL or EML-SMS.
9	**Access Code**		Code for unblocking the remote communication with the controller. Use InteliMonitor/DriveMonitor to enter or change access code.

IG-IB connection to internet (Ethernet)

Hint:
Your local LAN connection to the Internet is probably configured to enable access the servers located on the Internet, but to disable access in the reverse direction.

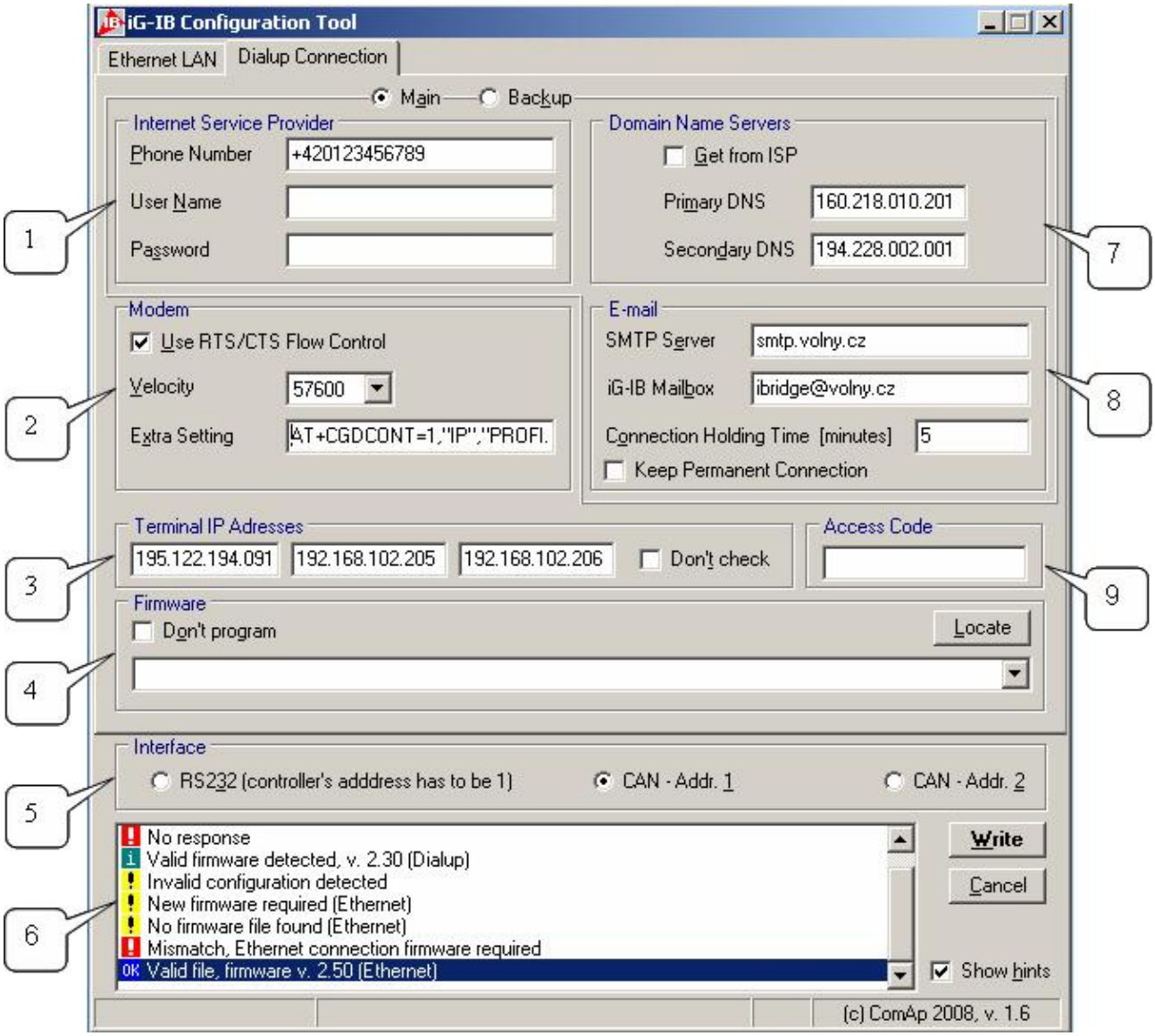
If iG-IB is to be visible from the Internet, then your IT manager should create a communication channel on the Internet access server.

The iG-IB uses the TCP-communication and listens at the port 23 (known as Telnet port).

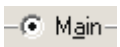
Hint:
Some providers have strict requirements for e-mail iG-IB Mailbox address = existing mailbox.

Dialup Connection Configuration

Use firmware V2.0 and higher for dialup connection.



Connection to ISP

There are two Internet Service Provider sets  and  .
Phone number is dialed using ATD command. The main ISP is dialed as first. The backup ISP is dialed when it is not possible to open the main ISP connection.
IP addresses for DNS servers can be obtained from ISP (“Get from ISP”) or explicitly set in configuration.

Modem initialization

Modem must accept following ASCII characters: Escape = ASCII 43 (=character +), ASCI 13 (= CR) and ASCII 10 (= LF). Then Following commands are sent to Modem.
1. ATZ
2. ATE0V0Q0S0=0
3. ATV0Q0X0S0=1

Active e-mail

Active e-mail can be sent only when operating modem is detected. IG-IB makes five open connection attempts on active e-mail request. There are three attempts to sent e-mail when connection is opened. Connection is opened for “Connection holding time” after e-mail is successfully sent.

Note for iG-IB firmware version 1.0 or 1.1 users: active e-mail sending may fail if selected SMTP server returns a multiline responses. Use firmware version 2.0 to solve this problem.

Configuration items

1	**Internet Service Provider**	Phone Number User Name Password	Data from local Internet provider
2	**Modem**	Use RTS/CTS Flow Control Velocity Extra Setting	Check if the modem requires RTS/CTS signals. Only when velocity auto detect is not active. Those “special” commands for modem configuration are sent to modem during init procedure.
3	**Terminal IP Addresses**	 Don’t check	Connection will be restricted to these terminal IP addresses, i.e. IG-IB won’t connect to any other address. Tick “***Don’t check***” if you want to allow connection to all terminal IP addresses.
4	**Firmware**		Firmware file name (*.bin) is displayed in this window when old iG-IB firmware version is detected
5	**Interface**	RS232 CAN1 CAN2	Connection to controller Connection to controller Two IG-IB’s on CAN bus
6	**Messages window**		Messages window contains messages regarding detected (actual) firmware version and configuration validity. Here you can also get information on configuration process.
7	**Domain Name Servers**	Get from ISP Primary DNS Secondary DNS	Get from your IT manager
8	**E-mail**	SMTP Server IG-IB mail box Connection Holding Time Keep Permanent Connection	 E-mail setting is necessary only when Active e-mails are enabled in Controller. It is used as a sender address when iG-IB sends active e-mail. After e-mail is successfully sent, waiting for operator response. If checked connection to ISP is never terminated. It is established immediately after IG-IB is initialized. If the communication is interrupted, it is re-established. It is necessary that the IP address of IG-IB is static so that the PC software (e.g. WinEdit) “knows” to which address to re-connect. If this option is not selected, the connection to ISP is established on request from terminal only. IP address obtained from ISP is then sent to the terminal via email (IP address can be dynamic in this case).
9	**Access Code**		Code for unblocking the remote communication with the controller. Use InteliMonitor/DriveMonitor to enter or change access code.

