



IntesisBox[®] Modbus Server

BACnet/IP Client

User's Manual

v10 r14 eng

Issue date: 15/10/2012

© Intesis Software S.L. 2012 All Rights Reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Intesis Software S.L.

Intesis Software S.L.
Milà i Fontanals, 1 bis
08700 Igualada
Spain

TRADEMARKS

All trademarks and tradenames used in this document are acknowledged to be the copyright of their respective holders.

Gateway for the integration of BACnet/IP based systems or devices into Modbus (RTU and TCP) control systems.

Typical application:

Integration of **Daikin VRV** Air Conditioners into a Modbus enabled control system (BMS, PLC, SCADA, Controller...).

Different models are available for this gateway, with the following **Order Codes**:

IBOX-MBS-BAC-100

Model supporting integration of 16 BACnet devices and 100 internal points.

IBOX-MBS-BAC-A

Model supporting integration of 16 BACnet devices and 500 internal points.

IBOX-MBS-BAC-B

Model supporting integration of 16 BACnet devices and 3000 internal points.

INDEX

1.	Description	5
1.1	Introduction	5
1.2	Functionality	6
1.3	Capacity of IntesisBox	7
1.4	Typical applications	8
2.	Modbus interface of IntesisBox	9
2.1	Description	9
2.2	Functions supported	9
2.3	Modbus RTU	9
2.4	Modbus TCP	10
2.5	Address Map	10
2.6	Points definition	10
3.	BACnet/IP interface	11
3.1	Description	11
3.2	Points definition	12
4.	LinkBoxMB. Configuration & monitoring tool for IntesisBox Modbus Server series	13
4.1	Introduction	13
4.2	Project definition	14
4.3	Connections configuration	18
4.4	Signals list	21
4.5	Sending the configuration to IntesisBox	24
4.6	Signals viewer	25
4.7	System commands	26
4.8	Files	27
4.9	Functionality considerations	27
5.	Setup process and troubleshooting	28
5.1	Pre-requisites	28
5.2	Setup procedure	28
6.	Connections	31
7.	BACnet/IP interface specifications	32
8.	Mechanical & Electrical characteristics	33
9.	Dimensions	34
10.	Annexes	35
10.1	Integration of Daikin VRV Air Conditioners into a Modbus enabled control system	35
10.2	Daikin DMS502B51 BACnet Gateway for Daikin Air Conditioning VRV system	36

1. Description

1.1 Introduction

This document describes the integration of Modbus enabled systems with BACnet ASHRAE 135 – 2001 Annex J - BACnet/IP protocol compatible devices or systems using the gateway *IntesisBox[®] Modbus - Server BACnet/IP Client*.

This document assumes that the user is familiar with Modbus and BACnet technology and technical terms.

From now on, and with the aim of easy the read of this document, the abbreviation "IntesisBox" is used instead of the full name IntesisBox[®] Modbus Server - BACnet/IP Client.

The aim of this integration is to make accessible signals and resources of any BACnet/IP based system from a system capable of "talking" Modbus protocol.

IntesisBox allows integrating devices "talking" BACnet/IP, into a supervision/control/automation system "talking" Modbus (PLC, SCADA, Controller...), and in general into any device or system with a Modbus master (TCP or RTU) interface.

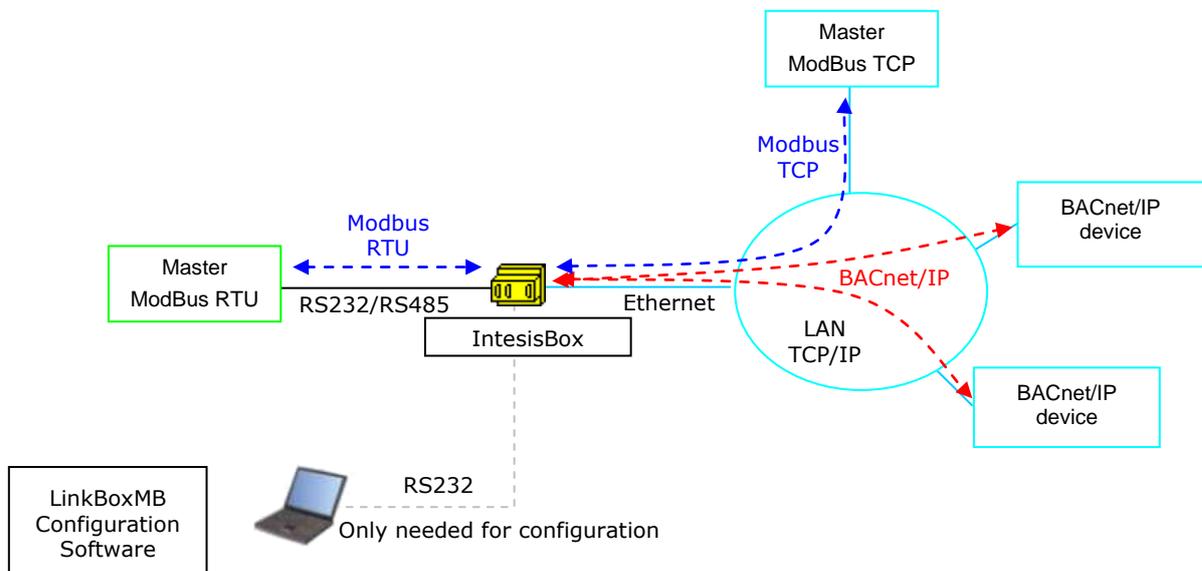
For this, IntesisBox acts as a BACnet/IP Client device in its BACnet interface, reading/writing points of other BACnet/IP device(s), and offering these point's values of other BACnet device(s) through its Modbus interface. The readings into the BACnet device(s) performed by IntesisBox can be by continuous polling or by subscription request (COV), configurable.

1.2 Functionality

Every one of the defined signals into IntesisBox is associated to a Modbus register address, with this, all the system is seen as a single Modbus slave unit from the Modbus system point of view.

From the BACnet system point of view, IntesisBox acts as a client device reading and writing points of other remote BACnet devices connected to the BACnet network.

IntesisBox is a BACnet client device, this means that no BACnet points of IntesisBox can be accessed from other BACnet devices connected to the BACnet network, IntesisBox just offer in its Modbus interface a mirror of those remote BACnet device's points declared into it.



Integration of BACnet and Modbus using IntesisBox Modbus Server - BACnet/IP Client gateway

The integration operation is as follow:

From the BACnet system point of view, every signal in IntesisBox has an associated object of a remote BACnet device, these points in the remote BACnet devices are read from IntesisBox using the present value property (by continuous polling process or by COV subscription requests, configurable), with this, IntesisBox maintains always a mirror of these remote BACnet device's point values in its memory.

Whenever a point in IntesisBox changes its value (written from Modbus side) the new value is written into the remote BACnet device's object associated (using also present_value property).

From the Modbus system point of view, after the start up process, IntesisBox listen for any read or write request received from Modbus system. If it is read request, IntesisBox serves the updated point's values in its memory. If it is a write request, performs the writing of the remote BACnet device's point associated immediately.

1.3 Capacity of IntesisBox

Element	Max. 100 version	Max. A version	Max. B version	Notes
Type of BACnet devices				Only those supporting <i>BACnet ASHRAE 135 – 2001 Annex J - BACnet/IP protocol</i> , commonly referred as BACnet/IP.
BACnet points supported	100	500	3000	Maximum number of points that can be defined into IntesisBox.
BACnet/IP devices supported	16	16	16	Maximum number of different BACnet/IP devices that can be defined into IntesisBox (to read/write points into them).

There are two different models of *IntesisBox® Modbus Server - BACnet/IP Client* with different capacity each of them.

- 100 model supporting integration of up to 100 internal data points and 16 BACnet devices. *Ref.: IBOX-MBS-BAC-100.*
- A model supporting integration of up to 500 internal data points and 16 BACnet devices. *Ref.: IBOX-MBS-BAC-A.*
- B model supporting integration of up to 3000 internal data points and 16 BACnet devices. *Ref.: IBOX-MBS-BAC-B.*

* If 32bit Modbus registers are used, the maximum number of points allowed by the LinkBoxMB license is reduced by half. In this case, the maximum number is 50, 250, and 1500 communication objects for the IBOX-MBS-BAC-100, the IBOX-MBS-BAC-A and the IBOX-MBS-BAC-B respectively.

IntesisBox model	Maximum Number of points	
	16 bit Modbus registers	32 bit Modbus registers
IBOX-MBS-KNX-100	100	50
IBOX-MBS-KNX-A	500	250
IBOX-MBS-KNX-B	3000	1500

1.4 Typical applications

One very typical application of IntesisBox is the integration of Daikin VRV Air Conditioners into a Modbus enabled control system (BMS, PLC, SCADA, Controller...). For this application, Daikin VRV system must be equipped with Daikin BACnet gateway (model DMS502B51), this Daikin gateway is normally commissioned by Daikin technical personnel, contact your nearest Daikin distributor for details.

