IntesisBox® KNX

Modbus TCP master

User's Manual

v10 r11 eng

Intesis software

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

Gateway for the integration of Modbus TCP slave devices with KNX TP-1 (EIB) control systems.

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

IBOX-KNX-MBTCP-100
Tiny model supporting connection to up to 5 Modbus TCP slave devices and 100 internal data points.

IBOX-KNX-MBTCP-A
Basic model supporting connection to up to 5 Modbus TCP slave devices and 500 internal data points.

IBOX-KNX-MBTCP-B
Extended model supporting connection to up to 5 Modbus TCP slave devices and 4000 internal data points.

INDEX

1. Description

1.1 Introduction

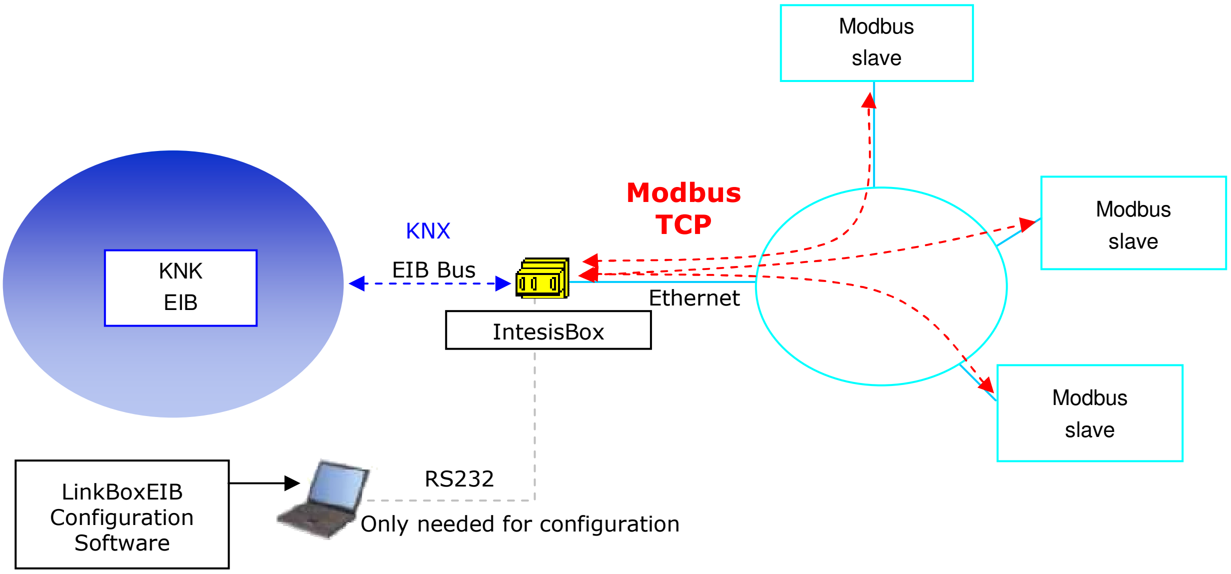
This document describes the integration of KNX TP-1 (EIB) systems with Modbus TCP slave devices or systems using the gateway *IntesisBox-KNX Modbus TCP master*.

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

From now on, and with the aim of easy the read of this document, just the word "IntesisBox" will be used instead of the full gateway model description "IntesisBox-KNX Modbus TCP master".

The aim of this integration is to make accessible signals and resources of any Modbus TCP slave device from any KNX TP-1 (EIB) control system.

For this, IntesisBox acts as a Modbus TCP master device in its Modbus interface, reading/writing points of the Modbus slave device(s), and offering this point's values through its KNX interface, acting in the KNX system as one more KNX device of the system. The readings of the Modbus slave device(s) is performed by IntesisBox by automatic continuous polling.



Integration of Modbus and KNX using
***IntesisBox-KNX Modbus TCP master* gateway**

1.2 Functionality

The integration operation is as follow:

From the Modbus system point of view, after the start up process, IntesisBox reads continuously the points configured to be read in the Modbus slave devices, and updates in its memory all the values received from the Modbus system.

Every one of the mentioned Modbus points is associated to a *KNX group address*, with this, all the Modbus system is seen as *one more KNX device* from the KNX system point of view, with the same configuration and operation characteristics.

When a change in any Modbus point occurs, a write telegram is sent to the KNX bus, of the associated *KNX Group*. See details in section 4.

When a telegram is received from the KNX bus, of a *KNX Group address* associated to a Modbus point, a message is sent immediately to the corresponding Modbus device to perform the corresponding action.

In the continuous polling of the Modbus devices, if a non response of the Modbus device is detected, the corresponding virtual signal inside IntesisBox will be activated indicating communication error with the Modbus device. These virtual signals indicating communication status in real time with the Modbus devices are also accessible from KNX, like the rest of the points of IntesisBox.

1.3 Capacity of IntesisBox

Element	Max. (Tiny version)	Max. (Basic version)	Max. (Extended version)	Notes
Type of Modbus slave devices				Those supporting *Modbus TCP/IP protocol*. Communication over Ethernet network.
Points	100	500	4000	Maximum number of points that can be defined into IntesisBox.
Modbus TCP slave devices	5	5	5	Maximum number of Modbus TCP slave devices that can be defined in IntesisBox.
Modbus slave address	1 to 254	1 to 254	1 to 254	Modbus slave address that can be defined in IntesisBox for each point. *For each point defined in IntesisBox, a Modbus TCP device to which belongs (1 to 5), and a Modbus slave address to which belongs (1 to 254) can be defined, wit this, the integration of a number of Modbus RTU slaves devices through a Modbus TCP to RTU converter is also possible.*
Virtual signals	1 per device	1 per device	1 per device	1 communication error signal per Modbus TCP slave device defined. Available from KNX.

There are three different models of *IntesisBox® KNX - Modbus TCP master,* with different capacity every one of them.

- Tiny model supporting integration of up to 100 internal data points and 5 Modbus TCP slave devices. *Ref.: IBOX-KNX-MBTCP-100.*

- Basic model supporting integration of up to 500 internal data points and 5 Modbus TCP slave devices. *Ref.: IBOX-KNX-MBTCP-A.*

- Extended model supporting integration of up to 4000 internal data points and 5 Modbus TCP slave devices. *Ref.: IBOX-KNX-MBTCP-B.*

2. KNX System

In this section, a common description for all IntesisBox KNX series gateways is given, from the point of view of KNX system which is called from now on *internal system* The Modbus system is also called from now on *external system*.

2.1 Description

IntesisBox KNX connects directly to the KNX TP-1 (EIB) bus and behaves as one more devices into the KNX system, with the same configuration and operational characteristics as other KNX devices.

Internally, the circuit part connected to the KNX bus is opto-isolated from the rest of the electronics.

IntesisBox-KNX receives, manages and sends all the telegrams related to its configuration to the KNX bus.

On receiving telegrams of KNX Groups associated to internal datapoints, the corresponding messages are sent to the external system (Modbus) to maintain both systems synchronised in every moment.

When a change in a signal of the external system is detected, a telegram is sent to the KNX bus (of the associated KNX group) to maintain both systems synchronised in every moment.

The status of the KNX bus is checked continuously and, if a bus drops down is detected, due to a failure in the bus power supply for example, when the KNX bus is restored again, IntesisBox will retransmit the status of all the KNX groups marked as *"T" Transmit*. Also the *Updates* of the groups marked as *"U" Update* will be performed. The behaviour of each individual point into IntesisBox is determined by the flags configured for the point. See details in section 4.

2.2 Points definition

Every internal datapoint to define has the following KNX properties:

Property	Description
Signal	Signal's Description. Only for informative purposes, allows identifying the signal comfortably.
EIS (Datapoint)	It is the KNX data type used to code the signal's value. It will depend on the type of signal associated in the external system in every case. In some integration it is selectable, in others it is fixed due to the intrinsic characteristics of the signal.
Group	It is the KNX group to which the point is associated. It is also the group to which the read (R), write (W), transmit (T) and update (U) flags are applied. Is the sending group.
Listening addresses	They are the addresses that will actuate on the point, apart of the main Group address.
R	Read. If this flag is activated, read telegrams of this group address will be accepted.
W	Write. If this flag is activated, write telegrams of this group address will be accepted.
T	Transmit. If this flag is activated, when the point's value changes, due to a change in the external system, a write telegram of the group address will be sent to the KNX bus.
U	Update. If this flag is activated, on IntesisBox start-up or after a KNX bus reset detection, read telegrams of the sending group will be sent to the KNX bus, and the value received will be sent to the external system as if it has been received by a KNX write telegram.
Active	If activated, the point will be active in IntesisBox, if not, the behaviour will be as if the point is not defined. This allows deactivating points without the need of delete them for possible future use.

These properties are common for all IntesisBox KNX series gateways. Although each integration may have specific properties according to the type of signals of the external system.

3. Modbus interface of IntesisBox

3.1 Description

Modbus TCP communication is characterised basically by the embedding of the Modbus RTU protocol into TCP/IP frames. This communication over TCP/IP allows faster communication and a longer distance between master and slave devices in comparison with RTU communication over serial line, and can use common TCP/IP infrastructure in buildings as well as communication over WAN or internet. It allows also the co-existence of one or more masters and of course one or more slave devices in a given network, all of them interconnected through a TCP/IP based network.

IntesisBox acts as master in the Modbus TCP network, and the other Modbus devices connected to the network communicating with IntesisBox must be always slave devices.

Up to 5 Modbus TCP slave devices can be defined in IntesisBox, to communicate with them.

For each point defined that belongs to a defined Modbus TCP slave device, a slave address from 0 to 255 can be also freely configured, this feature allows great flexibility, for example to integrate Modbus RTU slave devices connected in a serial line and with an RTU/TCP converter on top of this serial line, enabling the access to the RTU slaves' points through TCP/IP, in this case the RTU/TCP converter communicating in TCP identifies the destination of the point (slave address in the RTU network) by the contents of the slave address field.

Modbus TCP slave devices are characterised by their IP address, and their predefined *registers address map*, this address map specifies the address, type and characteristics of each internal point (commonly called *register*) of the Modbus slave device, these registers being accessible using Modbus TCP protocol.

Communication parameters of IntesisBox's Modbus TCP interface (IP address, Net Mask, Default router address, and TCP port) are fully configurable to adapt to any IP network and slave device.

Modbus TCP protocol defines different types of function codes to use to read/write different type of registers that can be found in Modbus devices, and also different data formats to encode values. See below (section 3.2) all the function codes and data formats supported by IntesisBox's Modbus interface.

Also the data encoding used for 16 bits registers (big-endian or little-endian) can be configured in IntesisBox's Modbus interface. This is the byte order for data encoding (MSB..LSB or LSB..MSB). This data encoding, although is specified as big-endian in Modbus protocol specification, it varies depending on manufacturer/type of slave.

All this gives great flexibility to integrate a wide range of Modbus slave devices that can be found in the market.

