ESSER
by Honeywell

IBOX Modbus Server

ESSER 8xxx/IQ8/FlexES fire panels

Installation Manual

r1 eng

11.2013

ES-MN-18002-01-2

Gateway for the integration of ESSER 8xxx/IQ8/FlexES fire panel series in Modbus enabled monitoring and control systems.

Order code:

IBOX-MBS-EDP-3: Up to 3000 points

IBOX-MBS-EDP-6: Up to 6000 points

IBOX-MBS-EDP-9: Up to 9000 points

INDEX

1 Description

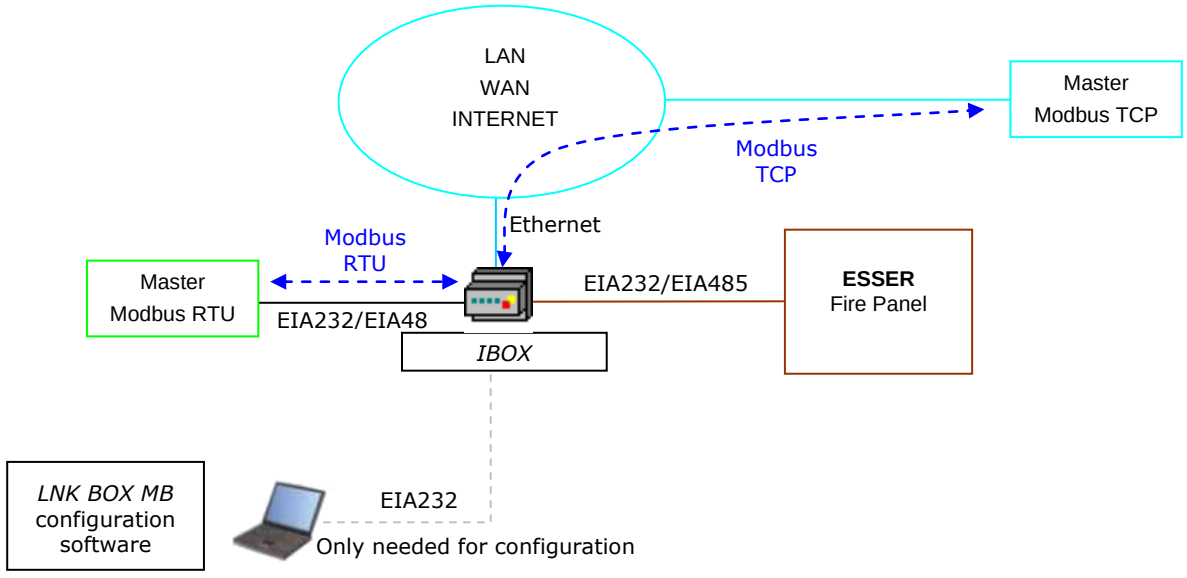
1.1 Introduction

Integration of ESSER 8xxx/IQ8/FlexES fire panels into a Modbus master device or system, using *IBOX Modbus Server - ESSER* gateway.

The aim of this integration is to make available state points of ESSER 8xxx, IQ8 and FlexES fire panels from a Modbus master device or system. From the Modbus system point of view, *IBOX* gateway works acting as a Modbus slave device responding to data polls coming from the Modbus master. From the ESSER system point of view, *IBOX* is acting as a serial device connected to its serial port serving the data received from ESSER to the Modbus side.

IBOX connects to the serial port of the ESSER panel, either through the EIA232 port or the EIA485 port, software selectable.

Once connected, *IBOX* is capable of connecting to all the panels present in the integration and connected to the same ESSERNET network as the *IBOX*.



Integration of ESSER 8xxx/IQ8/FlexES fire panels using *IBOX Modbus Server - ESSER*

1.2 Functionality

General overview:

The ESSER communication protocol is based on events. The states of the system elements (detectors, modules, etc.) are transmitted through the protocol in the form of events whenever they occur.

The role of *IBOX* consists in associate the elements of the ESSER system with Modbus register addresses.

IBOX has a configurable association of ESSER elements with Modbus register addresses.

The procedure of configuration of *IBOX* consists basically in the following:

- Introduction of the communication parameters for Modbus side and for ESSER side.
- Once this configuration has been done with the configuration software tool *LNK-BoxMB*, you have to download this configuration to *IBOX* via a serial or UDP connection and *IBOX* will reboot with the new configuration working.

IBOX can be configured as Modbus TCP slave or Modbus RTU (EIA232/EIA485) slave.

Up to 32 panels are supported.

The control of the ESSER panel is permitted, and commands toward the panel are permitted.

Also all the general events are detected by *IBOX* and translated to Modbus:

The integration operation is as follow:

Every time *IBOX* is connected and starts up, it proceeds with a polling cycle to allow the synchronism between the signals in the fire panel and in the *IBOX* registers.

Once *IBOX* is configured, connected to both systems (ESSER and Modbus) and the polling cycle is finished, it maintains a "keep alive" message with the panel, being this message the request/response of panel status, also it listens continuously the ESSER serial port for new events. With every event, the new status received is updated in the *IBOX* memory and become available to be read by the Modbus master device.

As mentioned before, the protocol in the serial port of the ESSER is based in spontaneous messages, that is, only changes of status are sent through the protocol whenever they occur. Because of this, when *IBOX* starts up, current status of elements is unknown, but the panel will inform to *IBOX* about the current state of the panel and elements when it receives the message requesting the status.

1.3 Capacity of IBOX

Element	Max.*	Notes
Number of Panels	32	This is the maximum number of panels supported independently on the version you have.
Number of Points	9000	Modbus registers

* There are different models of *IBOX Modbus Server – EDP* each one with different capacity. The table above shows the capacity for the top model (with maximum capacity).

Their order codes are:

- IBOX-MBS-EDP-3:
  - Model supporting up to 3000 Modbus registers (16 bit registers) or points
- IBOX-MBS-EDP-6:
  - Model supporting up to 6000 Modbus registers (16 bit registers) or points
- IBOX-MBS-EDP-9:
  - Model supporting up to 9000 Modbus registers (16 bit registers) or points

2 Modbus interface of *IBOX*

2.1 Description

IBOX acts as a slave device in its Modbus interface, this interface can be the Ethernet port (if using Modbus TCP), or the EIA232 port or the EIA485 port (if using Modbus RTU). To access the points and resources of the *IBOX* from Modbus system, you must specify as the Modbus register addresses, those inside *IBOX* corresponding to ESSER elements. See details of the Modbus address map below in this document.

2.2 Definition of signals

Every signal defined in *IBOX* corresponds to an ESSER element. Every possible element's status (FIRE, DISABLED, TEST…) in the ESSER system has a fixed numerical value in Modbus. This numerical value will be the point's value read from Modbus when the associated ESSER element is in this state. All the points are of type analog from the Modbus point of view.

2.3 Functions supported

Modbus functions 01 and 02 (*read coils* and *read digital inputs*) and are set to '0' when it is normal status (no alarms) and set to '1' when there is any other status.

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

Modbus function 06 must be used to write Modbus registers.

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

All the registers are of 2 bytes and its content is expressed in MSB..LSB.

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

3 ESSER interface of *IBOX*

This section describes the ESSER part of the *IBOX* configuration and functionality. This section assumes the user is familiar with ESSER technology and technical terms.

3.1 Main features

From the ESSER system point of view, *IBOX* acts as a serial device connected to its serial port, and serving the data received from ESSER to the Modbus side.

The ESSER communication protocol is based on events, the states of the system elements (detectors, modules, etc.) are transmitted through the protocol in the form of events whenever they occur.

IBOX ESSER interface use EDP protocol and can connect with FlexES panels through EIA232 or EIA485 connection. **EIA232 is the recommended connection** (TTY board may be required), although EIA485 can also be used with a direct connection to the fire panel.

Finally, if you don't have a FlexES panel, you will need the SEI KIT, which is provided by ESSER directly.