If you are interested in this application consult ANNEXES section for more details about the configuration of IntesisBox for this application. You can also contact Intesis Software to obtain sample projects for IntesisBox, these sample projects contain specific signal tables for Daikin VRV system, with these sample projects the configuration of IntesisBox for this application becomes easy and quick, almost plug & play.

2. Modbus interface of IntesisBox

This section describes the Modbus part of the IntesisBox's configuration and functionality. This section assumes the user is familiar with Modbus technology and technical terms.

2.1 Description

IntesisBox acts as a slave device in its Modbus interface, this interface can be the Ethernet port (if using Modbus TCP), or the RS232 port or the RS485 port (if using Modbus RTU). To access the points and resources of the IntesisBox from a Modbus master device, you must specify as the Modbus register addresses, those configured inside IntesisBox corresponding to BACnet points. See details below in this document.

It can only be one Modbus Mode active simultaneously in IntesisBox, Modbus RTU or Modbus TCP.

2.2 Functions supported

This part is common for Modbus RTU and TCP.

Modbus functions 03 and 04 (*read holding registers* and *read input registers*) can be used to read Modbus registers.

Modbus functions 06 and 16 (*Single Multiple Holding Registers* and *Write Multiple Holding Registers*) can be used to write Modbus registers.

If *poll records* are used to read or write more than one register, it is necessary that the range of addresses requested contains valid addresses, if not the corresponding Modbus error code will be returned.

The Modbus registers can be of 2 bytes (16 bits) or of 4 bytes (32 bits).

- For 2 bytes (16 bits) registers, its content is expressed always in MSB..LSB.
- For 4 bytes (32 bits) registers, its content (the way IntesisBox handles the byte order) can be one of 3 different options, configurable, see details below in this document (this is necessary because different Modbus master systems handle differently byte order of 32 bits registers).

Modbus error codes are fully supported, they will be sent whenever a non valid Modbus action or address is required.

2.3 Modbus RTU

Baud rate can be selected from 1200, 2400, 4800, 9600, 19200, 38400 and 56700.

Data Bits: 8, parity: none, Stop Bits: 1.

Modbus slave number can be configured. Physical connection (RS232 or RS485) can also be selected.

Only the lines RX, TX and GND of the RS232 connector are used (TX/RX+ and TX/RX- for RS485).

2.4 Modbus TCP

The TCP port to use can be configured (by default 502 is used).

The IP address, subnet mask and default router address to use by IntesisBox can be also configured.

2.5 Address Map

The Modbus address map is fully configurable, any point in the IntesisBox can be freely configured with the desired Modbus register address.

2.6 Points definition

Every point defined in IntesisBox has the following Modbus features associated to it, that can be configured (optionally or mandatory depending on the feature):

Feature	Description
Format (Data coding) *	Data type to encode the signal's value. It will depend on the BACnet type of signal associated to it in every case. Possible values are: 16 bits unsigned, 16 bits signed, 16 bits signed * 10, 32 bits unsigned, 32 bits signed, 32 bits float.
Point *	Is the Modbus register address for the point.
R/W *	Type of signal. Always from the Modbus system point of view. Possible values are: R (read only signal), R/W (read/write signal).

* Configuration for this feature is mandatory.

3. BACnet/IP interface

This section describes the BACnet part of the IntesisBox's configuration and functionality. This section assumes the user is familiar with BACnet technology and technical terms.

3.1 Description

The BACnet systems are represented in the form of *Devices* holding *Objects*, normally every physical device corresponds to a logical one. The *Objects* can be of different type depending on their task: analog input, analog output, digital input, etc. Up to 23 different types of objects. Every *Object* has different properties, the one more commonly used, and the one used to read and write values by IntesisBox is the *Present Value* property that indicates the real value of the object. Every Object of the same type in a *Device* is identified with its associated *Object Instance*.

These are the possible BACnet signals supported by IntesisBox:

Object Type	Property	Description
Analog Input	Present Value	Analog signal. i.e. Ambient temperature.
Analog Output	Present Value	Analog signal.
Analog Value	Present Value	Analog signal. i.e. Temperature set point value.
Binary Input	Present Value	Digital signal. i.e. ON/OFF status.
Binary Output	Present Value	Digital signal. i.e. ON/OFF command.
Binary Value	Present Value	Digital signal. i.e. ON/OFF status/command.
Multistate Input	Present Value	Multistate signal. i.e. Working mode status.
Multistate Output	Present Value	Multistate signal.
Multistate Value	Present Value	Multistate signal. i.e. Working mode command.

Multistate signals represent integer values, the same than binary ones but with more than 2 possible statuses. *Input* signals are read only signals.

Every signal is identified with its associated *Device* + *Object Type* + *Object Instance*.

IntesisBox simulates a virtual BACnet device inside the BACnet system acting as client device in the BACnet system. The communication with the rest of the BACnet devices is done via the Ethernet port of the gateway which implements the BACnet/IP protocol.

3.2 Points definition

Every point defined in IntesisBox has the following BACnet features associated to it:

Feature	Description
BACnet Device	<p>BACnet device to which belongs the point, from a list of BACnet devices that can be defined in IntesisBox (up to 16).</p> <p>For every BACnet device defined, a virtual signal is created automatically in IntesisBox to inform about the communication with the BACnet device, this signal is available also from the Modbus interface like the rest of points.</p>
BACnet object type	<p>BACnet object type for the point. It can be one of the following BACnet object types supported by IntesisBox :</p> <p>AI = Analog Input. AO = Analog Output. AV = Analog Value. DI = Digital Input. DO = Digital Output. DV = Digital Value. MI = Multistate Input. MO = Multistate Output. MV = Multistate Value.</p> <p><i>Consult documentation of BACnet device(s) to integrate for information about BACnet object types of the points desired to integrate.</i></p>
BACnet object instance	<p>BACnet object instance for the point.</p> <p><i>Consult documentation of BACnet device(s) to integrate for information about BACnet object instance of the points desired to integrate.</i></p>
BACnet object property	<p>BACnet object property for the point. The only property permitted is Present_Value. IntesisBox uses Present_Value property to read/write points into other BACnet devices.</p>

4. LinkBoxMB. Configuration & monitoring tool for IntesisBox Modbus Server series

4.1 Introduction

LinkBoxMB is a Windows™ compatible software developed specifically to monitor and configure IntesisBox Modbus Server series. It is possible to configure all external protocols available for IntesisBox Modbus Server and to maintain different customer's configurations based on a LinkBoxMB project for every different installation. Maintaining always on hard disk a copy of the last configuration files for every external protocol and customer, that is to say for every project.

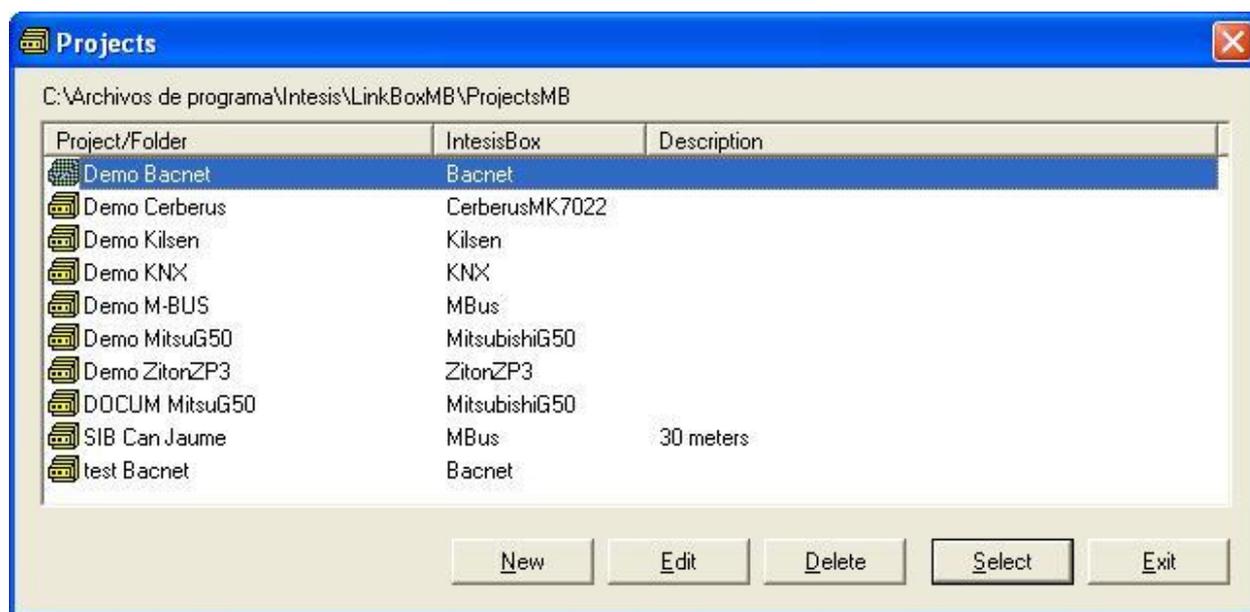
From LinkBoxMB, as well as configure the integration signals list and connection parameters for every external protocol, it is permitted to select the serial port to use to connect to IntesisBox and the use of some tools for monitoring and debugging de device. Some of these tools will be explained in this document but only some of them, the rest of available debugging tools and commands will not be explained here because they are for exclusive use under the recommendations of Intesis Software technical support.