3.2 Points definition

Each point defined in IntesisBox has the following Modbus features associated to it:

Feature	Description
Modbus Device	Modbus TCP device to which belongs the point, from a list of Modbus TCP slave devices that can be defined in IntesisBox (up to 5). For every Modbus TCP slave device defined, a virtual signal is created automatically in IntesisBox to inform about the communication with the device, this signal is available also from the KNX interface like the rest of the points.
Slave address	Address of the slave to which belongs the point, possible values from 0 to 255. This feature allows great flexibility, for example to integrate Modbus RTU slave devices connected in a serial line and with an RTU/TCP converter on top of this serial line, enabling the access to the RTU slaves' points through TCP/IP, in this case the RTU/TCP converter communicating in TCP identifies the destination of the point (slave address in the RTU network) by the contents of this slave address field.
Function code	One of the following Modbus function codes can be used: • 1- Read coils. • 2- Read discrete inputs. • 3- Read holding registers. • 4- Read input registers. • 5- Write single coil. • 6- Write single register. • 15- Write multiple coils. • 16- Write multiple registers.
Data coding format	One of the following Modbus data coding formats can be used: *Generic:* • 1 bit. • 16 bits unsigned. • 16 bits signed. • 16 bits signed (two's complement). • 32 bits unsigned. • 32 bits signed. • 32 bits signed (two's complement). • 32 bits IEEE. • 32 bits IEEE inverted. *Device Specific:* • Bit coded into 16 bits register. • 32 bits IEEE CIAT. • 32 bits Mod10K unsigned.

	• 48 bits Mod10K unsigned. • 64 bits Mod10K unsigned. • 32 bits Mod10K signed. • 48 bits Mod10K signed. • 64 bits Mod10K signed. • 32 bits Mod1k ION. • 32 bits ION signed. • 32 bits Invertomatic.
Register address	The Modbus register address inside the slave device for the point.
Bit inside the register	Bit inside the Modbus register (optional). The gateway allows bit decoding from generic 16 bits input/holding Modbus registers. *Bit coding into 16 bit input/holding Modbus registers is used for some devices to encode digital values into this type of registers, being these registers normally accessible using Modbus function codes 3 and 4 (read holding/input registers).*
Fractional part	Some slaves send analogy values which have fractional part (i.e. temperatures) into 16 bits integer registers, the way used to avoid missing the fractional part is multiplying the value by a power of 10 before send it. Using this feature of IntesisBox you can deal with the original value on the slave for these kinds of points. For example, if you specify Frac=2 for a given point, then IntesisBox will divide by 100 the value received from the slave in case of readings, and will multiply by 100 the value before sending it to the slave in case of writings.

4. LinkBoxEIB. Configuration & monitoring tool for intesisBox KNX series.

4.1 Introduction

LinkBoxEIB is a Windows compatible software tool developed specifically to monitor and configure IntesisBox-KNX series gateways. It is possible to configure all external protocols available for IntesisBox-KNX and to maintain different customer's configurations based on a LinkBoxEIB 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 LinkBoxEIB, as well as configure the integration points list and connection parameters for every external protocol, it is permitted also 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.

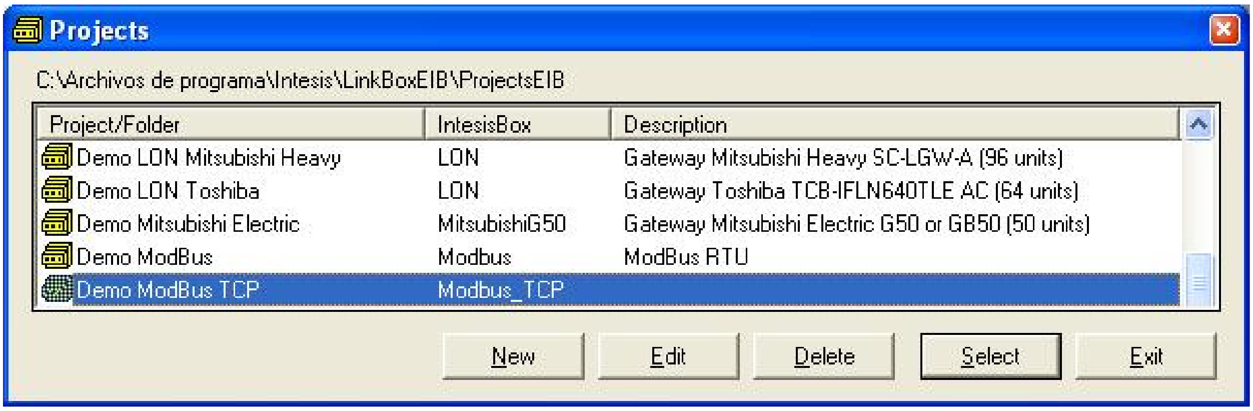
LinkBoxEIB allows configuring all IntesisBox-KNX series independently of the external system or protocol used. For every external system, LinkBoxEIB has a specific configuration window. Periodically, new free versions of LinkBoxEIB are released incorporating the latest developed integrations for external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

4.2 Project definition

The first step to do in LinkBoxEIB 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, loosing the configuration data for those previous installations. The projects folder is located in AppFolder\ProjectsEIB, where AppFolder is the installation folder of LinkBoxEIB (by default C:\Program Files\Intesis\LinkBoxEIB). Inside the projects folder, a new folder will be created for every project defined in LinkBoxEIB with the files needed for the project.

When you open LinkBoxEIB, the project selection window will appear inviting you to select a project or to create a new one. A demo project for every external protocol supported is provided with the standard installation of LinkBoxEIB.

You can create a brand new project, which will create a blank project (only basic parameters will be already configured), or you can select a demo project based on the external protocol desired and create a new project based on this demo project selected (all the configuration of the demo project will be copied into the new project created).



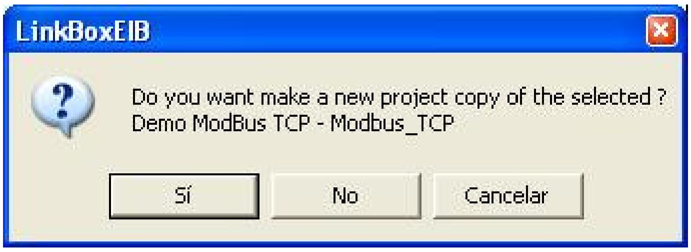
Projects

C:\Archivos de programa\Intesis\LinkBoxEIB\ProjectsEIB

Project/Folder	IntesisBox	Description
Demo LON Mitsubishi Heavy	LON	Gateway Mitsubishi Heavy SC-LGW-A (96 units)
Demo LON Toshiba	LON	Gateway Toshiba TCB-IFLN640TLE AC (64 units)
Demo Mitsubishi Electric	MitsubishiG50	Gateway Mitsubishi Electric G50 or GB50 (50 units)
Demo ModBus	Modbus	ModBus RTU
Demo ModBus TCP	Modbus_TCP	

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 push *New* button. You will be prompted to create a copy of the selected project (useful for similar installations) or create a brand new one.

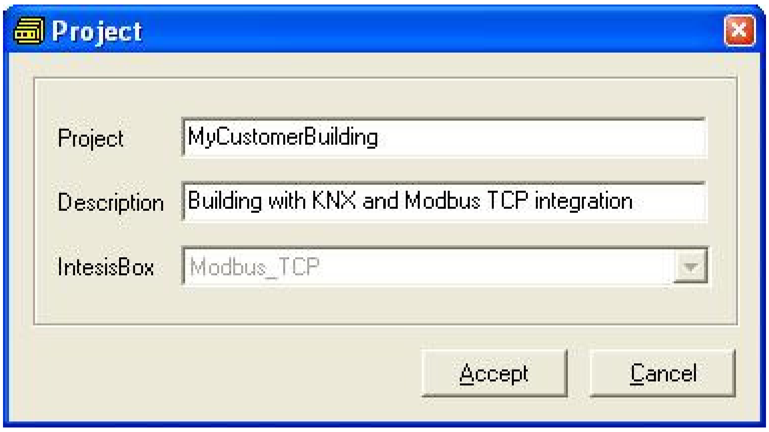


LinkBoxEIB

Do you want make a new project copy of the selected ?
Demo ModBus TCP - Modbus_TCP

If you select *Yes,* you will be prompted to specify a name and a description for the new project that will contain a copy of 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. This will create a brand new project based in the external protocol specified.



Project: MyCustomerBuilding

Description: Building with KNX and Modbus TCP integration

IntesisBox: Modbus_TCP

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.

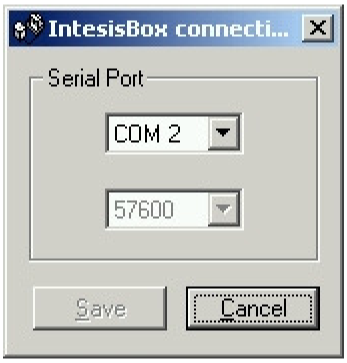
A description of the files created for a Modbus TCP protocol based project can be found in section *Files* below in this document.

From all the possibilities of LinkBoxEIB, 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 more comfortably in the office. Before any monitoring or downloading action to IntesisBox can be performed, the connection between IntesisBox and the PC running LinkBoxEIB must be established (working on-line). To do so follow these steps:

1. Make sure IntesisBox is powered-up and correctly connected to the KNX system via the EIB bus and to the Modbus TCP system via the Ethernet network (consult details for connection and pin assignments in section *Connection*s of this document).

2. Connect a free PC serial port to IntesisBox's serial port marked as ***PC Console***. (Use the standard serial cable supplied with IntesisBox or makes your own cable following the pin assignments specified in section *Connections* in this document).