Each Modbus register corresponds to a single element of the panel: detector, output or panels and loops. The value offered per each Modbus register reflects the state of the element associated in the panel, the possible values are from 0-Normal, 1-Alarm… to 7-TEST. Each element to detect is defined in a table indicating zone number, output or detector number, and type of element (detector, output, zone or loop).

Commands toward the panel are allowed for reset, disconnect and test the elements. When a communication error with the panel occurs it is indicate in the Comm. Error register.

4 *LNK-BoxMB*. Configuration & monitoring tool for *IBOX* Modbus Server series

4.1 Introduction

LNK-BoxMB is a Windows® compatible software developed specifically to monitor and configure *IBOX* Modbus Server series. It is possible to configure all external protocols available for *IBOX* Modbus Server and to maintain different customer's configurations based on a *LNK-BoxMB* 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 *LNK-BoxMB*, as well as configure the integration signals list and connection parameters for every external protocol, it is permitted to select the serial port to use to connect to *IBOX* Modbus Server 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 ESSER technical support.

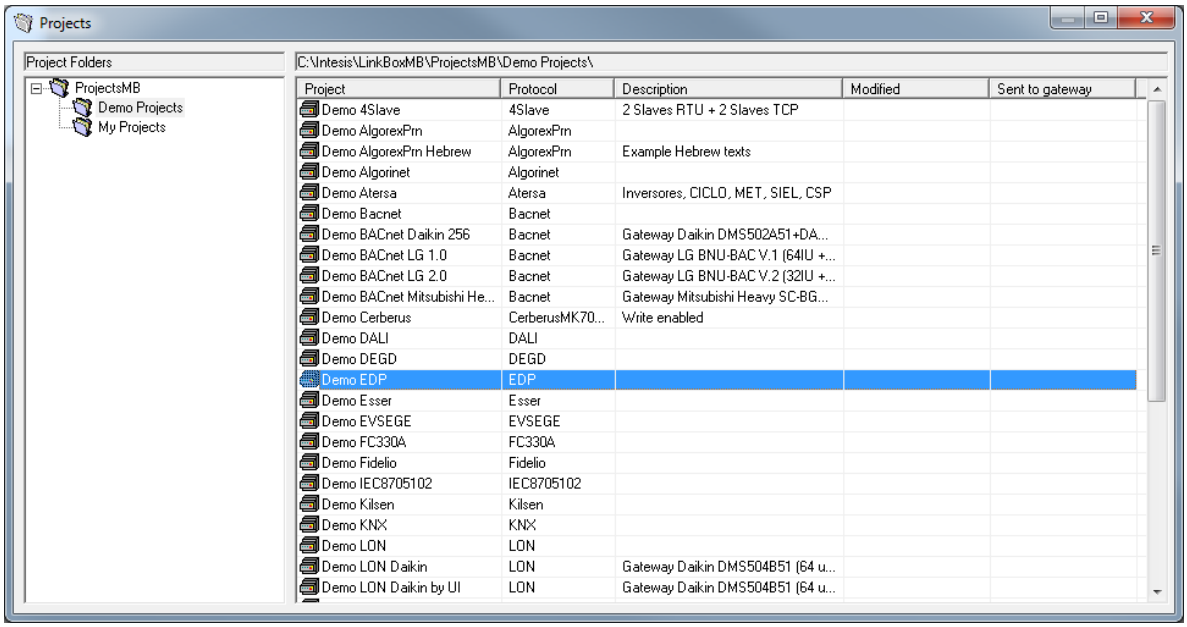
LNK-BoxMB allows configuring all *IBOX* Modbus Server series independently of the external system used. For every external system, *LNK-BoxMB* has a specific configuration window. Periodically, new free versions of *LNK-BoxMB* are released incorporating the latest developed integrations for external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

4.2 Project definition

The first step to do with *LNK-BoxMB,* for a new installation, is to create the installation's project giving a descriptive name to it. When you create a project, a new folder is created with the name of the project containing the configuration files needed depending on the external protocol selected for the project. It is strongly recommended that you create a new project for every installation, if not, overwriting of configuration files of previous installations using the same external protocol may occur, losing the configuration data for those previous installations.

The projects folder is located in AppFolder\ProjectsMB, where AppFolder is the installation folder of *LNK-BoxMB*. Inside the projects folder, a new folder will be created for every project defined in *LNK-BoxMB* with the files needed for the project.

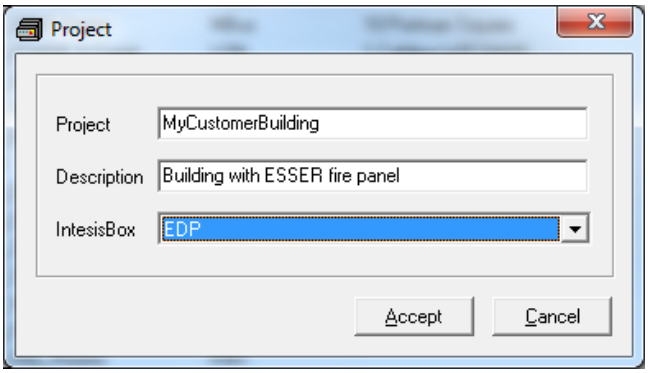
When you open *LNK-BoxMB*, the project selection window will appear inviting you to select a project or create a new one. A demo project for every external protocol supported is provided with the standard installation of *LNK-BoxMB*. You can create a new project or select a demo project based on the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



Project selection window

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

Specify a name and a description for the new project that will be based on the same external protocol than the selected one.



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.

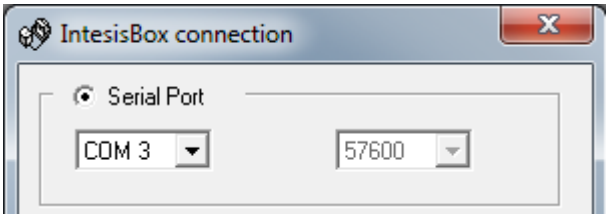
A description of the files created for an ESSER protocol based project can be found in section *Files* in this document.

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

1. Make sure *IBOX* is powered-up and correctly connected to the Modbus system via the Ethernet connection (Modbus TCP) or serial connection (Modbus RTU) and to ESSER panel via the EIA232 connection (consult details for connection and pin assignments in section *Connection*s of this document).

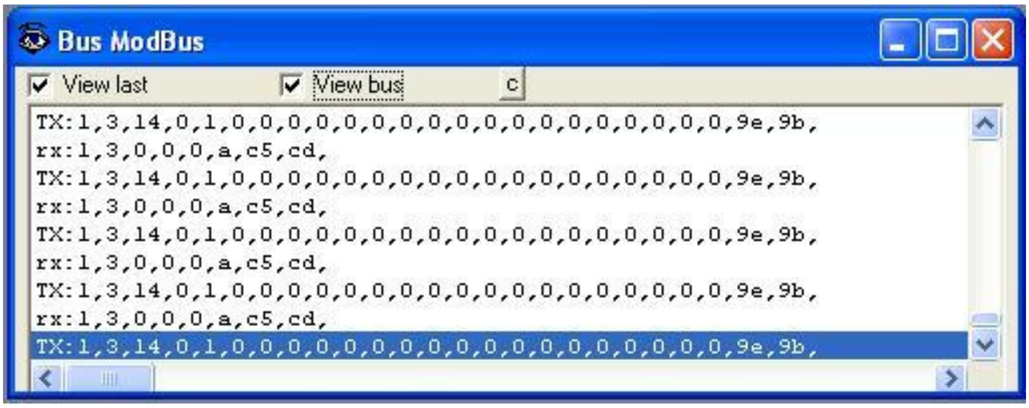
2. Connect a free PC serial port to the *IBOX* serial port marked as *PC Console*. (Use the standard serial cable supplied with the device or a customer's cable following the pin assignments specified in section *Connections* in this document).