LinkBoxMB allows configuring all IntesisBox Modbus Server series independently of the external system used. For every external system, LinkBoxMB has a specific configuration window. Periodically, new free versions of LinkBoxMB are released incorporating the latest developed integrations for external systems.

4.2 Project definition

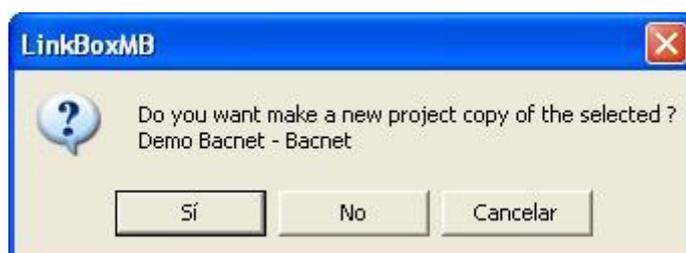
The first step to do in LinkBoxMB for a new installation is to create the installation's project giving a descriptive name to it. When you create a project, a new folder is created with the name of the project containing the configuration files needed depending on the external protocol selected for the project. It is strongly recommended that you create a new project for every installation, if not, overwriting of configuration files of previous installations using the same external protocol may occur, losing the configuration data for those previous installations. The projects folder is located in AppFolder\ProjectsMB, where AppFolder is the installation folder of LinkBoxMB (by default C:\Program Files\Intesis\LinkBoxMB). Inside the projects folder, a new folder will be created for every project defined in LinkBoxMB with the files needed for the project.

When you open LinkBoxMB, the project selection window will appear inviting you to select a project or create a new one. A demo project for every external protocol supported is provided with the standard installation of LinkBoxMB. You can create a new project or select a demo project based on the external protocol desired, and create a new one from the demo one selected.

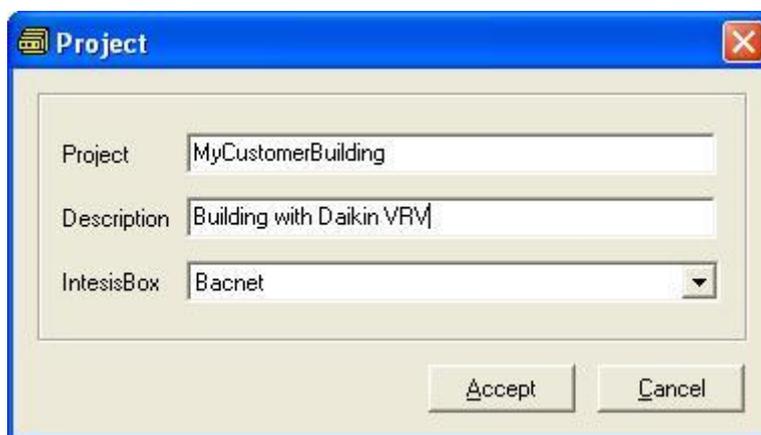


Project selection window

To create a new project, select a project using the same external protocol you want to use in the new project and click on *New* button. You will be prompted to create a copy of the selected project (useful for similar installations) or create a new one.



If you select *Yes* you will be prompted to specify a name and a description for the new project that will be based on the same external protocol than the selected one. If you select *No* you can specify a name, a description and an external protocol to use from the list of available external protocols.



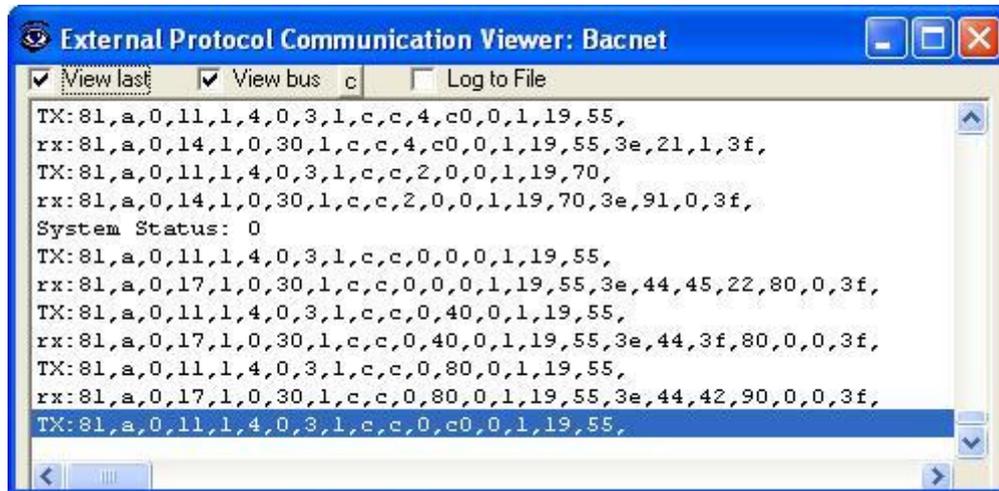
On *Accept*, a new folder will be created inside the projects folder with the name given to the project, this folder will contain the template configuration files if the project is a brand new one, or a copy of the configuration files if it is a copy of a selected one.

A description of the files created for a BACnet protocol based project can be found in section *Files* in this document.

From all the possibilities of LinkBoxMB, only changes in configuration for the integration and configuration file generation can be performed while disconnected from IntesisBox (working off-line), allowing you to do these tasks in the office more comfortably. Before any monitoring or downloading action to IntesisBox can be performed, the connection between IntesisBox and the PC running LinkBoxMB must be established (working on-line). To do so follow these steps:

1. Make sure IntesisBox is powered-up a correctly connected to the Modbus system via the Ethernet connection (Modbus TCP) or serial connection (Modbus RTU) and to the BACnet system via the Ethernet port (consult details for connection and pin assignments in section *Connections* of this document).
2. Connect a free PC serial port to the IntesisBox serial port marked as *PC Console*. (Use the standard serial cable supplied with the device or a customer's cable following the pin assignments specified in section *Connections* in this document).
3. Select in LinkBoxMB the PC serial port used for the connection to IntesisBox. Use menu Configuration -> Connection.

To monitor the communication between IntesisBox and the external system (BACnet in this case), select the menu *View -> Bus -> BACnet*. The *External protocol communication viewer* window will be opened. This window show in real time all the communication frames between IntesisBox and BACnet devices as well as debugging messages referent to external protocol (BACnet) sent by IntesisBox.



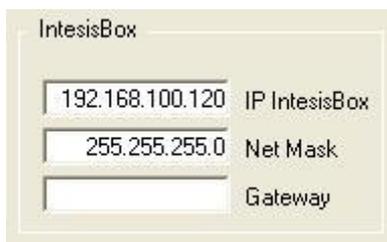
To configure the IntesisBox's connection parameters and the signals list, select menu *Configuration -> IntesisBox*. The *BACnet/IP client Configuration* window will be opened.

4.3 Connections configuration

Select the Connection tab to configure the connection parameters.

Three kinds of information are configured using this window, the referent to the IntesisBox IP parameters, the referent to the Modbus interface and the referent to the BACnet/IP interface.

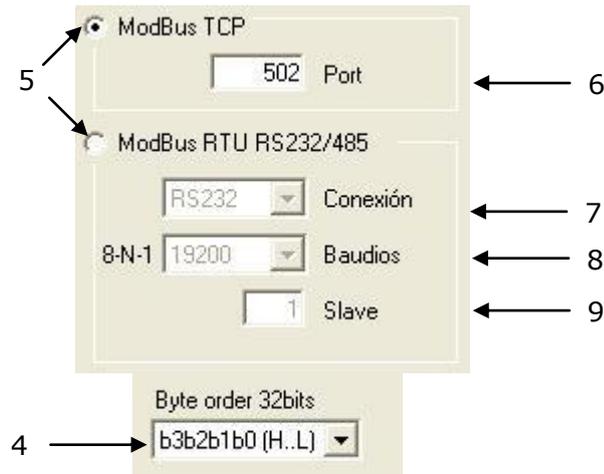
IntesisBox IP parameters:



IntesisBox IP parameters

1. Enter the IP address for IntesisBox.
2. Enter the IP netmask for IntesisBox.
3. Enter the default router address for IntesisBox, leave blank is there is no need of router.

Modbus interface configuration parameters:



Modbus Interface Configuration

4. Byte order of 32 bits registers, it can be one of these three possibilities:

- b3b2b1b0 (H...L).
- b0b1b2b3 (L...H).
- b1b0b3b2.

This has been implemented because Modbus master devices handle 32 bits registers in these three different ways depending on manufacturer and model. If your Modbus master device does not read correctly the 32 bits registers of IntesisBox and you do not know the way byte order is handled, then just try the three possibilities (do not forget to send again the configuration to IntesisBox whenever you change it).