3. Select in LinkBoxEIB the PC serial port used for the connection to IntesisBox. Use menu Configuration -> Connection.

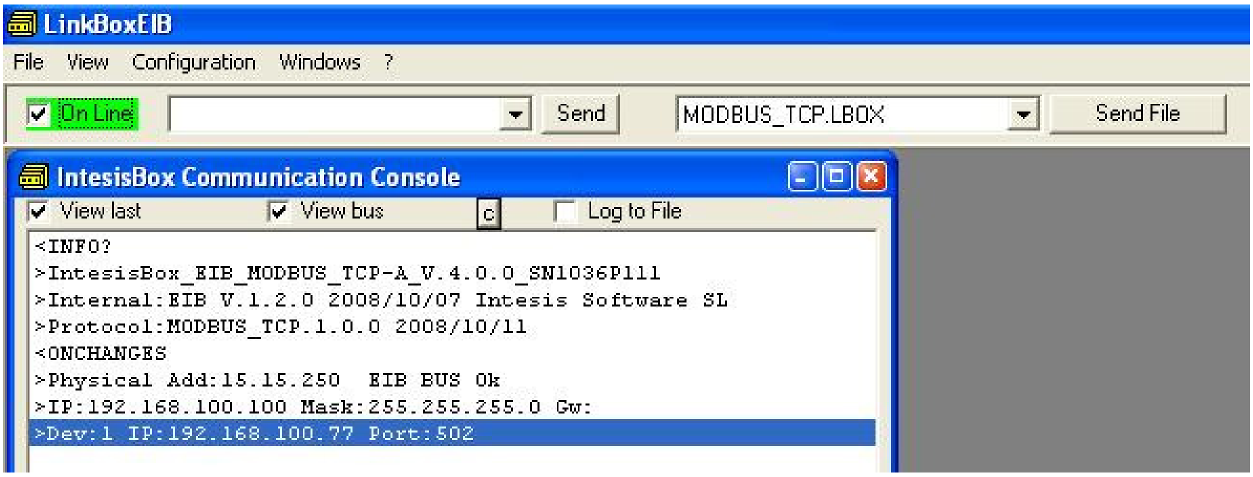


Serial Port: COM 2

57600

4. Check the checkbox *off-line* under the menu bar (it will change automatically to *on-line*) and LinkBoxEIB will ask for INFO about the IntesisBox connected to it via the serial connection, if the connection is ok then IntesisBox will respond with its identification

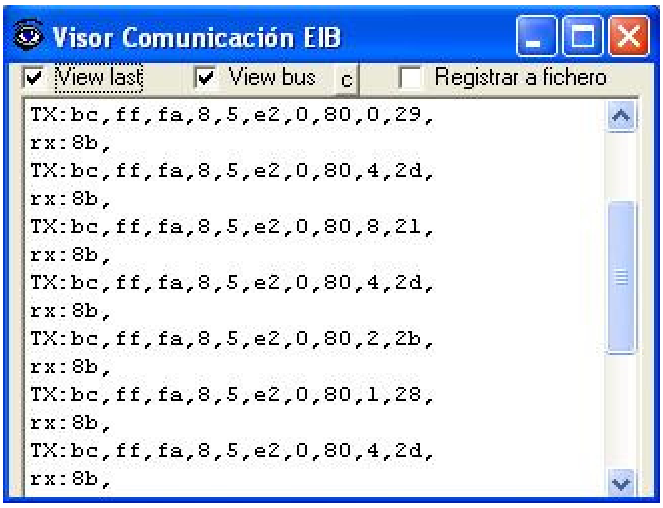
(this can be monitored in the *IntesisBox Communication Console* window, as showed below).



```
<INFO?
>IntesisBox_EIB_MODBUS_TCP-A_V.4.0.0_SN1036P111
>Internal:EIB V.1.2.0 2008/10/07 Intesis Software SL
>Protocol:MODBUS_TCP.1.0.0 2008/10/11
<ONCHANGES
>Physical Add:15.15.250  EIB BUS Ok
>IP:192.168.100.100 Mask:255.255.255.0 Gw:
>Dev:1 IP:192.168.100.77 Port:502
```

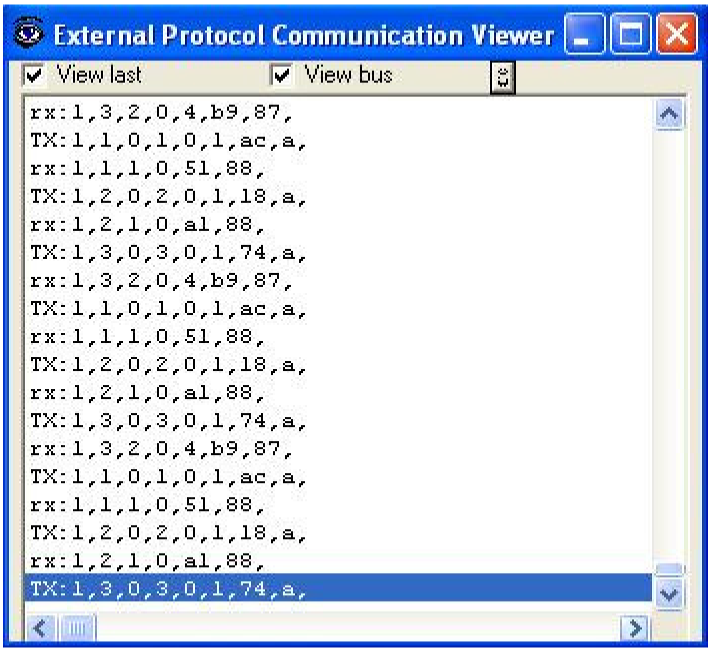
Once connected to IntesisBox, all the options of LinkBoxEIB are fully operative.

To monitor the communication between IntesisBox and the KNX system, select the menu *View -> Bus -> EIB*. The *EIB communication Viewer* window will be opened. This window shows in real time all the communication frames between IntesisBox and the KNX system as well as debugging messages referent to the internal protocol (KNX) sent by IntesisBox.



```
TX:bc,ff,fa,8,5,e2,0,80,0,29,
rx:8b,
TX:bc,ff,fa,8,5,e2,0,80,4,2d,
rx:8b,
TX:bc,ff,fa,8,5,e2,0,80,8,21,
rx:8b,
TX:bc,ff,fa,8,5,e2,0,80,4,2d,
rx:8b,
TX:bc,ff,fa,8,5,e2,0,80,2,2b,
rx:8b,
TX:bc,ff,fa,8,5,e2,0,80,1,28,
rx:8b,
TX:bc,ff,fa,8,5,e2,0,80,4,2d,
rx:8b,
```

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



☑ View last ☑ View bus

```
rx:1,3,2,0,4,b9,87,
TX:1,1,0,1,0,1,ac,a,
rx:1,1,1,0,51,88,
TX:1,2,0,2,0,1,18,a,
rx:1,2,1,0,a1,88,
TX:1,3,0,3,0,1,74,a,
rx:1,3,2,0,4,b9,87,
TX:1,1,0,1,0,1,ac,a,
rx:1,1,1,0,51,88,
TX:1,2,0,2,0,1,18,a,
rx:1,2,1,0,a1,88,
TX:1,3,0,3,0,1,74,a,
rx:1,3,2,0,4,b9,87,
TX:1,1,0,1,0,1,ac,a,
rx:1,1,1,0,51,88,
TX:1,2,0,2,0,1,18,a,
rx:1,2,1,0,a1,88,
TX:1,3,0,3,0,1,74,a,
```

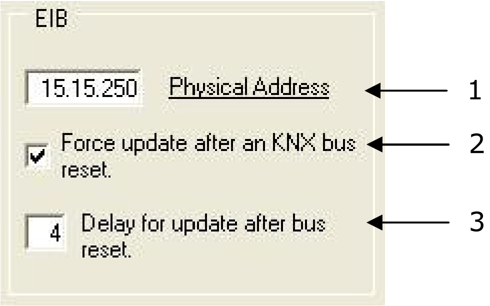
To configure the integration connection parameters, and the points list, select menu *Configuration -> IntesisBox*. The *Modbus TCP Configuration* window will be opened.

4.3 Connection configuration

Select the Connection tab to configure the connection parameters.

Three kinds of information are configured using this window: the parameters of the KNX interface, the IP parameters of IntesisBox, and the parameters of the Modbus TCP devices to communicate to.

KNX interface configuration parameters:



EIB

15.15.250 Physical Address

☑ Force update after an KNX bus reset.

4 Delay for update after bus reset.

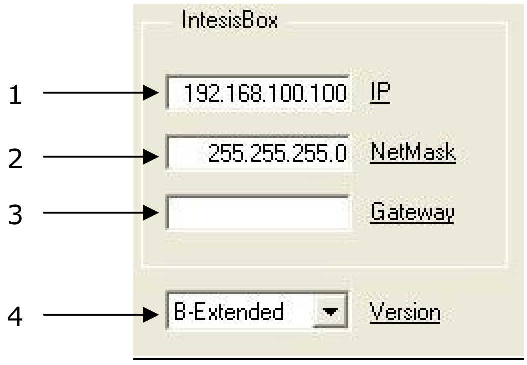
KNX configuration

1. Enter the physical address desired for IntesisBox inside the KNX network.

2. Check this if you want IntesisBox to force the readings in KNX of those points configured with "U" or "U2" flag after a KNX bus reset detection.

3. Delay (in seconds) to wait before perform the readings in KNX of those points configured with "U" or "U2" flag after a KNX bus reset detection (configure it at least to 4 seconds to allow all the devices in the KNX bus to start-up correctly after a bus power failure).

IP parameters of IntesisBox:



192.168.100.100 IP

255.255.255.0 NetMask

Gateway

B-Extended Version

1. Enter the IP address desired for IntesisBox.

2. Enter the Net Mask desired for IntesisBox.

3. Enter the default router address to use by IntesisBox if necessary.

4. Select the version of IntesisBox used.

   Remember you can identify the version of IntesisBox by the order code printed in the front label:

   - **IBOX-KNX-MBTCP-100**. Tiny model supporting up to 5 Modbus devices and 100 points.
   - **IBOX-KNX-MBTCP-A**. Basic model supporting up to 5 Modbus devices and 500 points.
   - **IBOX-KNX-MBTCP-B**. Extended model supporting up to 5 Modbus devices and 4000 points.

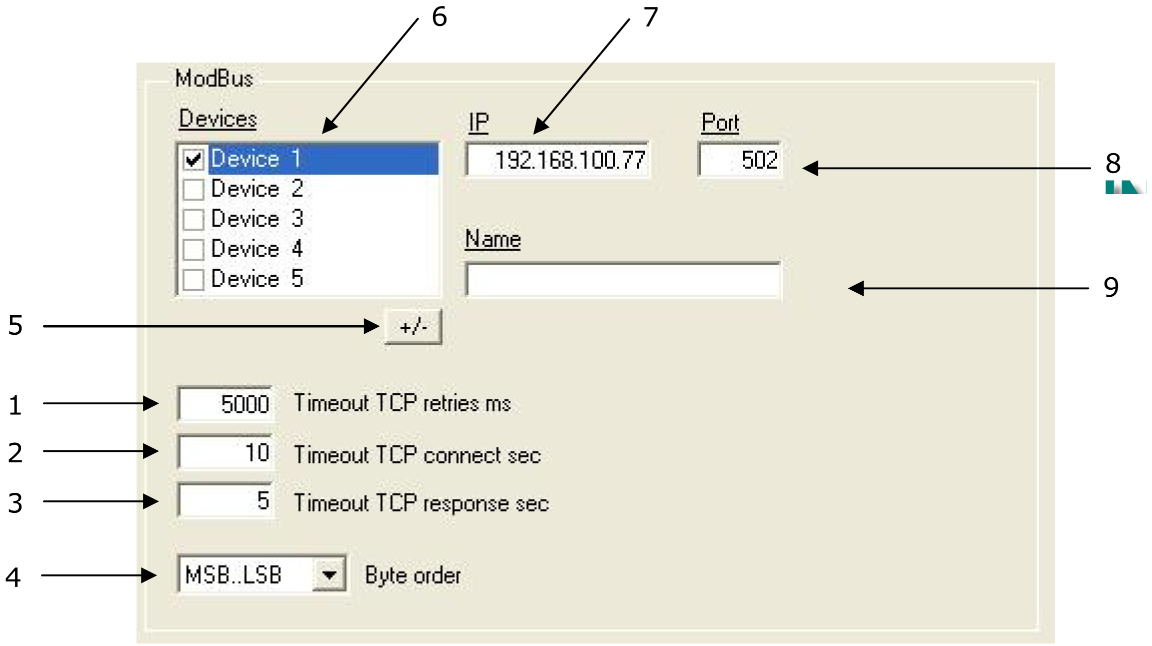
   You can identify also the version of IntesisBox by its identification given in response to an INFO command, it is something like this:

   IntesisBox_EIB_MODBUS_TCP-**100**… -> this is the tiny model.