Hint:

Some providers have strict requirements for e-mail iG-IB Mailbox address = existing mailbox.

IG-IB Interface

There are following connectors on iG-IB unit:

Modem / IB Config	RS232 interface for iG-IB configuration or Modem connection (not supported in SW version1.0)
Dongle	RS232 interface to Controller or for Dongle for multiple controllers communication (controller data are transferred via CAN bus)
CAN bus	Interface to one or group of controllers (IG/IS-NT, ID)
Ethernet to LAN	Interface to Ethernet

IG-IB Dongle

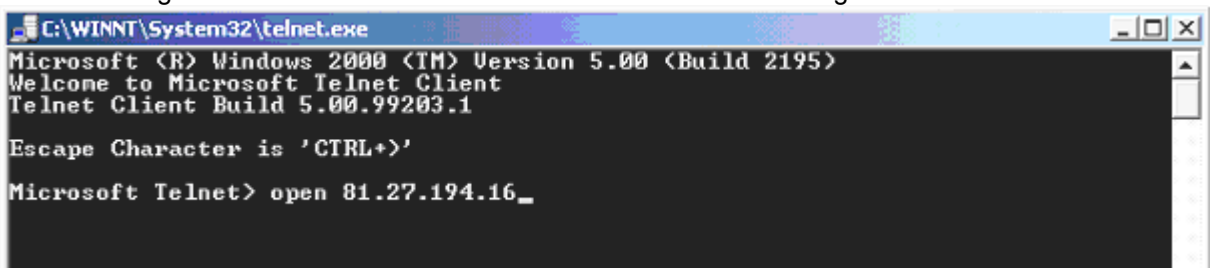
IG-IB Dongle limits number of accessible controllers.
Following addresses must be used for given dongle type (e.g. dongle IG-IB3 enables connection to controllers with addresses 1,2 and 3 only).

Dongle type	Number of accessible controllers
No dongle	1
Dongle IG-IB3	2 to 3
Dongle IG-IB7	2 to 7
Dongle IG-IB15	2 to 15
Dongle IG-IB32	2 to 32

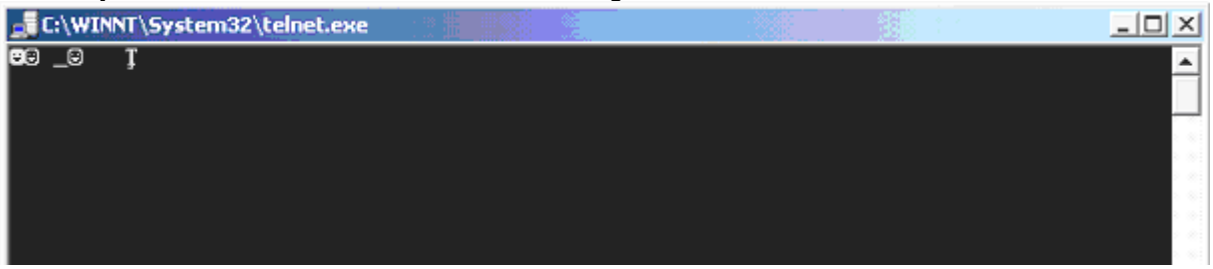
Hint:
From IG-IB-2.1 the dongle sets the limit of number of connected controllers (no specific order is required). If the controller is connected via RS232, the address 1 is supposed.

Internet Connection Diagnostics

Following method is Comap PC SW independent way how to check connection from PC via Internet to some controller.
The basic diagnostic of iG-IB to internet connection could be done using telnet software:



1) Start telnet software
2) In telnet window write command „open xxx.xxx.xxx.xxx“
3) If the connection is OK, you should get a nontext (binary - firmware version) answer. Depending on the font you use in the telnet window correct message looks like:

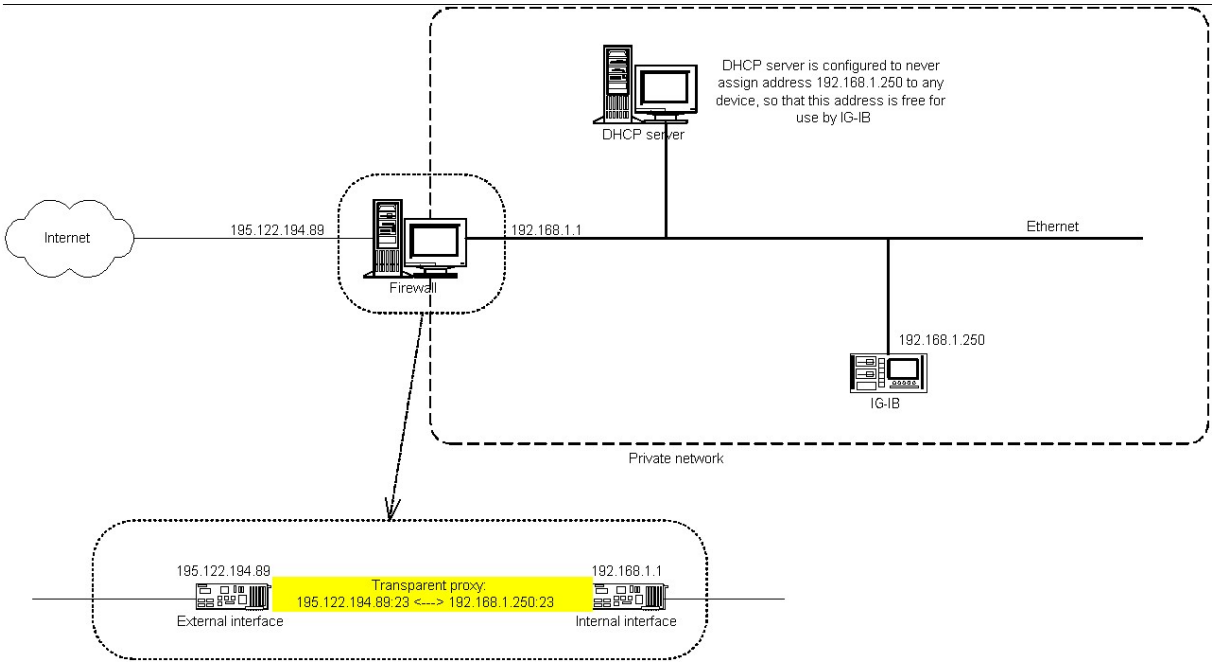


Hint:
Telnet software is included in standard Windows installation. To run telnet use Windows Start – Run button and write “telnet”, OK.

How to Access IG-IB behind Firewall

The solution principle is shown in attached picture. All addresses in this text and in the picture are fictive and will be different on real installation, port numbers are real. If the network software is configured as shown in picture, users from Internet will be able to connect from Comap PC SW to address 195.122.194.89 and communicate with IG-IB behind firewall. So in fact, we enter to PC SW different address than is configured in IG-IB, as is correctly noted in the manual.
Needed changes are these:
1. DHCP server in private network (if used): exclude address 192.168.1.250 from the address poll which are available to lease from this server, so that this address couldn't be assigned to any device in the network. This will make this address available for use by IG-IB.

2. Firewall: Configure transparent proxy channel from external interface 195.122.194.89, port 23 to protected network, internal address 192.168.1.250, port 23. This transparent proxy will do all network address translation needed when communicating from public network with internal private network.
Port 23 on interface 195.122.194.89 should be usually free for this use, as normally is this port used for telnet service and telnet is considered dangerous to use on firewalls.