5. Select the type of Modbus protocol to use (TCP or RTU).

If Modbus TCP is selected, then:

6. Enter the TCP port to use, by default 502.

If Modbus RTU is selected, then:

7. Select the type of connection used (RS232 or RS485).

8. Select the baud rate.

9. Enter the Modbus slave number to use by IntesisBox.

BACnet interface configuration parameters:

BACnet Interface Configuration

1. Check the BACnet devices you want to activate.

Select a device to configure its properties:

2. Enter the IP address configured in the BACnet device.
3. Enter the device number configured in the BACnet device.
4. Enter the type of reading desired (continuous Polling, or COV subscriptions). To minimise traffic in the BACnet network, COV is recommended (remember that the BACnet device must support COV subscription requests).
5. Enter the port number used by the BACnet device (by default 47808, which is BAC0).
6. Enter the BACnet device name (optional, just for identification purposes).
7. Select the gateway model used: basic or extended.

There are two models of IntesisBox supporting different number of points. You can identify the model of IntesisBox by the **order code** printed in the front label:

- **IBOX-MBS-BAC-A**. Basic model supporting up to 500 points.
- **IBOX-MBS-BAC-B**. Extended model supporting up to 4000 points.

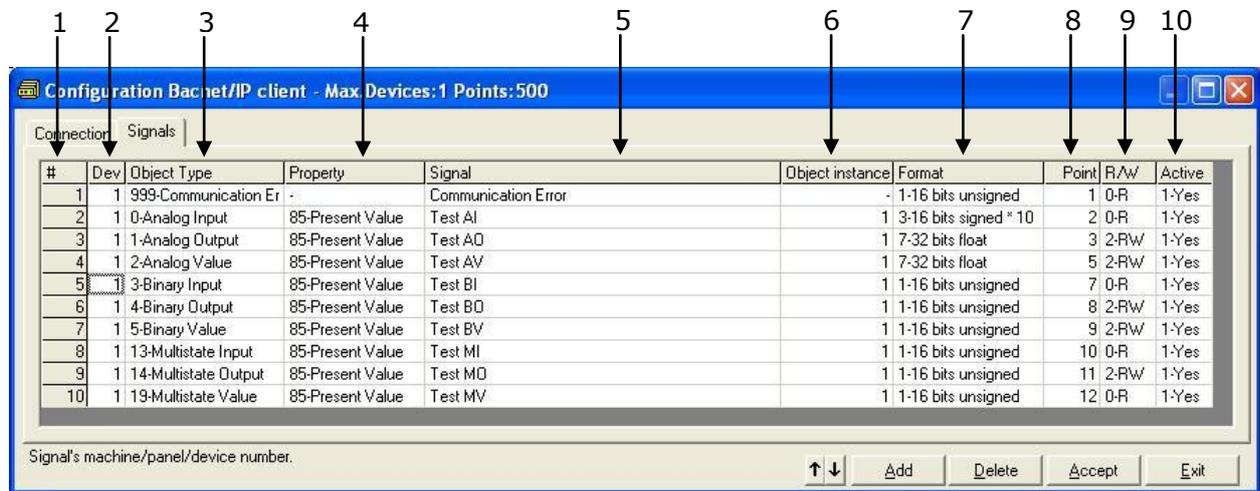
You can identify also the model of your IntesisBox through the identification given by it in response to an INFO command, it is something like this:

IntesisBox_MODBUS_SVR_BACNET_IP-**A**... -> this is the basic model (up to 500 internal points)

IntesisBox_MODBUS_SVR_BACNET_IP-**B**... -> this is the extended model (up to 4000 internal points)

4.4 Signals list

Select the Signals tab to configure the signals list (the IntesisBox's internal points).



Signals list

11

1. #. Signal's number (edit not permitted). Every row in the grid corresponds to a signal (point). Signals (rows in the grid) can be added or deleted selecting the desired row and clicking *Add* or *Delete* buttons. Deletion action can be executed for a single row or even for some consecutive rows, all the rows selected will be deleted. This column is used only to enumerate the rows in the grid (signals).
2. *Dev.* device number to which belongs the point. Referenced to the list of devices defined in *Connection Tab*. Note that this is not the device number configured in the BACnet device itself.
3. *Object Type*. BACnet object type for the signal. Edit using the mouse right-button-click pop-up menu available on the column as showed in the figure below. One of the showed object types can be selected:

- 0-AI
- 1-AO
- 2-AV
- 3-BI
- 4-BO
- 5-BV
- 13-MI
- 14-MO
- 19-MV

- AI = Analog Input.
- AO = Analog Output.
- AV = Analog Value.
- DI = Digital Input.
- DO = Digital Output.
- DV = Digital Value.

MI = Multistate Input.

MO = Multistate Output.

MV = Multistate Value.

Consult documentation of BACnet device(s) to integrate for information about BACnet object types of the points desired to integrate.

The *Communication error* virtual signals have always a predefined special Object type, not changeable.

4. *Object property*. It's the BACnet object property to use by IntesisBox to read/write points into BACnet devices. Only Present_Value property available. Edit not permitted.
5. *Signal*. Signal's description. Used to describe the signal at user level.
6. *Object Instance*. Object instance for the point into the BACnet device.
Consult documentation of BACnet device(s) to integrate for information about BACnet object instances of the points desired to integrate.
7. *Format*. Modbus data format to encode this signal's value. Edit using the mouse right-button-click menu available on the column, as showed in the figure below, and select one of the possible values.



1-16 bits unsigned
2-16 bits signed
3-16 bits signed * 10
5-32 bits unsigned
6-32 bits signed
7-32 bits float

8. *Point*. Modbus register address for the point. **Note that every address entered here corresponds to a 16 bits register, if you select a format of 32 bits for the point, then it takes two consecutive addresses in the address map, this means that, for example, if you have one point with address 1 and a format of 32 bits float, then address 2 can not be used for any other point, the subsequent free address to use will be 3.**

You can enter directly the address desired individually in every row or auto enumerate for some consecutive rows, for this last proceed as follow.

- Select using the left mouse button (clicking and dragging) the *Point* field of all the rows in the list to which you want to automatically assign addresses (must be consecutive rows).
- Click right mouse button over the selected fields and select *Auto Enumeration* option from the pop-up menu that will appear.

Point	R/W	Active
1		1-Yes
2		1-Yes
3		1-Yes
0	0-R	1-Yes
	1-RW	1-Yes
	1-RW	1-Yes
0	1-RW	1-Yes
	1-RW	1-Yes
	1-RW	1-Yes
	1-RW	1-Yes

- Enter the first address to assign.

The dialog box titled "Auto Enumeration" has a text input field labeled "Point" containing the number "4". Below the input field are "Accept" and "Cancel" buttons.

- Enter the increment between consecutive assignments. For example selecting 1 for the first address and an increment of 1, the addresses generated will be 1,2,3,4.. and so on.

The dialog box titled "Auto Enumeration" has a text input field labeled "Enter the increment" containing the number "1". Below the input field are "Accept" and "Cancel" buttons.

9. *R/W*. Signal type (from the Modbus system point of view). Edit not permitted. Possible values are *R* for read only signals or *R/W* for Read/write signals. This field will be automatically configured by LinkBoxMB according to the BACnet object type selected for the signal.
10. *Active*. Indicates if the signal is active or not for the integration. Possible values: 0-No, 1-Yes. Edit using the mouse right-button-click menu available on the column.
11. Buttons to move the selected row (or rows) up or down inside the grid. To move up or down inside the grid a single row or a group of consecutive rows, just select the row or rows using the left button of the mouse and push the desired up or down button. (This can be done also using the key combinations *ALT+arrow up* or *ALT+arrow down* instead of up or down buttons).

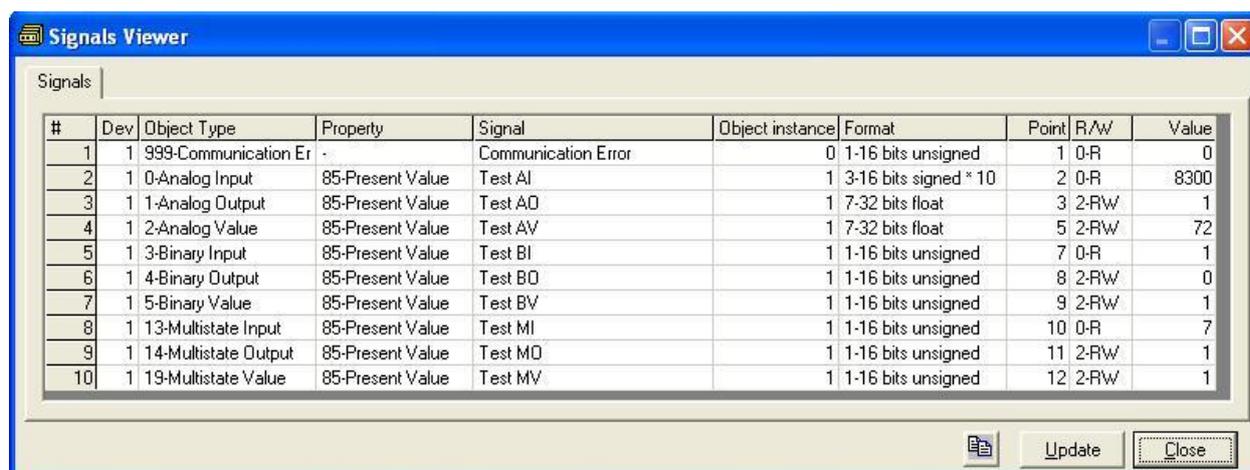
4.5 Sending the configuration to IntesisBox

When the configuration has been saved (button *Accept*) and the IntesisBox configuration binary file has been generated (remember to select yes when asked if you want to generate the IntesisBox file), to send the configuration file to IntesisBox click on the button **Send File**. The process of file transmission can be monitored in the *IntesisBox Communication Console* window. If the file transmission is ok, IntesisBox will reboot automatically with the new configuration loaded.