3. Select in *LNK-BoxMB* the PC serial port used for the connection to *IBOX*. Use menu Connection.

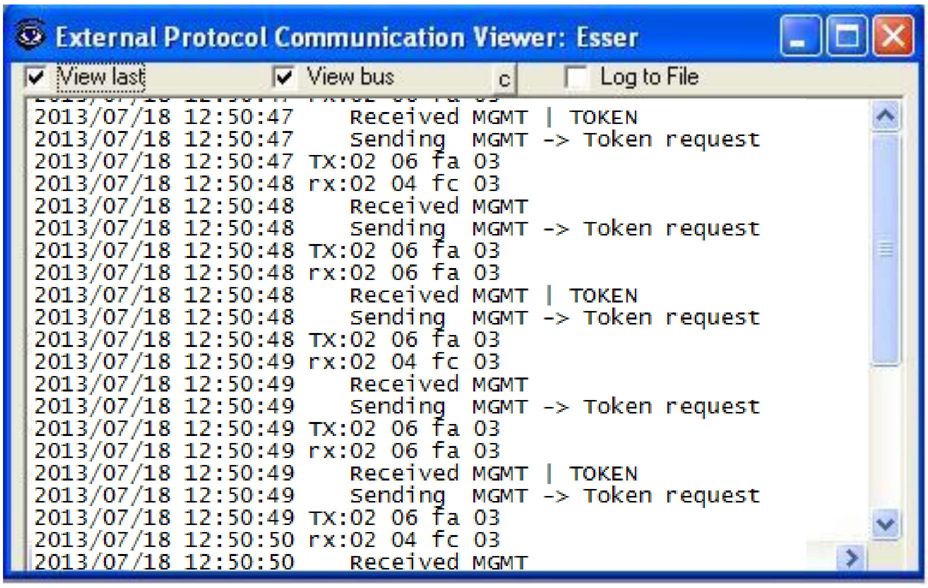


4. Click on the Connect button. This will start the communication process between LNK-BoxMB and IBOX. Once connected to *IBOX*, all the options of *LNK-BoxMB* are fully operative.

To monitor the communication between *IBOX* and the Modbus master device, select the menu *View -> Bus -> Modbus*. The *Modbus communication Viewer* window will be opened. This window show in real time all the communication frames between *IBOX* and the Modbus master device as well as debugging messages referent to internal protocol (Modbus) sent by *IBOX*.



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



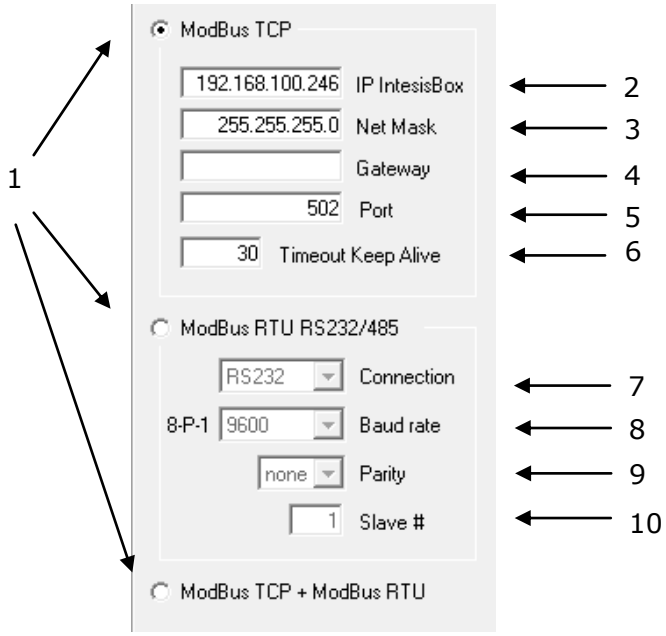
4.3 Connections configuration

To configure the *IBOX* connection parameters and the Modbus values for each possible state, select menu *Configuration -> IBOX*. The *ESSER Configuration* window will be opened.

Select the Connection tab to configure the connection parameters.

Two kinds of information are configured using this window, the referent to the Modbus side and the referent to the ESSER side.

Modbus side configuration parameters:



Modbus Interface Configuration

1. Select the type of connection desired (TCP, RTU or both).

If Modbus TCP is selected, then:

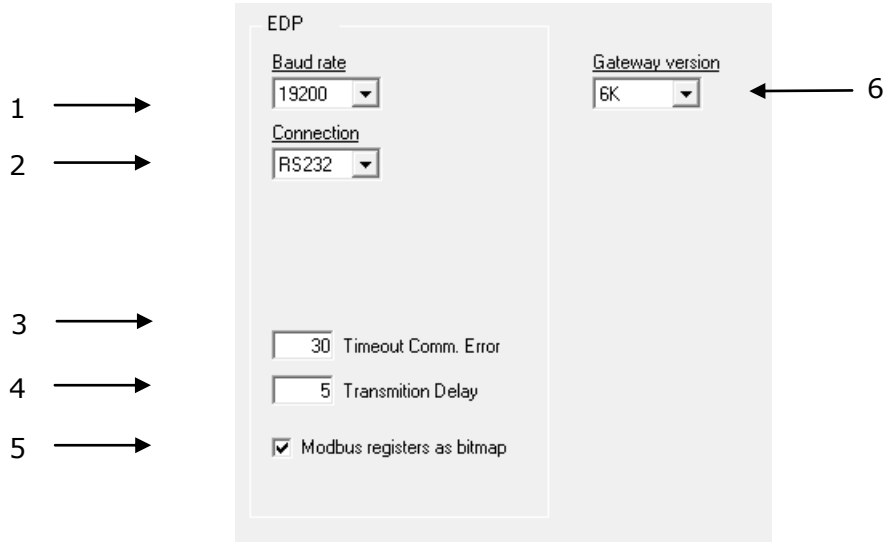
2. Enter the IP address for *IBOX*.
3. Enter the IP netmask for *IBOX*.
4. Enter the default router address to use by *IBOX*, leave blank if there is no need of router address.
5. Enter the TCP port to use, by default 502.
6. Enter the Timeout desired for the Keep Alive signal.

If Modbus RTU is selected, then:

7. Select the type of port to use (EIA232 or EIA485).
8. Select the baud rate to use.
9. Select the parity (none, even or odd).

10. Enter the Modbus slave number for *IBOX*.

ESSER side configuration parameters:



ESSER Configuration

1. Baud rate configured in the communication board, by default 19200.

2. ESSER to *IBOX* connection (EIA232/EIA485).

3. Timeout Comm. Error defines the amount of time (expressed in seconds) needed to consider that there has been a communication error. Its values may vary between 5 to 120 seconds.

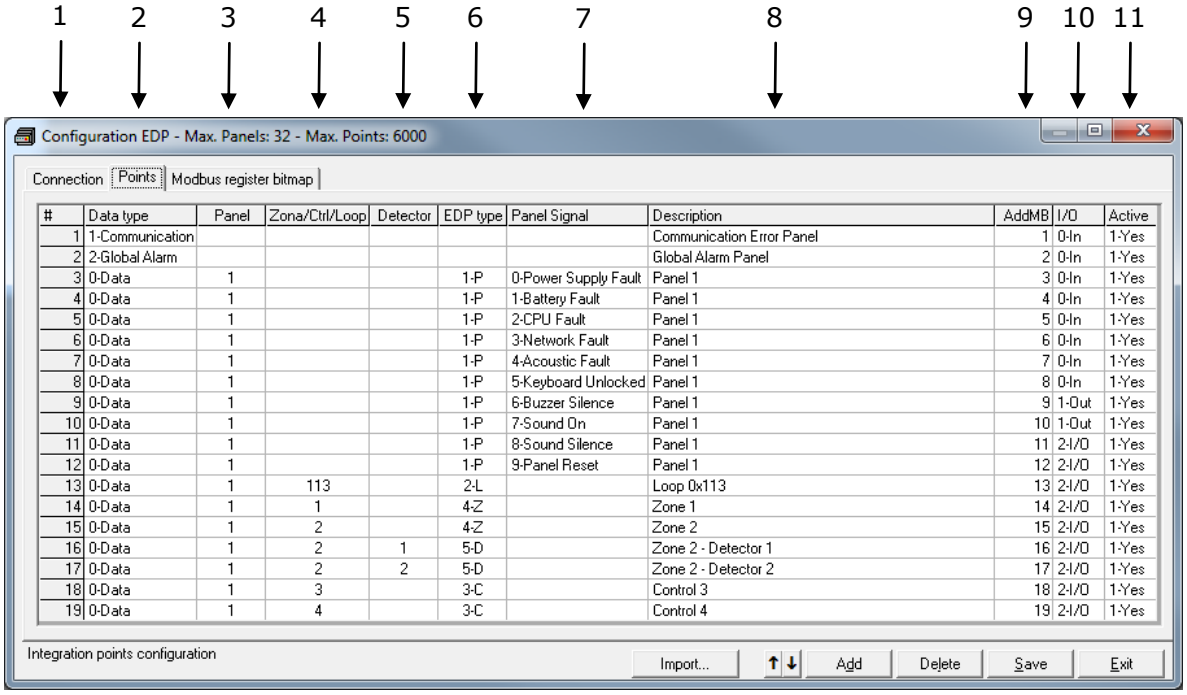
4. Transmission Delay defines the desired delay between Rx and Tx frames (expressed in miliseconds). Its values may vary between 1 to 100 ms.

