

## **FANinBOX 110V / 230V 1CH**

**4-speed ceiling fan controller @110VAC / @230VAC**

**ZCLFB110C1  
ZCLFB230C1**

Application program version: [1.3]  
User manual edition: [1.3]\_a

# CONTENTS

---

Contents .....	2
Document Updates .....	3
1 Introduction .....	4
1.1 FANinBOX .....	4
1.2 Installation.....	5
2 Configuration .....	6
2.1 General.....	6
2.2 Channel CH1 .....	9
2.2.1 Configuration.....	9
2.2.2 Initialization .....	13
2.2.3 Scenes.....	14
2.2.4 Timed Off.....	15
2.2.5 Lock.....	16
2.3 Logic Functions.....	18
2.4 Manual Control .....	19
ANNEX I: Communication Objects .....	22

## DOCUMENT UPDATES

---

Version	Changes	Page(s)
[1.3_a]	<b>Changes in the application program:</b> <ul style="list-style-type: none"><li>Update functional block Heartbeat: device recovery objects.</li></ul>	7, 8
[1.2_a]	<b>Changes in the application program:</b> <ul style="list-style-type: none"><li>Update of speed levels: up to 4 speeds.</li></ul>	10, 11
[1.1_a]	<b>Changes in the application program:</b> <ul style="list-style-type: none"><li>Internal optimization.</li></ul>	-

# 1 INTRODUCTION

---

## 1.1 FANinBOX

---

**FANinBOX** from Zennio is a KNX actuator specifically designed for the control of 230V or 110V fans, depending on the device.

Its wide variety of functions makes it a versatile and robust device.

- **Up to 4 speeds control for fans (and stop).**
- **Starting characteristic** to achieve an initial high torque.
- **Initial configuration of the fan speed level**, allows setting the initial state (after the power recovery, download or ETS restart).
- Customisable **timed Off**.
- **Error identification and notification.**
- **Lock** to enable/disable the fan control.
- **Scenes setting**. It is possible to define up to 5 scenes.
- 10 customisable, multi-operation **logic functions**.
- **Manual operation and supervision** of the fans through the on-board pushbuttons.
- **LED indicators** to show error situations.
- **Relays Switches Counter**.
- **Heartbeat** or periodical “still-alive” notification.

## 1.2 INSTALLATION

FANinBOX connects to the KNX bus through the on-board KNX connector. Once the device is provided with power from the KNX bus, both the individual address and the associated application program can be downloaded.

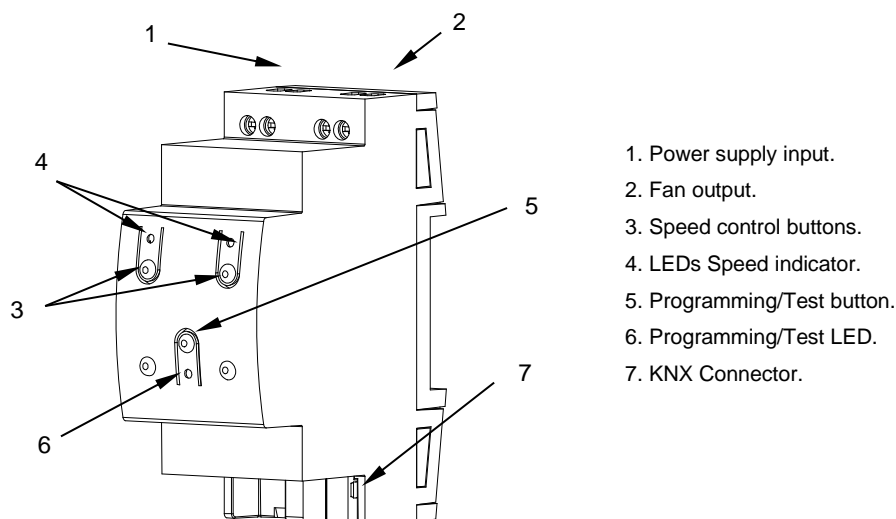


Figure 1 FANinBOX - Element Diagram.

The main elements of the device are described next:

- **Test/Prog. Pushbutton (5):** a short press on this button sets the device into the programming mode, making the associated LED (6) light in red.

**Note:** if this button is held while plugging the device into the KNX bus, the device will enter into **safe mode**. The LED will blink in red every 0.5 seconds.

- **Output Channels (2):** slots for the connection of the output lines (loads).
- **Neutral and Phase Inputs (1):** slots for the connection of the voltage wires (neutral and phase lines).

To get detailed information about the technical features of the device, as well as on the installation and security procedures, please refer to the corresponding **Datasheet**, bundled with the original package of the device and also available at [www.zennio.com](https://www.zennio.com).