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 (and PC) 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 – 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 Modem

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 – IG/IS-NT

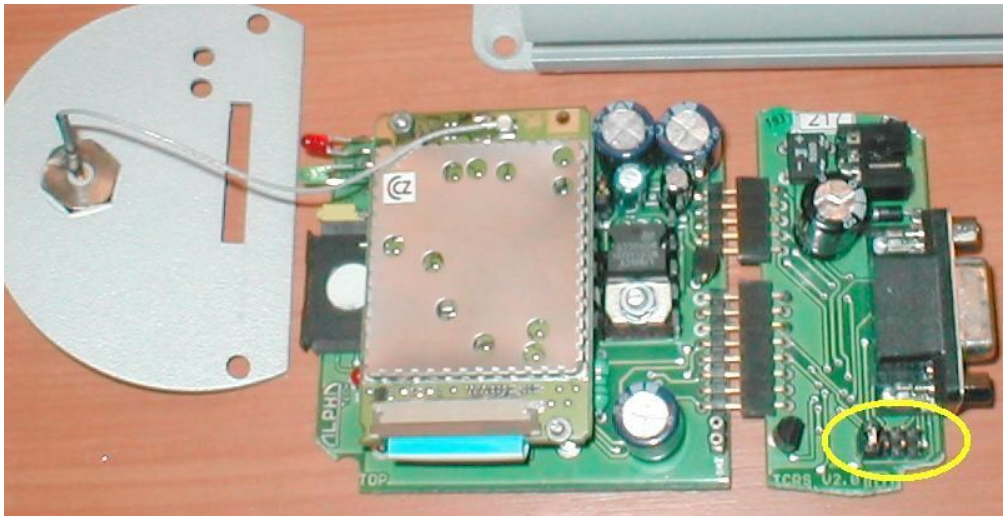
IG/IS-NT controllers provide a possibility to select whether a modem with active DSR signal (full modem cable connection - RXD, TXD, GND, RTS, CTS, DSR and DCD control signals available) or without this signal (3 wire connection – RxD, TxD and GND signals available) is connected.

In the controller the selection is done by **Comms settings**:*RS232(1/2) mode* setpoint.
Select MODEM (HW) for full cable connection or MODEM (SW) for 3 wire connection.

Place I-LB jumper “HW/SW control” when MODEM (SW) connection is selected.

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.



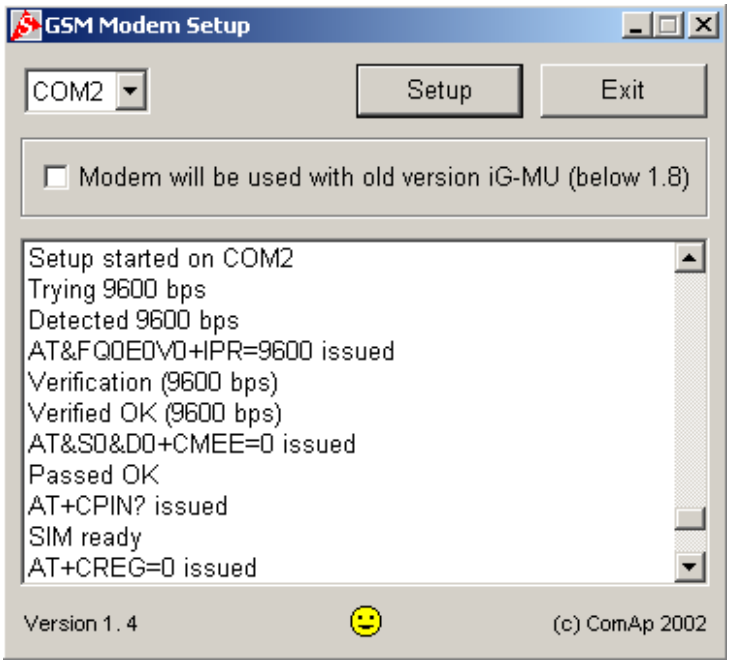
TC35 Alphatech GSM modem: the first jumper from the left is closed.

GSM Modem setup

Prior to start work with GSM modem run following program for GSM proper setup.
Program writes all the necessary AT commands to configure the GSM modem properly for use with IG-CU or IS-CU.

This program runs independent on other PC SW:

- Start MS Windows-Start-Program files – Comap – Tools – Gm_setup.exe.
- Select COM port
- Tick ☐ when GSM will be used with old iG-MU unit
- Press Setup button
- Follow commands in GSM Modem Setup window



Hint:
Selection “Modem will be used with old version of IG-MU (below 1.8)” sets GSM modem Baud rate to 38400 bps !!!

It is strongly recommended to use the same type of modem at both sides (IG and PC) of connection.

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 RS232()MdmIni parameter.
2. Send **AT+CPMS="SM","SM","SM"** command via hyperterminal or by means of RS232()MdmIni parameter.
3. Restart the modem.

Mobile Phone SIM Card Setting

Adjust SIM card on GSM modem in the following way:

- Enable data communication (it could be done by your local GSM operator),
- Set phone number SMS service center SC (recommended in international format in case of roaming),
- Set no PIN code required.

Hint:

All SMS on SIM card will be erased during GSM modem initialization. Valid SMS on SIM card will be served Except of adjusting and gen-set control command.
Both SIM cards must have DATA services enabled when communicating from controller to PC via two GSM modems.
The GSM tariff should have CSD (Circuit Switch Data) service enabled.

How to check SIM card DATA setting

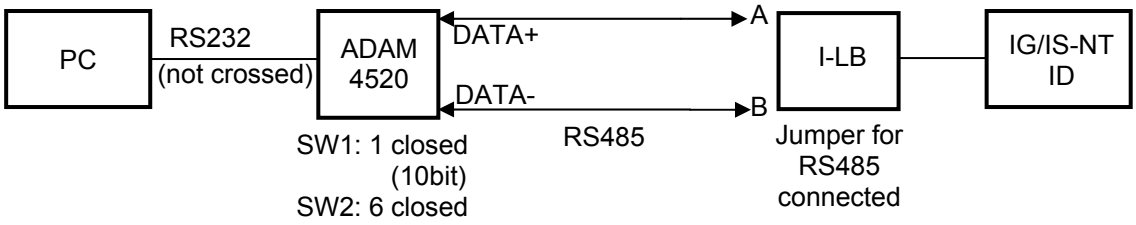
- Move SIM card from controller GSM modem to Mobile phone.
- Call from InteliMonitor to this Mobile phone and check (on Mobil phone) DATA call indication of incoming call.
- If phone does not indicate DATA - solve this with your GSM operator.

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

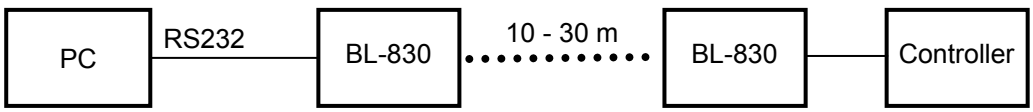
- External:
  ADAM 4520,
  ADVANTECH, (http://www.advantech.com/)
  - DIN rail, automatic RS485 bus supervision, no external data flow control signals, galvanic isolated, baud rate 19200 bps
  - When communication is working the LED on ADAM 4520 is going from full illumination to short darkness then again full illuminated
  - When communication of I-LB is working, PWR and RUN LEDs full red illuminated; TxD and RxD flashing when transmitting

- 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.