   IntesisBox_EIB_MODBUS_TCP-**A**… -> this is the basic model.

   IntesisBox_EIB_MODBUS_TCP-**B**… -> this is the extended model.

Modbus interface configuration parameters:



1. Timeout between TCP connection retries (from 100 to 30000 milliseconds).

2. Timeout waiting for a TCP connection (from 1 to 120 seconds).

3. Timeout waiting for a TCP response (from 1 to 120 seconds).

4. Byte order for data fields inside Modbus telegrams (LSB..MSB or MSB..LSB), it will depend on the slave, consult the slave documentation for details. If unknown just try the two possible choices and see if the values read make sense. This affects to all data fields of all slaves defined.

5. Use this button to define the number of Modbus slave devices to communicate to. Up to 5 devices maximum.

6. List of Modbus slave devices to communicate to. Check the devices you want to activate. Select a device to configure its properties.

For every Modbus device defined, the following properties must be entered:

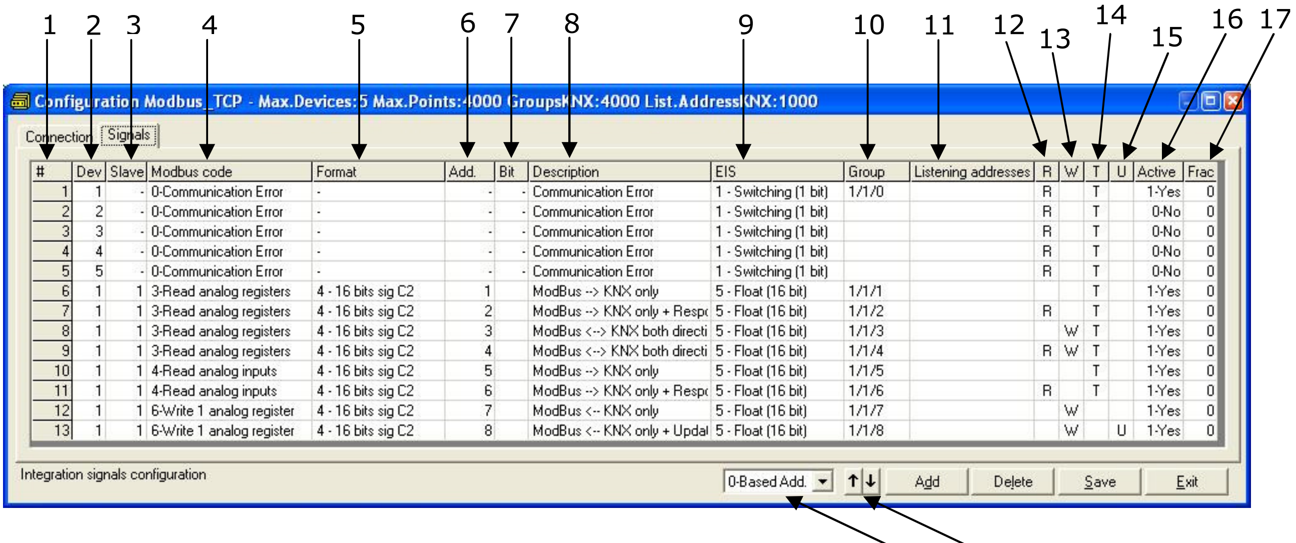
7. Enter the IP address of the device.

8. Enter the TCP port to use (502 by default).

9. Enter the device name (optional, just for identification purposes).

4.4 Points configuration

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


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Configuration Modbus_TCP - Max.Devices:5 Max.Points:4000 GroupsKNX:4000 List.AddressKNX:1000
Connection
Signals
Integration signals configuration
0-Based Add.
↑↓
Add
Delete
Save
Exit
18
19

#	Dev	Slave	Modbus code	Format	Add.	Bit	Description	EIS	Group	Listening addresses	R	W	T	U	Active	Frac
1	1	-	0-Communication Error	-	-	-	Communication Error	1 - Switching (1 bit)	1/1/0		R		T		1-Yes	0
2	2	-	0-Communication Error	-	-	-	Communication Error	1 - Switching (1 bit)			R		T		0-No	0
3	3	-	0-Communication Error	-	-	-	Communication Error	1 - Switching (1 bit)			R		T		0-No	0
4	4	-	0-Communication Error	-	-	-	Communication Error	1 - Switching (1 bit)			R		T		0-No	0
5	5	-	0-Communication Error	-	-	-	Communication Error	1 - Switching (1 bit)			R		T		0-No	0
6	1	1	3-Read analog registers	4 - 16 bits sig C2	1		ModBus --> KNX only	5 - Float (16 bit)	1/1/1				T		1-Yes	0
7	1	1	3-Read analog registers	4 - 16 bits sig C2	2		ModBus --> KNX only + Respo	5 - Float (16 bit)	1/1/2		R		T		1-Yes	0
8	1	1	3-Read analog registers	4 - 16 bits sig C2	3		ModBus <--> KNX both directi	5 - Float (16 bit)	1/1/3			W	T		1-Yes	0
9	1	1	3-Read analog registers	4 - 16 bits sig C2	4		ModBus <--> KNX both directi	5 - Float (16 bit)	1/1/4		R	W	T		1-Yes	0
10	1	1	4-Read analog inputs	4 - 16 bits sig C2	5		ModBus --> KNX only	5 - Float (16 bit)	1/1/5				T		1-Yes	0
11	1	1	4-Read analog inputs	4 - 16 bits sig C2	6		ModBus --> KNX only + Respo	5 - Float (16 bit)	1/1/6		R		T		1-Yes	0
12	1	1	6-Write 1 analog register	4 - 16 bits sig C2	7		ModBus <-- KNX only	5 - Float (16 bit)	1/1/7			W			1-Yes	0
13	1	1	6-Write 1 analog register	4 - 16 bits sig C2	8		ModBus <-- KNX only + Updal	5 - Float (16 bit)	1/1/8			W		U	1-Yes	0

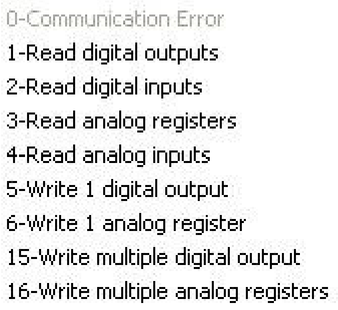
Signals list

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 slave number, it is just the order of the device (from top to bottom) in the devices list.

3. *Slave*. Slave number to which belongs the point. Consult slave documentation. In case you are integrating some Modbus RTU slave devices connected in a serial line and with an RTU/TCP converter on top of this serial line, enter here the slave number to which belongs the point.

4. *Modbus Code*. Modbus function code to be used by IntesisBox to read, to write or to read/write the point in the slave device. Consult the Modbus slave documentation to know what function codes are supported to read and write points. For points of type *read only* you should specify function codes 1, 2, 3 or 4. For points of type *write only*, you should specify function codes 5, 6, 15 or 16. For points of type *read/write*, see below how to configure *read/write* points. Edit using the mouse right-button-click pop-up menu available on the column as showed in the figure below. One of the showed function codes can be selected:

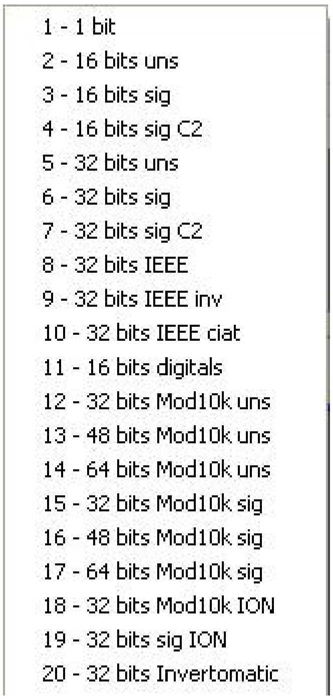


0-Communication Error
1-Read digital outputs
2-Read digital inputs
3-Read analog registers
4-Read analog inputs
5-Write 1 digital output
6-Write 1 analog register
15-Write multiple digital output
16-Write multiple analog registers

Function code *0 - Communication error* is a virtual function code associated to the *communication error* virtual signals, it can not be selected for normal use. The rest are general Modbus function codes that can be freely selected.

Consult documentation of Modbus device(s) to integrate for information about function codes supported to read/write their internal points.

5. *Format*. Modbus data format for the point. Edit using the mouse right-button-click pop-up menu available on the column as showed in the figure below. One of the showed data formats can be selected:



1 - 1 bit
2 - 16 bits uns
3 - 16 bits sig
4 - 16 bits sig C2
5 - 32 bits uns
6 - 32 bits sig
7 - 32 bits sig C2
8 - 32 bits IEEE
9 - 32 bits IEEE inv
10 - 32 bits IEEE ciat
11 - 16 bits digitals
12 - 32 bits Mod10k uns
13 - 48 bits Mod10k uns
14 - 64 bits Mod10k uns
15 - 32 bits Mod10k sig
16 - 48 bits Mod10k sig
17 - 64 bits Mod10k sig
18 - 32 bits Mod10k ION
19 - 32 bits sig ION
20 - 32 bits Invertomatic

Formats 1 to 9 are generic Modbus data formats, formats 10 to 20 are *Device specific*. Note that *Format* cell will be filled according to the *Modbus Code* selected for the point, only some of the possible Format values can be selected depending on Modbus Code previously selected.

Consult documentation of Modbus device(s) to integrate for information about Modbus data format of the points desired to integrate

6. *Address*. Is the Modbus register address to use by IntesisBox to read/write the point into the Modbus device.