## 2 CONFIGURATION

---

### 2.1 GENERAL

---

The general configuration of FANinBOX requires setting some general parameters:

- The **manual control** type, in the case that operating the channels through the on-board pushbuttons is necessary for testing or for other purposes.
- Whether to send **error notifications** or not to the bus.
- Enabling or disabling the **Logic Functions** module.
- **Heartbeat** or periodical “still-alive” notification.

---

#### ETS PARAMETERISATION

---

After importing the corresponding database in ETS and adding the device into the topology of the desired project, the configuration process begins by entering the Parameters tab of the device.

The tab tree on the left shows the “**General**” tab in the first place, containing the following parameters:

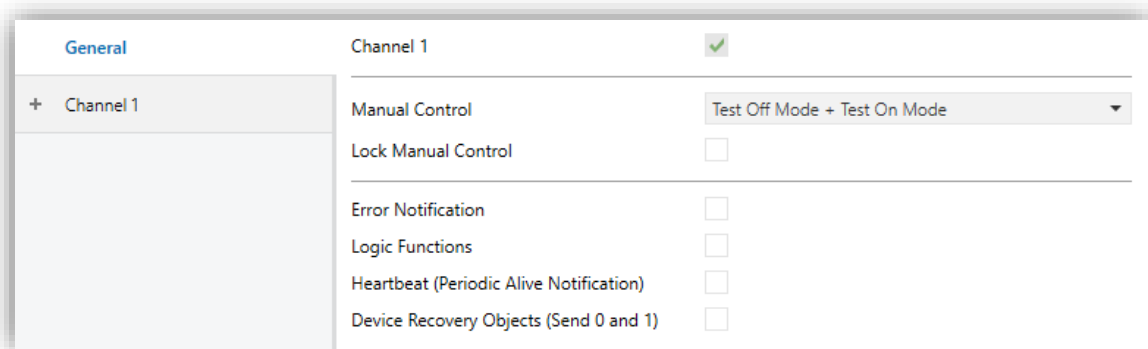
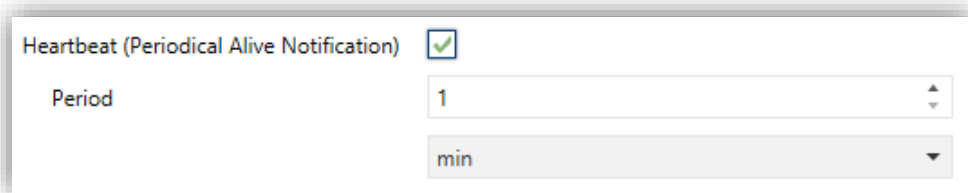


Figure 2 General - Configuration.

- **Channel 1** [[enabled](#)]<sup>1</sup>: read-only parameter to make it evident that the “Channel 1” tab is always enabled in the tab tree on the left. See section 2.2 for details.
- **Manual Control**: see section 2.4 for details.
- **Error Notifications** [[disabled/enabled](#)]: enables or disables the object “**Power Supply Failure**”. This object is sent to the bus with a value of “1” every 30 seconds when FANinBOX detects a power failure and with a value of “0” (once only) when it is reset.

In addition, regardless of whether this parameter has been enabled or not, FANinBOX notifies the error by flashing on one of the indicator LEDs and also disconnects the channel.

- **Logic Functions** [[disabled/enabled](#)]: enables or disables the “Logic Functions” tab, which contains specific parameters for the case the Logic Functions module is required. Please see section 2.3 for details.
- **Heartbeat (Periodical Alive Notification)** [[disabled/enabled](#)]: this parameter lets the integrator incorporate a one-bit object to the project (“**[Heartbeat] Object to Send ‘1’**”) that will be sent periodically with a value of “1” to notify that the device is still working (still alive).



The screenshot shows a configuration window for 'Heartbeat (Periodical Alive Notification)'. At the top, there is a checked checkbox. Below it, the 'Period' is set to '1' in a dropdown menu, and the unit is set to 'min' in another dropdown menu.

Figure 3. Heartbeat (Periodical Alive Notification)

**Note:** The first sending after download or bus failure takes place with a delay of up to 255 seconds, to prevent bus overload. The following sendings match the period set.

- **Device Recovery Objects (Send 0 and 1)** [[disabled/enabled](#)]: this parameter lets the integrator activate two new communication objects (“**[Heartbeat]**

<sup>1</sup> The default values of each parameter will be highlighted in blue in this document, as follows: [[default/rest of options](#)].

**Device Recovery**”), which will be sent to the KNX bus with values “0” and “1” respectively whenever the device begins operation (for example, after a bus power failure). It is possible to parameterise a certain **delay** [0...255] to this sending.