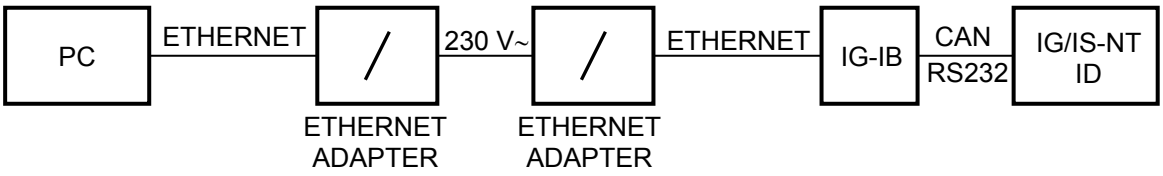
RS232 Bluetooth adapter



Recommended adapter

BL-830 (http://www.brainboxes.com/category/bluetooth.aspx) – doesn't work with Windows Vista

Converter 230 V AC ↔ TCP/IP

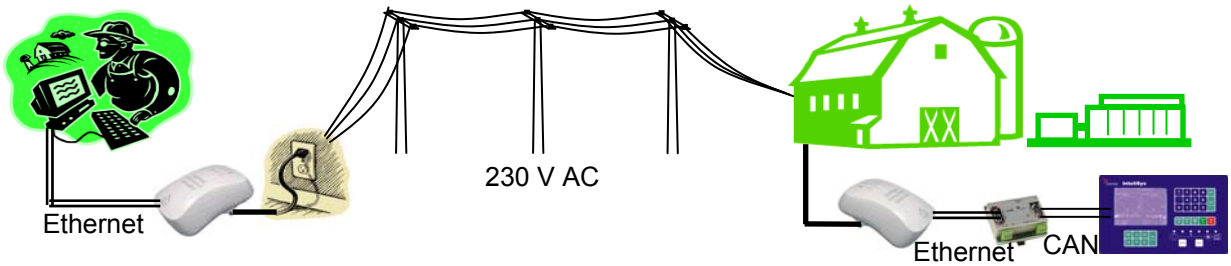


- For installations where IG-IB is used but internet connection is not available
- The connection can be established using electric grid (230 V AC) using Ethernet/230VAC converters
- Can be used for distances up to 200 meters

Recommended converter

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

Example



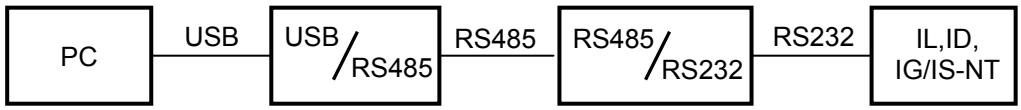
Converter USB ↔ RS232

- 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

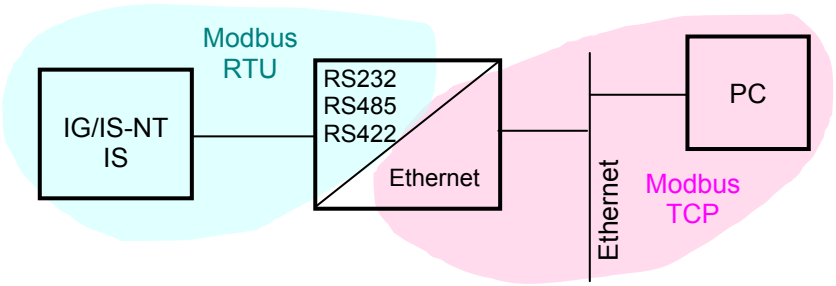


- Extends distance between PC and controller up to 1200 meters

Recommended converter

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

Converter Modbus RTU ↔ TCP



- For connecting of Modbus serial devices (e.g. Inteli controllers) to an Ethernet network

Recommended converter

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

Recommended settings

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/)

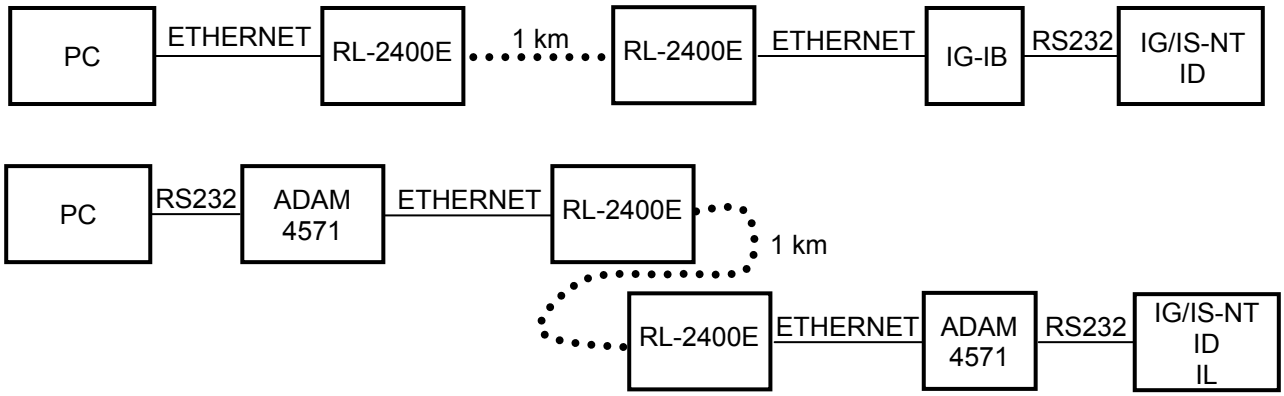
The isolator UC232 can be used instead of UC232-7. The only difference is that UC232 needs external power supply. It can be 5V stabilized or 7-17V unstabilized. The power supply voltage must be specified in the order. Suitable 5V power supply is also available from the Papouch company.

- 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

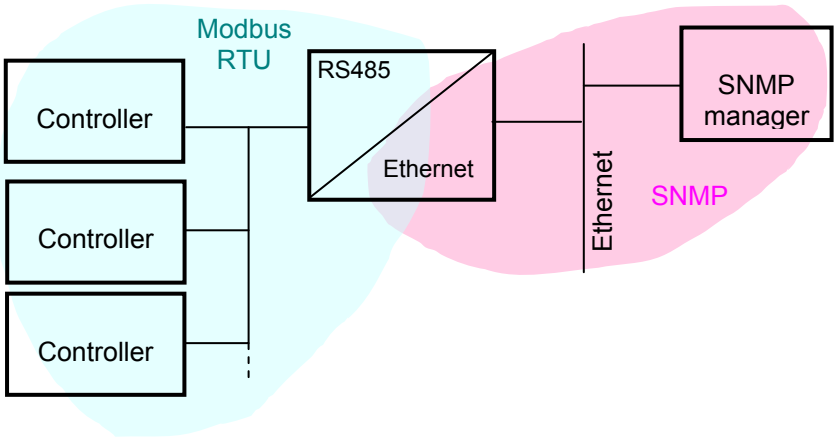


- Useful when the control room is distant from the site
- Can be more economical than to hard wire it

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	
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	

Modbus Register(s)	Com.Obj.	Name	Dim	Type*	Decimals
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.

