

IntesisBox[®] LON KNX

User's Manual

Issue date: 25/05/2011
r2 eng

© Intesis Software S.L. 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 - 1º
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 KNX (TP-1) systems or devices into LON (TP/FT-10) control systems

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

IBOX-LON-KNX-200

Tiny model, supporting:

- 200 communication objects at KNX side (bindable to 8000 group addresses)

IBOX-LON-KNX-A

Basic model, supporting:

- 500 communication objects at KNX side (bindable to 8000 group addresses)

IBOX-LON-KNX-B

Extended model, supporting:

- 4000 communication objects at KNX side (bindable to 8000 group addresses)

INDEX

1.	Description	5
1.1	Introduction	5
1.2	Capacities of IntesisBox	6
2.	IntesisBox hardware	7
2.1	Power Supply	7
2.2	LEDs.....	8
2.3	Buttons.....	8
3.	LinkBoxLON. Configuration & monitoring tool for IntesisBox LON series.....	9
3.1	Installing LinkBoxLON	9
3.2	Connecting the PC with the IntesisBox	9
3.2.1	USB connection setup	9
3.2.2	Ethernet connection setup	10
3.2.3	First communication between LinkBoxLON and IntesisBox	11
3.2.4	Changing IP settings.....	12
3.2.5	Restoring default IP settings.....	13
3.3	Creating a configuration for IntesisBox.....	13
3.3.1	LON interface configuration	15
3.3.2	KNX interface configuration	23
3.3.3	Creating bindings between LON and KNX	29
3.4	Sending the configuration to the gateway.....	33
3.5	Firmware updates	34
4.	Mechanical & electrical characteristics	36
5.	Dimensions.....	37
6.	Annexes	38
6.1	Exporting group addresses from an ETS project in a CSV file	38
6.2	Exporting group addresses from an ETS project in an ESF file	39
6.3	Using the secondary LON interface	39

1. Description

1.1 Introduction

The *IntesisBox LON-KNX* is a device meant to enable interworking between KNX (TP-1) and LON (TP/FT-10) control networks.

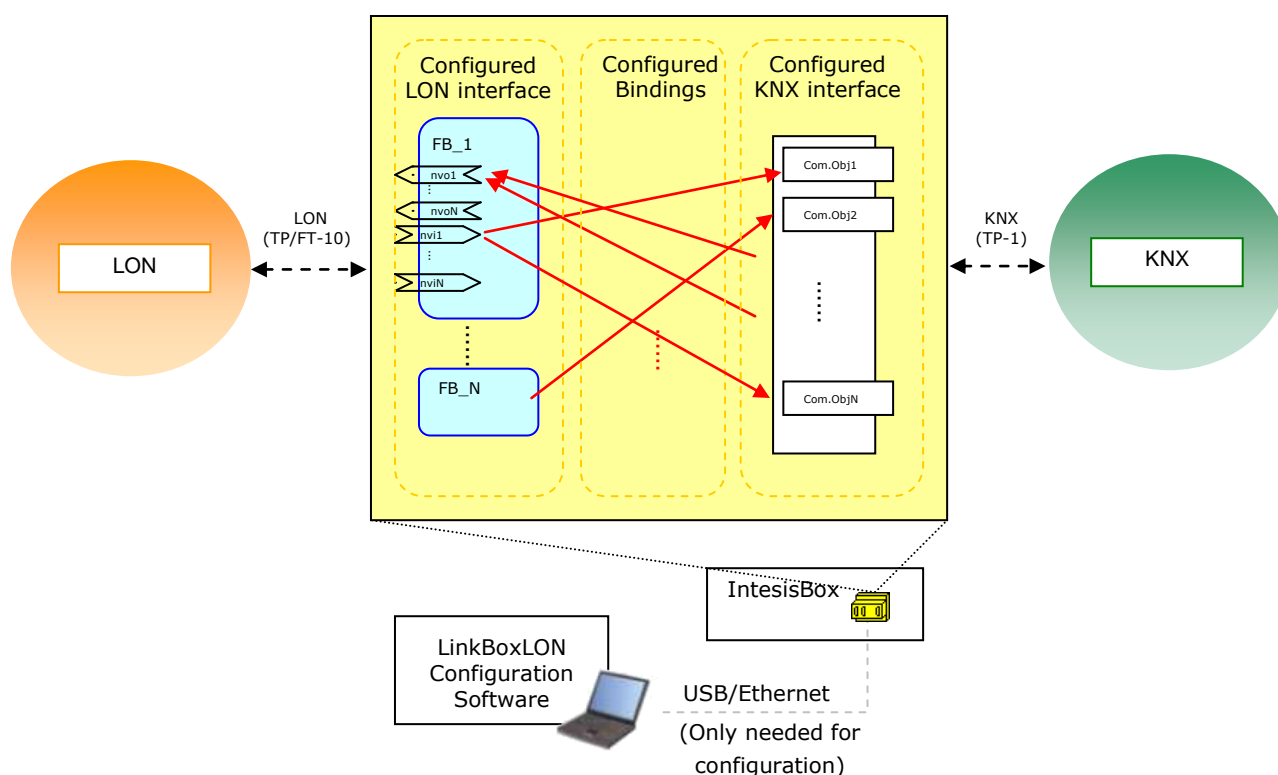
This ending is achieved by effectively implementing both a Network Variables' (NVs) interface on the LON side and a Communication Objects' interface on the KNX side. Datapoints on either side (NVs at LON or Communication Objects at KNX) are internally bound within the *IntesisBox LON-KNX*, allowing the information to be moved from KNX to LON and viceversa.

Thus, *IntesisBox LON-KNX* will be seen as a regular LON device on the LON TP/FT-10 network and a regular KNX device at the KNX TP-1 network.

The set of Communication Objects on the KNX interface, the set of NVs at the LON interface and the bindings between them are defined in the *IntesisBox LON-KNX* at configuration time. This document describes the procedure of performing this configuration, by using *LinkBoxLON* Windows-based tool.

Familiarity with both KNX and LON technologies and technical terms is assumed throughout the document.

From now on, with the aim of easing the read of this document, the word "IntesisBox" will be used instead of the full gateway model description "IntesisBox LON - KNX".



1.2 Capacities of IntesisBox

There are three different models of *IntesisBox*® *KNX – LON*, which vary in the number of KNX communication object they support:

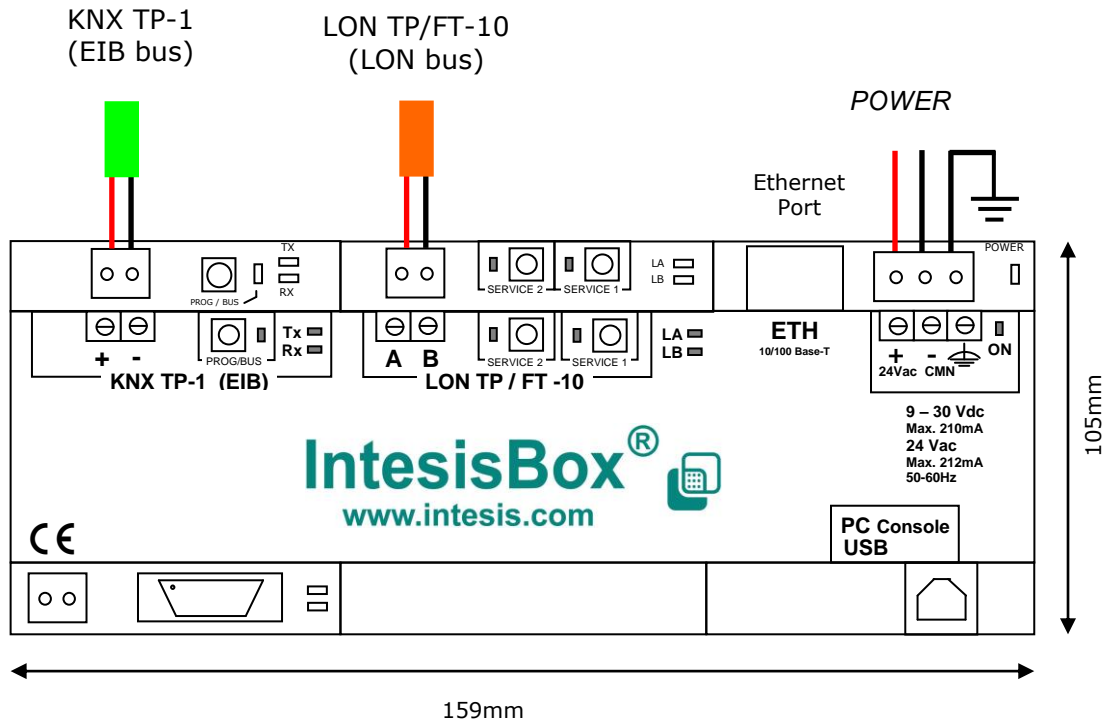
- Tiny model, *Ref.:IBOX-LON-KNX-200*
- Basic model, *Ref.:IBOX-LON-KNX-A*
- Extended model, *Ref.:IBOX-LON-KNX-B*

Concept	Max. (Tiny version)	Max. (Basic version)	Max. (Extended version)	Notes
LON (TP/FT-10) interface				
Supported number of network variables	4000	4000	4000	
Supported number of address table entries	512	512	512	This number poses as the maximum number of LON devices to which the IntesisBox can <i>talk</i>
Number of LON interfaces	2	2	2	2 physical interfaces allow to communicate with IntesisBox using both Enhanced Command Set (ECS) and Legacy network management
KNX (TP-1) interface				
Supported number of communication objects	200	500	4000	
Supported number of group addresses	8000	8000	8000	Each address can be freely associated to any communication object

2. IntesisBox hardware

Device uses a standard casing which allows DIN EN60715 TH35 rail mounting. Its plastic meets standard PC UL 94 V0.

Sizes are 159mm x 105mm x 58mm.



Ensure proper space for all connectors when mounted. In normal operation only power supply, LON TP/FT-10 and KNX TP-1 terminals will be connected. During setup & configuration of the device, USB and Ethernet ports will be needed.

2.1 Power Supply

IntesisBox can be powered from either an AC or a DC power supply, with following requirements:

- 9 to 30Vdc +/-10% 2.5W
- 24VAC +/-10% 2.5VA

Power connector is a 2 pole plug-in screw terminal block.

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

- Use floating DC power supplies or with the negative terminal connected to earth.
Never use a DC power supply with the positive terminal connected to earth.
- Use AC power supplies only if they are floating and not powering any other device

2.2 LEDs

LED	Behavior								
POWER ON (red)	ON when the device is powered up								
LON SERVICE1 / SERVICE2 (yellow)	ON during device startup (5 to 20s, depending on number of configured datapoints) BLINKING 1s ON/OFF when the LON interface is not commissioned OFF when the LON interface is commissioned								
KNX PROGRAMMING (red)	OFF normally ON when the KNX interface is in programming mode								
KNX TX (yellow) / RX (green)	Show activity at KNX bus. BLINK when a KNX frame is being received or sent								
LA and LB (green)	<p>Both are ON during startup sequence BLINKING LB AND LA ALTERNATIVELY after startup when there is no valid configuration loaded on the device (first time is plugged in and/or after a firmware update) After startup, LA reflects the status of LON bus and LB reflects the status of KNX bus in the following manner:</p> <table border="1"> <tr> <td>LB ON</td><td>No KNX bus voltage detected at KNX port</td></tr> <tr> <td>LB BLINKING</td><td>IntesisBox is performing KNX bus group addresses' READs after device startup or bus recovery</td></tr> <tr> <td>LB OFF</td><td>KNX bus voltage detected at KNX port, and KNX side of the gateway has been updated (READs have been triggered)</td></tr> <tr> <td>LA¹ BLINKING</td><td>IntesisBox is requesting polled-nvi's to bound devices at LON side</td></tr> </table>	LB ON	No KNX bus voltage detected at KNX port	LB BLINKING	IntesisBox is performing KNX bus group addresses' READs after device startup or bus recovery	LB OFF	KNX bus voltage detected at KNX port, and KNX side of the gateway has been updated (READs have been triggered)	LA ¹ BLINKING	IntesisBox is requesting polled-nvi's to bound devices at LON side
LB ON	No KNX bus voltage detected at KNX port								
LB BLINKING	IntesisBox is performing KNX bus group addresses' READs after device startup or bus recovery								
LB OFF	KNX bus voltage detected at KNX port, and KNX side of the gateway has been updated (READs have been triggered)								
LA ¹ BLINKING	IntesisBox is requesting polled-nvi's to bound devices at LON side								

2.3 Buttons

Button	Functionality
LON SERVICE1 / SERVICE2 BUTTONS	Regular LON service button (sends a broadcast service pin message to the LON network, telling neuron-id and program-id)
KNX PROGRAMMING BUTTON	Set the KNX side of the IntesisBox in programming mode / withdraw programming mode. Programming mode for KNX is still not used by current implementation of the protocol.

¹ LA and LB are meant to have symmetric functionality with regard to LON and KNX respectively. This is: ON – bus failure, OFF – bus connected, BLINKING – requesting initial values. Though, failure in the connection of LON cannot be detected by IntesisBox. This way, except during IntesisBox startup, LA will generally be OFF, when not BLINKING.

3. LinkBoxLON. Configuration & monitoring tool for IntesisBox LON series

All interaction between the PC and the IntesisBox is performed by means of the software tool LinkBoxLON, provided along with the hardware device itself.