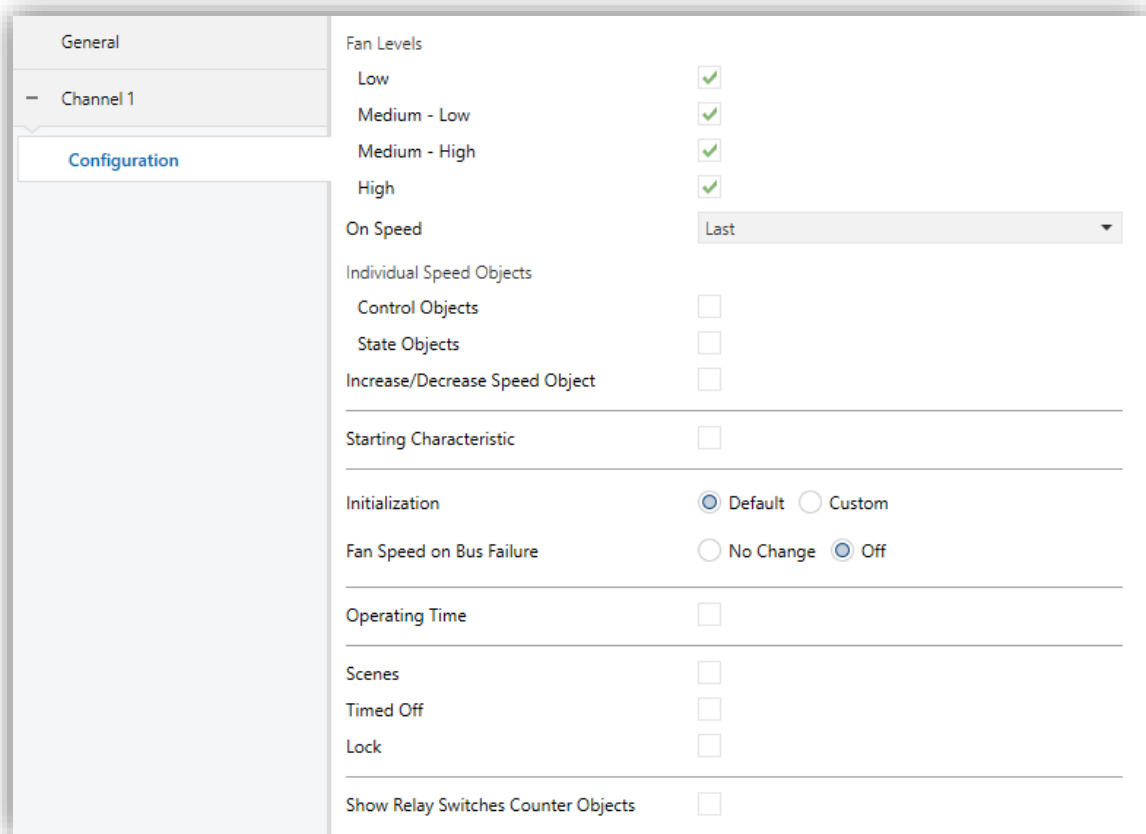


## 2.2 CHANNEL CH1

### 2.2.1 CONFIGURATION

In the Channel 1 configuration it is possible to set the **fan levels** to be controlled, to enable the control and status speed objects, as well as enabling other more specific functionalities that will be explained in the following sections: **starting characteristic**, **initial configuration**, **scenes**, **timed off**, **lock**, etc.

#### ETS PARAMETERISATION



The screenshot displays the configuration interface for Channel CH1. The left sidebar shows a tree view with 'General' and 'Channel 1' expanded, with 'Configuration' selected. The main area contains the following settings:

Setting	Value/Status
Fan Levels	
Low	<input checked="" type="checkbox"/>
Medium - Low	<input checked="" type="checkbox"/>
Medium - High	<input checked="" type="checkbox"/>
High	<input checked="" type="checkbox"/>
On Speed	Last
Individual Speed Objects	
Control Objects	<input type="checkbox"/>
State Objects	<input type="checkbox"/>
Increase/Decrease Speed Object	<input type="checkbox"/>
Starting Characteristic	<input type="checkbox"/>
Initialization	<input checked="" type="radio"/> Default <input type="radio"/> Custom
Fan Speed on Bus Failure	<input type="radio"/> No Change <input checked="" type="radio"/> Off
Operating Time	<input type="checkbox"/>
Scenes	<input type="checkbox"/>
Timed Off	<input type="checkbox"/>
Lock	<input type="checkbox"/>
Show Relay Switches Counter Objects	<input type="checkbox"/>

Figure 4 Channel CHx.

Regardless of the setting, the following is always displayed:

- “[CH1] On/Off”: allows turning on/off the fan.
- “[CH1] On/Off (Status)”: notifies the on/off fan status.
- “[CH1] Speed: Percentage Control”: percentage fan speed control.

- “[CH1] Speed: Percentage Control (Status)”: percentage fan speed status.