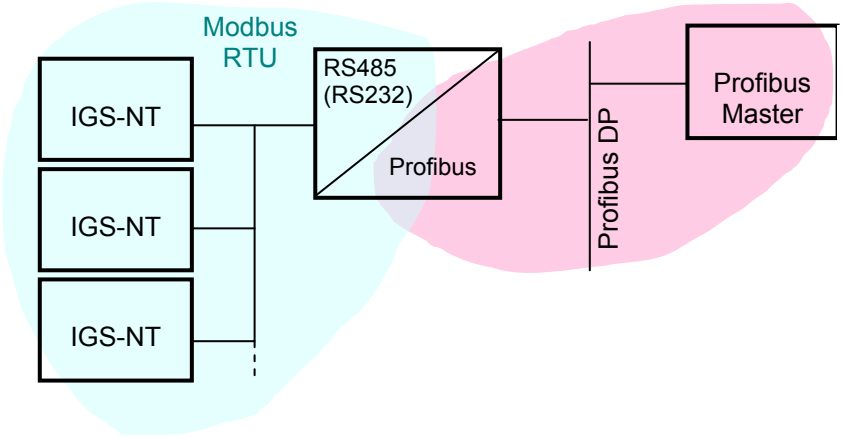
Controller settings

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 NT line 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

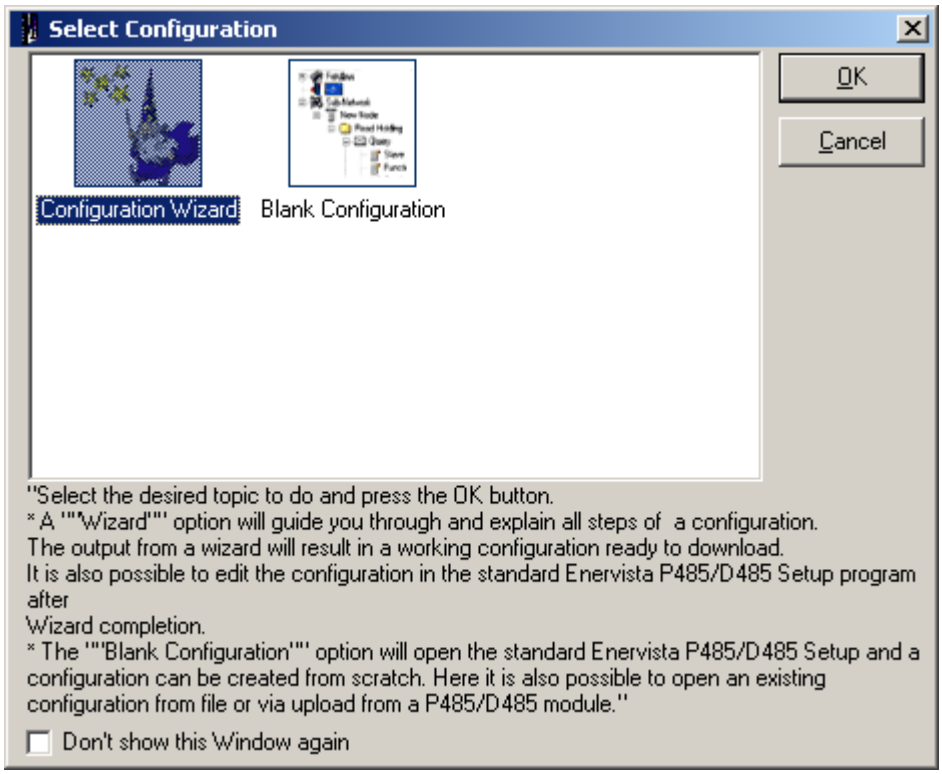
- Use EnerVista software to setup the converter. It can be downloaded from the web page http://pm.geindustrial.com/download/download.asp?id=p485&file=1.
- To configure the converter in the Profibus network, download the description file http://www.geindustrial.com/products/software/d485/P48509E5.zip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

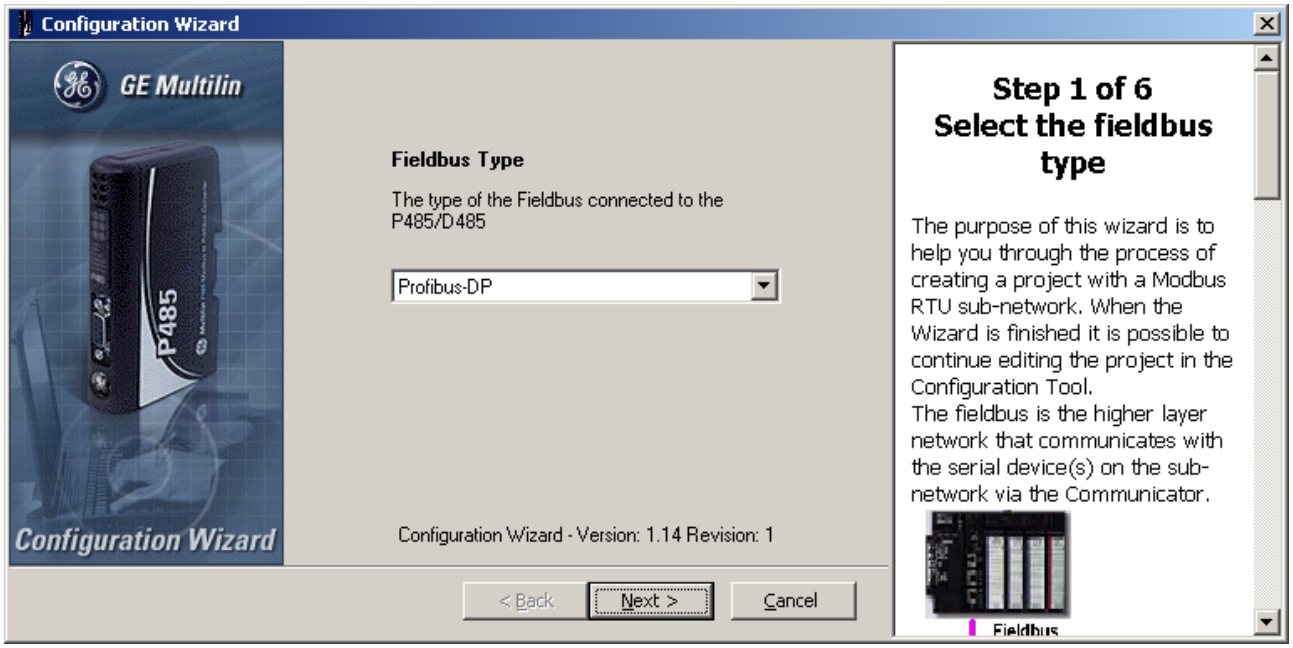
1. The physical layer for Modbus communication is select table. The selected type (RS232/RS485) and speed must be same in the P485 and controller, see Controller settings.
2. Use RS485 in case more controllers are connected to the P485.
3. A *Device* mentioned in the wizard represents a controller type (IG-NT, IM-NT, IS-NT). 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):