This section explains how to use this software tool to configure the IntesisBox and monitor its operation.

3.1 Installing LinkBoxLON

Periodically, new free versions of LinkboxLon are released including improvements, fixes, support for new firmware versions of IntesisBox or support for newer products of IntesisBox LON series family of devices. Check the link in the installation sheet supplied with the device to ensure that you have latest version of the tool.

It is supplied in the shape of a self-extracting setup utility. Supported operating systems are Windows XP, Windows Vista and Windows 7.

3.2 Connecting the PC with the IntesisBox

LinkBoxLON can communicate with the IntesisBox using one of either the FTDI-based USB port or the Ethernet port.

3.2.1 USB connection setup

Download and unzip FTDI driver for IntesisBox from www.intesis.com.

Power up the IntesisBox and using the supplied 1.8m USB cable, connect it to the PC where LinkBoxLON is installed.

Windows will detect the new device as "FT232R USB UART" and will then open up the "Found New Hardware Wizard", asking whether to connect to Windows Update service to search for its driver. Select "No, not this time" and click "Next".

In next window of the wizard, select "Install from a list or specific location (Advanced)" and click "Next".

Finally, it will ask for the search location for the driver. Select the radio button "Search for the best driver in these locations" and browse to the location in which you unzipped the FTDI driver. Click next.

During installation of the driver, windows will inform that the driver "has not passed Windows Logo testing to verify its compatibility with Windows XP". You can safely ignore it and click "Continue anyway".

Right after the driver for "FT232R USB UART" device has been installed, Windows will detect a new device over again, this time called "USB Serial Port". This is a new COM port that has been installed in your system by the previous driver, with which you will get to communicate with IntesisBox. You will have to follow the process you just followed for "FT232R USB UART" over again.

Once both drivers are installed, Windows will have assigned a new COM port for IntesisBox. The COM port number that will be used will vary from one system to another and needs to be checked specifically.

In order to know which COM port has been assigned go to "Control Panel" in your system and (in "Classic View" mode) double click the icon "System".

Then, open up the "Device Manager" in tab "Hardware". Expanding the category "Ports (COM & LPT)" of the "Device Manager" you will find all COM ports installed in your system. In particular, you have to look for a device called "USB Serial Port (COMx)". This tells you which COM port you need to select to communicate with IntesisBox.

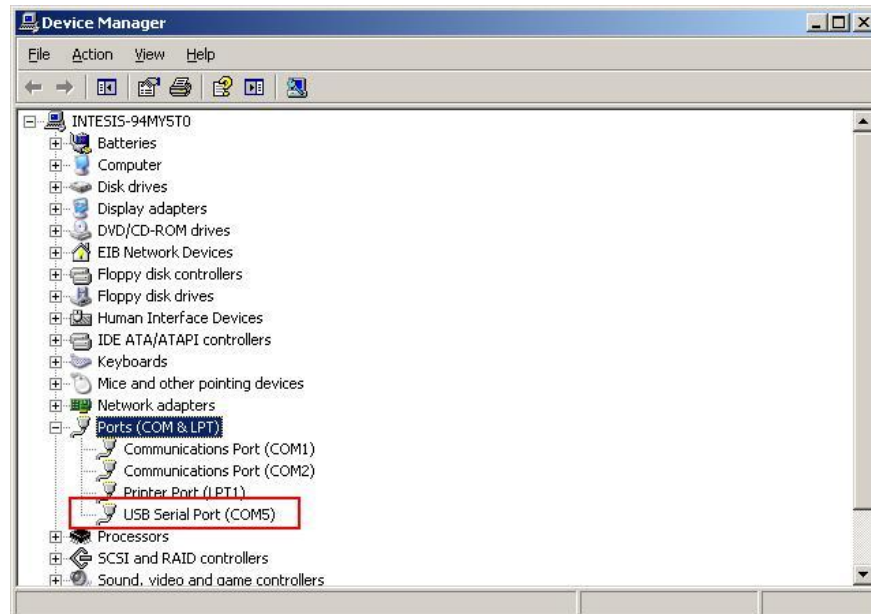


Figure 1 - Detecting IntesisBox COM port

Next step is starting LinkBoxLON for checking that you can effectively communicate with IntesisBox. In section 3.2.3 you will find the details on how to check it.

3.2.2 Ethernet connection setup

Power up IntesisBox and connect suitable Ethernet cable:

- Use a patched Ethernet cable if you are connecting your PC's Ethernet adapter directly to the IntesisBox
- Or use a straight Ethernet cable if you are to connect to the IntesisBox to a switch or router in your LAN

IntesisBox comes from factory with configured IP address 192.168.150.254. If the IP address of your PC is in the subnet 192.168.150.* you are ready to communicate with IntesisBox. If your PC has an IP address in a different subnet, you will need to add a route in Windows routing table, by doing following:

Choose "Start->Run..." and enter:

cmd.exe

This will open up a Windows console shell, with which you can add the route to the IntesisBox with following command line:

```
route add 192.168.150.254 %COMPUTERNAME%
```

In Windows Vista you may have trouble using environment variable %COMPUTERNAME%. If this is your case, try issuing the same command changing %COMPUTERNAME% for the IP of your PC (you can get the IP from your computer with shell command ipconfig).