The “Configuration” screen of each channel contains the following parameters:

- **Fan Levels** [[Low](#) / [Medium-Low](#) / [Medium-High](#) / [High](#)]: sets the fan speeds.

When more than one fan level is selected the following parameter shows up:

- **On Speed:** [[Last](#) / [Level 1](#) / [Level 2](#) / [Level 3](#) / [Level 4](#)]: fan speed set when the fan is turned on through the object “[C1] On/Off”.

Speed levels will be assigned in order from lowest to highest, according to the speeds selected in the checkboxes, for example:

- 3 Speeds (Medium-Low, Medium-High, High):
  - Medium-Low → Level 1.
  - Medium-High → Level 2.
  - High → Level 3.
- 2 Speeds (Low, High):
  - Low → Level 1.
  - High → Level 2.

- **Individual Speed Objects:**

- **Control Objects** [[disabled/enabled](#)]: enables a 1-bit control object “[CH1] Speed: Level X” for each level.
- **State Objects** [[disabled/enabled](#)]: enables a 1-bit status object “[CH1] Speed: Level X (Status)” for each level.

- **Increase/Decrease Speed Object** [[disabled/enabled](#)]: enables the one-bit object “[CH1] Speed: 1-Bit Control” for increasing or decreasing the speed level sequentially. In addition, a **cyclical** control [[disabled/enabled](#)] can be set (a further step once reaching the maximum level activates the minimum level again) or not.

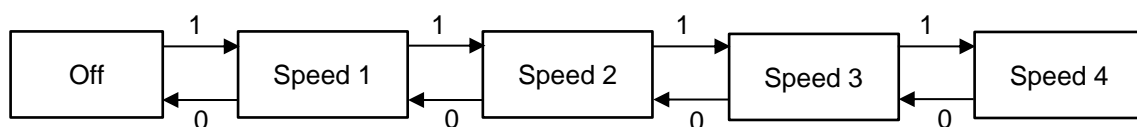


Figure 5. Non-cyclical fan step control

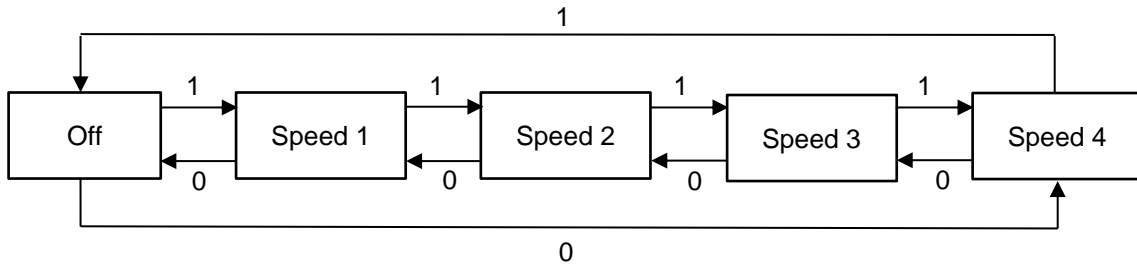


Figure 6. Cyclical fan step control

- **Starting Characteristic** [*disabled/enabled*]: allows to select a certain fan speed level for a limited time when the fan switches on. This option is useful for some engines that require a higher supply in the start-up and need to step through a higher speed level (e.g. 2, 3 or 4) for some time before they switch to lower speeds. When enabled, the following parameters show up:
  - **Starting Fan Speed at Switch On** [*Medium-Low / Medium-High / High*]: sets the starting speed.
  - **Minimum Dwell Period in Starting Fan** [*1...5...65535*][s]: sets the time for which the previous speed will be maintained.

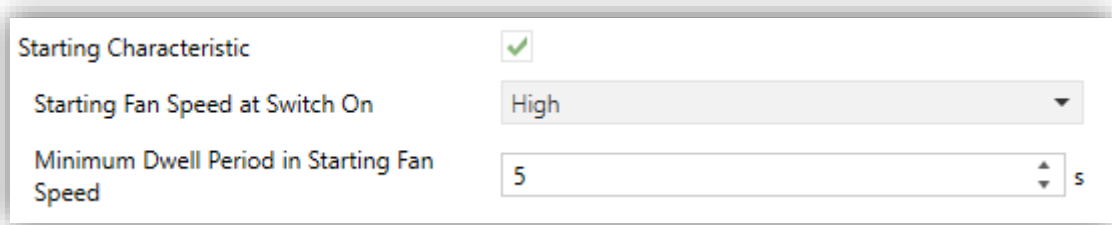


Figure 7. Starting Characteristic.