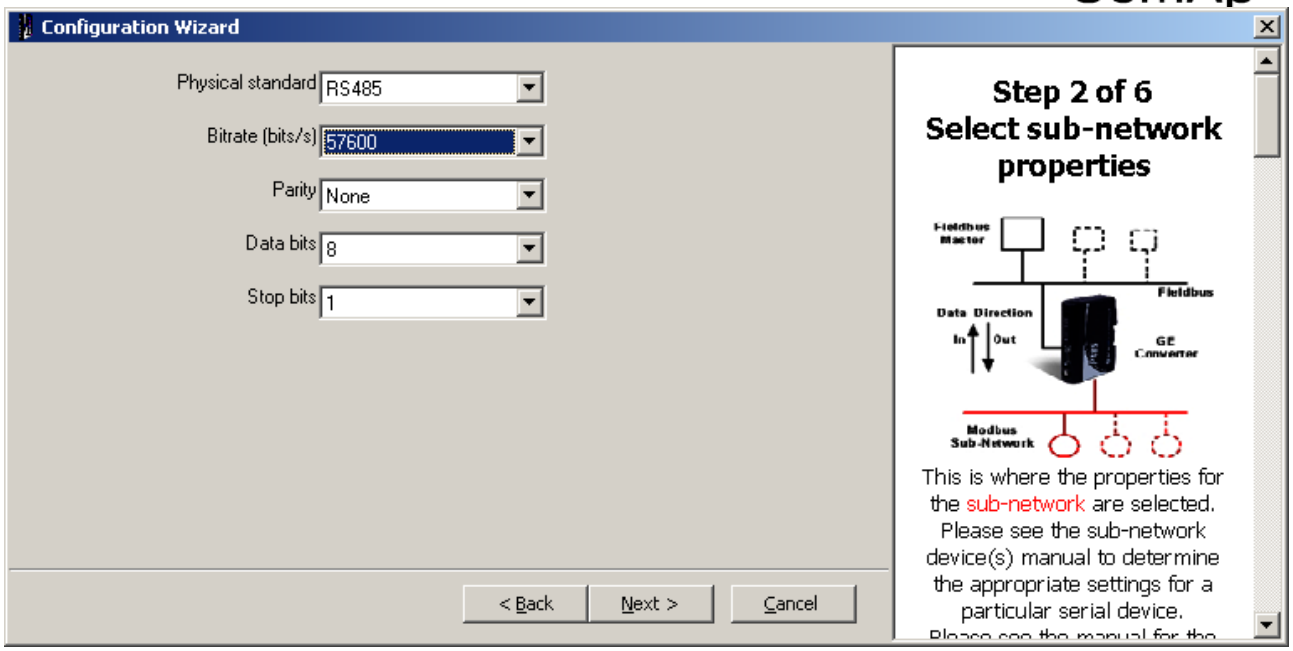
1. Select wizard.



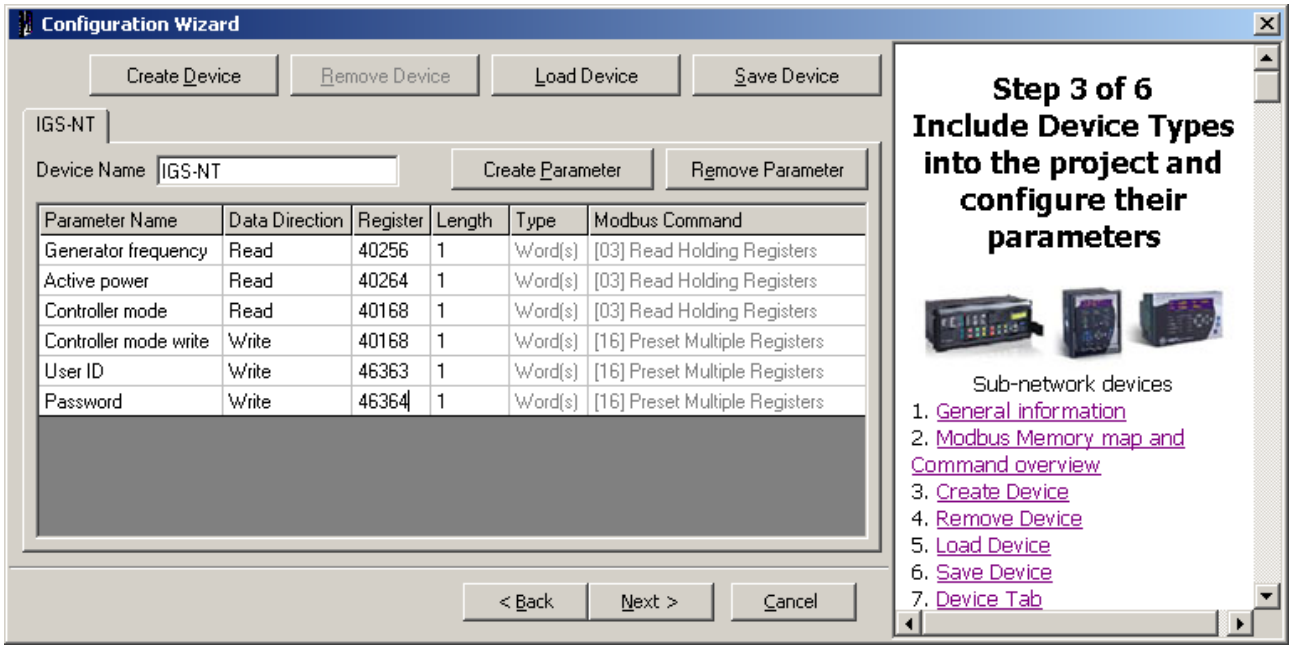
2. Select fieldbus type.



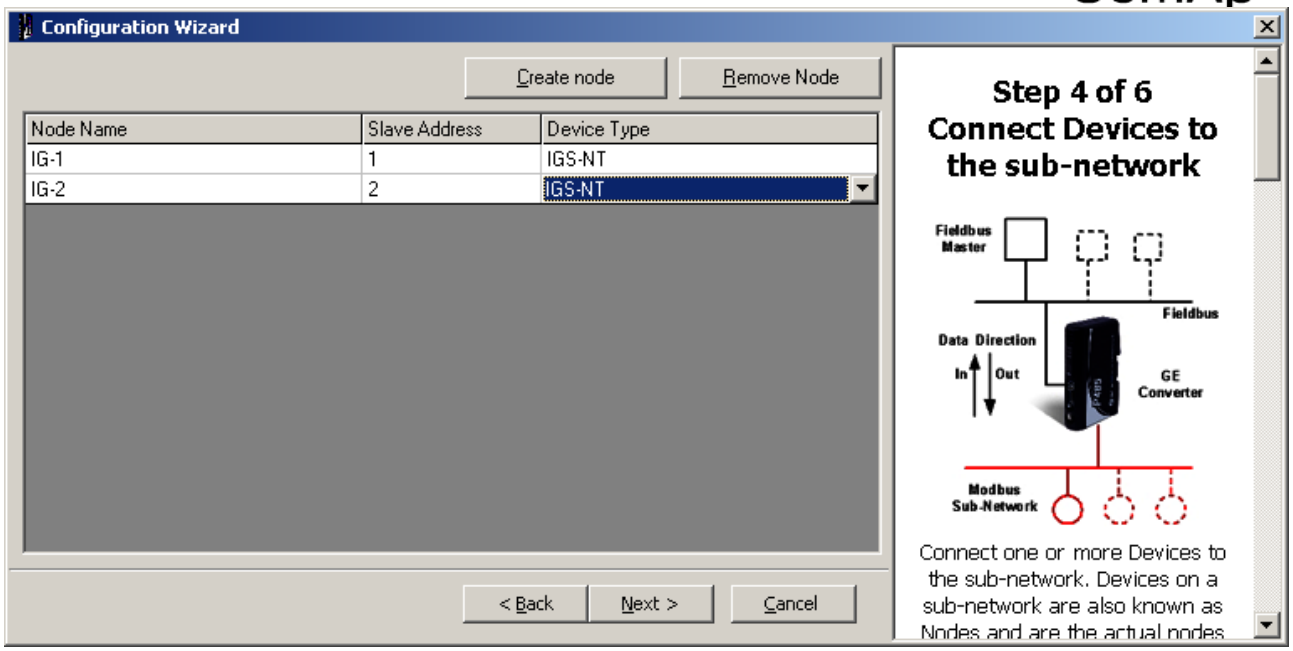
3. Select physical layer and communication parameters for Modbus.