Finally, ping 192.168.150.254 from the command shell so to check that your PC can see IntesisBox in the network:

```
C:\>ping 192.168.150.254
```

```
Pinging 192.168.150.254 with 32 bytes of data:
```

```
Reply from 192.168.150.254: bytes=32 time<1ms TTL=255
Reply from 192.168.150.254: bytes=32 time<1ms TTL=255
Reply from 192.168.150.254: bytes=32 time<1ms TTL=255
Reply from 192.168.150.254: bytes=32 time=1ms TTL=255
```

```
Ping statistics for 192.168.150.254:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Next step is starting LinkBoxLON for checking that you can effectively communicate with IntesisBox. In next section you will find the details on how to check it.

3.2.3 First communication between LinkBoxLON and IntesisBox

Start LinkBoxLON by choosing Start->Programs->Intesis->LinkBoxLON->LinkBoxLON.

Right after program start, it will open a project manager asking which project to choose. After installation you will only have a sample project called "Demo LON-KNX". Click "Select" to continue with this project.

In order to connect to IntesisBox, first you need to setup the communication parameters in LinkBoxLON (COM port for USB link or IP settings for Ethernet). Click the button "Serial" from the button bar below the menu and enter appropriate settings in "IntesisBox connection" window.



Figure 2 - LinkBoxLON button bar / "Connect" and "Serial" buttons

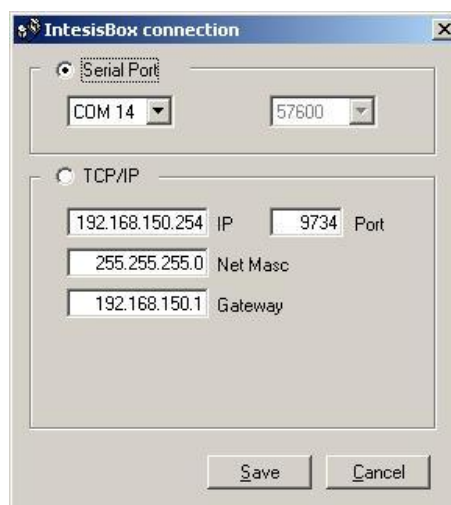


Figure 3 - IntesisBox connection settings window

Click "Save" when you are done with the settings. Button "Serial" that you have clicked on the button bar will change to "TCP/IP" if you selected this type of link.

Now, having IntesisBox properly connected to the PC (USB or Ethernet) and powered up, click button "Connect" (on the left side next to "Serial"). You should get an output similar to the one in Figure 4 in window "IntesisBox Communication Console".

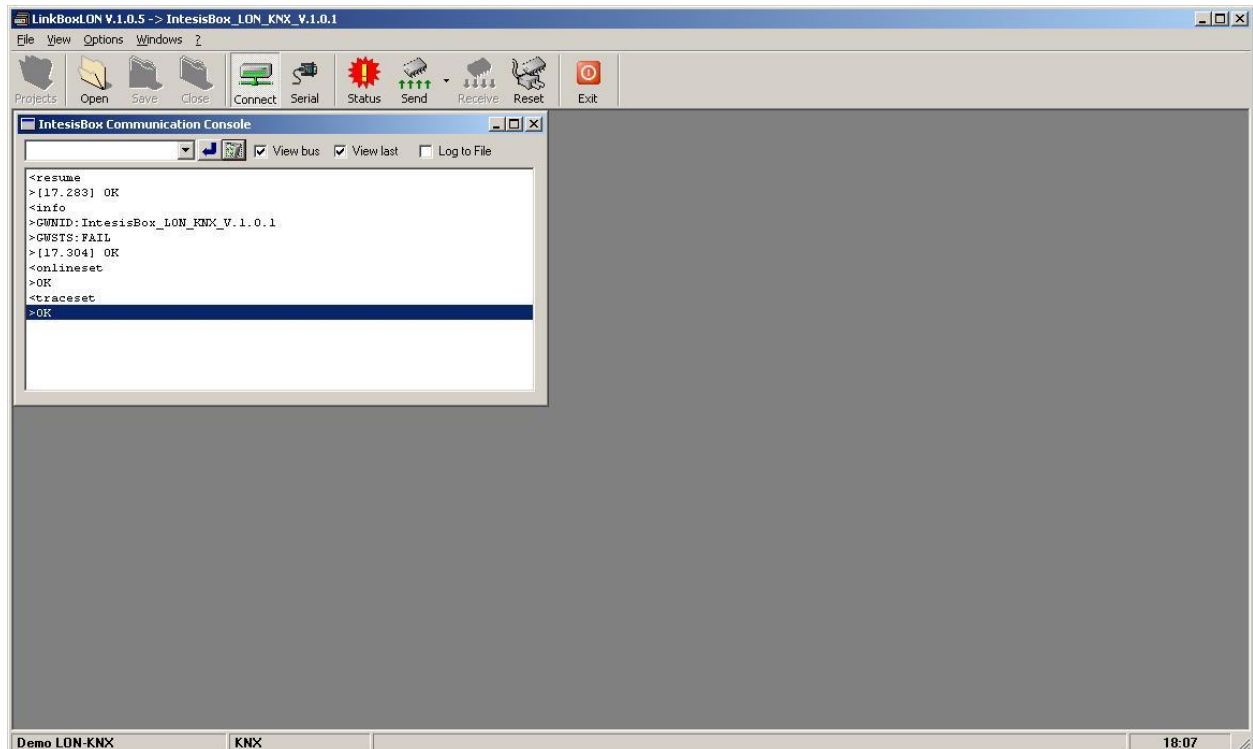


Figure 4 - First connection to IntesisBox

In the text shown at "IntesisBox Communication Console" window you can see the version string for your IntesisBox: "IntesisBox_LON_KNX_V.1.0.1" in the example above.

After connection, you can also see that the icon "Status" has turned red with an exclamation mark. This means that your IntesisBox does not have a valid configuration loaded yet.

By clicking (unchecking) button "Connect" you will disconnect from IntesisBox.

If using USB connection, it is of special importance that you ensure closing the link with IntesisBox before unplugging the USB cable from the PC: unplugging it while having open IntesisBox' COM port may crash your system, obliging you to reboot.

3.2.4 Changing IP settings

Once connected to IntesisBox, either by using USB cable or Ethernet, you can change the IP address that the device is using at its Ethernet interface.

In order to do so, connect to the IntesisBox and click button "Serial" (or "TCP/IP") next to "Connect" in LinkBoxLON button bar. This will open up the connection settings with two more buttons. They provide functionality to send IP settings to the gateway or to retrieve them from it.

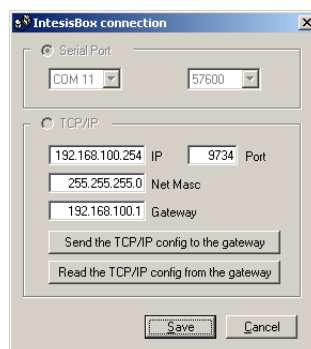


Figure 5 - IntesisBox connection settings window with TCP/IP setup buttons

At this point you can enter new IP settings for IntesisBox (fields IP, Net Mask, Default Gateway and Port) and push the button "Send..." to update IntesisBox configuration.

Changes in IP configuration will take effect next time IntesisBox is rebooted. In order to do so, click "Save" to close IntesisBox connection window and click button "Reset" on LinkBoxLON button bar. This will send a reboot command to IntesisBox, which will start the reboot process.

If connection with IntesisBox was being held using TCP/IP, it will automatically disconnect during reboot, requiring you to press the "Connect" button over again – wait a few seconds for the device to restart the IP interface.

If link was a USB connection it will not disconnect and you will be informed of the whole reboot process in "IntesisBox Communication Console". After device startup, you may want to connect to IntesisBox with the new IP settings: Change connection settings to TCP/IP by clicking "Serial" button and reconnect by unchecking and checking "Connect" button.

3.2.5 Restoring default IP settings

You can always recover the device to factory IP settings, using following procedure:

- Unpower IntesisBox by removing its power connector
- While keeping button "KNX PROGRAMMING" pressed, insert the power connector over again
- Keep the button "KNX PROGRAMMING" pressed until red LED "KNX PROGRAMMING" lights up – it will take 10 to 15s. At this point, IntesisBox will delete the current network configuration and reboot with default/factory IP settings
- Wait for the boot process to finish (LA and LB off)

This will delete IP settings. Note that current configuration in the gateway (LON and KNX interfaces and bindings between them) will not be deleted by this process.

3.3 Creating a configuration for IntesisBox

When starting LinkBoxLON, it will always first open up a project manager window, letting you choose the configuration project you want to work with.

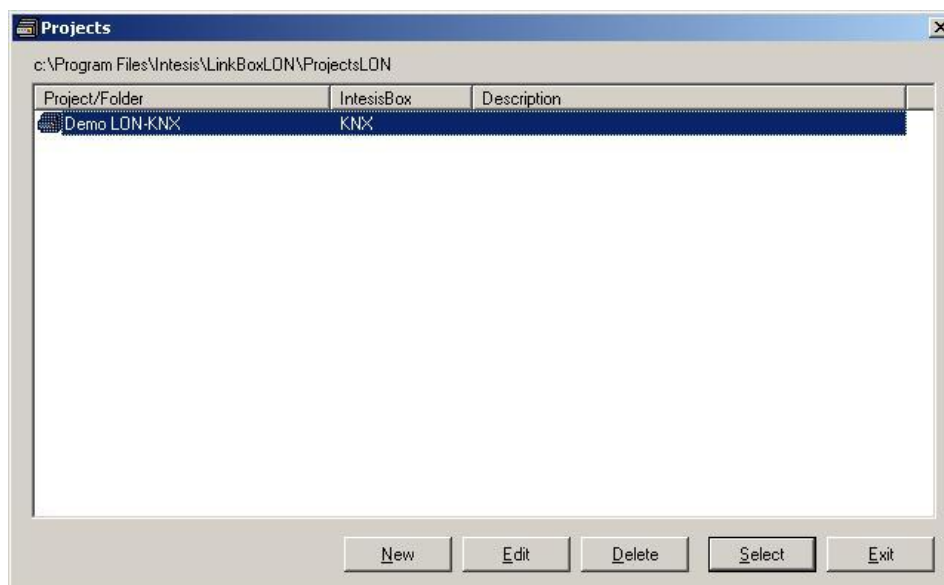


Figure 6 - LinkBoxLON startup project manager window

After installation the only available project will be "Demo LON-KNX" with a simple configuration for IntesisBox. Create a new project by clicking button "New" in the project manager. This will create a new project, performing a copy of the currently selected one ("Demo LON-KNX" in this case). You will be prompted to enter a name and description after clicking "New".

Note that each project has its own folder in "ProjectsLON" folder, inside LinkBoxLON installation directory. You can always backup your projects from there.

Once the new project has been created, choose it from the list and click "Select". This will set it as the current project.

After that, press button "Open" from the button bar to open projects configuration. You will get a window for each of the interfaces of the gateway: one for LON and one for KNX.

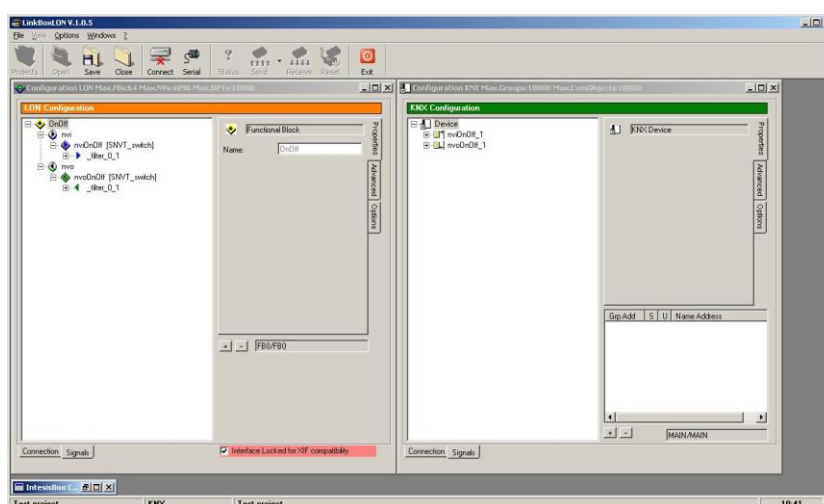


Figure 7 - Project configuration windows

Each interface –LON or KNX– can be configured independently, so they will be explained separately in next subsections.

3.3.1 LON interface configuration

Left window lets you setup all parameters for LON interface on IntesisBox.

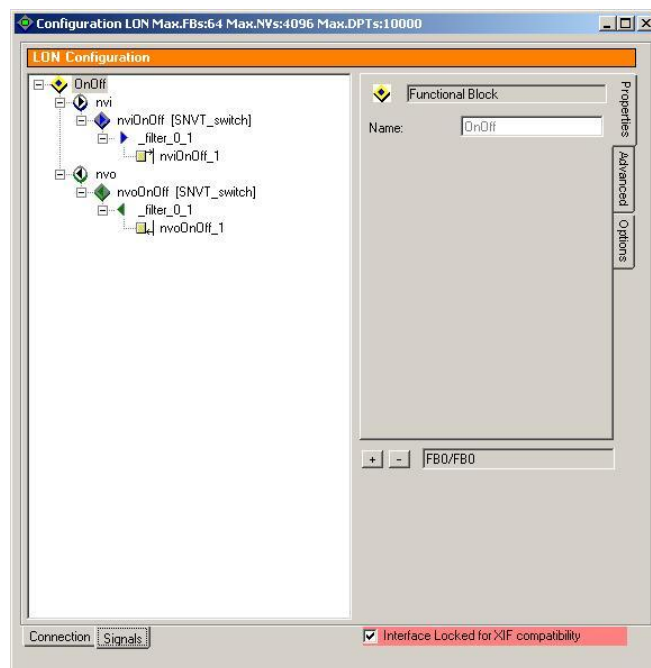


Figure 8 - LON interface configuration window

In this window, tab "Connection" shows device global parameters and tab "Signals" lets you define each of the variables in which this interface will consist.

At the end of the configuration, a XIF file will be generated for it (which you will be able to import using a network management tool – like LonMaker). Thus, most changes you do in this window will result in a different interface and a different XIF file. In order to prevent accidental changes this interface is "locked" by default, so that no changes can be performed.

The checkbox in the lower part of the window, showing "Interface Locked for XIF compatibility", lets you unlock the interface so that you can change it, losing compatibility with any previously existing XIF file.

3.3.1.1 Connection tab

By clicking connection tab you can access the global settings for the LON interface

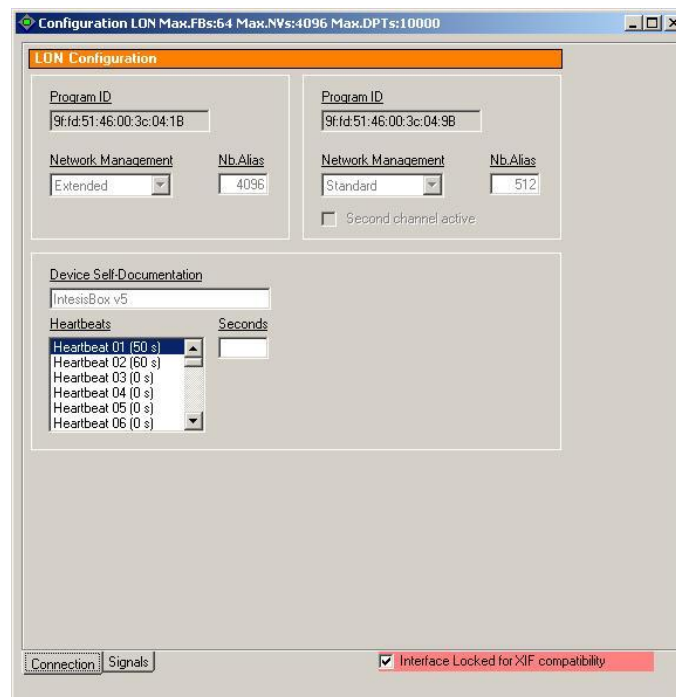


Figure 9 - Connection tab

Uncheck the mark in the "Interface Locked for XIF compatibility" label so that you can modify these parameters.

Following parameters are accessed here:

- **Program ID:** The program ID will be automatically generated by LinkBoxLON. Each time the project is saved it will be updated – in case that the LON interface (and its associated XIF file) has changed.