Consult documentation of Modbus device(s) to integrate for information about register addresses of the points desired to integrate

7. *Bit*. Bit used inside the Modbus register to encode the digital value for the point. The gateway allows bit decoding from generic 16 bits input/holding Modbus registers. Bit coding into 16 bit input/holding Modbus registers is used for some devices to encode digital values into this type of registers, being these registers normally accessible to be read/write using Modbus function codes 3 and 4 (read holding/input registers). Specifying bit number is only necessary if data format *11- 16 bits digitals* have been selected.
8. *Description*. Signal's description. Used to describe the signal, for identification purposes.
9. *EIS*. KNX data type (Data point) to encode the signal's value. Edit using the mouse right-button-click pop-up menu available on the column. Possible values are: switching (1 bit), dimming (4 bit), float (16 bit), scaling (8 bit), drive control (1 bit), priority (2 bit), float IEEE (32 bit), counter (8 bit), counter (16 bit), counter (32 bit), ASCII char (8 bit).
10. *Group*. Main EIB group address for the point. Format: P/I/S or P/S. Flags *R,W,T,U* explained below will only apply for this main EIB group address, not for listening addresses (if defined). Is the sending group address.
11. *Listening addresses*. EIB group addresses that will be listen by IntesisBox for this point, that is to say, if IntesisBox receives an EIB telegram with destination one of these listening addresses, then the telegram will be taken into account and the corresponding action will be performed on this point. Format: P/I/S or P/S, if more than one is entered then they must be separated by comma.
12. *R*. Indicates if this signal is allowed to be read from KNX system. Possible values: "R" or blank. "R" means flag activated. Edit using the mouse right-button-click pop-up menu available on the column. Freely configurable, with some necessary restrictions (see below).
13. *W*. Indicates if this signal is allowed to be written from KNX system. Possible values: "W" or blank. "W" means flag activated. Edit using the mouse right-button-click pop-up menu available on the column. Freely configurable, with some necessary restrictions (see below).
14. *T*. Indicates if this signal will generate a telegram sending to the KNX system following a change of the point's value, that is to say, any change of value of this point received from the Modbus device will be transmitted to the KNX system if this flag is activated. Possible values: "T" or blank. "T" means flag activated. Edit using the mouse right-button-click pop-up menu available on the column. Freely configurable, with some necessary restrictions (see below).
15. *U*. Indicates if this signal will be updated whenever IntesisBox starts up or after an EIB bus reset. "U" means flag activated for the main EIB group address (a read of the main EIB group address will be performed in the KNX system for the update). "U2" means flag activated for the first listening address defined (a read of the first listening address defined for the point will be performed in the KNX system for the update). Blank means flag not activated. Edit using the mouse right-button-click pop-up menu available on the column. Freely configurable, with some necessary restrictions (see below).
16. *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.
17. *Frac*. Fractional part to consider for the point's value when read/write.

    Some Modbus devices encode for example temperature values (integer + fractional part) in common 2-bytes Modbus registers (using data format 16 bits signed two's complement for example), the problem of using 2-bytes registers is that no fractional part can be encoded. To avoid this problem, the real value of the device is sent multiplied by 10 as just an integer part (a real value of 25.1 will be sent as 251 for example). For this kind of points for example you can specify a value of 1 in *Frac* column, and then the value received by IntesisBox on readings will be divided by 10 and thus will be the real value in the device (including integer and fractional part), and for writings IntesisBox will multiply by 10 the value prior to send it to the Modbus device.

18. Some Modbus devices use 0-based register addresses while others use 1-based register addresses in the Modbus communication (for 0-based devices, the register address 100 is specified as 99 in the Modbus communication frames). Select *0-Based add* if your Modbus devices use 0-based address map (also called JBus), or select *1-Based add* if your Modbus devices use 1-based address map.
19. 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).

KNX flags

All the mentioned KNX Flags (R,W,T,U,U2) related to the KNX part are configurable for every point in IntesisBox, with the following necessary restrictions:

- Points defined with Modbus function codes 1, 2, 3 and 4 (reads) have the following configuration possibilities for KNX flags:
  - Flag T is mandatory (is automatically activated).
  - Flag R is optional.
  - Flag W can be activated only if Modbus function code selected is *1-Read digital outputs* or *3-Read analog registers.*
  - *F*lags U and U2 are forbidden.
  - If flag W is activated and IntesisBox receives a write request for this group address from KNX, then the associated Modbus function code to write will be used to write the new value received into the Modbus slave device (see below *how to configure read/write points*).

- Points defined with Modbus function codes 5, 6, 15 and 16 (writes) have the following configuration possibilities for KNX flags:
  - Flags R and T are forbidden.
  - Flag W is mandatory (is automatically activated).
  - Flags U and U2 are optional.

How to configure read/write points

First of all is important to take into account that different names for Modbus function codes are used in technical literature depending on the manufacturer of the Modbus device. The following table shows the equivalence between nomenclature for function codes used by Intesis Software in IntesisBox and the used in Modbus protocol specification.

Function code	IntesisBox	Modbus protocol specification
01	Read digital outputs	Read Coils
02	Read digital inputs	Read Discrete Inputs
03	Read analog registers	Read Holding Registers
04	Read analog inputs	Read Input Registers
05	Write 1 digital output	Write Single Coil
06	Write 1 analog register	Write Single Register
15	Write multiple digital outputs	Write Multiple Coils
16	Write multiple analog registers	Write Multiple Registers

Given a point in a Modbus slave device, if this point allows to be read and written, different Modbus function codes must be used for read and for write actions (consult the slave documentation for details of what function codes must be used for read and for write). Use the following criteria for configuration of this kind of points in IntesisBox:

1. If the Modbus function code to use for read is *03* and the function code to use for write is *06* (which is very common), then select the function code *3-Read analog registers* in column *Modbus Code* and activate KNX flag W. With this, IntesisBox will use function code 03 for read the point in every polling cycle, and whenever a write request for the point is received from KNX, the new value will be written in the Modbus slave device using function code 06.

2. If the Modbus function code to use for read is *01* and the function code to use for write is *05* (which is also very common), then select the function code *1-Read digital outputs* in column *Modbus Code* and activate KNX flag W. With this, IntesisBox will use function code 01 for read the point in every polling cycle, and whenever a write request for the point is received from KNX, the new value will be written in the Modbus slave device using function code 05.

3. If the Modbus function code to use for read and the function code to use for write are different than 01-05 or 03-06 (sometimes found with specific devices), then you have to declare two points in IntesisBox to perform the read and the write separately. The way to configure this is better explained using an example.

   Imagine you have a device in which a given analog point (register address 100 for example) of type read/write must be read using function code 03 and must be write using function code 16. Let's imagine also that this point must be linked to the KNX group address 1/0/45.

   To be able to read and write this Modbus point from KNX using a single group address you have to configure two points into IntesisBox like the following:

Nb	Dev	Modbus Code	Format	Add.	Bit	Description	EIS	Group	Listening add.	R	W	T	U	Active	Frac.
1	1	3-Read analog registers	4 - 16 bits sig C2	100		example of R/W Modbus point	5 - Float (16 bit)	1/0/45		R		T		1-Si	1
2	1	16-Write multiple analog registers	4 - 16 bits sig C2	100		example of R/W Modbus point	5 - Float (16 bit)		1/0/45		W			1-Si	1

The important configuration parameters to obtain the desired functionality are highlighted in green colour, the rest of configuration parameters are irrelevant in this example. Note that both points must have the same Modbus Address and the same Modbus Format.

Useful tips for a more quick and comfortable use and configuration of the signals table:

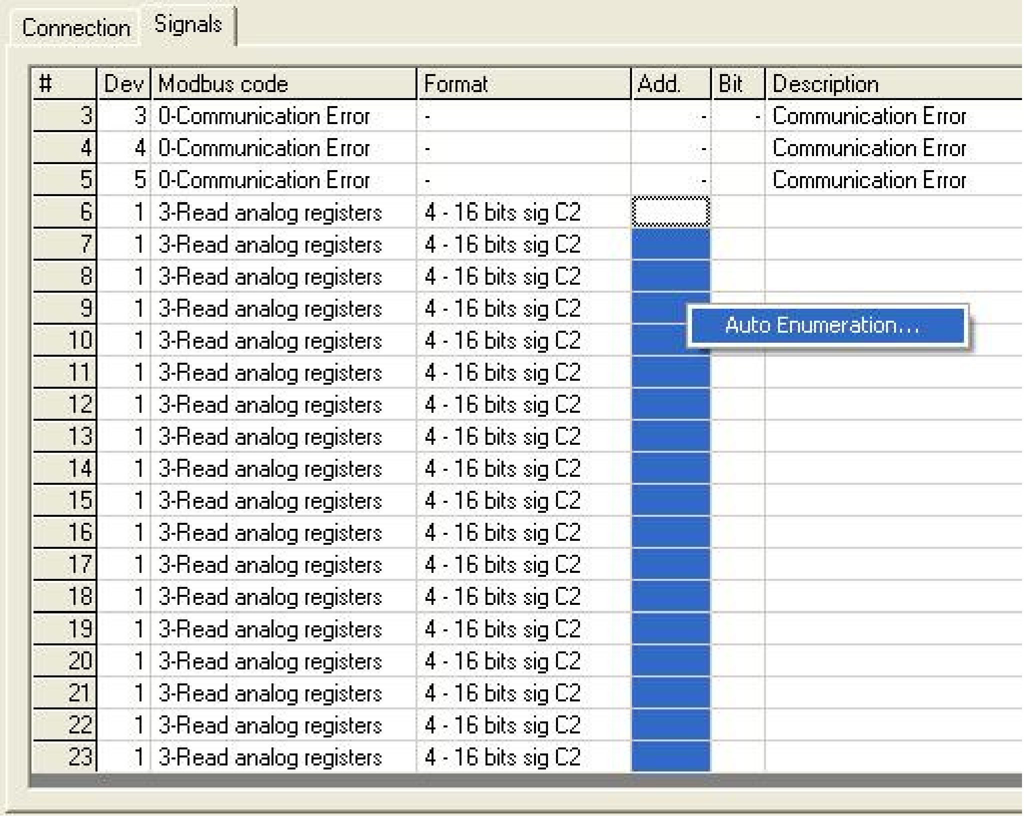
The columns *Modbus Code, Format, EIS, R, W, T, U* and *Active* can be modified selecting one or just more than one consecutive cells in the same column and using the contextual menu appearing with mouse right button click over the cells selected.

The columns *R, W, T, U* and *Active* can be also modified using mouse-left-button double-click over the desired cell.

In columns *Dev, Slave, Addr* and *Frac* you can enter the desired value individually per cell or you can auto enumerate some consecutive cells, for this last follow these steps:

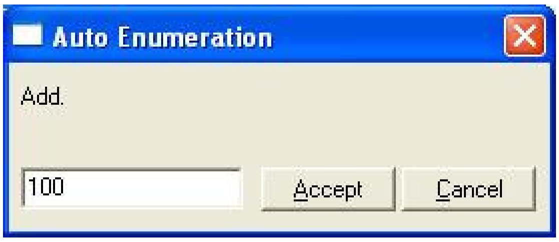
1. Select using the left mouse button (clicking and dragging) the field (Dev, Slave, Addr, Frac) of all the rows in the list to which you want to automatically assign values (must be consecutive rows).

2. Click right mouse button over the selected fields and select *Auto Enumeration* option from the pop-up menu that will appear.