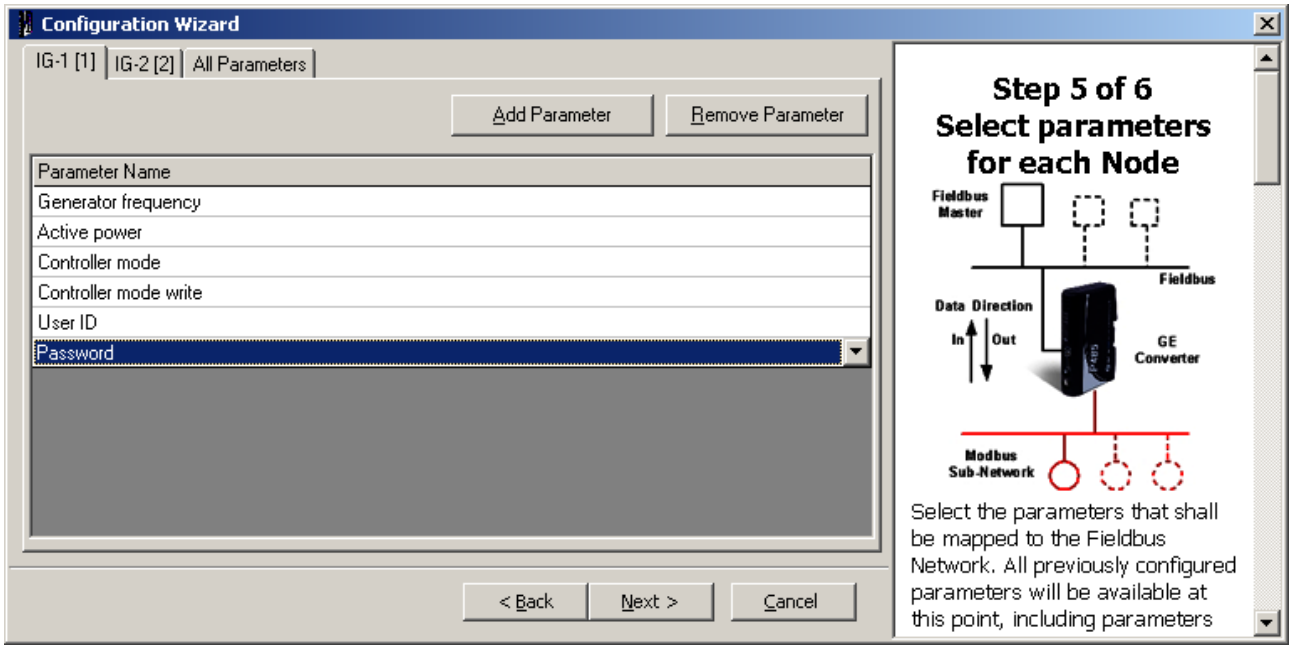
4. Define IGS-NT Device, it's Parameters and related Modbus registers.



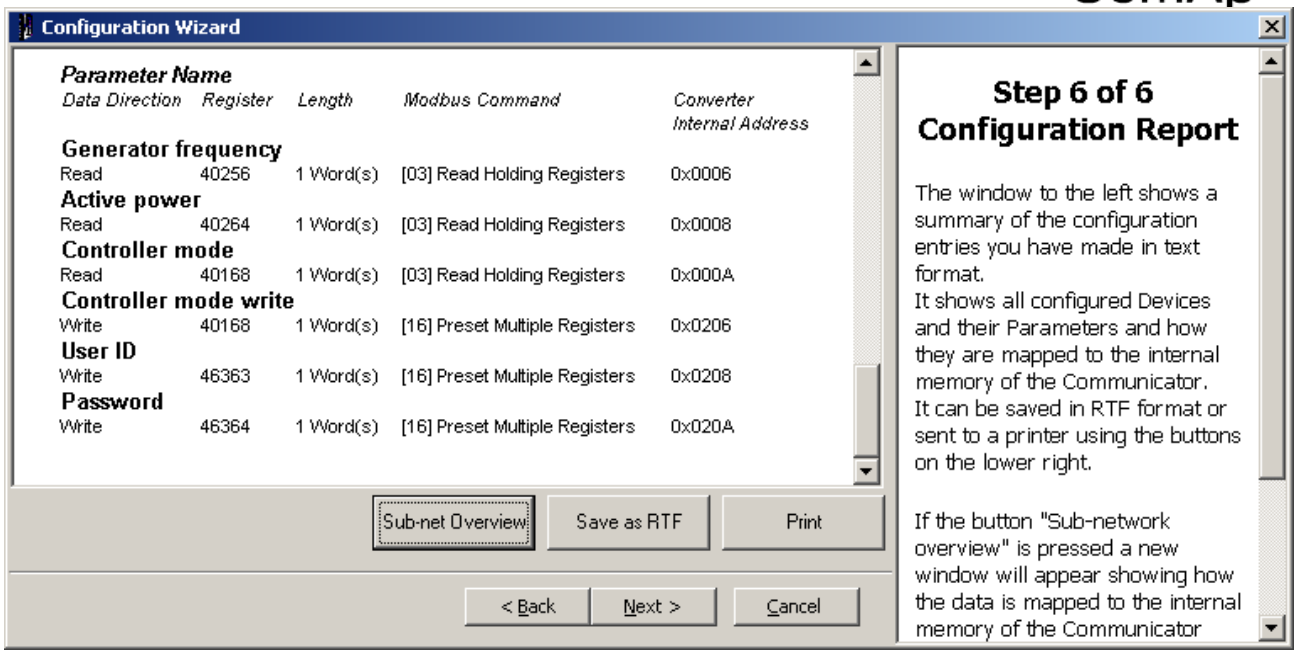
5. Define nodes connected to the Modbus network.



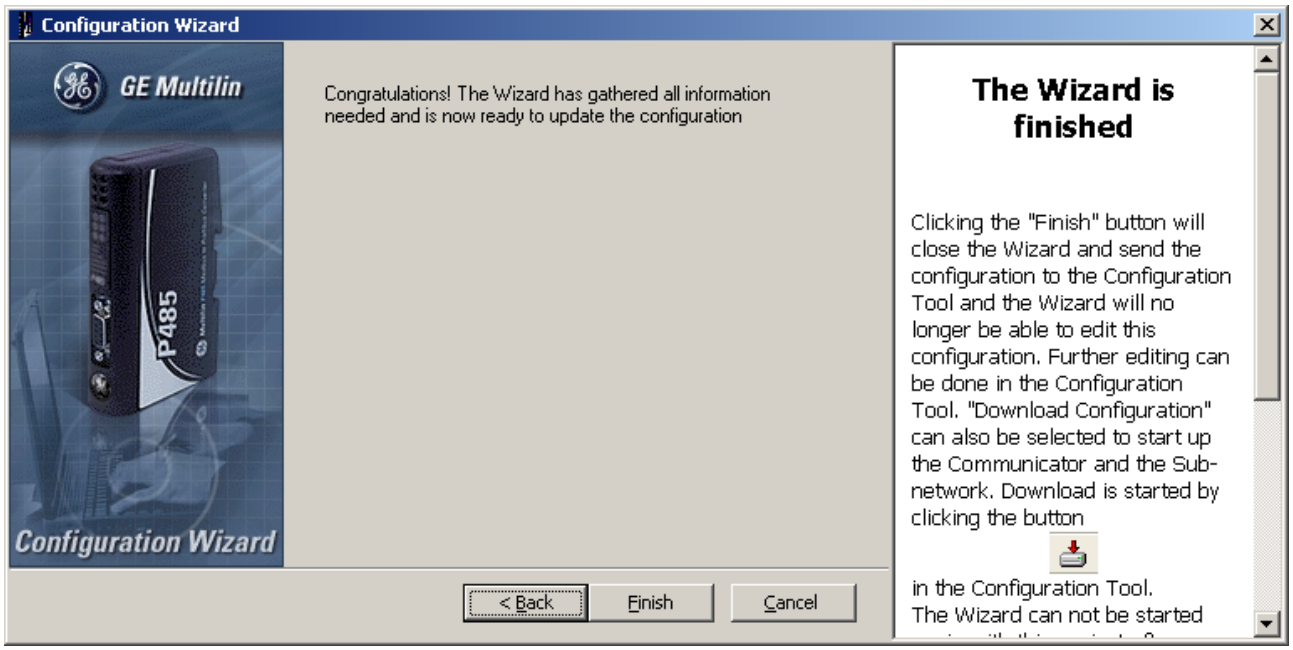
6. For each node select parameters that have to be mapped to the Profibus network.