- **Initialization** [*Default / Custom*]: sets whether to perform the default action or a custom action during the device start-up. The latter should be configured from a specific parameter screen (see section 2.2.2).
- **Fan Speed on Bus Failure** [*No Change / Off*]: sets whether the fan speed on bus failure will be “Off” or “No change”.
- **Operating Time Counter** [*disabled/enabled*]: offers the possibility to record the time the fan remains on. When enabled, the following parameters appear:

Operating Time	<input checked="" type="checkbox"/>
Seconds	<input type="checkbox"/>
Hours	<input checked="" type="checkbox"/>
Initial Operating Time	<input checked="" type="radio"/> Keep Current Value <input type="radio"/> Set Value
Periodic Sending (0 = Disabled)	60 <input checked="" type="radio"/> min <input type="radio"/> h

Figure 8. Operating Time

- **Seconds** [*disabled/enabled*]: enables the object “[CH1] Operating Time (s)”, corresponding to the counter log (in seconds) of the time that the fan remains on.
- **Hours** [*disabled/enabled*]: enables the object “[CH1] Operating Time (h)”, corresponding to the counter log (in hours) of the time that the fan remains on.
- **Initial Operating Time** [*Keep Current value / Set Value*]: sets the initial value of the counter log after an ETS download. When selecting “Set Value”, an additional textbox appears to enter the desired **value** [*0...3600*][s] [*0...65535*][h].
- **Periodic Sending** [*0...60...1440*][min][*0...1...24*][h]: sets a periodic sending for the enabled objects (“[CH1] Operating time (s)” and/or “[CH1] Operating time (h)”). This period is common for both objects.

It is allowed both reading and updating the value of the counters by writing in the enabled objects (for example, the user can reset them by writing the value 0 in any of them, updating both objects at the same time).

- **Scenes** [*disabled/enabled*]: activates or deactivates the Scenes function, which should be configured from a specific parameter screen (see section 2.2.3).

- **Timed Off** [[disabled/enabled](#)]: activates or deactivates the Timed Off function, which should be configured from a specific parameter screen (see section 2.2.4).
- **Lock** [[disabled/enabled](#)]: activates or deactivates the Lock function, which should be configured from a specific parameter screen (see section 2.2.5).
- **Show Relay Switches Counter Objects** [[disabled/enabled](#)]: enables two communication objects to keep track of the number of switches performed by each of the relays (“**[Relay X] Number of Switches**”) and the maximum number of switches carried out in a minute (“**[Relay X] Maximum Switches per Minute**”).

#### Notes:

- *Note that FANinBOX has three internal relays for each channel to set the desired fan speed.*
- *After updating to version 1.3 the values of these objects are reset.*

## 2.2.2 INITIALIZATION

---

This function is provided to let the integrator specify the desired fan speed after FANinBOX recovers from a KNX bus failure or a power supply failure.

In case the integrator feels comfortable with the **default** initialisation settings (fan off after an ETS download, and previous fan speed level after a bus recovery), it will not be necessary to configure this function.

### ETS PARAMETERISATION

---

If “**Initialization**” (in the Configuration tab; see section 2.2.1) was set to “Custom” a specific entry (“**Initialization**”) will be added to the tab tree. Otherwise, FANinBOX will implement the already described default initialisation.

Initial Speed	Last
Send Status Object	<input checked="" type="checkbox"/>
Delay (0 = Disabled)	1 s

Figure 9 Initial Configuration

- **Initial Speed** [[Last](#) / [Off](#) / [Level 1](#) / [Level 2](#) / [Level 3](#) / [Level 4](#)]: fan speed both after a bus recovery or a power supply recovery.
- **Send Status Object** [[disabled](#)/[enabled](#)]: when enabled the following parameter shows up:
  - **Delay** [[0](#)...[1](#)...[3600](#)][s]: time (seconds) that elapses between the start-up of the device at the beginning of sending objects to the bus.

### 2.2.3 SCENES

Up to **five scenes** can be defined so that, when the corresponding value is received from the bus, the fan adopts a certain speed level.

It is important to bear in mind that executing a scene is equivalent to sending the analogous orders to the corresponding objects. Therefore, the result will depend on the initial state when the scene is executed.

For example, if a speed selection order is received during a lock state, the order will be ignored, as it would happen in case of receiving the request through the analogous communication object.

This device permits the **scene recording**, although it will not be possible to record any states that, for the current configuration, may not be available for configuration in ETS during the scene parameterisation.

### ETS PARAMETERISATION

Once Scenes has been enabled from the “Configuration” tab (see section 2.2.1), a new tab named “Scenes” is added to the tab tree on the left, containing the following options:

Scene 1	<input checked="" type="checkbox"/>
Number	1
Speed	Off
Scene 2	<input type="checkbox"/>

Figure 10 Scenes

- **Scene “n”** [*disabled/enabled*]: enables or disables scene “n”, which should be then configured through the following additional parameters:
  - **Number** [*1...64*]: sets the value (1-64) that, when received through object “[C1] Scene”, will trigger the adoption of the configuration defined next.
  - **Speed** [*Off / Level 1 / Level 2 / Level 3 / Level 4*]: fan speed applied when executing the scene.