#	Dev	Modbus code	Format	Add.	Bit	Description
3	3	0-Communication Error	-	-	-	Communication Error
4	4	0-Communication Error	-	-		Communication Error
5	5	0-Communication Error	-	-		Communication Error
6	1	3-Read analog registers	4 - 16 bits sig C2			
7	1	3-Read analog registers	4 - 16 bits sig C2			
8	1	3-Read analog registers	4 - 16 bits sig C2			
9	1	3-Read analog registers	4 - 16 bits sig C2			
10	1	3-Read analog registers	4 - 16 bits sig C2			
11	1	3-Read analog registers	4 - 16 bits sig C2			
12	1	3-Read analog registers	4 - 16 bits sig C2			
13	1	3-Read analog registers	4 - 16 bits sig C2			
14	1	3-Read analog registers	4 - 16 bits sig C2			
15	1	3-Read analog registers	4 - 16 bits sig C2			
16	1	3-Read analog registers	4 - 16 bits sig C2			
17	1	3-Read analog registers	4 - 16 bits sig C2			
18	1	3-Read analog registers	4 - 16 bits sig C2			
19	1	3-Read analog registers	4 - 16 bits sig C2			
20	1	3-Read analog registers	4 - 16 bits sig C2			
21	1	3-Read analog registers	4 - 16 bits sig C2			
22	1	3-Read analog registers	4 - 16 bits sig C2			
23	1	3-Read analog registers	4 - 16 bits sig C2			

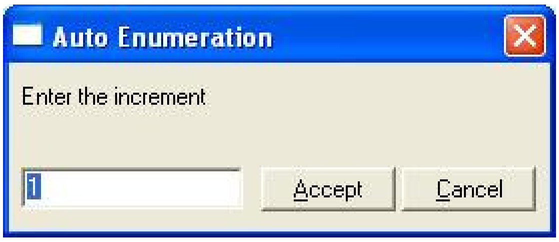
3. Enter the first value to assign.



Add.

100

4. Enter the increment between consecutive assignments. For example selecting 100 for the first value and an increment of 1, the values generated will be 100, 101, 102, 103, 104.. and so on. To assign the same value to all the rows (useful to assign the same Device number in the column *Dev* for some consecutive rows) just select the desired value and an increment of 0.


Auto Enumeration

Enter the increment

1

4.4.1 Remember

- If "R" is not activated, the KNX group address could not be read by Read Requests from KNX.
- If "W" is not activated, no write on the group address could be done from KNX, neither on the links (listening addresses).
- If "U" is activated, after IntesisBox start-up or a KNX bus reset, READ Requests will be sent to KNX to update the sending group.
- The groups defined as just links, will take the EIS of the first linked group.
- The data of the groups read from KNX due to read requests between other KNX devices, will be treated as writes on the groups (standard behaviour BCU1).
- The signals that are inputs to KNX must be configured as: T (mandatory), R (optional).
- The signals that are outputs from KNX must be configured as: W (mandatory), U (optional).
- Automatic type conversion is performed by IntesisBox, for example if an EIS5 is received by a group of type EIS1, then it is performed the conversion EIS1=(EIS5<>0), or EIS5=EIS9 and EIS9=EIS5, EIS6=EIS5(0..255), etc.
- It is recommended that listening addresses associated to different groups maintain the same EIS in all the groups, if not, non desired conversions can be performed.
- A write from KNX is propagated to the external system through the group address and also through the listening addresses.
- A write from the external system is NOT propagated to KNX through the listening addresses, but DO update the local sending groups that are used as listening addresses.
- If a sending group is used as listening address with other local sending groups, a write from the external system will update the sending group but not the listening addresses.

4.4.2 Restrictions

- It is allowed group numbers in format P/I/S, P/S or directly the group number coded.
- It is not allowed duplicated sending groups (column Group).
- Group 0 is not allowed, it is used for signals without sending group.
- NO signal is allowed with none of R-W-T-U flags activated.
- Empty groups are allowed, but only if they have just W activated and one or more listening addresses.
- Duplicated groups in the same listening address field are not allowed.
- It is not allowed a listening address that is the same as the sending group (circular reference).
- Listening addresses are not allowed if the flag W is not activated. Without W activated, the listening addresses would not work.
- Only those EIS defined are allowed.

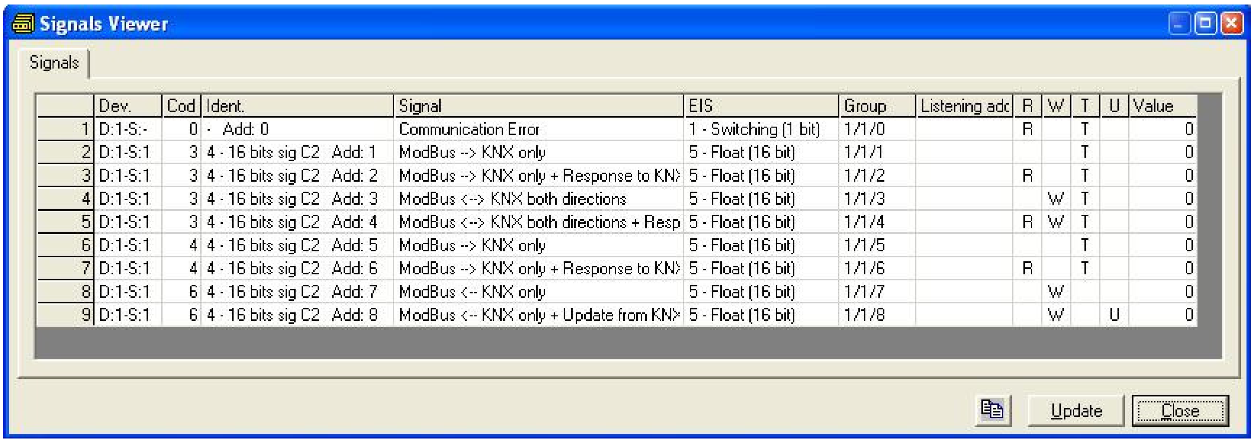
4.5 Saving the configuration and sending it to IntesisBox

When the configuration is finished, click on button *Save* to save it to the project folder on hard disk. You will be prompted to generate the configuration file to send to IntesisBox, if you select *Yes*, the binary file containing the configuration for IntesisBox will be generated and saved also into the project folder.

Once the configuration has been saved and the configuration file for IntesisBox has been generated, to send this 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.

4.6 Signals viewer

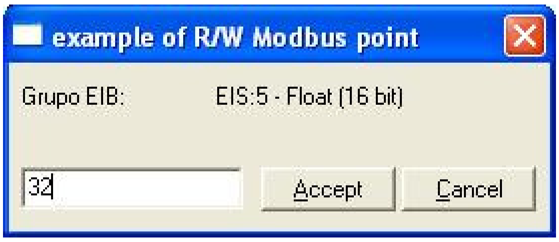
Once IntesisBox is running with the correct configuration, to supervise the status of the configured signals, select menu *View -> Signals*. The Signals Viewer window will be opened. This window shows all the active IntesisBox's signals with its main configuration parameters and its real time value in 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 signal values, from this moment the signal values will be maintained updated until the connection is closed.



	Dev.	Cod	Ident.	Signal	EIS	Group	Listening adc	R	W	T	U	Value
1	D:1-S:-	0	- Add: 0	Communication Error	1 - Switching (1 bit)	1/1/0		R		T		0
2	D:1-S:1	3	4 - 16 bits sig C2 Add: 1	ModBus --> KNX only	5 - Float (16 bit)	1/1/1				T		0
3	D:1-S:1	3	4 - 16 bits sig C2 Add: 2	ModBus --> KNX only + Response to KN>	5 - Float (16 bit)	1/1/2		R		T		0
4	D:1-S:1	3	4 - 16 bits sig C2 Add: 3	ModBus <--> KNX both directions	5 - Float (16 bit)	1/1/3			W	T		0
5	D:1-S:1	3	4 - 16 bits sig C2 Add: 4	ModBus <--> KNX both directions + Resp	5 - Float (16 bit)	1/1/4		R	W	T		0
6	D:1-S:1	4	4 - 16 bits sig C2 Add: 5	ModBus --> KNX only	5 - Float (16 bit)	1/1/5				T		0
7	D:1-S:1	4	4 - 16 bits sig C2 Add: 6	ModBus --> KNX only + Response to KN>	5 - Float (16 bit)	1/1/6		R		T		0
8	D:1-S:1	6	4 - 16 bits sig C2 Add: 7	ModBus <-- KNX only	5 - Float (16 bit)	1/1/7			W			0
9	D:1-S:1	6	4 - 16 bits sig C2 Add: 8	ModBus <-- KNX only + Update from KN>	5 - Float (16 bit)	1/1/8			W		U	0

The signals viewer can be used although only one system is connected to the IntesisBox, *KNX* or *Modbus*, 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 has T activated, its value will be updated and a telegram will be sent to KNX indicating the new value, the same way as if it has been received from *Modbus* system. If the signal has W activated, the new value entered will be sent to the external system, Modbus in this case, the same way as if it has been received from *KNX*.



Grupo EIB: EIS:5 - Float (16 bit)

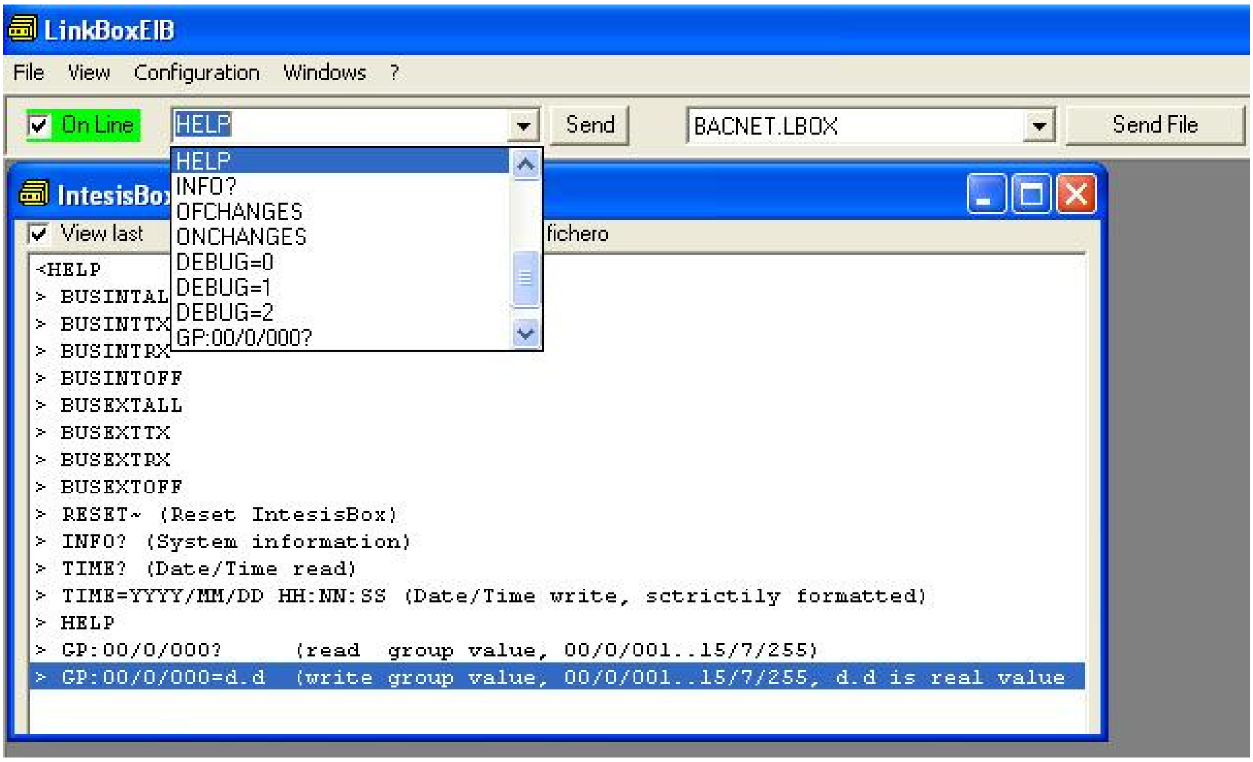
32

This tool is very useful to test any of the systems connected to the IntesisBox, KNX and *Modbus* 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