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

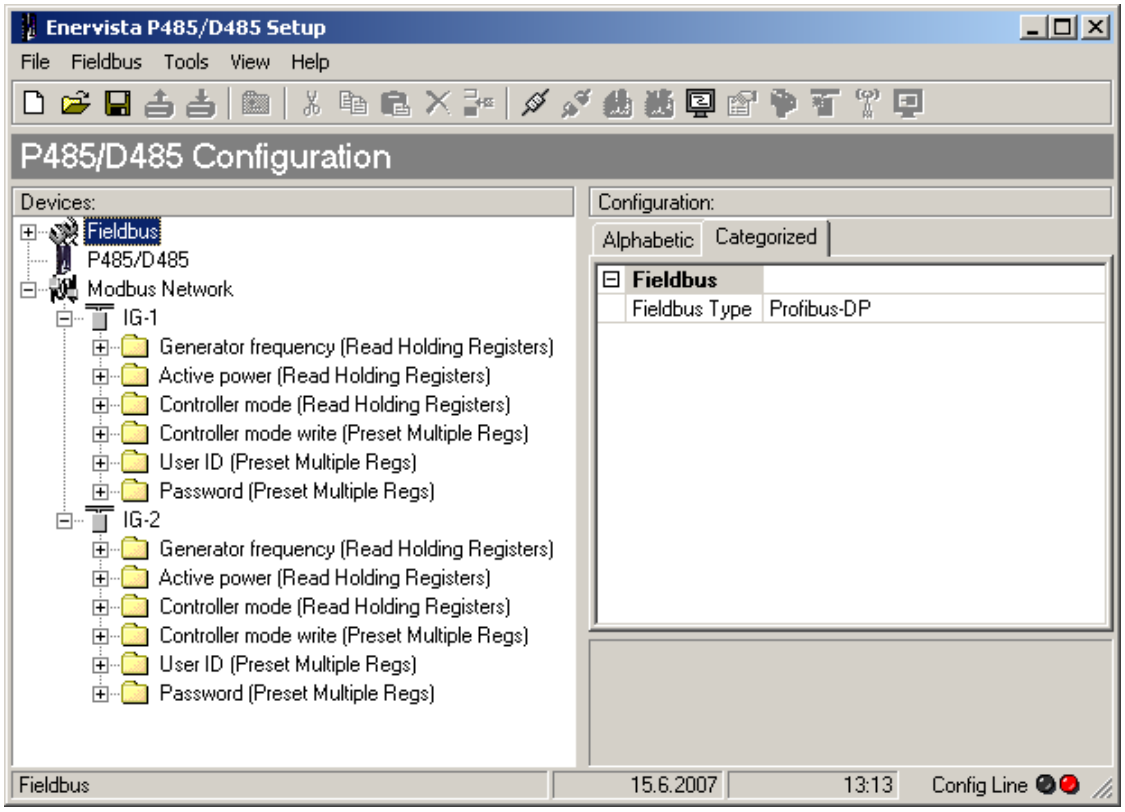


8. The configuration using wizard is finished.

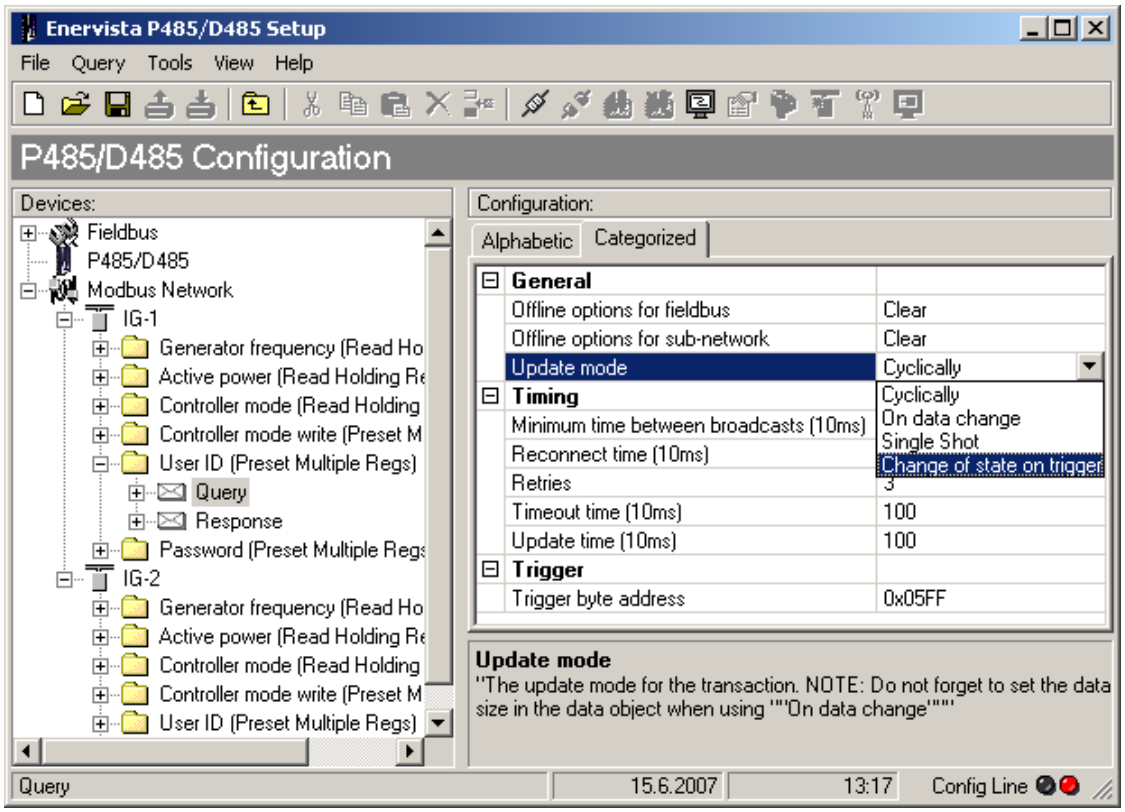


9. Save the configuration to a file.

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.