Remember that saving the configuration and generating the IntesisBox file only saves to the hard disk on the PC the configuration files. **Do not forget to send the configuration file to the IntesisBox using button *Send File*.**

4.6 Signals viewer

Once IntesisBox is running with the correct configuration, to supervise the status of the configured signals, select menu *View -> Signals*. The Signals Viewer window will be opened. This window shows all the active IntesisBox's signals with its main configuration parameters and its real time value in the column Value. After a reset of IntesisBox or after sending a configuration file to the IntesisBox, all the signal's values will be updated automatically in the signals viewer, in case you connect to the IntesisBox when it is already running, you should press the *Update* button to get updated values, press just once the button to update all the signals values, since that moment the signal values will be maintained updated until the connection is closed.



#	Dev	Object Type	Property	Signal	Object instance	Format	Point	R/W	Value
1	1	999-Communication Er	-	Communication Error	0	1-16 bits unsigned	1	0-R	0
2	1	0-Analog Input	85-Present Value	Test AI	1	3-16 bits signed * 10	2	0-R	8300
3	1	1-Analog Output	85-Present Value	Test AO	1	7-32 bits float	3	2-RW	1
4	1	2-Analog Value	85-Present Value	Test AV	1	7-32 bits float	5	2-RW	72
5	1	3-Binary Input	85-Present Value	Test BI	1	1-16 bits unsigned	7	0-R	1
6	1	4-Binary Output	85-Present Value	Test BO	1	1-16 bits unsigned	8	2-RW	0
7	1	5-Binary Value	85-Present Value	Test BV	1	1-16 bits unsigned	9	2-RW	1
8	1	13-Multistate Input	85-Present Value	Test MI	1	1-16 bits unsigned	10	0-R	7
9	1	14-Multistate Output	85-Present Value	Test MO	1	1-16 bits unsigned	11	2-RW	1
10	1	19-Multistate Value	85-Present Value	Test MV	1	1-16 bits unsigned	12	2-RW	1

The signals viewer can be used although only one system is connected to the IntesisBox, *Modbus* or *BACnet*, and is very useful for supervision and test.

It is possible to force a specific value to any signal for test purposes, to do so just double click on the row and select the desired value and Accept in the Data Test window. If the signal is of type R or RW, then the value entered will be available to be read from Modbus master device, if the signal is of type RW, then the value entered will be sent to the BACnet system.

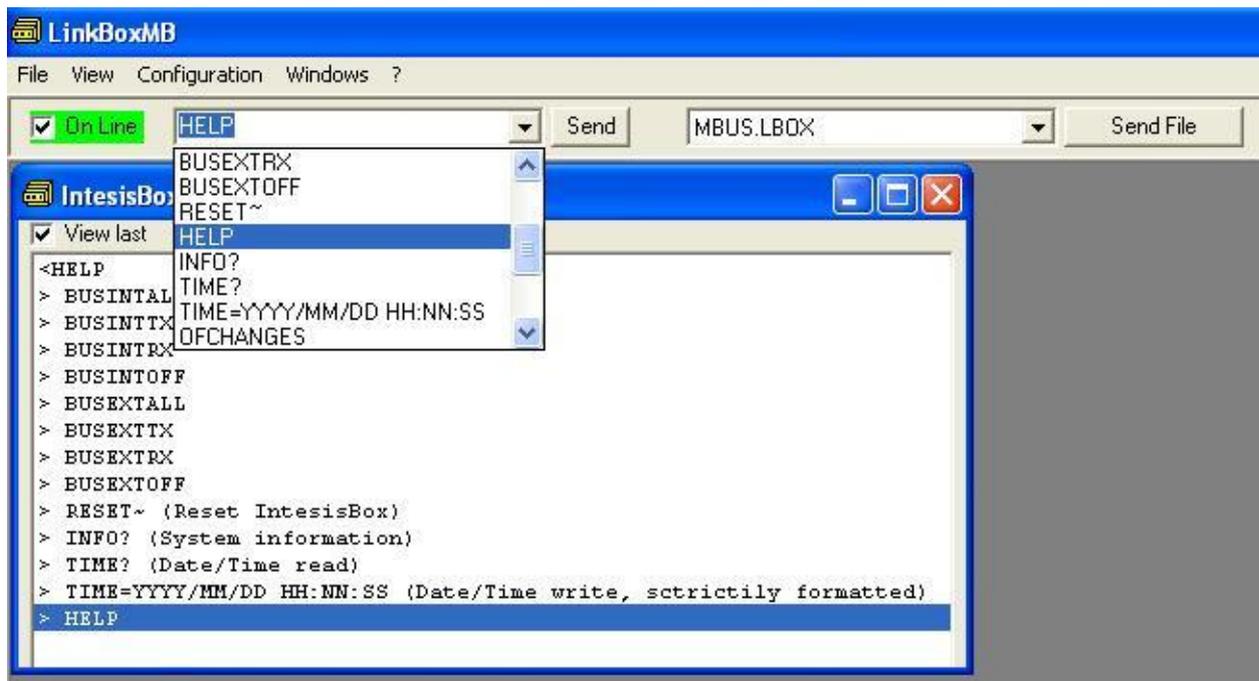


This tool is very useful to test any of the systems connected to IntesisBox, *Modbus* and *BACnet* without the need to actuate on the real signals.

The signals viewer window has a button to copy to the Windows Clipboard all the contents of the window (in tab separated text format).

4.7 System commands

LinkBoxMB includes an option to send to IntesisBox a set of system commands for debugging and control purposes; this list is available in the commands list as shown in the figure below. To send a command to IntesisBox just select it from the list, or type it with the correct format, and press *Enter* or click on button *Send*. IntesisBox will act accordingly with the command received; the process can be monitored in the IntesisBox Communication Console window. The use of some of these commands can be critical for IntesisBox normal functioning, having this in mind use only these commands following the recommendations of Intesis Software technical support. A list of the more commonly used commands and the way to use them will be returned by IntesisBox after sending the HELP command.



4.8 Files

LinkBoxMB saves the integration configuration in the following files inside the project folder:

PROJECT.INI	.ini file containing general information referent to the project
BACNET.INI	.ini file containing the information referent to the connection window and other special adjustments
BACNET.DAT	Text file (tab separated values) with the signals information (signals list). This file can be edited (with Excel for example) to change the configuration quicker and easier. Later on, when selecting <i>Configuration - > IntesisBox</i> in LinkBoxMB, if the changes have been made respecting the correct format, all the changes in the configuration done from Excel can be seen in the signals list.
BACNET.LBOX	Binary file created from the information in the files described above. This is the file uploaded to the gateway.

It is strongly recommended to back up the project folder containing these files in external media, once the installation process is finished. This way you will be able to do future configuration changes in case of reinstallation of LinkBoxMB due, for example, to a failure of the hard disk in the PC where it was previously installed.

The configuration cannot be downloaded from the gateway to LinkBoxMB, only can be uploaded; the upload file BACNET.LBOX does not contain all the integration information, as for example the signals description.

4.9 Functionality considerations

IntesisBox returns some useful information about internal parameters in response to an INFO command:

Referent to IntesisBox:

- Firmware version and serial number.
- Protocols versions.
- IP address configured.
- NetMask configured.
- Default Gateway configured.

Referent to Modbus TCP:

- Port configured.

Referent to Modbus RTU:

- Connection type.
- Serial port settings.

* See *Release notes* of the latest firmware version for up to date information on IntesisBox Modbus Server - BACnet/IP Client gateway and LinkBoxMB functionality.

5. Setup process and troubleshooting

5.1 Pre-requisites

It is necessary to have the Modbus master device operative and well connected to the Modbus port of IntesisBox, remember to respect the maximum of 15 meters cable distance if using RS232 communication.

It is necessary to have the BACnet/IP device or network operative and ready to be connected to the Ethernet port of IntesisBox.

Connectors, connection cables, PC for LinkBoxMB, and other auxiliary material, if needed, are not supplied by Intesis Software for this standard integration. The items supplied by Intesis Software for this integration are:

- IntesisBox Modbus Server device with BACnet/IP client external protocol firmware loaded.
- LinkBoxMB software to configure IntesisBox.
- Console cable needed to download the configuration to IntesisBox.
- Product documentation.

If requested, Intesis Software also can supply:

- Standard plug-in power supply 220Vac 50Hz to power IntesisBox (European plug type).