Besides running a scene, it is also possible to save it: if the device receives an order to save the scene (values 128-191 through “[CH1] Scene”), the current speed level of the fan will be saved, but only if the value corresponds to any of the parameterised scenes (if not, the order will be ignored).

## 2.2.4 TIMED OFF

The Timed Off function allows automatically turn off the fan controlled by the channel after a certain configurable time.

### ETS PARAMETERISATION

Once the Timed Off function has been enabled from the “Configuration” tab (see section 2.2.1), a new tab named “Timed Off” is added to the tab tree on the left, containing the following options:

Figure 11. Timed Off.

The only parameters contained by the **Timed Off** screen are:

- **Value** [0 = Deactivate; 1 = Activate / 0 = Activate; 1 = Deactivate]: sets the polarity of the object “[CH1] Timed Off” which will active or deactivate the timed off.
- **Delay** [10...3600][s][1...1000][min][1...24][h]: sets the time that should elapse before turning off the fan. It is possible to change this time using the communication object “[CH1] Timed Off: Delay”.

The Timed Off function counts the time elapsed after the object “[CH1] Timed Off” receive the activation value. If this count either exceeds the parameterised timeout or the value of the object “[CH1] Timed Off: Delay”, FANinBOX will turn the fan off. The count will stop if the object “[CH1] Timed Off” receive the deactivation value.

## 2.2.5 LOCK

This function permits locking the channel when receiving the lock order through a specific one-bit communication object. From that moment, any action being executed by the channel will stop and the orders during will be ignored.

FANinBOX will unlock the channel when receiving the unlock value through the lock object. Any request received during the lock state will not be run by the channel after the unlock event.

It is possible to set the fan speed when locking and when unlocking.

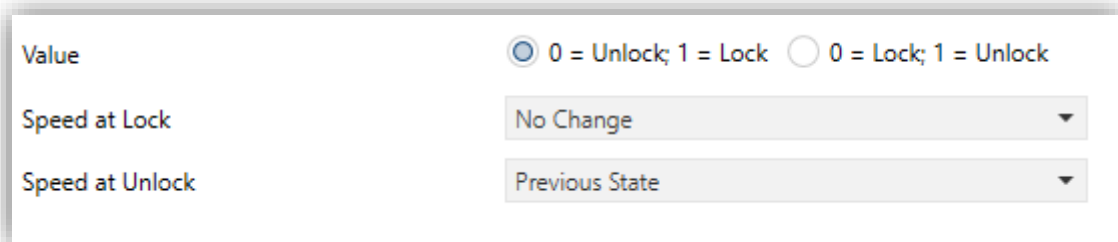
After a power failure, the channel will maintain the lock state and the fan speed level (the **initial configuration** will not apply in this case; see section 2.2.2).

Note that the **Timed Off** (section 2.2.4) function will not be available during the lock.



## ETS PARAMETERISATION

Once the Lock function has been enabled from the “Configuration” tab (see section 2.2.1), a new tab named “Lock” is added to the tab tree on the left, containing the following options:



The screenshot shows a configuration window for the 'Lock' function. It contains three rows of settings:

- Value:** Two radio button options. The first is selected and labeled '0 = Unlock; 1 = Lock'. The second is labeled '0 = Lock; 1 = Unlock'.
- Speed at Lock:** A dropdown menu currently showing 'No Change'.
- Speed at Unlock:** A dropdown menu currently showing 'Previous State'.

Figure 12. Lock

- **Value** [0 = Unlock; 1 = Lock / 0 = Lock; 1 = Unlock]: sets the polarity of the object “[CH1] Lock” which will active or deactivate the channel lock.
- **Speed at Lock** [No Change / Off / Level 1 / Level 2 / Level 3 / Level 4]: sets fan speed to be applied when a lock order is received.
- **Speed at Unlock** [Previous State / Off / Level 1 / Level 2 / Level 3 / Level 4]: sets fan speed to be applied when an unlock order is received.

## 2.3 LOGIC FUNCTIONS

---

This module makes it possible to perform numeric and binary operations to incoming values received from the KNX bus, and to send the results through other communication objects specifically enabled for this purpose.

FANinBOX can implement **up to 10 different and independent functions**, each of them entirely customisable and consisting in **up to 4 consecutive operations each**.

The execution of each function can depend on a configurable **condition**, which will be evaluated every time the function is **triggered** through specific, parameterisable communication objects. The result after executing the operations of the function can also be evaluated according to certain **conditions** and afterwards sent (or not) to the KNX bus, which can be done every time the function is executed, periodically or only when the result differs from the last one.

Please refer to the specific user manual “**Logic Functions**” (available in the FANinBOX product section at the Zennio homepage, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameter.