5. Modbus registers as bitmap lets select if the Modbus registers are coded as a bitmap (several states can be read at the same time) or as values (just one single value can be read at the same time).

6. Gateway version: Please select here 3K, 6K or 9K if you own an IBOX-MBS-EDP-3, IBOX-MBS-EDP-6 or IBOX-MBS-EDP-9 respectively.

4.4 Signals

Select the *Points* tab for a description of the *IBOX* datapoints.



#	Data type	Panel	Zona/Ctrl/Loop	Detector	EDP type	Panel Signal	Description	AddMB	I/O	Active
1	1-Communication						Communication Error Panel	1	0-In	1-Yes
2	2-Global Alarm						Global Alarm Panel	2	0-In	1-Yes
3	0-Data	1			1-P	0-Power Supply Fault	Panel 1	3	0-In	1-Yes
4	0-Data	1			1-P	1-Battery Fault	Panel 1	4	0-In	1-Yes
5	0-Data	1			1-P	2-CPU Fault	Panel 1	5	0-In	1-Yes
6	0-Data	1			1-P	3-Network Fault	Panel 1	6	0-In	1-Yes
7	0-Data	1			1-P	4-Acoustic Fault	Panel 1	7	0-In	1-Yes
8	0-Data	1			1-P	5-Keyboard Unlocked	Panel 1	8	0-In	1-Yes
9	0-Data	1			1-P	6-Buzzer Silence	Panel 1	9	1-Out	1-Yes
10	0-Data	1			1-P	7-Sound On	Panel 1	10	1-Out	1-Yes
11	0-Data	1			1-P	8-Sound Silence	Panel 1	11	2-I/O	1-Yes
12	0-Data	1			1-P	9-Panel Reset	Panel 1	12	2-I/O	1-Yes
13	0-Data	1	113		2-L		Loop 0x113	13	2-I/O	1-Yes
14	0-Data	1	1		4-Z		Zone 1	14	2-I/O	1-Yes
15	0-Data	1	2		4-Z		Zone 2	15	2-I/O	1-Yes
16	0-Data	1	2	1	5-D		Zone 2 - Detector 1	16	2-I/O	1-Yes
17	0-Data	1	2	2	5-D		Zone 2 - Detector 2	17	2-I/O	1-Yes
18	0-Data	1	3		3-C		Control 3	18	2-I/O	1-Yes
19	0-Data	1	4		3-C		Control 4	19	2-I/O	1-Yes

Points list

This window is just for information purposes about the datapoints existing into the *IBOX* and its functionality.

1. *#.* Signal's number (edit not permitted). Every line in the grid corresponds to a signal (group of ESSER points). Signals (lines in the grid) can be added or deleted selecting the desired line and clicking *Add* or *Delete* buttons. Special signals (see below) are fixed (deletion not permitted). This column is used only to enumerate the lines in the grid (signals).

2. *Data type*. Indicates de type of data. It can take 3 different values:
   - Communication Error: It is set to '1' if there is a communication error between *IBOX* and the fire panel.
   - Global alarm: It is set to '1' if any of the elements in the panel is set to "fire". It can only be reset with a 'Panel Reset' (either from the panel itself or from the *IBOX*).
   - Data: If makes references to the rest of signals associated with the ESSER fire panel.

3. Panel. Indicates to which panel the signal is refered to. It may take values from 1 to 32.

4. *Zone/Control/Loop*. Zone number or Control group number (0..9999) of this ESSER point. Remember that Control and Zone share the addressing, therefore having "Control 1" and "Zone 1" is not possible.

5. *Detector*. Detector number (1..32) inside the zone of this ESSER point.
6. *EDP* Type. It indicates the type of ESSER point:
    - 1 - Panel
    - 2 – Loop
    - 3 – Control Group (Outputs)
    - 4 – Zone (It can be one single detector or a group of detectors)
    - 5 – Detector

7. Panel signal. This only applies if the EDP Type is a panel. It can be set to 10 different values:
    - 0 – Power Supply Fault
    - 1 – Battery Fault
    - 2 – CPU Fault
    - 3 – Network Fault
    - 4 – Acoustic Fault
    - 5 – Keyboard Unlocked
    - 6 – Buzzer Silence
    - 7 – Sound On
    - 8 – Sound Silence
    - 9 – Panel Reset

8. *Description*. Point's descriptive name (optional). Only used to describe the point at user level.

9. *AddMB.* Modbus address (1..30000).

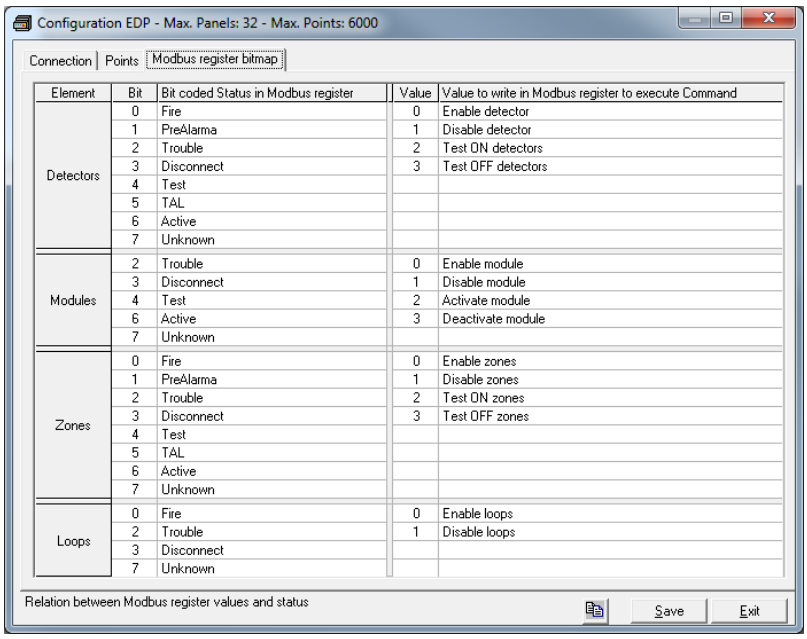
10. *I/O*. Data direction. (0-Input, 1-Ouput, 2-In/Out). Edit not permitted.

11. *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.

12. Import. This button can be used to import files directly generated by the fire panel system. Before importing the file, it is necessary to open the file and save it in CSV format limited by comas (with CSV-MSDOS format). You can make this kind of manipulation with Excel, for instance.

4.5 Modbus register bitmap/values

The last tab of the configuration window shows the different relations between the Modbus registers values and the ESSER signal status. Information shown in this tab changes according to the checkbox "Modbus registers as bitmap". Remember that this checkbox is in the connection tab.