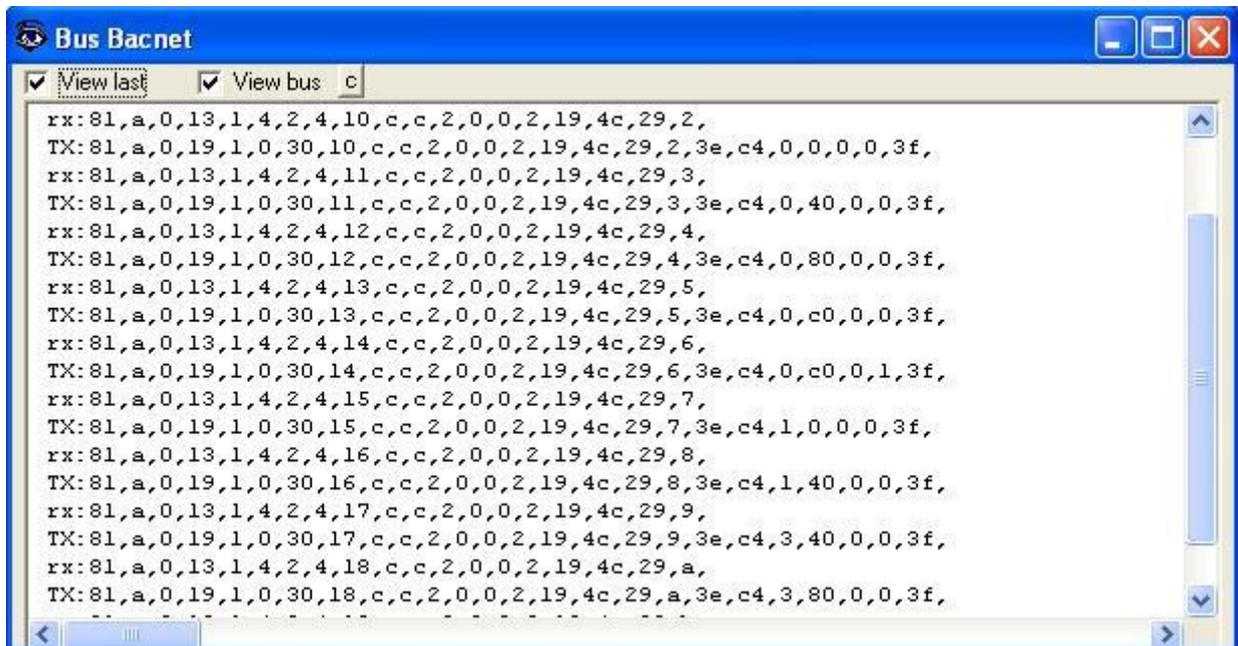
5.2 Setup procedure

1. Install LinkBoxMB on your laptop, use the setup program supplied for this and follow the instructions given by the Installation wizard.
2. Install IntesisBox in the desired installation site. The mounting can be on DIN rail or on a stable not vibrating surface (DIN rail mounted inside a metallic industrial cabinet connected to ground is recommended).
3. Connect the communication cable coming from the Modbus master device to the port marked as **Modbus** of IntesisBox (used either RS232, RS485 or Ethernet port depending on the type of Modbus communication to use). (See details for this communication cable in section *Connections* of this document).
4. Connect the Ethernet cable to the port marked as **BACnet/IP** of IntesisBox. (See details for this communication cable in section *Connections* of this document).
5. Power up IntesisBox. The supply voltage can be 9 to 30 Vdc or just 24 Vac. You can use also the standard plug-in power supply 220/125VAC-12VDC/300mA supplied with the device (if requested). Take care of the polarity of the supply voltage applied.

WARNING! In order to avoid earth loops that can damage IntesisBox and/or any other equipment connected to it, we strongly recommend:

- The use of DC power supplies, floating or with the negative terminal connected to earth. **Never use a DC power supply with the positive terminal connected to earth.**

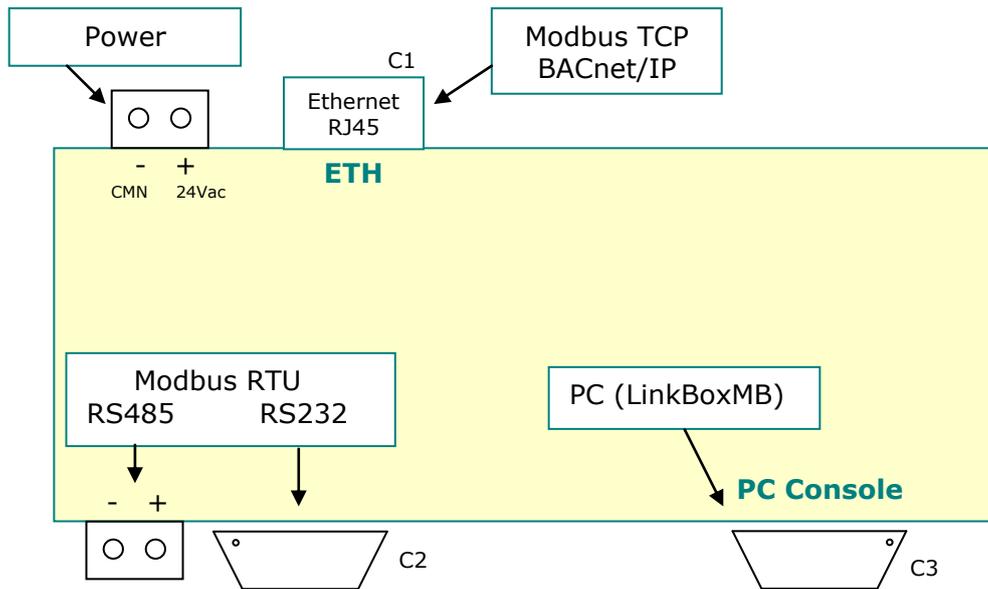
10. Open the *External Protocol Communication Viewer* window (menu View -> Bus -> BACnet) and check that there is communication activity, some TX frames and some other rx frames as showed in the figure below. This means that the communication with the BACnet devices is ok.



```
Bus Bacnet
View last View bus c
rx: 81,a,0,13,1,4,2,4,10,c,c,2,0,0,2,19,4c,29,2,
TX: 81,a,0,19,1,0,30,10,c,c,2,0,0,2,19,4c,29,2,3e,c4,0,0,0,0,3f,
rx: 81,a,0,13,1,4,2,4,11,c,c,2,0,0,2,19,4c,29,3,
TX: 81,a,0,19,1,0,30,11,c,c,2,0,0,2,19,4c,29,3,3e,c4,0,40,0,0,3f,
rx: 81,a,0,13,1,4,2,4,12,c,c,2,0,0,2,19,4c,29,4,
TX: 81,a,0,19,1,0,30,12,c,c,2,0,0,2,19,4c,29,4,3e,c4,0,80,0,0,3f,
rx: 81,a,0,13,1,4,2,4,13,c,c,2,0,0,2,19,4c,29,5,
TX: 81,a,0,19,1,0,30,13,c,c,2,0,0,2,19,4c,29,5,3e,c4,0,c0,0,0,3f,
rx: 81,a,0,13,1,4,2,4,14,c,c,2,0,0,2,19,4c,29,6,
TX: 81,a,0,19,1,0,30,14,c,c,2,0,0,2,19,4c,29,6,3e,c4,0,c0,0,1,3f,
rx: 81,a,0,13,1,4,2,4,15,c,c,2,0,0,2,19,4c,29,7,
TX: 81,a,0,19,1,0,30,15,c,c,2,0,0,2,19,4c,29,7,3e,c4,1,0,0,0,3f,
rx: 81,a,0,13,1,4,2,4,16,c,c,2,0,0,2,19,4c,29,8,
TX: 81,a,0,19,1,0,30,16,c,c,2,0,0,2,19,4c,29,8,3e,c4,1,40,0,0,3f,
rx: 81,a,0,13,1,4,2,4,17,c,c,2,0,0,2,19,4c,29,9,
TX: 81,a,0,19,1,0,30,17,c,c,2,0,0,2,19,4c,29,9,3e,c4,3,40,0,0,3f,
rx: 81,a,0,13,1,4,2,4,18,c,c,2,0,0,2,19,4c,29,a,
TX: 81,a,0,19,1,0,30,18,c,c,2,0,0,2,19,4c,29,a,3e,c4,3,80,0,0,3f,
```

In case there is no response from the BACnet devices to the frames sent by IntesisBox check that they are operative and reachable from the network connection used by IntesisBox, check the IntesisBox's BACnet/IP interface using any commercial BACnet browser connected to the same Ethernet network. See details for the communication cable between IntesisBox and BACnet system in section *Connections* of this document.