## 2.4 MANUAL CONTROL

---

The output channel of FANinBOX can be manually operated and verified by means of the **on-board pushbuttons and LEDs** (two per channel), located on the top side of the device.



Manual operation can be done in two different ways, named as **Test On Mode** (for testing purposes during the configuration of the device) and **Test Off Mode** (for a normal use, anytime). Whether both, only one, or none of these modes are available needs to be parameterised in ETS. Moreover, it is possible to enable a specific binary object for locking and unlocking the manual control in runtime.

### **Note:**

- *The **Test Off mode** will be active (unless it has been disabled by parameter) after a download or a reset with no need of a specific activation – the pushbuttons will respond to user presses from the start.*
- *On the contrary, switching to the **Test On mode** (unless disabled by parameter) needs to be done by long-pressing the Prog/Test button (for at least three seconds), until the LED is no longer red and turns yellow. From that moment, once the button is released, the LED light will remain green to confirm that the device has switched from the Test Off mode to the Test On mode. After that, an additional press will turn the LED yellow and then off, once the button is released. This way, the device leaves the Test On mode. Note that it will also leave this mode if a bus power failure takes place.*

### **Test Off Mode**

Under the Test Off Mode, the fan can be controlled through both their communication objects and the actual pushbuttons located on the top of the device.

When one of these buttons is pressed, the fan will behave as if an order had been received through the communication object “[CH1] Speed: 1-Bit Control”, and will also send the status objects when required. Thus, each press on the left button  increases the fan speed by one level and each press on the right button  decreases it.

Regarding the lock, timed off and scene functions, the device will behave under the Test Off mode as usual. Button presses during this mode are entirely analogous to the reception of the corresponding orders from the KNX bus.

## Test On Mode

After entering the Test On mode, it will only be possible to control the fan through the on-board pushbuttons. Orders received through communication objects will be ignored.

Under the Test On Mode, the fan will react to button presses in the same way as for the Test Off Mode.

The lock, timed off and scene functions as well as any orders received from the KNX bus will not have an effect over the fan status while the device is under the Test On mode. The status objects will not be sent to the bus, either.

On the contrary, lock objects will be re-evaluated after leaving the Test On mode, so any changes that may have taken place in Test On will be considered when leaving.

Note that, for safety reasons, the Test On Mode will not be available in the event of a power failure. On the other hand, if that error is detected during the Test On Mode, the device will automatically leave this mode.

**Important:** *the device is delivered from factory with both manual modes (Test Off and Test On) enabled by default.*

---

## ETS PARAMETERISATION

---

The **Manual Control** is configured from the General configuration tab (see section 2.1).

Manual Control	Test Off Mode + Test On Mode
Manual Control Lock	<input checked="" type="checkbox"/>
Value	<input type="radio"/> 0 = Lock; 1 = Unlock <input checked="" type="radio"/> 0 = Unlock; 1 = Lock
Initialization	Last Value

Figure 13. Manual Control

- **Manual Control** [*Disabled / Only Test Off Mode / Only Test On Mode / [Test Off Mode + Test On Mode](#)*]. Depending on the selection, the device will permit using the manual control under the Test Off, the Test On, or both modes. Note that, as stated before, using the Test Off mode does not require any special action, while switching to the Test On mode does require long-pressing the Prog/Test button.
  
- **Manual Control Lock** [*enabled/disabled*]: unless the above parameter has been “Disabled”, the Lock Manual Control parameter provides an optional procedure for locking the manual control in runtime. When this checkbox is enabled, object “**Manual Control Lock**” turns visible, as well as two more parameters:
  - **Value** [*0 = Lock; 1 = Unlock / [0 = Unlock; 1 = Lock](#)*]: defines whether the manual control lock/unlock should take place respectively upon the reception (through the aforementioned object) of values “0” and “1”, or the opposite.
  
  - **Initialization** [*Unlocked / Locked / [Last Value](#)*]: sets how the lock state of the manual control should remain after the device start-up (after an ETS download or a bus power failure). “Last Value” (default; on the very first start-up, this will be Unlocked).