- **Network Management:** Lets you choose which subset of network management commands can be used to interact with the device: either standard (legacy) set of network management commands or the newer Enhanced/Extended Command Set for network management (ECS). This will depend on the kind of network management or supervisory tool you plan to use to deal with IntesisBox: For example, Echelon's LonMaker can deal with enhanced command set for network management. TAC's Vista cannot. If your tool supports it you will prefer to use the enhanced command set for network management: when not using it your device is limited to 15 address table entries, not letting you talk to more than 15 different LON devices. When using the enhanced command set you get 512 address table entries. The effect of choosing "extended" when the network management tool does not support it will be that this tool will not be able to access part of the configuration structures defined on the LON device. This will prevent it from doing several simple tasks, as could be writing a network variable.
- **Second channel active:** Only applicable for those hardware versions implementing a second LON device. Activates the second LON interface. This interface will implement exactly the same device as the first one, BUT allowing different network management option. More information on the meaning and purpose of this second channel is detailed in Appendix 6.3.
- **Alias entries:** Number of alias network variables that the network management tool will be able to define on your LON device. When using legacy network management, a big number of alias table entries will result in larger commissioning times.
- **Device self-documentation:** Self-documentation string that identifies IntesisBox in the LON network
- **Heartbeats:** It lets you define up to 50 heartbeats, to be used later in each network variable you want. In tab "signals" you will be able to associate to each network

variable one of the heartbeats in the list. This heartbeat defines the maximum period at which the network variable will be updated to the network in the case of an output network variable. For the case of an input network variable, it defines the period at which this value will be polled from its source (only has sense in the case of polled-nvi's). Value "0" in heartbeat disables it.

3.3.1.2 Signals tab

In this window you can define which network variables will implement IntesisBox as a LON device.

As in any LON device, network variables will be grouped into functional blocks. At least a single functional block containing a single network variable is needed.

Also for each network variable, you can define which fields you want to bind to the KNX side of the gateway.

Figure 10 shows the way this hierarchy is represented.

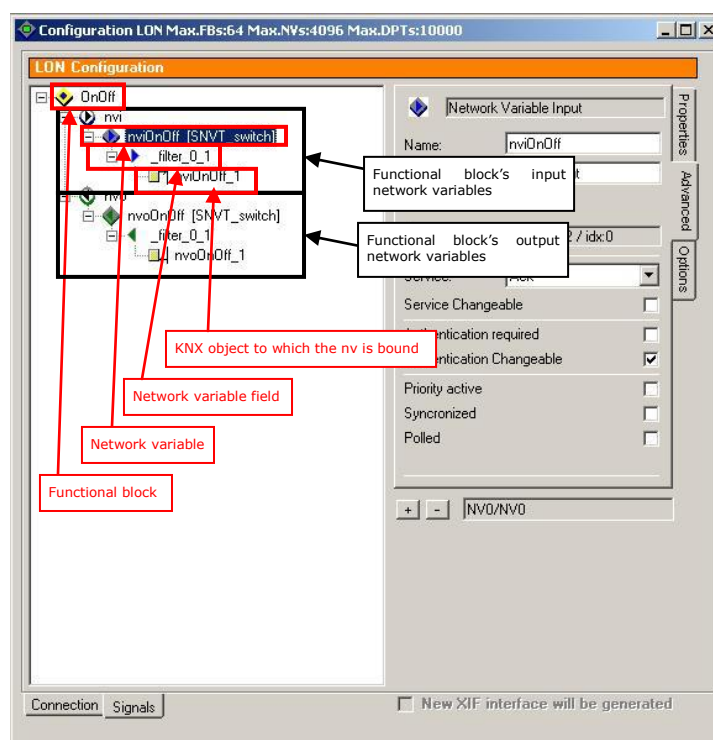








Figure 10 - Functional block hierarchy

The elements that can be managed through this window and its meaning is following:

- **Functional blocks** (FBs – represented by icon - **FB's input network variables (nvi) section** (represented by icon 

© Intesis Software S.L. - All rights reserved
This information is subject to change without notice

- **Input network variable** (represented by icon  or  / whether it is input or output): Part of a network variable (field of it) that can be bound to a KNX object. They correspond to fields as in LonMark definition's of the network variable. These fields are bound to KNX objects by doing a drag and drop of them on the KNX objects window.
- **Network variable filter or datapoint** (represented by icon  or ): Same as network variable field, with the difference that they do not correspond to LonMark's definition of the NV they apply to. They stand as fields able to perform transformations on the value of the NV they attach to, enabling a more convenient way of handling NV values when bound to KNX objects. The whole concept of filter is explained in section 3.3.1.2.1. These filters are bound to KNX objects by doing a drag and drop of them on the KNX objects window.

A new network variable is created right clicking an existing one. When doing so, following dialog will appear:

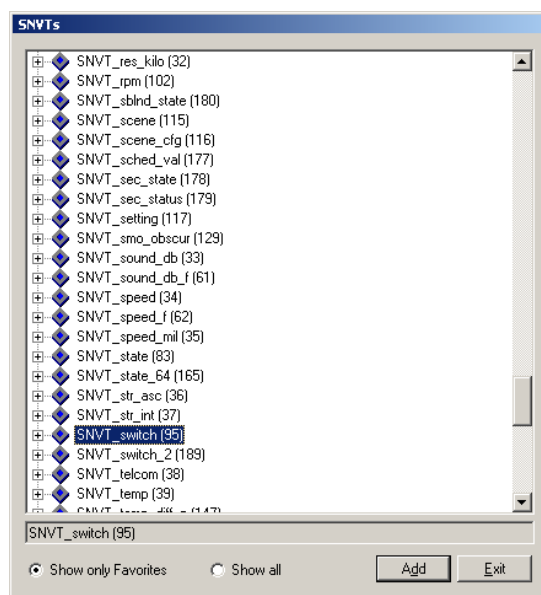


Figure 11 - Selecting SNVT for a new NV

In it you will select which SNVT you want for the new network variable.

You can setup this dialog to only show the SNVT's you are to use in the project: right-click each SNVT you don't want to see each time this dialog opens and select "Hide from Favorites". This way, "Show only favorites" (default) or "Show all" radio buttons select whether you want to see the whole set of SNVTs or discard those that are not your "favorites". At any time you can enable a given SNVT as favourite by choosing radio-button "Show all", right-clicking the SNVT and selecting "Show in favorites".

Note that each type can be expanded by a "+" sign so that you can see the fields and filters you can integrate from the new SNVT.

If you click the button "Add" while having a SNVT name selected, the new NV will be created having all available fields and filters, so that you are able to bind any of them to any KNX communication object.

Instead, if you click the button "Add" while having selected a field or filter under a SNVT name, the new NV will be created having only this filter.

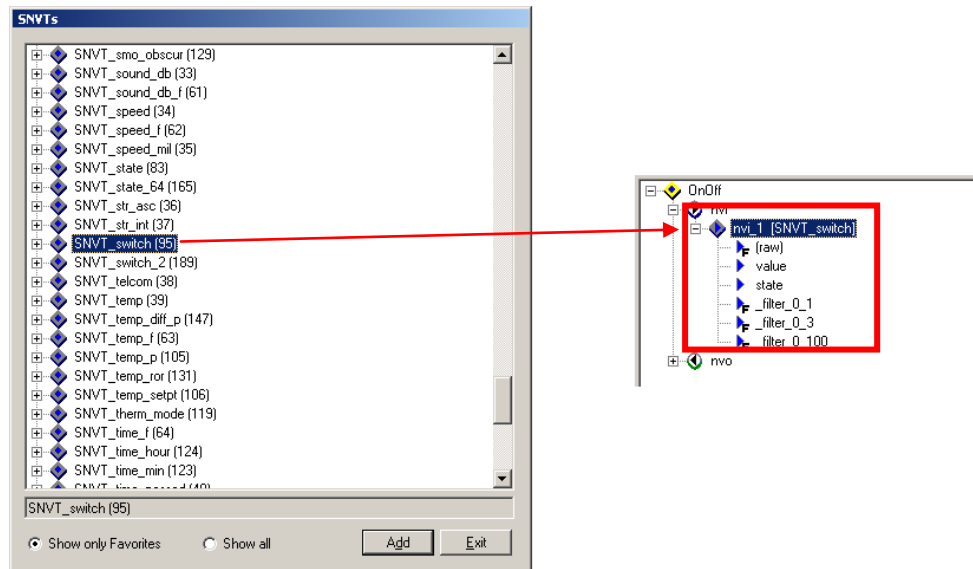


Figure 12 - Creating a new NV with all available fields/filters

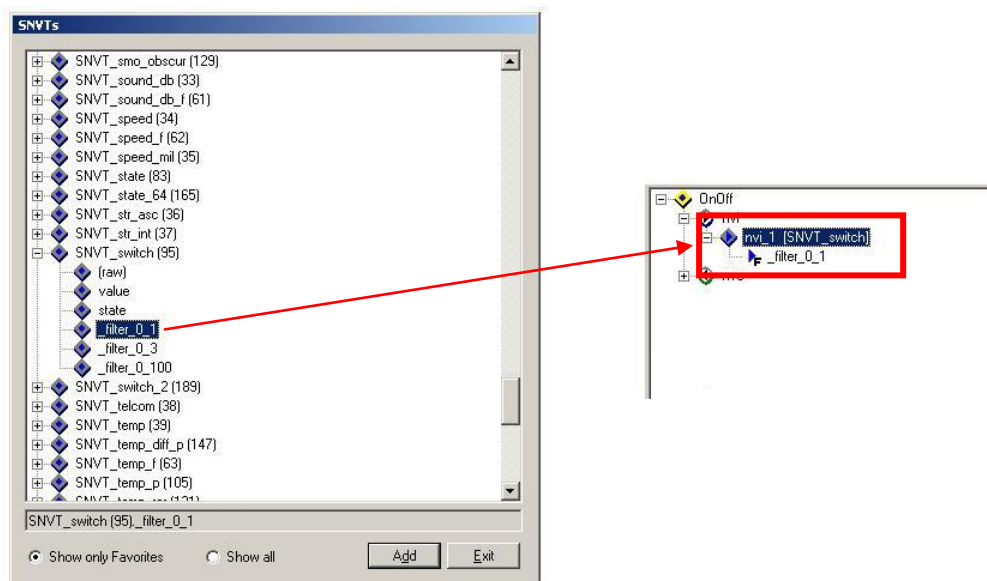


Figure 13 - Creating a new NV with a single field/filter

Each of these objects (functional blocks, network variables, datapoints) has a set of properties that will be given a default value at the time of creating them. They can be accessed and changed using tabs "Properties" and "Advanced" on the right pane of the window.

Following is a list of all of the available properties and their effect in the LON device, sorted out by object they belong to.

Property name	Default value	Description
Functional block		
Name (Properties tab)	FB_#	Name of the functional block that will be seen in network management tools (e.g. LonMaker)
Input or Output Network Variable		
Name (Properties tab)	nvi_# / nvo_#	Name of the network variable that will be seen in network management tools (e.g. LonMaker)
Self-Document (Properties tab)	(empty)	Self-documentation string for the network variable
Heartbeat (Properties tab)	(none)	Sets which heartbeat (from table in tab "Connection") will use this nv. In the case of nvi's, each time the heartbeat is elapsed the nvi will be polled. Note it has only sense in case of polled nvi's (default) In the case of nvo's, each time the heartbeat is elapsed the nvo will be propagated. Note it has only sense in case of non-polled nvo's (default)
Service (Advanced tab)	Acknowledged	Default service type that will be used when binding this NV. Possible values are: Acknowledged, Unacknowledged and Unacknowledged with repetitions
Service Changeable (Advanced tab)	Checked	Tells whether the service type that will be used when binding this NV can be changed by network management tools, or always the default should be used
Authentication required (Advanced tab)	Unchecked	When checked, this NV requires an authenticated binding
Authentication changeable (Advanced tab)	Checked	When checked, authentication flag can be changed by network management tools
Priority active (Advanced tab)	Unchecked	When checked, this network variable will use priority messages. Priority flag can be changed by network management tools in any case
Synchronous (Advanced tab)	Unchecked	In case of fast changes of this network variable, tells whether all values should be processed (synchronous) – or older unprocessed values can be discarded if newer values arrive before they are processed (non-synchronous)
Polled (Advanced tab)	Checked	Tells whether the nv is a polled. For nvi's it implies that it is able of polling the nvo it is attached to. It must also be taken into account that it consumes an address table entry. For nvo's it implies that it should not update its value to the network when it changes (instead, a polled nvi should poll it periodically)

Datapoints (or network variable field/filter)		
GW Data Type (Properties tab)	Depends on nv field/filter	Data type that will contain this object inside the gateway. The whole topic of gw data types is explained in 3.3.3.1
Synchronized (Properties tab)	Unchecked	Same as "synchronous", but applied to the datapoint itself: whether new values received in this datapoint discard unprocessed older ones. It should only be set if "Synchronous" flag for its corresponding NV has been set