6. Connections



IntesisBox (RJ45 F)	C1	Modbus TCP & BACnet/IP Connection	Master TCP BACnet/IP Device (RJ45 F)
Cable (RJ45 M)			Cable (RJ45 M)
		Ethernet	1 device
		Cable UTP/FTP Cat5 Crossed	
		Cable UTP/FTP Cat5 Straight	Hub N devices

IntesisBox (DB9 M)	C2	Modbus RTU Connection	Master RTU (DB9 M)
Cable (DB9 F)			Cable (DB9 F)
RX	2	RS-232 (Crossed)	2 RX
TX	3		3 TX
GND	5	or RS-485	5 GND
Cable (DB9 F)			
TX/RX+			TX/RX+
TX/RX-			TX/RX-

IntesisBox (DB9 F)	C3	PC Connection (LinkBoxMB)	PC (DB9 M)
Cable (DB9 M)			Cable (DB9 F)
TX	2	RS-232 (Straight)	2 RX
RX	3		3 TX
GND	5		5 GND

7. BACnet/IP interface specifications

BACnet Conformance Class Supported: Class 3

BACnet Standard Application Services Supported:

	Initiate Requests	Executes Requests
ConfirmedCOVNotification		X
Subscribe COV	X	
UnconfirmedCOVNotification		X
ReadProperty	X	
WriteProperty	X	
Who-Is	X	
I Am		X

BACnet Standard Object Types Supported:

Analog Input
 Analog Output
 Analog Value
 Binary Input
 Binary Output
 Binary Value
 Multi-state Input
 Multi-state Output
 Multi-state Value

Data Link Layer Option: BACnet/IP, (Annex J)

No Segmented Requests/Responses Supported

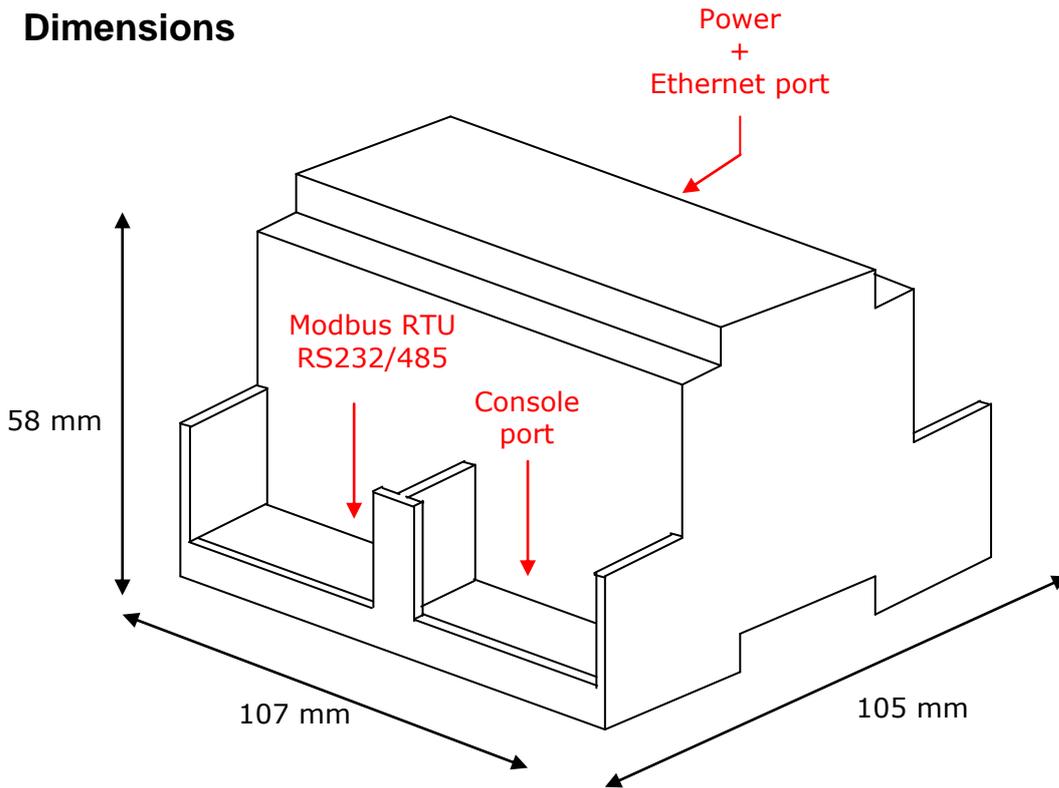
8. Mechanical & Electrical characteristics



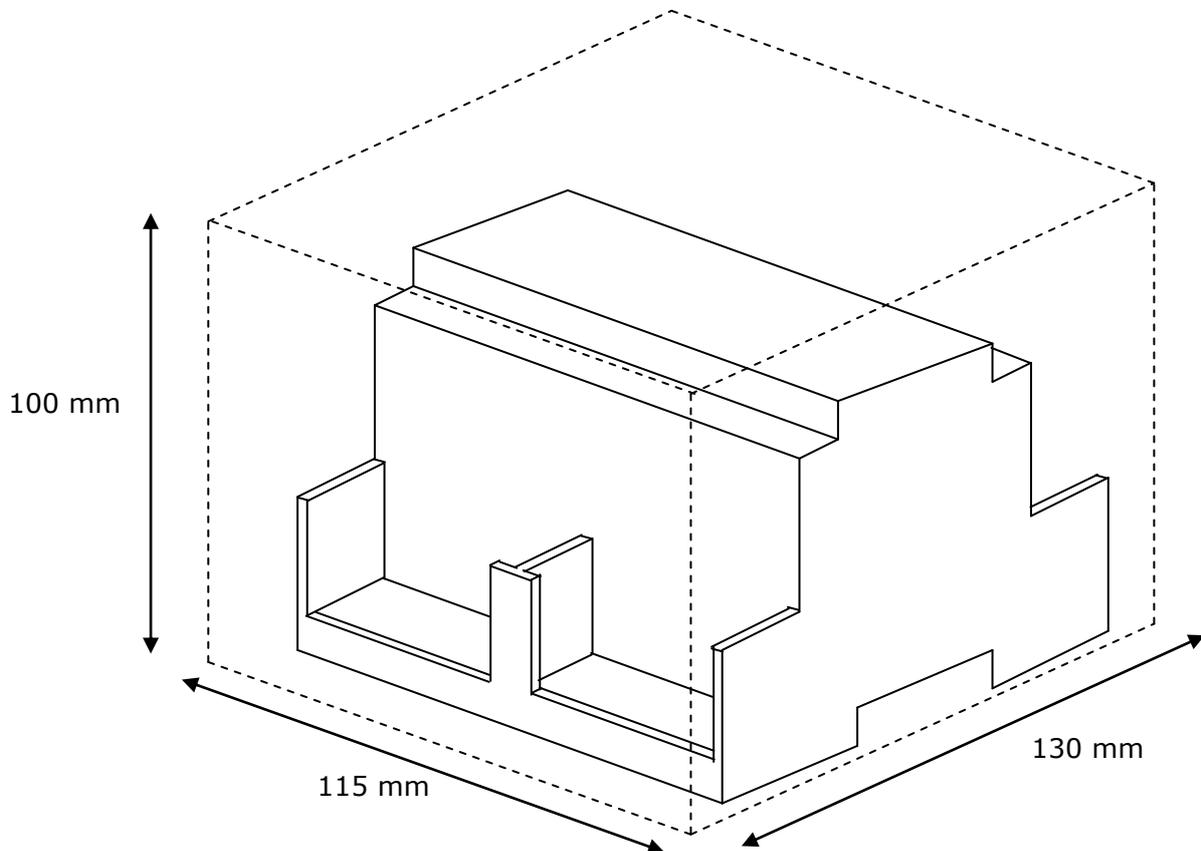
Enclosure	Plastic, type PC (UL 94 V-0). Dimensions: 107mm x 105mm x 58mm.
Colour	Light Grey. RAL 7035.
Power	9 to 30Vdc +/-10% 1.4W. 24Vac +/-10% 1.4VA. Plug-in terminal block for power connection (2 poles).
Mounting	Surface. Wall. DIN rail EN60715 TH35.
Modbus RTU ports	1 x Serial RS232 (DB9 male DTE). 1 x Serial RS485 (Plug-in screw terminal block 2 poles).
Modbus TCP & BACnet/IP port	1 x Ethernet 10BT RJ45.
LED indicators	1 x Power. 2 x Serial port (Modbus RTU) activity (Tx, Rx). 2 x Ethernet port link and activity (LNK, ACT).
Console port	RS232. DB9 female connector (DCE).
Configuration	Via console port. ¹
Firmware	Allows upgrades via console port.
Operational temperature	-40°C to +70°C
Operational humidity	5% to 95%, non condensing
Protection	IP20 (IEC60529).
RoHS conformity	Compliant with RoHS directive (2002/95/CE).
Certifications	CE

¹ Standard cable DB9male-DB9female 1,8 meters long is supplied with the device for connection to a PC COM port for configuring and monitoring the device. The configuration software, compatible with Windows® operating systems, is also supplied.