LinkBoxEIB 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 operation, 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.



```
<HELP
> BUSINTAL
> BUSINTTX
> BUSINTRX
> BUSINTOFF
> BUSEXTALL
> BUSEXTTX
> BUSEXTRX
> BUSEXTOFF
> RESET~ (Reset IntesisBox)
> INFO? (System information)
> TIME? (Date/Time read)
> TIME=YYYY/MM/DD HH:NN:SS (Date/Time write, sctrictily formatted)
> HELP
> GP:00/0/000?      (read  group value, 00/0/001..15/7/255)
> GP:00/0/000=d.d   (write group value, 00/0/001..15/7/255, d.d is real value
```

4.8 Files

LinkBoxEIB saves the IntesisBox configuration in the following files inside the project folder:

PROJECT.INI	.ini file containing general information referent to the project
MODBUS_TCP.INI	.ini file containing the information referent to the connection window and other special adjustments
MODBUS_TCP.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 *Configuration -> IntesisBox* in LinkBoxEIB, 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.
MODBUS_TCP.LBOX	Binary file created from the information in the files described above. This is the file really uploaded to IntesisBox.

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 LinkBoxEIB due, for example, to a failure of the hard disk in the PC where LinkBoxEIB was previously installed.

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

5. Setup process and troubleshooting

5.1 Pre-requisites

It is necessary to have a KNX TP-1 (EIB) system operative and ready to be connected to the KNX port of IntesisBox.

It is necessary to have the IP network connection near IntesisBox with all Modbus TCP slave devices connected to this network.

Connectors, connection cables, and PC for LinkBoxEIB are not supplied by Intesis Software for this standard integration. The items supplied by Intesis Software for this integration are:

- IntesisBox-KNX gateway with KNX internal protocol and Modbus TCP master external protocol firmware loaded.
- Console cable. Standard DB9F-DB9M cable 1.8 meters long.
- LinkBoxEIB software.
- Product documentation.

5.2 Setup procedure

1. Install LinkBoxEIB on your laptop.

2. Install IntesisBox in the desired installation site. The mounting can be on DIN rail or on a stable not vibrating surface (DIN rail mounting inside a metallic industrial cabinet connected to ground is recommended).

3. Connect the KNX TP-1 (EIB) bus cable to the port marked as **KNX TP-1 (EIB)** of IntesisBox. (See details for this bus cable in section *Connections* of this document).

4. Connect the communication cable coming from the Ethernet network or the Modbus TCP device to the port marked as **ETH** of IntesisBox, use a CAT-5 crossed cable if you connect directly to the Modbus TCP device, or a straight cable if you connect to a hub or switch. (See details for this communication cable in section *Connections* of this document).

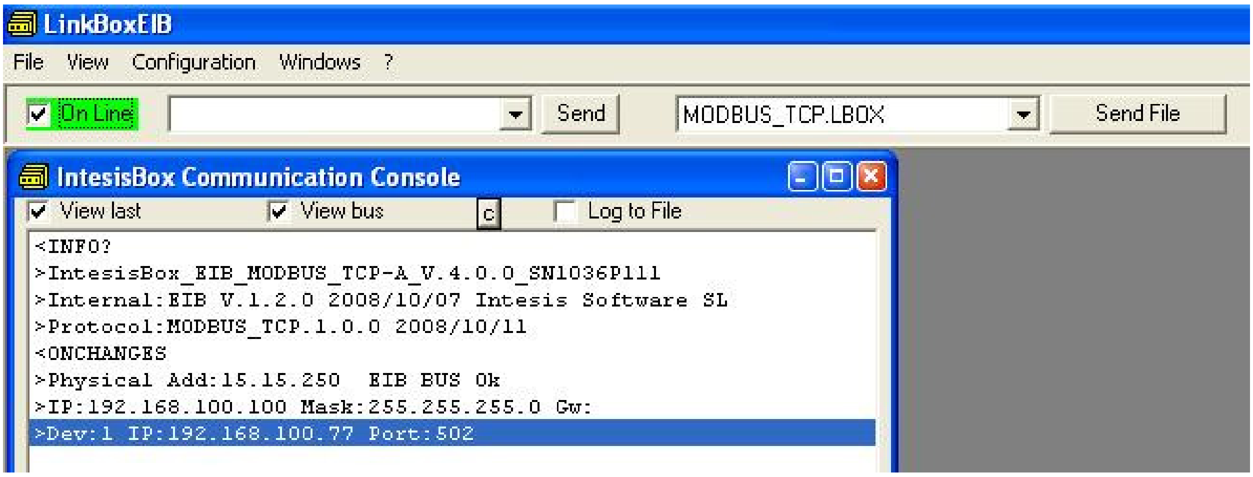
5. Power up IntesisBox using a standard power supply 220/125VAC-12VDC/300mA for example.

   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**.
   - The use of AC power supplies only if they are floating and not powering any other device.

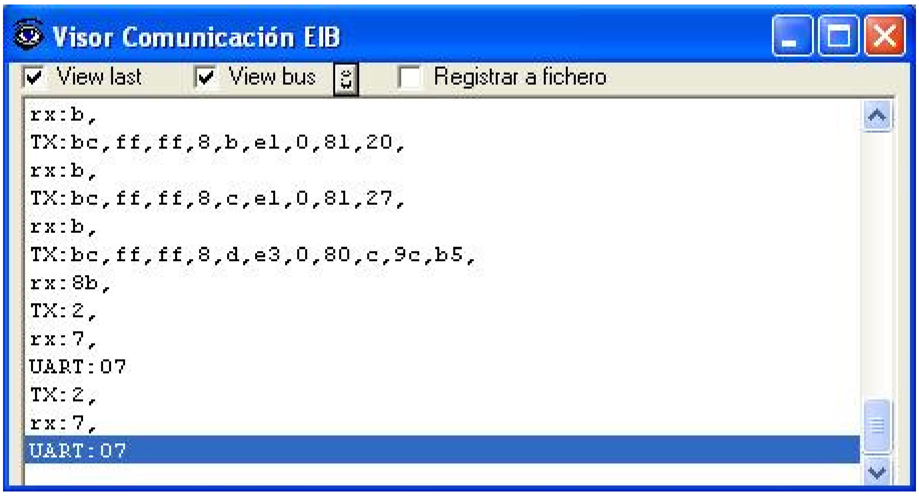
6. Connect the communication cable coming from the serial port of your laptop PC to the port marked as **PC Console** of IntesisBox. (See details for this communication cable in section *Connections* of this document).

7. Open LinkBoxEIB, create a new project selecting a copy of the one named **DEMO Modbus TCP** and give it the name desired, select the serial port used to connect to IntesisBox (menu Configuration -> Connection) and switch working mode to *on-line* (checkbox *off-line/on-line*). The IntesisBox identification must appear in the *IntesisBox communication console* window as showed below.



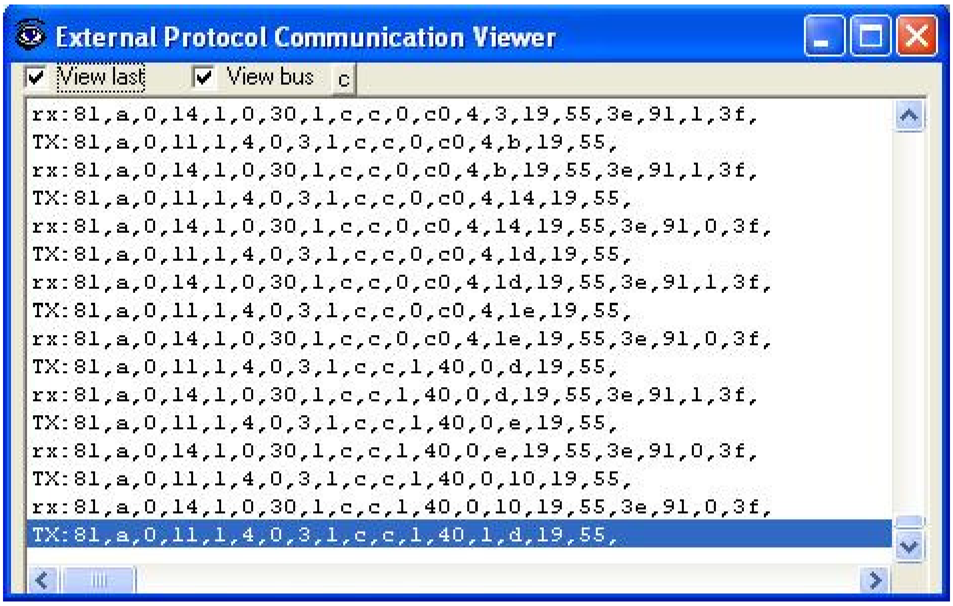
```
<INFO?
>IntesisBox_EIB_MODBUS_TCP-A_V.4.0.0_SN1036P111
>Internal:EIB V.1.2.0 2008/10/07 Intesis Software SL
>Protocol:MODBUS_TCP.1.0.0 2008/10/11
<ONCHANGES
>Physical Add:15.15.250  EIB BUS Ok
>IP:192.168.100.100 Mask:255.255.255.0 Gw:
>Dev:1 IP:192.168.100.77 Port:502
```

8. Open the *EIB Communication Viewer* window (menu View -> Bus -> EIB) and check that there is communication activity, some TX frames and some other rx frames. This means that the communication with the KNX system is ok. In case there is no communication activity between IntesisBox and the KNX system check that KNX bus is operative and well connected to the IntesisBox.



```
rx:b,
TX:bc,ff,ff,8,b,e1,0,81,20,
rx:b,
TX:bc,ff,ff,8,c,e1,0,81,27,
rx:b,
TX:bc,ff,ff,8,d,e3,0,80,c,9c,b5,
rx:8b,
TX:2,
rx:7,
UART:07
TX:2,
rx:7,
UART:07
```