Always propagate (Properties tab)	Checked	When checked, every time that this datapoint is updated (either from its corresponding nvi or an incoming communication object) its value will be propagated, without concern on whether its value has changed or not from its previous one. If not checked, the value will only be propagated if there is a change.

Once the LON interface has been planned and configured using LinkBoxLON its XIF file can be obtained: Every time the configuration is saved (button "Save" in the button bar) an XIF file for it is generated in the project folder. Going to tab "Options" in "Signals" pane of LON interface configuration, you will see a "View XIF..." button. Pressing it opens the folder in which the XIF is placed (configuration project folder) with Windows Explorer.

You can use this XIF file with your network management tool at the time of integrating IntesisBox into a LON network.

For those configurations making use of the second LON interface two XIF files will be generated, one for each interface. Naming of the XIF file is as follows:

Projectname_chX_yyy.XIF

Where chX will be ch1 or ch2 (channel 1 implements the LON interface whose service button is labelled as SERVICE1 in the IntesisBox – channel 2 implements the LON interface with label SERVICE2) – yyy will be "ext" if Enhanced/Extended Command Set for network management has been chosen for this channel, or "std" if Standard/Legacy Network Management has been used. See Appendix 6.3 for more regarding the second LON interface.

At this point, you should define its KNX interface and finally define the bindings between both. In following sections both topics are covered.

3.3.1.2.1 Network variable filters

Filters address common issues related to the translation of NV types from LON to any other protocol (in this case, KNX).

The most typical of them is the case of the *SNVT_switch*: In most LON devices and following LonMark's guidelines, basic ON/OFF operations are controlled or monitored using variable *SNVT_switch*. This SNVT has two fields, for which following table applies:

Operation	Value	State
ON	0,5% – 100%	1
OFF	0%	1
OFF	Any (0% – 100%)	0

Thus, at the time of moving this information from / to KNX clearly appears a basic problem: whereas in KNX ON/OFF operations are usually controlled by means of a single *1-bit Switching* ON/OFF object, LON imposes that two values instead of a single one are to be handled with.

In terms of the LON-KNX gateway this would mean that two communication objects would be required to support an ON/OFF operation.

In order to overcome this limitation a so-called "filter" has been implemented. In particular "_filter_0_1" allows mapping the fields "value" and "state" for the *SNVT_switch* in a single datapoint, applying following tables:

For an output nv:

<i>nviOnOff._filter_0_1</i>		<i>nviOnOff.value</i>	<i>nviOnOff.state</i>
0	→	0	0
1	→	100	1

For an input nv:

<i>nviOnOff._filter_0_1</i>		<i>nviOnOff.value</i>	<i>nviOnOff.state</i>
0	←	x	0
0	←	0	x
0	←	x	-1
1	←	x > 0	1

This way, filters are defined for certain network variables to ease the process of integrating them.

In particular, following is the list of filter implemented at the time of writing this document (software version "IntesisBox_LON_KNX_V.1.0.1").

SNVT.filter_name	Purpose
SNVT_switch._filter_0_3	Translates <i>SNVT_switch</i> (0%, 0), (33%, 1), (66%, 1) and (100%, 1) groups of (value, state) into values 0, 1, 2 and 3 respectively
SNVT_switch._filter_0_100	Translates <i>SNVT_switch</i> (0%, 0) - (x%, 1) groups of (value, state) into values x (meant to handle percentage values from a single 0-100% communication object).
SNVT_alarm.(date_time)	Concentrates fields "year", "month", "day", "hour", "minute" and "second" from within a <i>SNVT_alarm</i> into a single DATE_TIME type ¹ datapoint
SNVT_date_cal.(date_time)	Concentrates fields "hour", "minute" and "second" from within a <i>SNVT_date_time</i> into a single DATE_TIME type ¹ datapoint
SNVT_reg_val_ts.(date_time)	Concentrates fields "year", "month", "day", "hour", "minute" and "second" from within a <i>SNVT_reg_val_ts</i> into a single DATE_TIME type ¹ datapoint
SNVT_time_stamp.(date_time)	Concentrates fields "year", "month", "day", "hour", "minute" and "second" from within a <i>SNVT_time_stamp</i> into a single DATE_TIME type ¹ datapoint

¹ The whole topic of datapoint types is covered in detail in section 3.3.3, where bindings between LON and KNX datapoints is explained

Number of filters may vary in future software releases.

3.3.2 KNX interface configuration

Window on the right side relates to KNX interface and lets you define the interface of IntesisBox as KNX device.

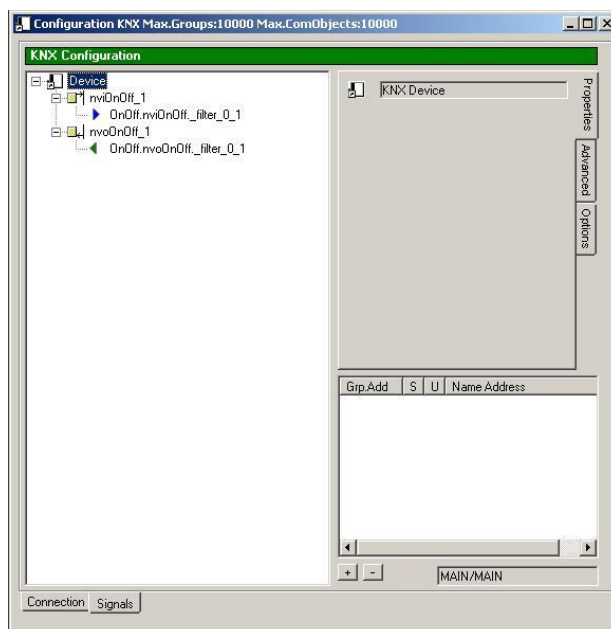


Figure 14 - KNX interface configuration window

As in LON side, tab "Connection" relates to global parameters of the KNX device implemented by IntesisBox and tab "Signals" lets you define the Communication Object interface it will show to the bus.

3.3.2.1 Connection tab

By clicking "Connection" tab you will access following configuration parameters:

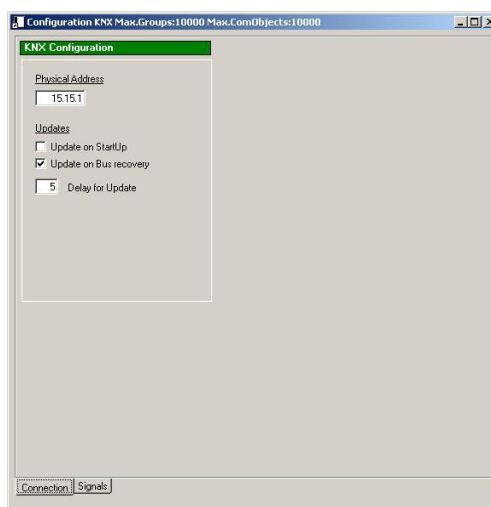


Figure 15 - Connection tab

- **Physical Address:** Physical address of IntesisBox as a KNX device
- **Update on startup:** If checked, IntesisBox will trigger READs for all Communication Objects for which an update address has been set, so to get an initial value for them.
- **Update on bus recovery:** Same as "update on startup", but to be applied every time that the bus is recovered. Note that if "update on bus recovery" is enabled, in particular it will also update values on startup (first time bus is detected as connected)
- **Delay for updates:** Applicable only when "Update on startup" or "Update on bus recovery" have been set. It represents the number of seconds for which, after startup or bus recovery, READs will be triggered. This is meant to reduce as much as possible traffic load in the KNX network when bus is recovered – in case that there is other KNX devices that also may request READs at startup.

3.3.2.2 Signals tab

This window is used to define the Communication Objects that will be implemented in the KNX interface of IntesisBox.

It is divided in three areas, shown in the figure below:

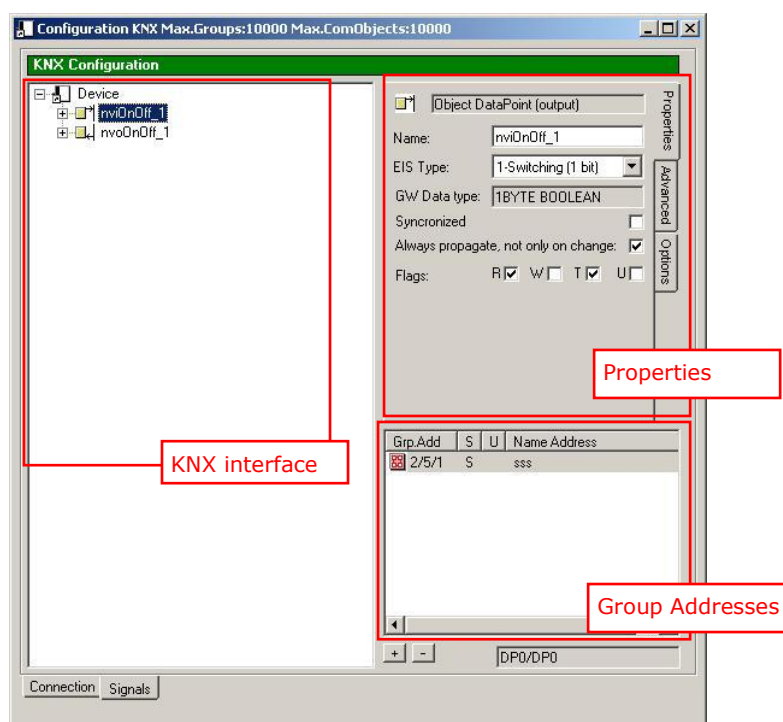


Figure 16 - Signals tab window

In the **KNX interface** area Communication Objects (COs) can be defined and selected. **Properties** area shows the properties of actual selected Communication Object. **Group Addresses** area is used to both see and change Group Addresses (GAs) associations of actual selected Communication Object.

Communication Objects can be created at KNX interface area by right-clicking an existing object – the new object will be created as a copy of the existing one.

By right-clicking they can also be deleted - at least one Communication Object needs to be defined at KNX interface, so it will not let you delete the last one.

Following is a list of all properties (in **Properties** area) applicable to a single Communication Object:

Property name	Default value	Description
Name (Properties tab)	Object_#	Name of the communication object
EIS type (Properties tab)	-	EIS type / datapoint implemented by the CO. Available datapoint types are: <ul style="list-style-type: none"> • EIS-1 Switching (1bit) • EIS-2 Dimming (4bit) • EIS-3 Time (4bytes) • EIS-4 Date (4bytes) • EIS-5 Float (2bytes) • EIS-6 Scaling (1byte) • EIS-7 Drive (1bit) • EIS-8 Priority (2bit) • EIS-9 IEEE float (4byte) • EIS-10 Counter (2byte) • EIS-11 Counter (4byte) • EIS-13 ASCII char (1byte) • EIS-14 Counter (1byte) • EIS-15 String (14byte)
GW Data type (Properties tab)	Depends on EIS type	Data type that will contain this object inside the gateway. The whole topic of gw data types is explained along with bindings in section 3.3.3.1
Synchronized (Properties tab)	Unchecked	Tells whether new values received in this CO discard unprocessed older ones.