9. Dimensions



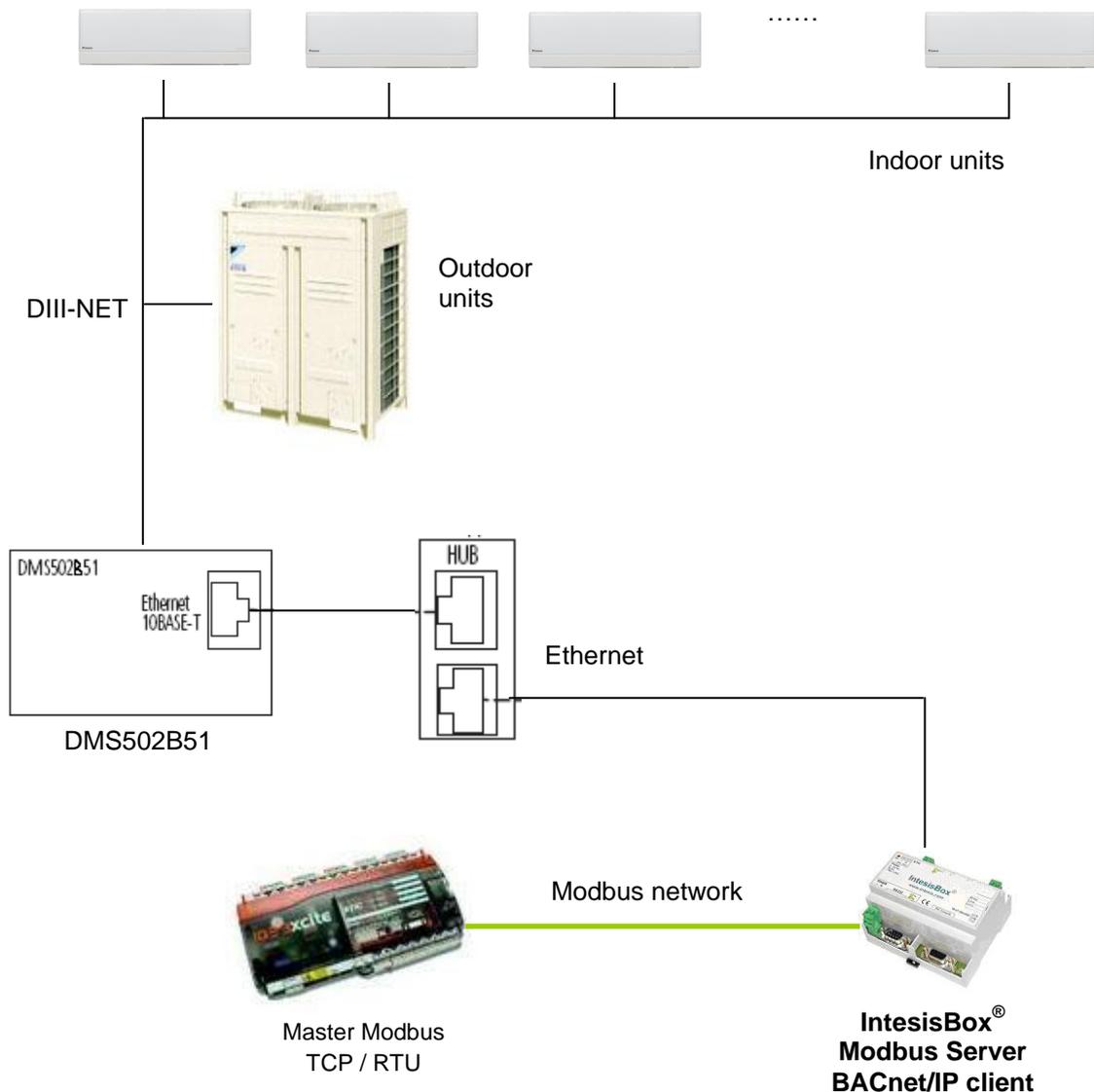
Recommended available space for its installation into a cabinet (wall or DIN rail mounting), with space enough for external connections:



10. Annexes

10.1 Integration of Daikin VRV Air Conditioners into a Modbus enabled control system

For this application, Daikin VRV Air Conditioning system must be equipped with Daikin BACnet gateway (model DMS502B51), this Daikin gateway is normally commissioned by Daikin technical personnel, contact your nearest Daikin distributor for details.



10.2 Daikin DMS502B51 BACnet Gateway for Daikin Air Conditioning VRV system

This gateway allows connecting up to 256 VRVs in 4 groups of 64 (with the corresponding cards). It is important to consult Daikin documentation to configure successfully the integration.

The signals available for every VRV internal unit are showed in the following list. As an example, a selection of *Addresses* for internal Modbus points in IntesisBox have been selected, and also the appropriate values for field *Format* (*R/W* field is fix and depends on the object type of every signal). Note that *Property* field must be always *85-Present Value*.

Object Type	Property	Signal	Object instance	Format	Point	R/W
4-Binary Output	85-Present Value	1-StartStopCommand_000	1	1-16 bits unsigned	2	2-RW
3-Binary Input	85-Present Value	2-StartStopStatus_000	2	1-16 bits unsigned	3	0-R
3-Binary Input	85-Present Value	3-Alarm_000	3	1-16 bits unsigned	4	0-R
13-Multistate Input	85-Present Value	4-MalfunctionCode_000	4	1-16 bits unsigned	5	0-R
14-Multistate Output	85-Present Value	5-AirConModeCommand_000	5	1-16 bits unsigned	6	2-RW
13-Multistate Input	85-Present Value	6-AirConModeStatus_000	6	1-16 bits unsigned	7	0-R
14-Multistate Output	85-Present Value	7-AirFlowRateCommand_000	7	1-16 bits unsigned	8	2-RW
13-Multistate Input	85-Present Value	8-AirFlowRateStatus_000	8	1-16 bits unsigned	9	0-R
0-Analog Input	85-Present Value	9-RoomTemp_000	9	3-16 bits signed * 10	10	0-R
2-Analog Value	85-Present Value	10-TempAdjust_000	10	3-16 bits signed * 10	11	2-RW
3-Binary Input	85-Present Value	11-FilterSign_000	11	1-16 bits unsigned	12	0-R
5-Binary Value	85-Present Value	12-FilterSignReset_000	12	1-16 bits unsigned	13	2-RW
5-Binary Value	85-Present Value	13-RemoteControlStart_000	13	1-16 bits unsigned	14	2-RW
5-Binary Value	85-Present Value	14-RemoteControlAirConModeSet_000	14	1-16 bits unsigned	15	2-RW
5-Binary Value	85-Present Value	16-RemoteControlTempAdjust_000	16	1-16 bits unsigned	16	2-RW
3-Binary Input	85-Present Value	20-CommunicationStatus_000	20	1-16 bits unsigned	17	0-R
2-Analog Value	85-Present Value	22-AirDirectionCommand_000	22	1-16 bits unsigned	18	2-RW
0-Analog Input	85-Present Value	23-AirDirectionStatus_000	23	1-16 bits unsigned	19	0-R
4-Binary Output	85-Present Value	24-ForcedThermoOFFCommand_000	24	1-16 bits unsigned	20	2-RW
3-Binary Input	85-Present Value	25-ForcedThermoOFFStatus_000	25	1-16 bits unsigned	21	0-R
4-Binary Output	85-Present Value	26-EnergyEfficiencyCommand_000	26	1-16 bits unsigned	22	2-RW
3-Binary Input	85-Present Value	27-EnergyEfficiencyStatus_000	27	1-16 bits unsigned	23	0-R
3-Binary Input	85-Present Value	28-ThermoStatus_000	28	1-16 bits unsigned	24	0-R
3-Binary Input	85-Present Value	29-CompressorStatus_000	29	1-16 bits unsigned	25	0-R
3-Binary Input	85-Present Value	30-IndoorFanStatus_000	30	1-16 bits unsigned	26	0-R
3-Binary Input	85-Present Value	31-HeaterStatus_000	31	1-16 bits unsigned	27	0-R

To calculate the *Object Instance* for Daikin points use the following formula:

Object Instance = (Indoor Unit Number X 256) + Point Number

Indoor Unit Number: 0..255

Point Number: 1..31

Example:

Indoor Unit Number: 1 (Remember 256 numbered 0..255)

Point Number: 9-RoomTemp

Object Instance = (1 X 256) + 9 = 265

Note: In the Daikin system configuration, sometimes the VRVs are configured as masters and sometimes as slaves, the ones configured as slaves do not accept commands because they work in parallel with their masters.

Important: Contact Intesis Software for some example projects containing signals configuration for integration of Daikin VRV systems with Daikin DMS502B51 BACnet Gateway.

They will be something like the following:

BACnet Daikin 5 units, 5 VRV units with all the signals.

BACnet Daikin 25 units, 25 VRV units with all the signals.

BACnet Daikin 256 units, 256 VRV units with all the signals.

BACnet Daikin 256 units LE, 256 VRV units with only StartStopCommand, StartStopStatus and TempAdjust signals.

Create a copy of the project most suitable for your application and, after that, it will only be necessary to activate those signals needed for your integration and configure your Modbus master device to poll the internal *Modbus register Addresses* configured in IntesisBox corresponding to the signals to integrate. Very easy and quick, almost plug & play.

Requirements: The information needed to configure the integration is:

- The Daikin Gateway must be configured in mode BACnet/IP
- IP address of the Daikin Gateway
- Net Mask of the Daikin Gateway
- UDP Port of the Daikin Gateway
- Device number of the Daikin Gateway
- List of all the VRVs of the installation containing the physical location in the building and the number (0..255) for every VRV
- List of the slave VRVs and their corresponding masters (no commands supported by these slave VRVs).