Element	Bit	Bit coded Status in Modbus register	Value	Value to write in Modbus register to execute Command
Detectors	0	Fire	0	Enable detector
	1	PreAlarma	1	Disable detector
	2	Trouble	2	Test ON detectors
	3	Disconnect	3	Test OFF detectors
	4	Test		
	5	TAL		
	6	Active		
	7	Unknown		
Modules	2	Trouble	0	Enable module
	3	Disconnect	1	Disable module
	4	Test	2	Activate module
	6	Active	3	Deactivate module
	7	Unknown		
Zones	0	Fire	0	Enable zones
	1	PreAlarma	1	Disable zones
	2	Trouble	2	Test ON zones
	3	Disconnect	3	Test OFF zones
	4	Test		
	5	TAL		
	6	Active		
	7	Unknown		
Loops	0	Fire	0	Enable loops
	2	Trouble	1	Disable loops
	3	Disconnect		
	7	Unknown		

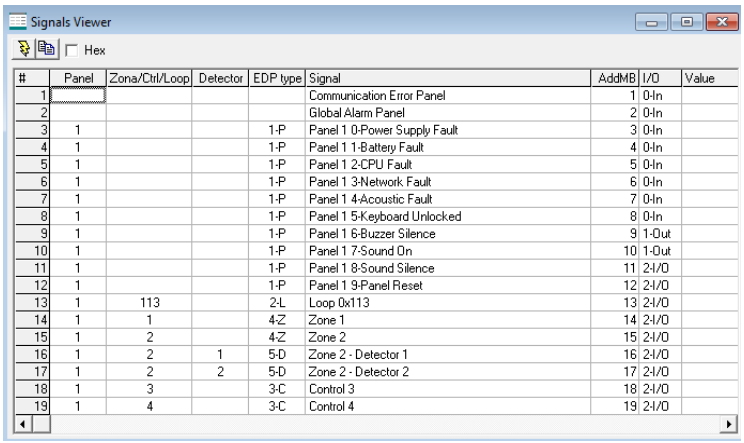
4.6 Sending the configuration to IBOX

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

Remember that saving the configuration and generating the *IBOX* bin file only saves to the hard disk on the PC the configuration files. **Do not forget to send the configuration binary file to the *IBOX* (using button *Send File)* after saving the configuration.**

4.7 Signals viewer

Once *IBOX* is running with the correct configuration, to supervise the status of the configured signals, select *Signals on the bar menu*. The Signals Viewer window will be opened. This window shows all the active *IBOX* signals with its main configuration parameters and its real time value in the column Value. After a reset of *IBOX* or after sending a configuration file to the *IBOX*, all the signal's values will be updated automatically in the signals viewer, in case you connect to the *IBOX* 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.



#	Panel	Zona/Ctrl/Loop	Detector	EDP type	Signal	AddMB	I/O	Value
1					Communication Error Panel	1	0-In	
2					Global Alarm Panel	2	0-In	
3	1			1-P	Panel 1 0-Power Supply Fault	3	0-In	
4	1			1-P	Panel 1 1-Battery Fault	4	0-In	
5	1			1-P	Panel 1 2-CPU Fault	5	0-In	
6	1			1-P	Panel 1 3-Network Fault	6	0-In	
7	1			1-P	Panel 1 4-Acoustic Fault	7	0-In	
8	1			1-P	Panel 1 5-Keyboard Unlocked	8	0-In	
9	1			1-P	Panel 1 6-Buzzer Silence	9	1-Out	
10	1			1-P	Panel 1 7-Sound On	10	1-Out	
11	1			1-P	Panel 1 8-Sound Silence	11	24/0	
12	1			1-P	Panel 1 9-Panel Reset	12	24/0	
13	1	113		2-L	Loop 0x113	13	24/0	
14	1	1		4-Z	Zone 1	14	24/0	
15	1	2		4-Z	Zone 2	15	24/0	
16	1	2	1	5-D	Zone 2 - Detector 1	16	24/0	
17	1	2	2	5-D	Zone 2 - Detector 2	17	24/0	
18	1	3		3-C	Control 3	18	24/0	
19	1	4		3-C	Control 4	19	24/0	

The signals viewer can be used although only one system is connected to the *IBOX*, *ESSER* 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. The new value entered will be available through the *Modbus* interface, the same way as if it has been received from the ESSER panel.

This tool is very useful to test the communication in the ESSER panel from the Modbus side and vice versa.

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

4.8 Files

LNK-BoxMB saves the integration configuration in the following files inside the project folder:

PROJECT.INI	.ini file containing general information referent to the project
EDP.INI	.ini file containing the information referent to the connection parameters and other special adjustments
EDP.DAT	Text file (tab separated values) with the signals information. This file can be edited (with Excel for example) to change the configuration quicker and easier. Later on, when selecting *Configuration -> IBOX* in *LNK-BoxMB*, if the changes have been made respecting the correct format, all the changes in the configuration done from Excel can be seen in the groups list.
EDP.LBOX	Binary file created from the information in the four files described above. This is the file really downloaded to *IBOX*.

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 *LNK-BoxMB* due, for example, to a failure of the hard disk in the PC where *LNK-BoxMB* was installed.

The configuration cannot be uploaded from IBOX to LNK-BoxMB, only can be downloaded, the download file EDP.LBOX does not contain all the integration information, as for example the signals description.

5 Set-up process and troubleshooting

5.1 Pre-requisites

It is necessary to have the Modbus master device operative and well connected to the Modbus port of *IBOX*, remember to respect the maximum of 15 meters cable distance if using EIA232 communication.

It is necessary to have the ESSER panel with an EIA232 port operative and at a distance of *IBOX* installation site of 15 meters maximum (due to EIA232 communication).

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

- *IBOX* Modbus Server device with ESSER *EDP* external protocol firmware loaded.
- *LNK-BoxMB* software to configure *IBOX*.
- Console cable needed to download the configuration to *IBOX*.
- Product documentation.

5.2 Set-up procedure

1. Install *LNK-BoxMB* on your laptop, use the setup program supplied for this and follow the instructions given by the Installation wizard.

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

3. Connect the communication cable coming from the Modbus master device to the port marked as **Modbus** of *IBOX* (used EIA232, EIA485 or Ethernet port depending on the type of Modbus communication to use). (See details for this communication cable in section *Connections* of this document).

4. Connect the communication cable coming from the EIA232 or EIA485 port of the ESSER to the port marked as **ESSER** of *IBOX*. (See details for this communication cable in section *Connections* of this document).

5. Power up *IBOX*. The supply voltage can be 9 to 30 Vdc or just 24 Vac. Take care of the polarity of the supply voltage applied.

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

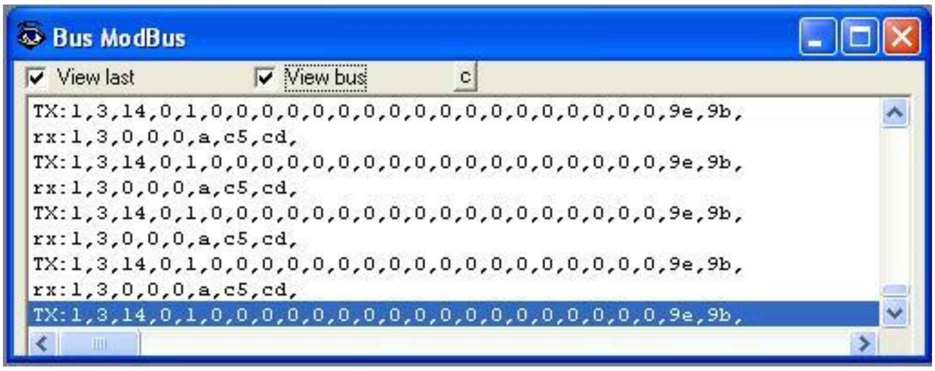
   - The use of DC power supplies, floating or with the negative terminal connected to earth. **Never use a DC power supply with the positive terminal connected to earth**.
   - The use of AC power supplies only if they are floating and not powering any other device.