Always propagate (Properties tab)	Checked	When checked, every time that this CO is updated (either from KNX or an incoming value from an nvi) its value will be propagated, without concern on whether its value has changed or not from its previous one. If not checked, the value will only be propagated if there is a change.
Flag R (Properties tab)	-	The value of this object can be read via the bus if this flag is set ¹
Flag W (Properties tab)	-	The value of the object can be modified via the bus if this flag is set ¹ . When setting this flag, Communication Object's icon will show up a small incoming arrow (📡) representing that this value is meant to be written from bus.
Flag T (Properties tab)	-	A telegram will be transmitted when the object value has been modified if this flag is set ¹ . When setting this flag, Communication Object's icon will show up a small outgoing arrow (📡) representing that this value is meant to be written from bus.
Flag U (Properties tab)	-	Value response telegrams are interpreted as a write command and hence the value of the group object will be updated if this flag is set ¹

¹ RWTU flags have the same functionality as in typical KNX network management tools.

Back propagate (Advanced tab)	Checked	If checked will behave as follows: When this object is updated from LON interface it will send its value back to LON (additionally to update it towards KNX). This functionality is explained along with bindings in section 3.3.3
---	---------	--

Special care needs to be taken when choosing values for CO's flags: They will define whether it is an input, output or input/output communication object. This needs to be consistent with the internal binding in the gateway of the communication object (i.e., whether it is bound to an nvi, an nvo or both at LON side).

Finally, associations with one or more group addresses can be done with each communication object. Next section covers the topic on group addresses management and association.

3.3.2.3 Group Addresses

Group addresses (GAs) hierarchy is managed in a separate window.

Right-click **group addresses** area in Signals tab and select "Group Addresses...". This will open up following window:

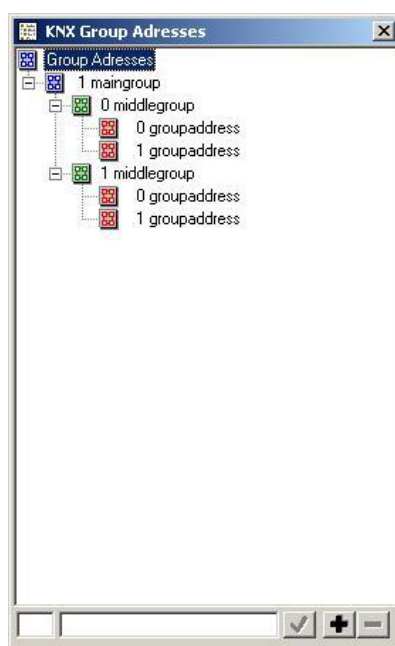


Figure 17 - Group Addresses management window

In it you will define the group addresses to be used in your project. They have format "main group/middle group/group address".

Create a new main group, middle group or group address in following way:

- Select the parent object for which you want to create a child (label "Group Address" for creating a middle group, a main group for creating a middle group and a middle group for creating a group address)
- Enter the number for the main group, middle group or group address in the lower left textbox
- Enter the name for this object in the textbox in the lower middle part of the window
- Press button "+"

Note that the object will be added in the hierarchy and the number in the lower left textbox will be increased. This aids the process of creating several correlative objects, by successively pressing button "+".

You can delete any object in the window by selecting it and pressing button "-". Also, selecting an object you can edit its current name in the textbox below: Use the validation button once you have changed its name.

Addresses can also be imported from a .csv file in the format typically generated by ETS tool. This process is covered in section 3.3.2.4.

Once you have created suitable addresses for your project you can associate them to your communication objects. To do so follow these steps:

1. Select the communication object on which you want to associate group addresses in KNX interface window.
2. Select the group address you want to associate to this object, then drag and drop it to Group Address pane in KNX configuration window

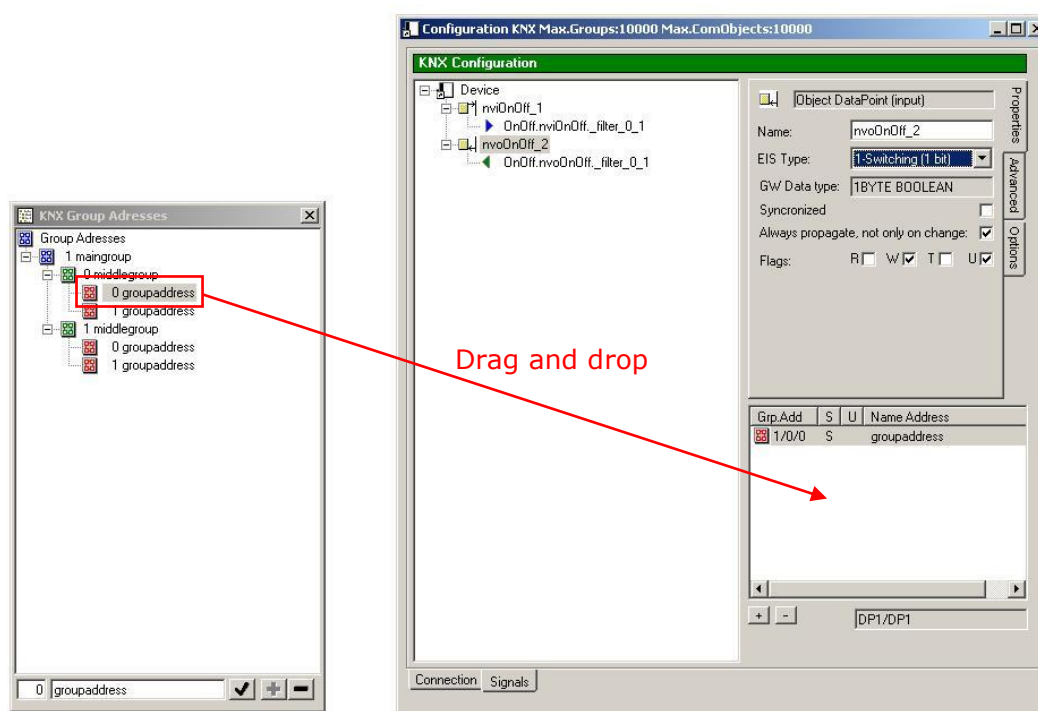


Figure 18 - Adding Group Addresses to a Communication Object

Note that you can add more than a single address to a single communication object.

For each address in an object you can set up which one of them is the **Sending Address** and which one is the **Update Address**.

The **Sending Address** is the group address through which updates of its associated communication object will be sent – provided that its "T" flag is set.

The **Update Address** is the group address on which the value will be requested (READ frame) at bus recovery / startup – provided that its "U" flag is set.

Right-click group addresses in its pane of the KNX interface window to set one of them as Sending Address or Update Address.

3.3.2.4 Importing Group Addresses and ESF files

Additionally to manually creating group addresses and communication objects, LinkBoxLON allows you to import group addresses' data of an existing project from following file formats:

- ETS CSV file containing group addresses (you need to export them in format 1/1 TAB separated, in ETS)
- ETS ESF file (typically exported to OPC servers)

Group addresses importer can be open by going to tab "Options" in KNX configuration window. By pressing button "Import Group Address..." following dialog will open:

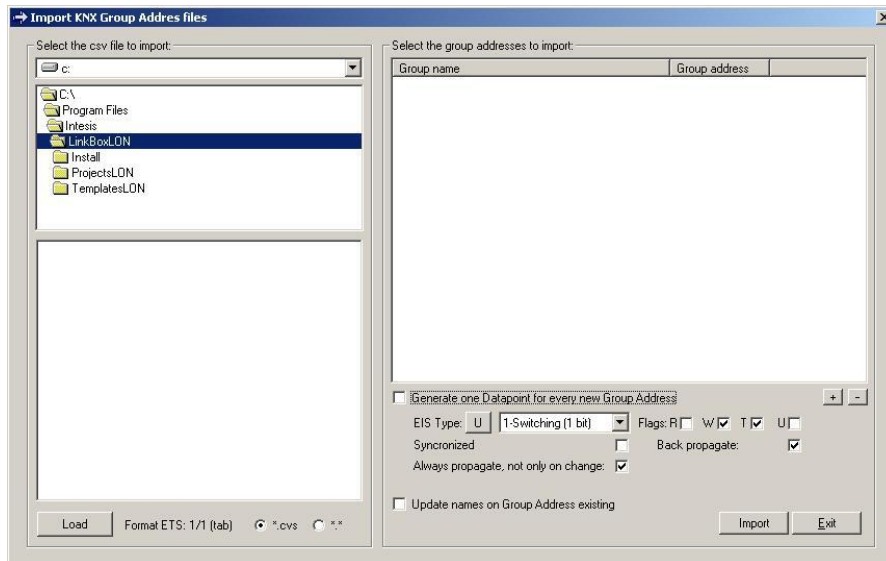


Figure 19 - Group Addresses importer tool

In it you can browse to the location of your source .csv file containing the addresses you want to import.

Double-clicking the file or by pressing the "Load" button will load all group addresses found in the file.

From the list, you can either individually select the addresses to import, or make a selection of multiple items with regular Ctrl+click / Shift+click operations.

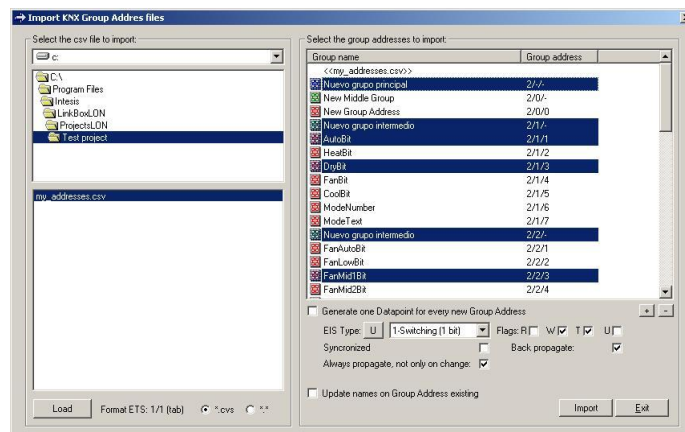


Figure 20 - Selecting which group addresses to import

Pressing the import button will import them, without closing the import dialog.

Note that the dialog offers you the option to generate a single datapoint (Communication Object) for each imported group address – in the same dialog the parameters for those new communication objects can be set.

Check “Update names on existing Group Addresses” if you want to discard old (existing) names when collision of existing and new group addresses in the project occur.

Regarding ESF file importer, it works in the same fashion as the Group Addresses importer. The only relevant difference between is that ESF file contains information of several EIS types for the imported addresses.

3.3.3 Creating bindings between LON and KNX

Having both “LON Configuration” and “KNX Configuration” windows open and in its respective “Signals” tab, you can bind datapoints from both sides by doing a drag and drop operation, from one window to the other:

- Drag a LON datapoint (NV's filter or field) and drop it onto a KNX datapoint (Communication Object)
- Drag a KNX datapoint (Communication Object) and drop it onto a LON datapoint (NV's filter or field)

The effect of either operation will be the same.

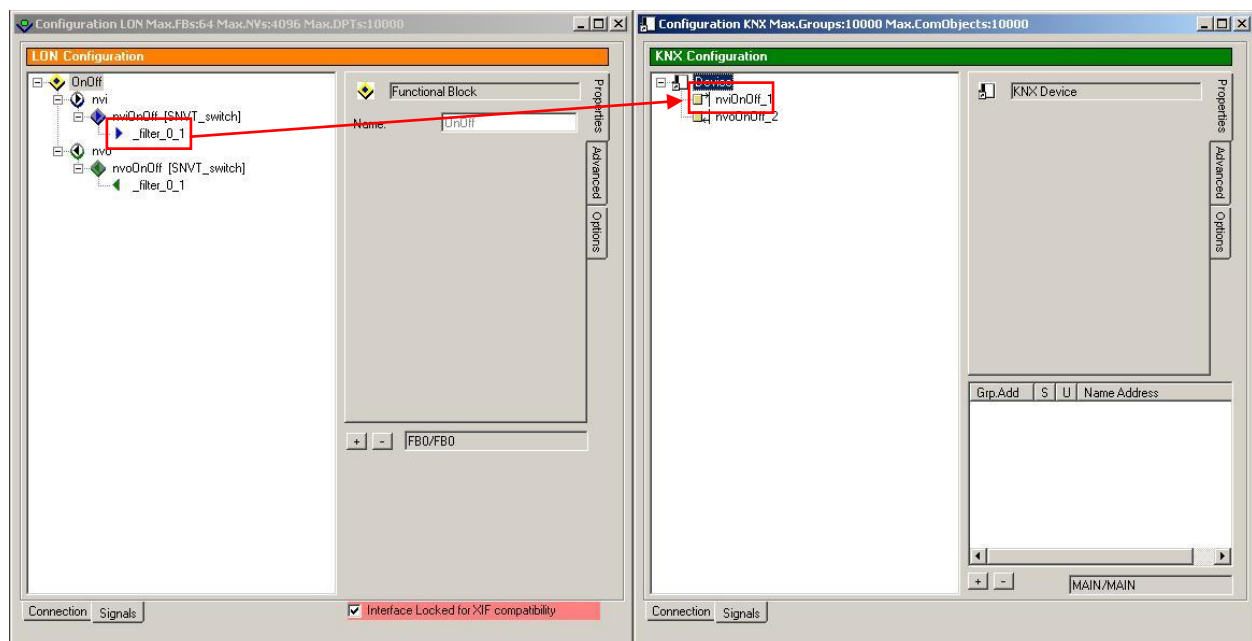


Figure 21 - Drag and drop to do a binding between LON and KNX datapoints

After the binding has been done both LON Configuration and KNX Configuration windows will show that they are bound by adding the Communication Object or the NV field/filter as a child of the datapoint of the opposite protocol they are bound to, as shown in figure below.

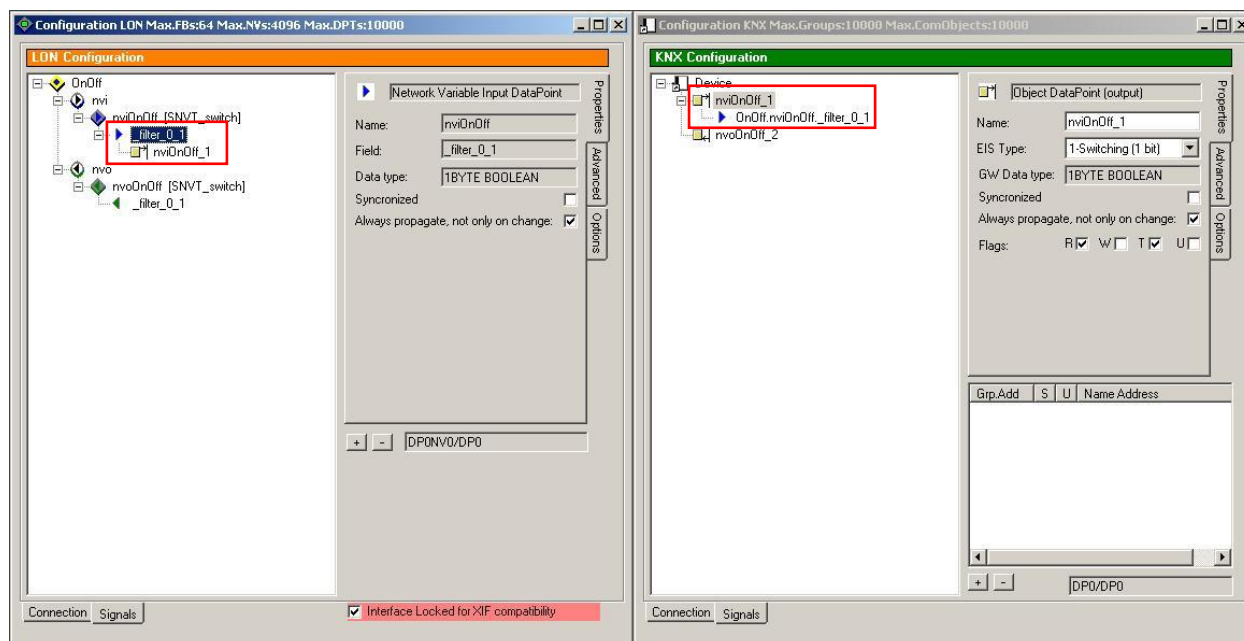


Figure 22 - Representation of bound datapoints at both sides

There is no limitation in the number of bindings between datapoints: Any single datapoint at LON/KNX side can be bound to any number of datapoints at KNX/LON side. This means that, for instance, you can bind an input NV to, say, different 10 communication objects that will be updated to the bus (by its respective Sending GA) when the nvi value is updated – if their “T” flag is appropriately set. Likewise, you could bind 10 communication objects – with its “W” flag being set – to a single nvo – so that this nvo value is updated when any of the 10 communication objects is received on the bus (an alternative to this last configuration could be declaring a single communication object with 10 associated group addresses).