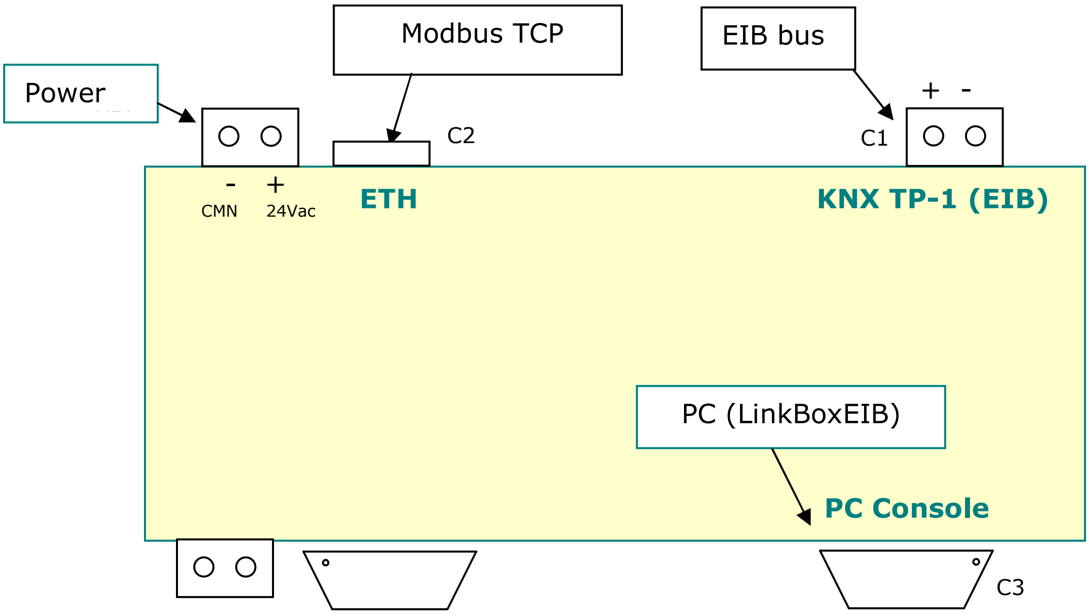
9. Open the *External Protocol Communication Viewer* window (menu View -> Bus -> External system) and check that there is communication activity, some TX frames and some other rx frames as showed in the figure below. This means that the communication with the Modbus slave devices is ok.



```
rx:81,a,0,14,1,0,30,1,c,c,0,c0,4,3,19,55,3e,91,1,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,0,c0,4,b,19,55,
rx:81,a,0,14,1,0,30,1,c,c,0,c0,4,b,19,55,3e,91,1,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,0,c0,4,14,19,55,
rx:81,a,0,14,1,0,30,1,c,c,0,c0,4,14,19,55,3e,91,0,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,0,c0,4,1d,19,55,
rx:81,a,0,14,1,0,30,1,c,c,0,c0,4,1d,19,55,3e,91,1,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,0,c0,4,1e,19,55,
rx:81,a,0,14,1,0,30,1,c,c,0,c0,4,1e,19,55,3e,91,0,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,1,40,0,d,19,55,
rx:81,a,0,14,1,0,30,1,c,c,1,40,0,d,19,55,3e,91,1,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,1,40,0,e,19,55,
rx:81,a,0,14,1,0,30,1,c,c,1,40,0,e,19,55,3e,91,0,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,1,40,0,10,19,55,
rx:81,a,0,14,1,0,30,1,c,c,1,40,0,10,19,55,3e,91,0,3f,
TX:81,a,0,11,1,4,0,3,1,c,c,1,40,1,d,19,55,
```

In case there is no response from the Modbus devices to the frames sent by IntesisBox, check that they are operative and configured with the correct communication parameters, check the IntesisBox's Modbus interface connection. See details for the communication cable between IntesisBox and Modbus devices in section *Connections* of this document.

6. Connections



IntesisBox (Terminal block 2C)	C1	KNX TP-1 (EIB) Connection		EIB bus
Cable				
+		↔		+
-		↔		-

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

IntesisBox (RJ45 F)	C2	Modbus TCP devices Connection Ethernet		Modbus Device (RJ45 F)
Cable (RJ45 M)		Cable UTP/FTP Cat5 Crossed		Cable (RJ45 M)
		↔		1 device
		Cable UTP/FTP Cat5 Straight		
		↔	Hub	N devices

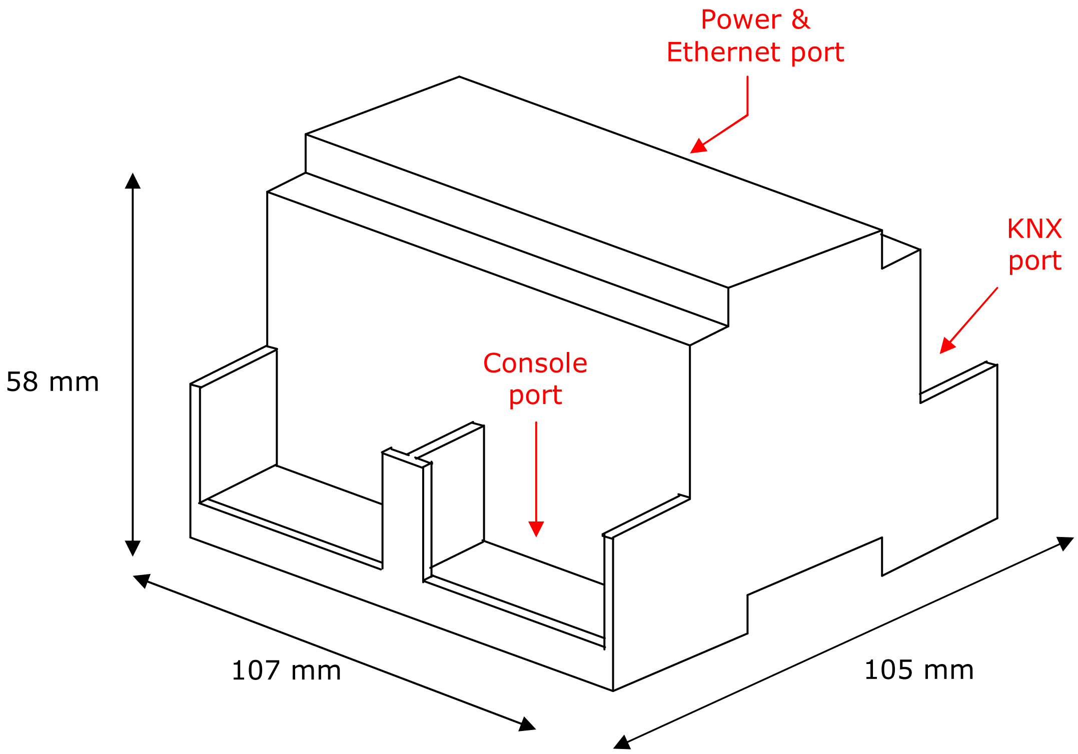
7. Mechanical & electrical characteristics

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 bloc for power connection (2 poles).
Mounting	Surface. Wall. DIN rail EN60715 TH35.
Modbus TCP port	1 x Ethernet 10BT RJ45.
KNX port	1 x KNX TP1 (EIB) port opto-isolated. Plug-in terminal bloc (2 poles).
LED indicators	1 x Power. 2 x KNX port activity (Tx, Rx). 2 x Ethernet port link and activity (LNK, ACT). 1 x KNX programming/bus.[1]
Push buttons	1 x KNX programming.[1]
Console port	RS232. DB9 female connector (DCE).
Configuration	Via console port.[2]
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

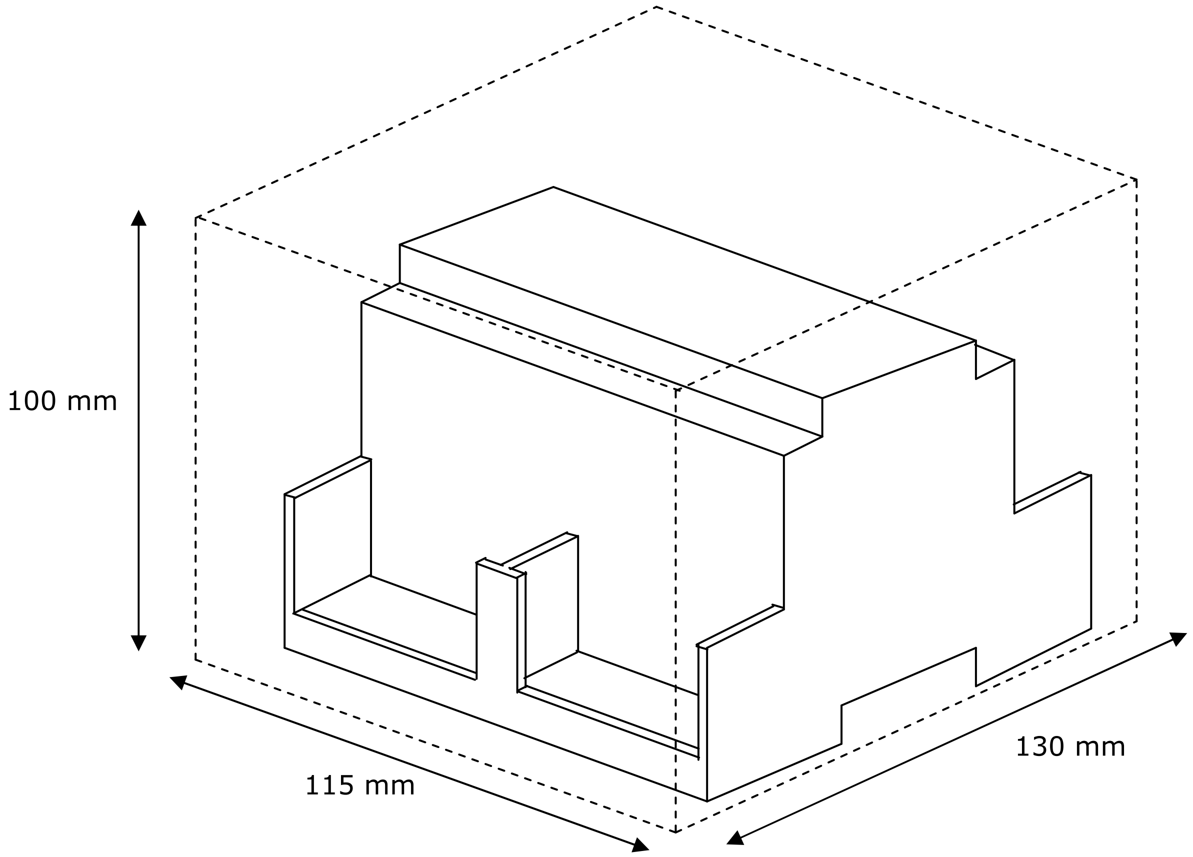
[1] Not operational for the moment. Reserved for future use.

[2] 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.

8. Dimensions



Free space recommended installing the device into a cabinet (wall or DIN rail mounting), with spacing enough for external connections:


100 mm
115 mm
130 mm