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

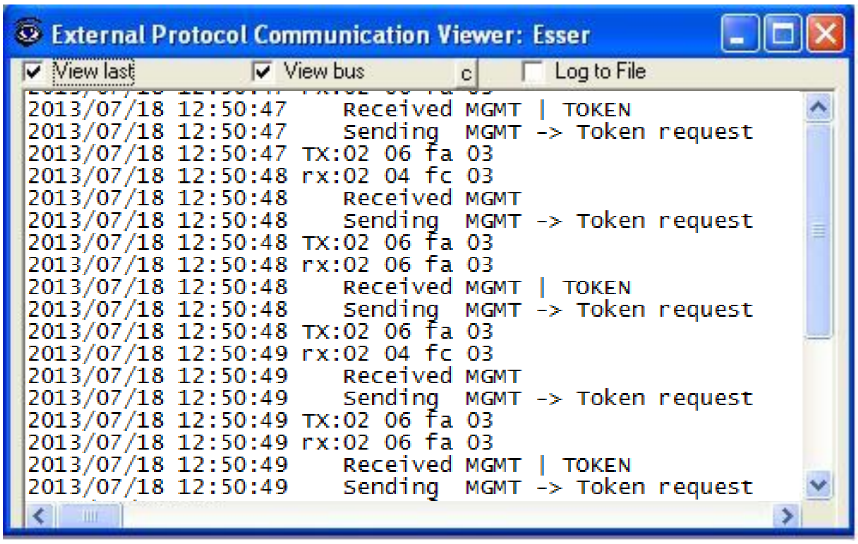
7. Open *LNK-BoxMB*, create a new project (see section

8. Modify the configuration as desired, save it and download the configuration file to *IBOX* as explained before.

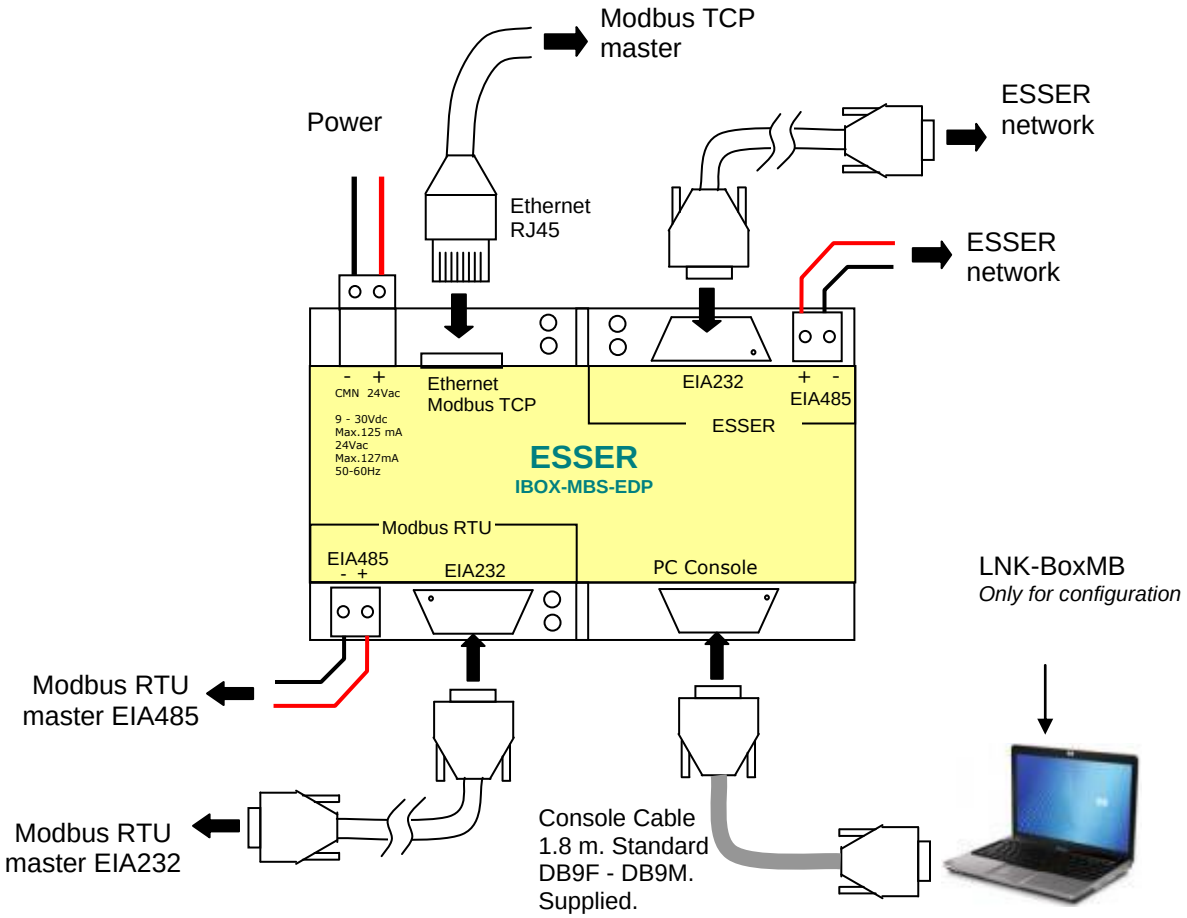
9. Open the *Modbus Communication Viewer* window (menu View -> Bus -> Modbus) and check that there is communication activity, some TX frames and some other rx frames. This means that the communication with the Modbus master device is ok. In case there is no communication activity between *IBOX* and the Modbus master device check that it is operative, check the baud rate, and check also the communication cable used to connect both devices. (See details for this communication cable in section *Connections* of this document).



10. Open the *External Protocol Communication Viewer* window (menu View -> Bus ESSER) and check that there is communication activity, some RX frames as showed in the figure below. This means that the communication with the ESSER panel is ok. In case of no communication activity between *IBOX* and ESSER, check that the EIA232 port of ESSER panel is operative and well configured, and check also the communication cable used to connect both devices. (See details for this communication cable in section *Connections* of this document).



6 Connections



Only one of the ESSER ports of the interface (EIA232 or EIA485) can be used simultaneously, the port to use must be configured within LNK-BoxMB software.

- *EIA232: Connect the serial EIA232 cable coming from the essernet network (for instance a SEI KIT network interface) to the ESSER EIA232 connector of the interface, this is a male DB9 connector (DTE) in which only lines TX, RX and GND are used, see pinout details in section 10. Respect the maximum distance of 15 meters for this EIA232 line.*

- *EIA485: Connect the EIA485 bus coming from essernet network (for instance a FlexES panel control) to the ESSER EIA485 connector of the interface. Respect the polarity. Remember the characteristics of the standard EIA485 bus: maximum distance of 1200 meters, maximum 32 device connected, and a 120 ohms terminator resistor in each end of the bus.*