Additionally, you can bind a single communication object to both an nvi and an nvo (or a multiplicity of them) at the same time, so that the value of this object can be updated from both KNX and LON: Whenever the KNX object is written the nvo will be updated – and whenever the nvi is written the KNX object will also be updated. In this case, you may want the property “**Backpropagation**” for the KNX object to be **enabled**. This will setup that, once the nvi value is updated from the LON bus –and consequently sent to KNX as well– you also want this value to be sent *back* to the LON side of the gateway, so to update the nvo the KNX object is bound to. If **backpropagation** is not enabled for the KNX object in this binding, the nvo will only be updated whenever the value at KNX side is updated (i.e. there is no *turning-back* of values within the gateway for this binding).

Figure below depicts this process for both cases (backpropagation disabled and enabled for the KNX object).

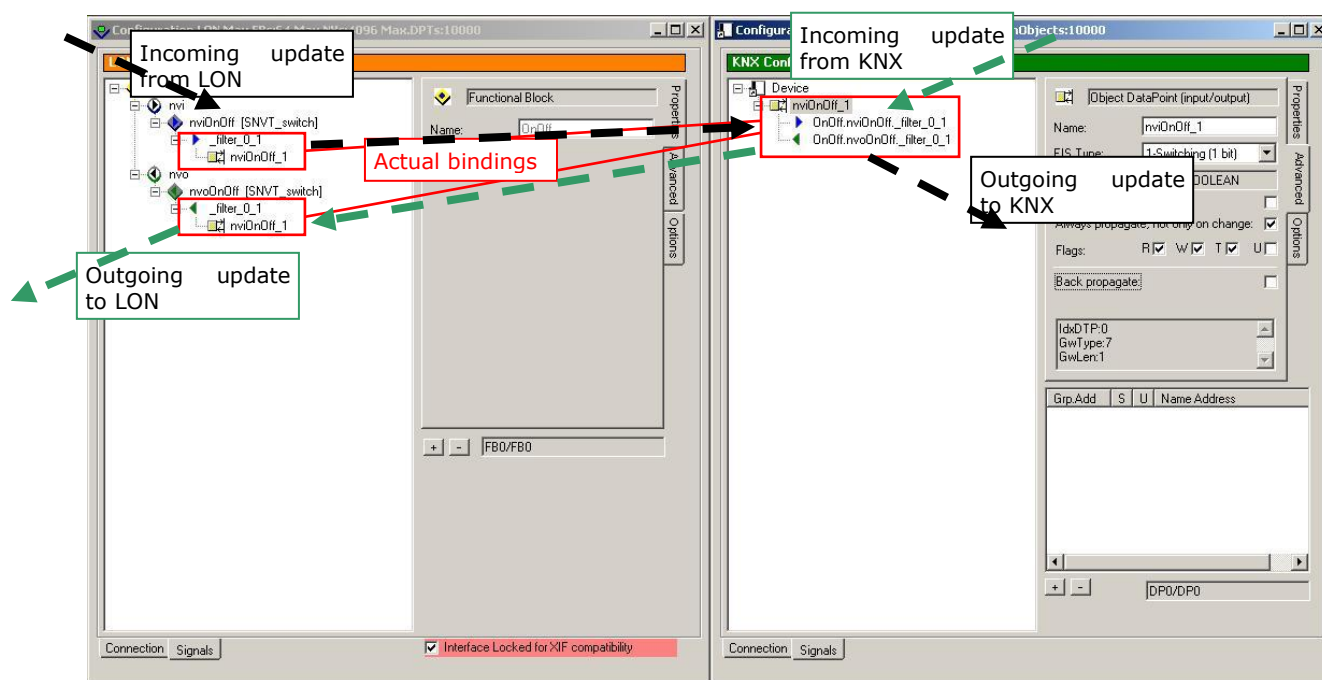


Figure 23 - Binding an nvi and an nvo to the same comm. object - no backpropagation

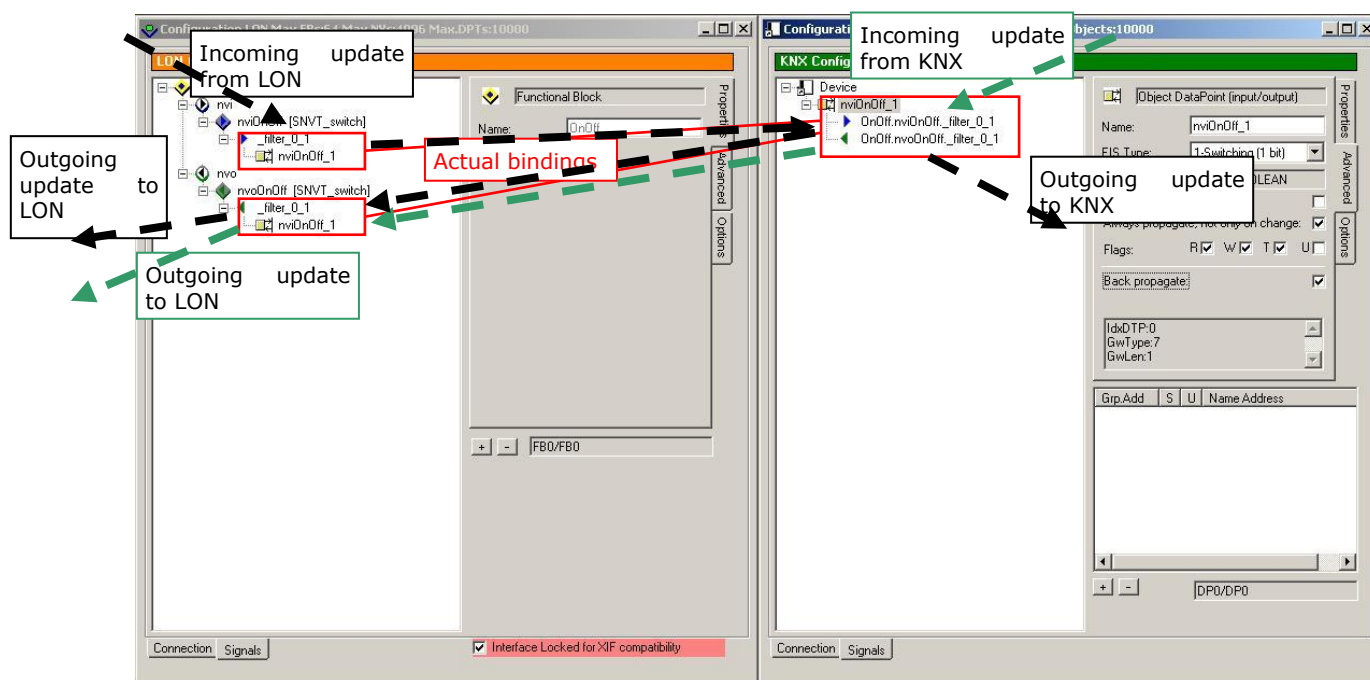


Figure 24 - Binding an nvi and an nvo to the same comm. object - backpropagation

By default, all communication objects are created having its property "**Backpropagation**" as enabled – so behaviour depicted in Figure 24 will be the default one.

Other important considerations to be taken into account when doing bindings are following:

- You need to setup KNX flags accordingly with its bound LON datapoint: Typically, you will want at least "R" flag to be enabled if the datapoint is bound to a field/filter from an nvi / at least the "W" flag enabled if the datapoint is bound to a field/filter from an nvo: The most common configuration is having those communication objects bound to nvi field/filters with flags "R" and "T" enabled – and those communication objects bound to nvo field/filters with flags "W" and "U" enabled.

- Appropriately set property “**always propagate**” at each datapoint: If it is enabled for both the incoming LON/KNX and outgoing KNX/LON datapoint bound to it, whenever value is received at LON/KNX it will be updated/triggered to KNX/LON – regardless of being the same value as the previous one. If only changes are to be transmitted by the gateway disable **always propagate** property for the applying datapoints. This property is **enabled** by default.
- Values of NV fields at LON side are always interpreted at **application layer**. This means that values propagated within the gateway are those resulting of applying a, b and c scaling factors – as per LonMark’s definition of them: In short, for a *SNVT_switch.value* field, the value propagated in the gateway is a 0-100% value and NOT the 0-200 value that is transmitted in the bus.
- Datapoint types must match **minimally**. This means that you cannot bind, for instance, numeric formats to strings, or date-time datapoints to numeric formats. Data type considerations are important and are detailed in next section.

3.3.3.1 Gateway data type and bindability

Depending on the type and ranges of value they contain, each datapoint for both KNX and LON falls into following types or categories:

- Numeric – 1 byte signed
- Numeric – 1 byte unsigned
- Numeric – 2 byte signed
- Numeric – 2 byte unsigned
- Numeric – 4 byte signed
- Numeric – 4 byte unsigned
- Numeric – Boolean
- Numeric – 4 byte IEEE float
- String
- Date-time

At LON side the datapoint type goes intrinsically defined by the field it implements: For instance, following types are hard-coded for *SNVT_switch* possible fields and filters:

- *SNVT_switch.value*: 1 byte unsigned
- *SNVT_switch.state*: 1 byte signed
- *SNVT_switch._filter_0_1*: Boolean
- *SNVT_switch._filter_0_3*: 1 byte unsigned
- *SNVT_switch._filter_0_100*: 1 byte unsigned

At KNX side the datapoint type depends on the EIS type implemented by the KNX object, in the following manner:

- EIS-1 Switching (1bit): Numeric – Boolean
- EIS-2 Dimming (4bit): Numeric – 1 byte unsigned
- EIS-3 Time (4byte): Date-time
- EIS-4 Date (4byte): Date-time
- EIS-5 Float (2byte): Numeric – 4 byte IEEE float
- EIS-6 Scaling (1byte): Numeric – 1 byte unsigned
 - Values for this object inside the gateway are translated into the range 0-100% - being 0 to 255 the bus values
- EIS-7 Drive (1bit): Numeric – Boolean
- EIS-8 Priority (2bit): Numeric – 1 byte unsigned
- EIS-9 Float (4byte): Numeric – 4 byte float

- EIS-10 Counter (2byte): Numeric – 2 byte signed / unsigned¹
- EIS-11 Counter (4byte): Numeric – 4 byte signed / unsigned¹
- EIS-13 ASCII char (1byte): Numeric – 1 byte unsigned
- EIS-14 Counter (1byte): Numeric – 1 byte signed / unsigned¹
- EIS-15 String (14byte): String

In any case, LON or KNX, datapoint type can be seen at **Properties** pane for each datapoint, in its respective configuration window.

According to this list of data types, following bindings between datapoint types are possible:

- **Date-time** type datapoints can only be bound with datapoints of the same type
- **String** type datapoints can only be bound with datapoints of the same type:
 - Size will be truncated in case of mismatch (e.g. SNVT_str_asc LON datapoint, with 31 bytes in size, can be bound to EIS 15 of KNX – but size will be truncated to 14 bytes when transmitted to KNX).
- Any size **Numeric** type datapoints can be bound between them
 - When moving a datapoint from higher resolution (e.g. 4byte IEEE float) to lower resolution (e.g. 1byte signed) information will be lost (for the example case, the only data that would successfully be transferred would be input values of -128 to 127, i.e. those that fit in the targeted data type at the other side of the gateway). Thus, binding a datapoint in the direction from higher resolution to lower resolution is NOT recommended, unless it is guaranteed that the input datapoint will never take values able to overflow targeted datapoint at the other side of the gateway.
 - Binding a datapoint in the direction “lower resolution” to “higher resolution” poses no problem.
 - Sign considerations apply: It is not recommended to bind signed type datapoints to unsigned types, unless it is guaranteed that the source will never make use of negative out-of-target's-range values.

3.4 Sending the configuration to the gateway

Once one configuration has been created and saved it can be sent to IntesisBox by pressing “Send” button on LinkBoxLON's button bar.



Figure 25 - Button bar / "Send" button

This will trigger following operations:

- Download a set of files containing the configuration data in ASCII format to IntesisBox
- Reboot IntesisBox

Since IntesisBox is rebooted, if its connection link is being held through Ethernet/TCP, it will loose connection – wait a few seconds for the device to restart the TCP/IP interface and reconnect.

¹ Either signed or unsigned can be chosen as a property of the EIS object in KNX Configuration window (button “U” in front of label “EIS Type”). KNX definition of this datapoint type approves both a unsigned or a signed implementation, provided that the definition is consistent within the KNX network.

Once the configuration files are sent and IntesisBox has rebooted, button bar's status icon should change to a green happy face, telling that the gateway has properly restarted with a valid configuration.

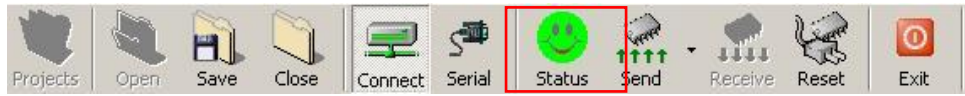


Figure 26 - Status after successful configuration of the gateway

At this point, your device is ready to be tested into the LON & KNX networks.

At LON side you will need to commission the device with a network management tool so to be able to operate with its interface. Once IntesisBox is configured and commissioned in its LON side, if you download a new configuration which implements a new LON interface you will need to commission it over again (i.e., last commissioning information will be dropped, since it applied to an old external interface) – Checkbox "Interface Locked for XIF compatibility" at LON Configuration window prevents you from doing any change in LON's configuration, so to avoid accidental changes.

Remember disconnecting from IntesisBox by pressing button "Connect" before unplugging the USB cable.