RS-232 (SEI KIT) Port	***IBOX EIA232 ESSER***
2 (Tx)	*2 (Rx)*
3 (Rx)	*3 (Tx)*
7 (GND)	*5 (GND)*

RS-485 PCB Port	***IBOX EIA485 ESSER***
RS485-1a (TX/RX) +	*(TX/RX)+*
RS485-1b (TX/RX) -	*(TX/RX) -*

NOTE: *More information on connection examples can be found in section 10.*

7 Mechanical & electrical characteristics

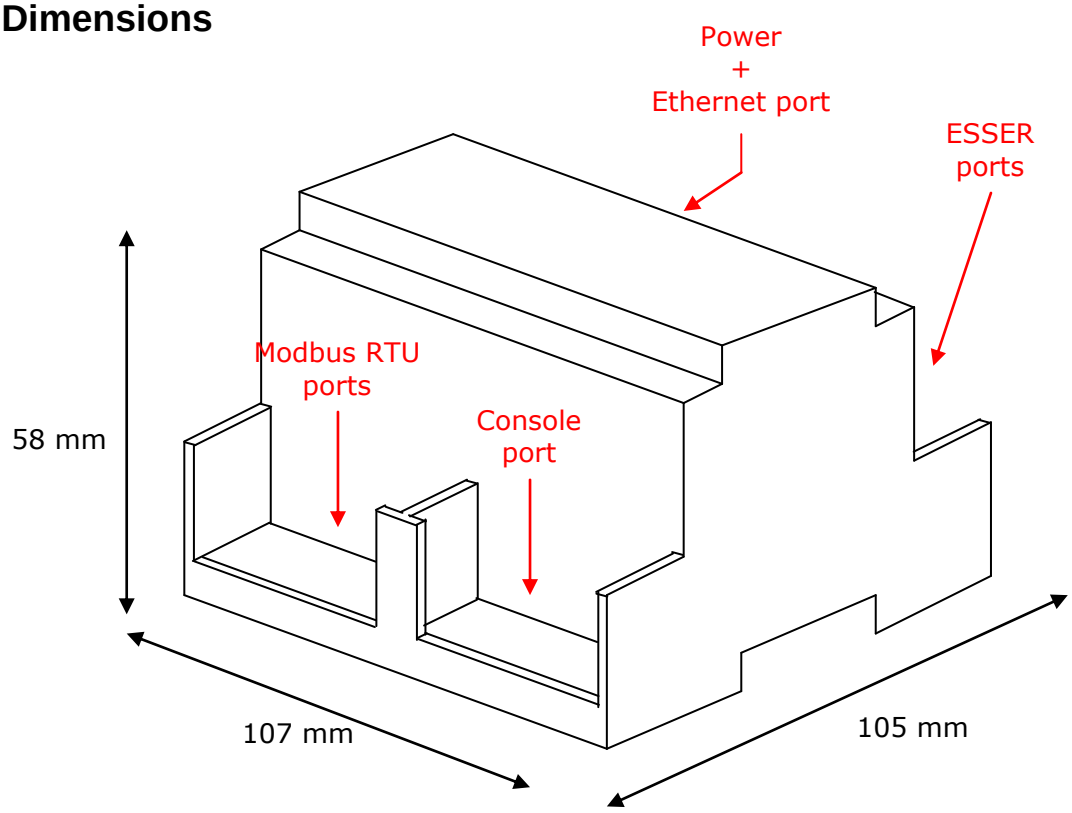
Enclosure	Plastic, type PC (UL 94 V-0). Dimensions: 107mm x 105mm x 58mm.
Colour	Light Grey. RAL 7035.
Power	9 to 30Vdc +/-10%, Max.: 125mA. 24Vac +/-10% 50-60Hz, Max.: 127mA Must use a NEC Class 2 or Limited Power Source (LPS) and SELV rated power supply. Plug-in terminal block for power connection (2 poles).
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.
Modbus TCP port	1 x Ethernet 10Base-T (RJ45).
Modbus RTU ports	1 x Serial EIA232 (DB9 male DTE). SELV 1 x Serial EIA485 (Plug-in screw terminal block 2 poles). SELV
ESSER ports	1 x Serial EIA232 (DB9 male DTE). SELV 1 x Serial EIA485 (Plug-in screw terminal block 2 poles). SELV
LED indicators	1 x Power. 2 x ESSER serial port activity (Tx, Rx). 2 x Modbus RTU serial port activity (Tx, Rx). 2 x Ethernet port link and activity (LNK, ACT).
Console port	EIA232. DB9 female connector (DCE). SELV
Configuration	Via console port.[1]
Firmware	Allows upgrades via console port.
Functional temperature range	0°C to +70°C
Functional humidity range	5% to 95%, non-condensing
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

8 Functional characteristics

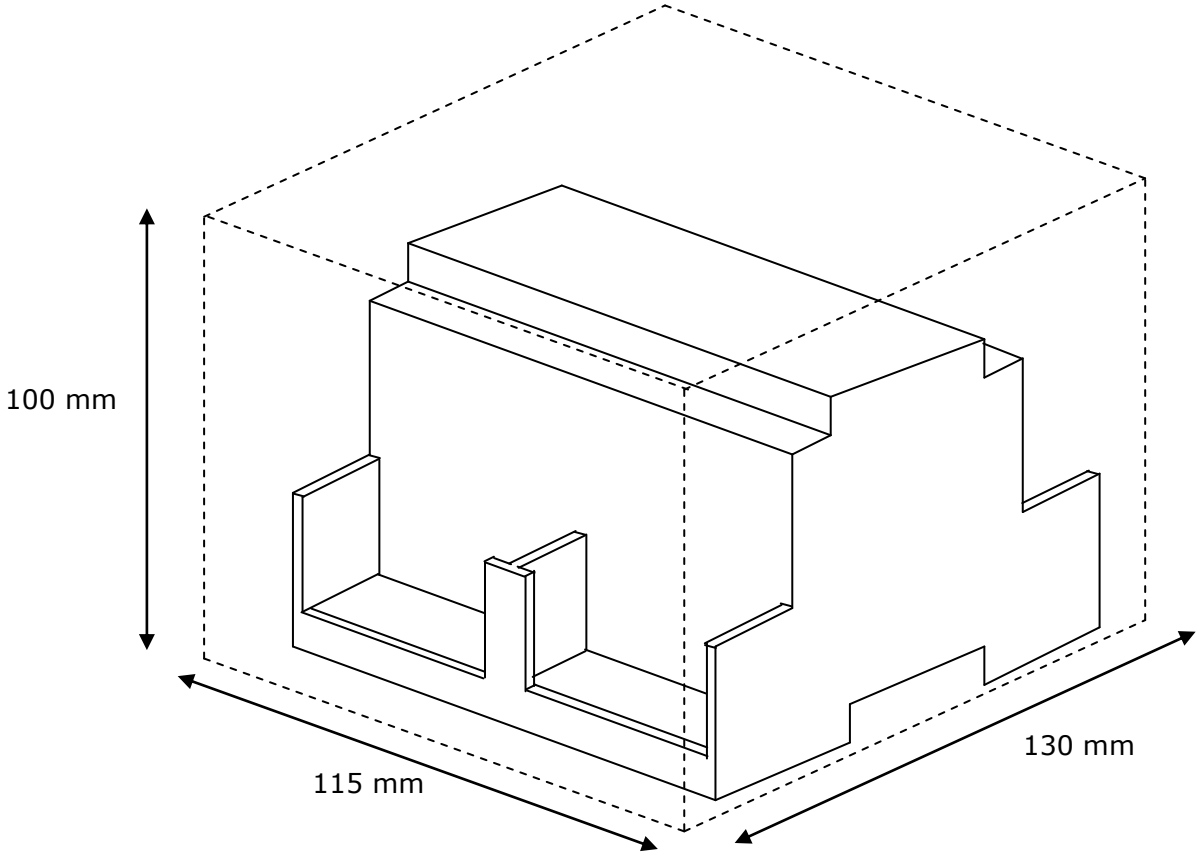
ESSER interface	
Type	Serial connection.
Configuration parameters	• Baud rate, data bits, parity. • EIA232/R485. • Waiting timeout for a response of the ESSER panel. • Waiting timeout for change status point back to normal. • Modbus registers as bitmap
Interactivity with ESSER system	• ESSER points can be read and write from the gateway. • Modbus value reflecting each possible state is fully configurable.
Modbus	

interface	
Device type	Slave.
Modbus modes supported	TCP, RTU EIA232 or EIA485.
Modbus TCP configuration parameters	• IP address. • Subnet mask. • Default gateway. • TCP port.
Modbus RTU configuration parameters	• EIA232/EIA485. • Baud rate, parity. • Slave number.
Points	
Modbus data types	All the points are of data type UNSIGNED INT in the Modbus interface.