3.5 Firmware updates

New releases of firmware may be downloaded on the device using LinkBoxLON.

Prior to firmware download, you need to be connected to IntesisBox either using the USB or Ethernet cables.

Then you can download your firmware image by going to "Options->Firmware update...". Then you need to select the file to download in your filesystem. Typically the firmware image name will have format "imagelonknx_vX_Y_Z.firm".

Once the firmware file has been selected following dialog will appear:

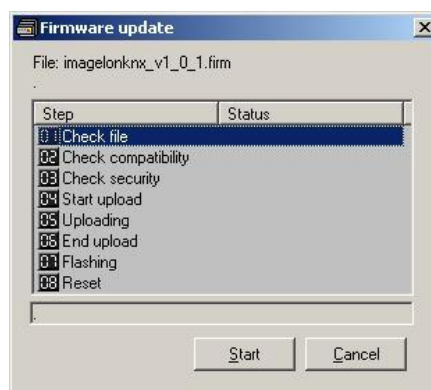


Figure 27 - Firmware update dialog

Press the start button to proceed with the firmware download. The process may take up to several minutes (about 2-3 minutes if connected using Ethernet, 5-6 if connected through USB).

After successful firmware download, dialog will show up as follows:

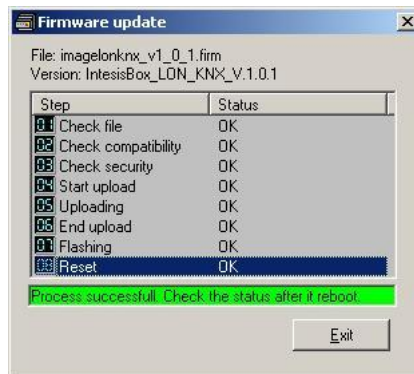


Figure 28 - Firmware update dialog - after fw has been downloaded

Click "Exit" to leave this dialog. If connection was being held through Ethernet you will have to reconnect by pressing "Connect" button (IP settings may have been reset to factory defaults – this is IP address 192.168.150.254).

After reboot you may find that "Status" icon on the button bar goes back to show the exclamation mark in red, telling that the gateway has no valid configuration. In either case ("Status" icon showing an exclamation mark in red or a happy face in green) you need to download IntesisBox's configuration over again after a successful firmware download (press "Send" button).

Firmware update process will fail if the selected file was wrong – in this case you will need to repeat the process with a correct file.

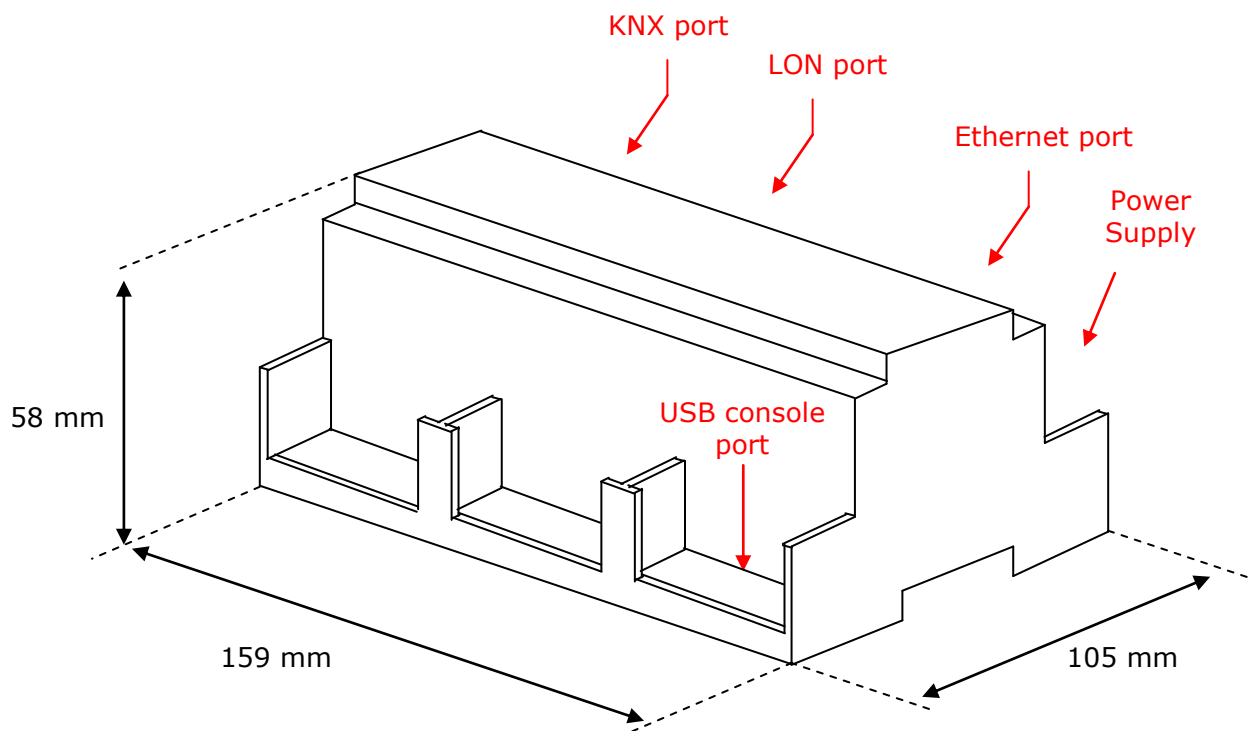
4. Mechanical & electrical characteristics

Enclosure	Plastic, type PC (UL 94 V-0). Dimensions: 159mm x 105mm x 58mm.
Color	Grey. RAL 7035.
Power	De 9 a 30Vcc +/-10%, Máx.: 210mA 24Vca +/-10% 50-60Hz, Máx.: 212mA Must use a NEC Class 2 or Limited Power Source (LPS) and SELV rated power supply. Plug-in screw terminal block (2 poles) for 12 AWG cable.
Terminal wiring (for power supply and low-voltage signals)	Per terminal: solid wires or stranded wires (twisted or with ferrule) 1 core: 0.5mm ² ... 2.5mm ² 2 cores: 0.5mm ² ... 1.5mm ² 3 cores: not permitted
Mounting	Wall DIN rail EN60715 TH35.
LON Port	1 x LON TP/FT-10. Plug-in screw terminal block (2 poles) for 12 AWG cable. TNV-1
KNX Port	1 x KNX TP1 (EIB) opto isolated. Plug-in screw terminal block (2 poles) for 12 AWG cable. TNV-1
LED indicators	1 x Power. 2 x KNX port activity (Tx, Rx). 2 x Gateway status: LA (LON status), LB (KNX status) 1 x KNX programming. ¹ 2 x LON Service LED.
Switches	1 x KNX programming. ¹ 2 x Service LON.
Console port	USB. Type B female standard connector. SELV
Ethernet port	1 x Ethernet 10/100Base-T connector RJ45.
Configuration	Via USB console port and/or Ethernet interface ²
Firmware	Allows upgrades via USB console port and/or Ethernet interface
Operational temperature	0°C to +60°C
Operational humidity	10% to 90%, non condensation
Protection	IP20 (IEC60529).
RoHS conformity	Compliant with RoHS directive (2002/95/CE).
Norms and standards	CE conformity to EMC directive (2004/108/EC) and Low-voltage directive (2006/95/EC) EN 61000-6-2 EN 61000-6-3 EN 60950-1 EN 50491-3

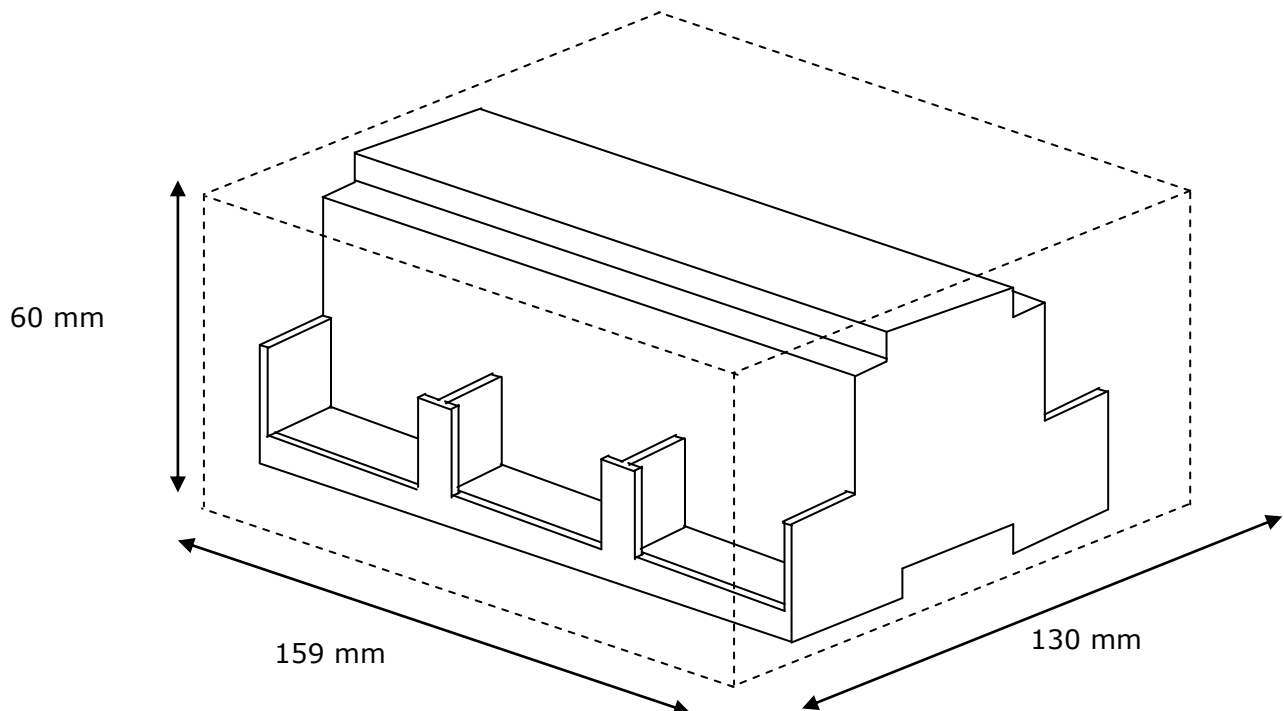
¹ Not operational for the moment. Reserved for future use.

² Standard cable USB type A-B of 1,8 meters length is supplied with the device for connection to a PC USB port for configuring and monitoring the device. The configuration software, compatible with Windows® operating systems, is also supplied.

5. Dimensions



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



6. Annexes

6.1 Exporting group addresses from an ETS project in a CSV file

Using ETS 3.0 you can export the set of group addresses used in a project, so that you are able to import them in LinkBoxLON.

Start with opening ETS 3.0 and opening the project whose group addresses you are to export.

Firstly, ensure that you see the "Group Addresses" window for your project. If it is not open, you can open it by going to menu option "View->Project Views->Group Addresses".

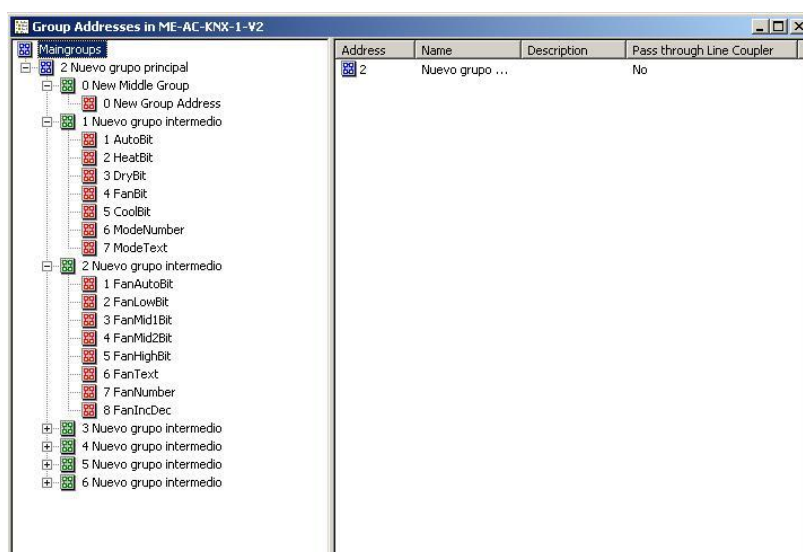


Figure 29 - Group Addresses window for an ETS project

In it, right-click the point in the hierarchy from which you want to export addresses (i.e. only those addresses that are children of your selected object will be exported). In the contextual menu for this object select "Import Group Addresses...".

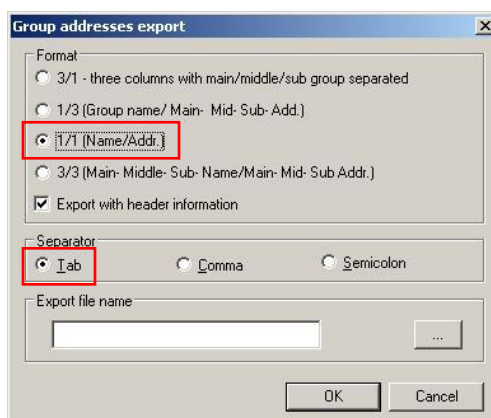


Figure 30 - Group addresses export dialog

Then, in the "Group Addresses Export" dialog that appears, you should select radio buttons for Format: "1/1 (Name/Addr)" and Separator: "Tab". Finally, enter the filename for the resulting file in the textbox below (or use its browse "...") and click OK.

6.2 Exporting group addresses from an ETS project in an ESF file

You can also import in LinkBoxLON your ETS 3.0 project in ESF format. ESF file is designed to be used by the former KNX OPC server from the KNX association.

Start with opening ETS 3.0 and opening the project for which the ESF file is to be generated.

Choose menu option "File->Extract Data...".

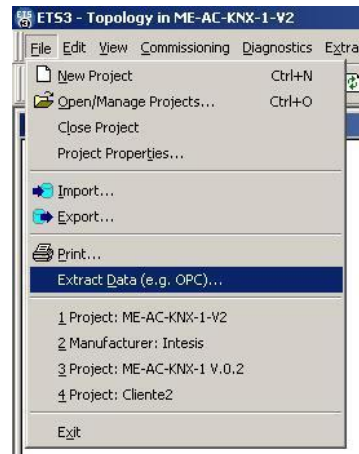


Figure 31 - "Extract Data" menu option in ETS 3.0

Then click "Export..." in "Export to OPC Server" pane in the dialog below.

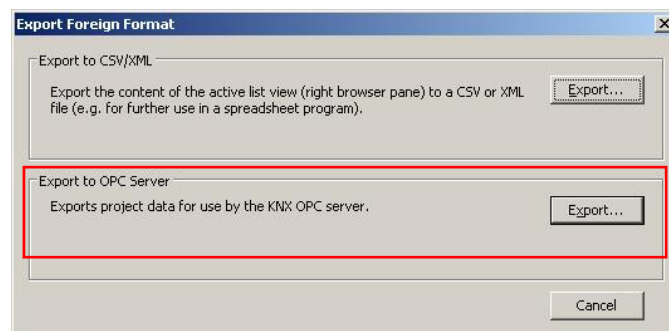


Figure 32 - Export Foreign Format dialog

This will open a "Save As..." dialog in which you can point to the location where you want the ESF file, which will be generated by just clicking "Save" on it.

6.3 Using the secondary LON interface

As mentioned throughout the document, IntesisBox allows choosing between using Legacy Network Management or Enhanced Command Set-based Network Management in its LON interface.

If possible, it is also preferable to work with Enhanced Command Set (ECS), since it increases the number of available address entries (512 entries for IntesisBox) – i.e., the number of devices that IntesisBox can talk to.

Though, some network management tools may not support communicating with LON devices using the enhanced command set, thus forcing the configuration of Legacy Network Management. In this case, the number of address table entries is limited to 15 devices, imposed by previous CEA709 standard definitions, which is rather poor.

In order to address this limitation, hardware versions implementing a second LON interface are also available for IntesisBox, which let you choose different network management options for each one of them, while they implement the same network variable interface. This is done in such a way that, when an NV value is updated on one of the interfaces, it is also updated in the other one –and viceversa– i.e. the two interfaces are implementing *two versions* of the same device - having two different neuron-id's.

This allows you to do configurations like the following:

- One of the interfaces set to "Standard/Legacy Network Management" – meant to be supervised by a tool that does not support enhanced/extended NM.
- The other interface set to "Extended Network Management" – allowing you to use the 512 address table entries specified for IntesisBox.

Both interfaces are available on the same physical channel – accessed at the 2-pole terminal connector for LON. Though, they do have different service button and LED, marked with SERVICE1 and SERVICE2 on IntesisBox PCB.

Each interface has its own XIF file (generated when saving the configuration using LinkBoxLON) – you can distinguish them by their name: "ch1" for channel 1, "ch2" for channel 2. Keywords "ext" or "std" also tell you about the type of network management implemented in the channel the XIF applies to.

Note that, at the time of integrating the device into a LON network the device will need to be commissioned twice: each interface needs to be commissioned separately, as if each one was a single device.

Also you will need a tool supporting Enhanced Command Set, such as LonMaker, in order to be able to commission an interface implementing it.