9 Dimensions

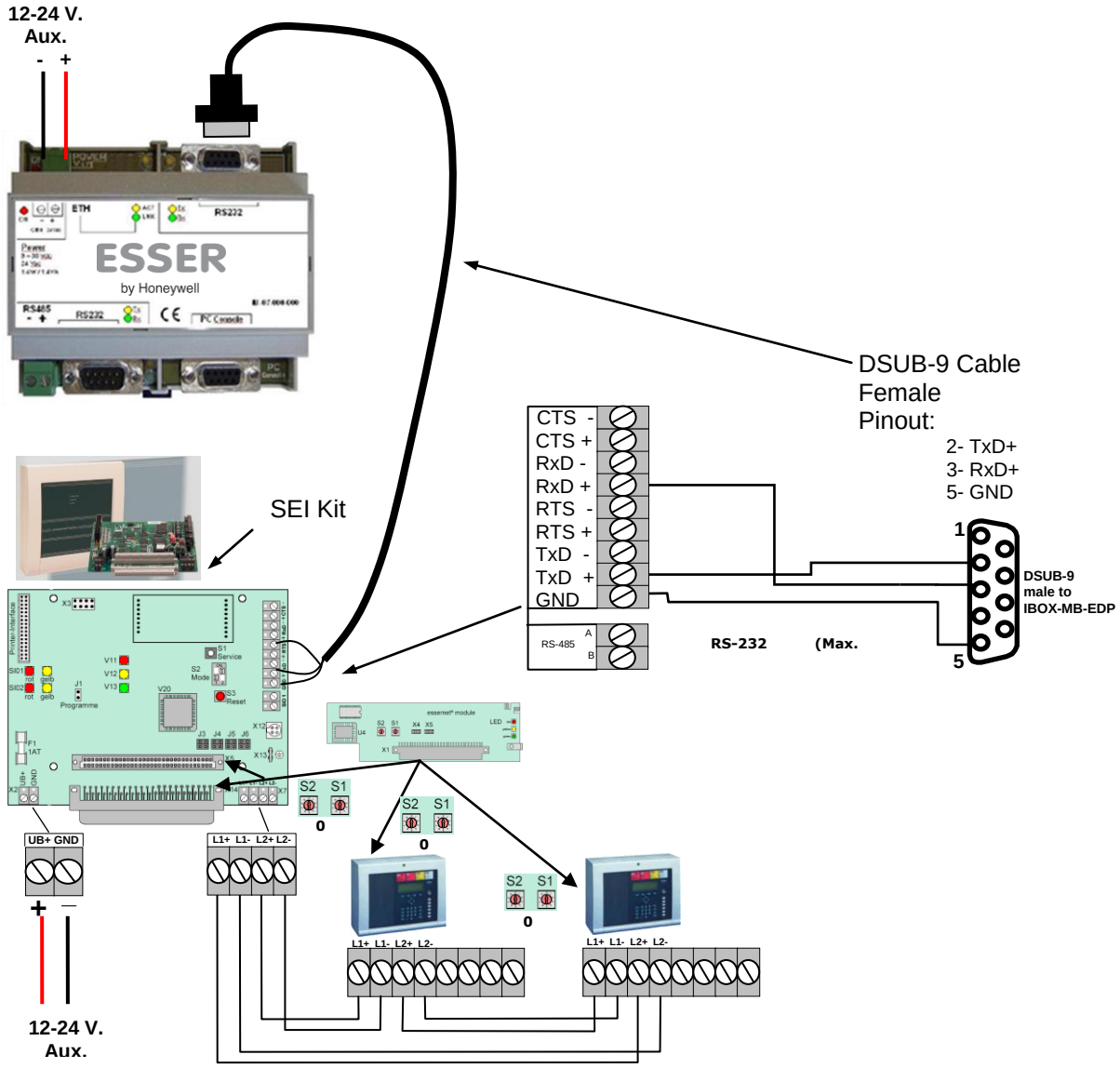


Recommended available space for its installation into a cabinet (wall or DIN rail mounting), with space enough for external connections:



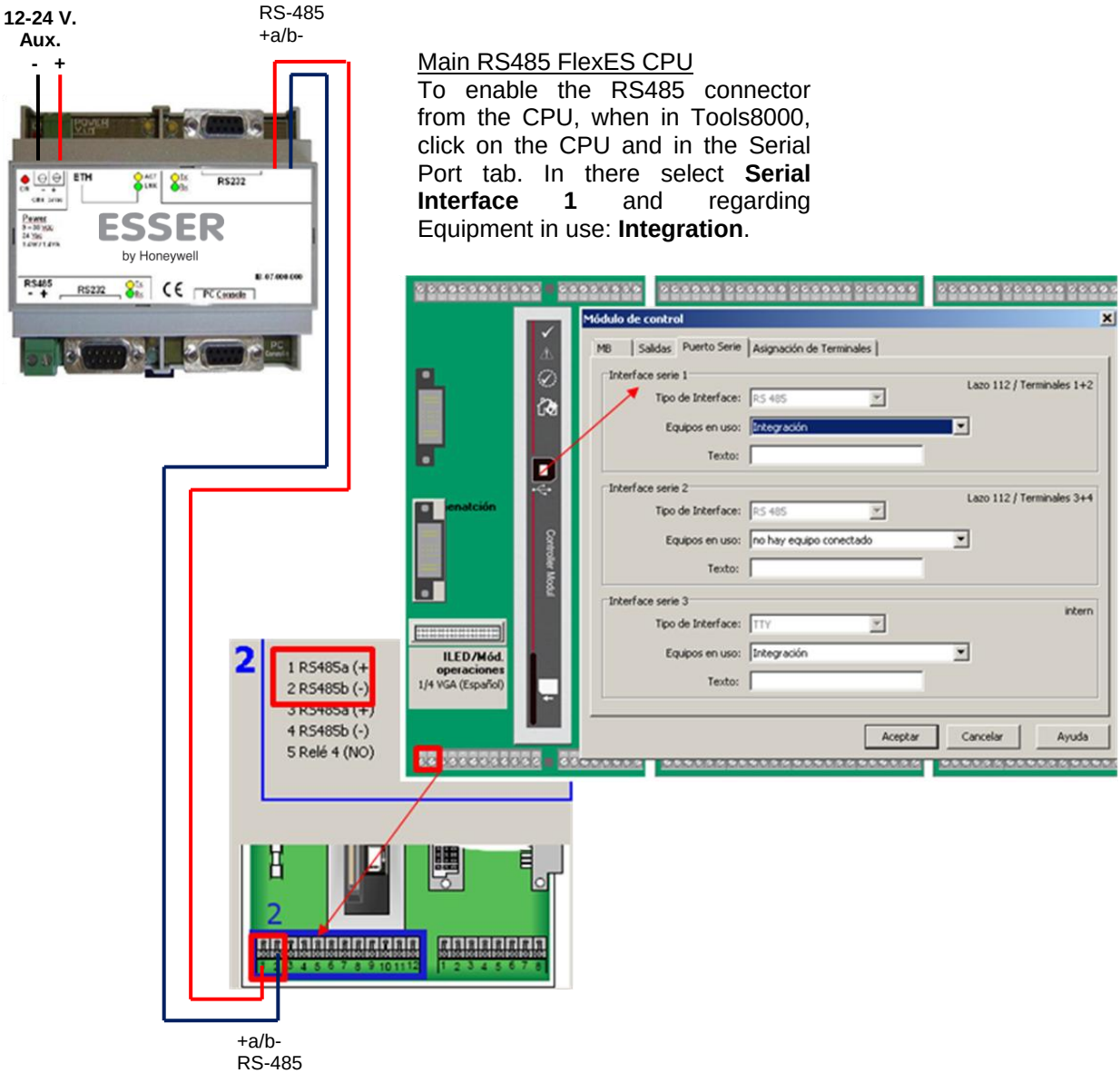
10 Connection examples

10.1 SEI KIT integration



Puerto RS-232 (SEI KIT)	IBOX RS232 ESSER
2 (Tx)	2 (Rx)
3 (Rx)	3 (Tx)
7 (GND)	5 (GND)

10.2 FlexES integration



Main RS485 FlexES CPU

To enable the RS485 connector from the CPU, when in Tools8000, click on the CPU and in the Serial Port tab. In there select **Serial Interface 1** and regarding Equipment in use: **Integration**.

Puerto RS-485 Placa base	**IBOX RS-485 ESSER**
RS485-1a (TX/RX) +	(TX/RX)+
RS485-1b (TX/RX) -	(TX/RX) -

Honeywell Life Safety Iberia

C/Pau Vila, 15-19 · 08911 Badalona (Barcelona)
Tel,: 902 03 05 45 · Fax: (+34) 934 658 635
International Tel.: +34 93 24 24 236

www.esser-es
www.honeywelllifesafety.es | infohlsiberia@honeywell.com