## ANNEX I: COMMUNICATION OBJECTS

- “Functional range” shows the values that, with independence of any other values permitted by the bus according to the object size, may be of any use or have a particular meaning because of the specifications or restrictions from both the KNX standard or the application program itself.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	Name	Function
1	1 Bit		C--T-	DPT_Trigger	0/1	[Heartbeat] Object to Send '1'	Sending of '1' Periodically
2	1 Bit		C--T-	DPT_Trigger	0/1	[Heartbeat] Device Recovery	Send 0
3	1 Bit		C--T-	DPT_Trigger	0/1	[Heartbeat] Device Recovery	Send 1
4	1 Bit	I	C-W--	DPT_Enable	0/1	Lock Manual Control	0 = Unlock; 1 = Lock
	1 Bit	I	C-W--	DPT_Enable	0/1	Lock Manual Control	0 = Lock; 1 = Unlock
5	1 Bit	O	CR-T-	DPT_Alarm	0/1	Power Supply Failure	0 = No error; 1 = Error
6, 25, 44, 63	1 Byte	I	C-W--	DPT_SceneControl	0-63; 128-191	[CHx] Scene	0 - 63 (Run 1 - 64); 128 - 191 (Save 1 - 64)
7, 26, 45, 64	1 Bit	I	C-W--	DPT_Switch	0/1	[CHx] On/Off	0 = Off; 1 = On
8, 27, 46, 65	1 Bit	O	CR-T-	DPT_Switch	0/1	[CHx] On/Off (Status)	0 = Off; 1 = On
9, 28, 47, 66	1 Byte	I	C-W--	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control	0% = Off; [0.2...33.5]% = Level 1; [33.6...66.8]% = Level 2; [66.9...100]% = Level 3
	1 Byte	I	C-W--	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control	0% = Off; [0.2...50.3]% = Level 1; [50.4...100]% = Level 2
	1 Byte	I	C-W--	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control	0% = Off; [0.2...100]% = Level 1
	1 Byte	I	C-W--	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control	0% = Off; [0.2...25.2]% = Level 1; [25.3...50.3]% = Level 2; [50.4...75.4]% = Level 3; [75.5...100]% = Level 4
10, 29, 48, 67	1 Byte	O	CR-T-	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control (Status)	Off = 0%; Level 1 = 33%; Level 2 = 67%; Level 3 = 100%
	1 Byte	O	CR-T-	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control (Status)	Off = 0%; Level 1 = 50%; Level 2 = 100%
	1 Byte	O	CR-T-	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control (Status)	Off = 0%; Level 1 = 100%
	1 Byte	O	CR-T-	DPT_Scaling	0% - 100%	[CHx] Speed: Percentage Control (Status)	Off = 0%; Level 1 = 25%; Level 2 = 50%; Level 3 = 75%; Level 4 =

							100%
11, 30, 49, 68	1 Bit	I	<b>C - W - -</b>	DPT_Switch	0/1	[CHx] Speed: Level 1	0 = Off; 1 = On
12, 31, 50, 69	1 Bit	O	<b>CR - T -</b>	DPT_Switch	0/1	[CHx] Speed: Level 1 (Status)	0 = Off; 1 = On
13, 32, 51, 70	1 Bit	I	<b>C - W - -</b>	DPT_Switch	0/1	[CHx] Speed: Level 2	0 = Off; 1 = On
14, 33, 52, 71	1 Bit	O	<b>CR - T -</b>	DPT_Switch	0/1	[CHx] Speed: Level 2 (Status)	0 = Off; 1 = On
15, 34, 53, 72	1 Bit	I	<b>C - W - -</b>	DPT_Switch	0/1	[CHx] Speed: Level 3	0 = Off; 1 = On
16, 35, 54, 73	1 Bit	O	<b>CR - T -</b>	DPT_Switch	0/1	[CHx] Speed: Level 3 (Status)	0 = Off; 1 = On
17, 36, 55, 74	1 Bit	I	<b>C - W - -</b>	DPT_Switch	0/1	[CHx] Speed: Level 4	0 = Off; 1 = On
18, 37, 56, 75	1 Bit	O	<b>CR - T -</b>	DPT_Switch	0/1	[CHx] Speed: Level 4 (Status)	0 = Off; 1 = On
19, 38, 57, 76	1 Bit	I	<b>C - W - -</b>	DPT_Step	0/1	[CHx] Speed: 1-Bit Control	0 = Decrease; 1 = Increase
20, 39, 58, 77	1 Bit	I	<b>C - W - -</b>	DPT_Start	0/1	[CHx] Timed Off	0 = Deactivate; 1 = Activate
	1 Bit	I	<b>C - W - -</b>	DPT_Start	0/1	[CHx] Timed Off	0 = Activate; 1 = Deactivate
21, 40, 59, 78	2 Bytes	I	<b>C - W - -</b>	DPT_TimePeriodSec	0 - 65535	[CHx] Timed Off: Delay	[10...65535]s
22, 41, 60, 79	1 Bit	I	<b>C - W - -</b>	DPT_Enable	0/1	[CHx] Lock	0 = Unlock; 1 = Lock
	1 Bit	I	<b>C - W - -</b>	DPT_Enable	0/1	[CHx] Lock	0 = Lock; 1 = Unlock
23, 42, 61, 80	4 Bytes	I/O	<b>CRWT -</b>	DPT_LongDeltaTimeSec	-2147483648 - 2147483647	[CHx] Operating Time (s)	Time in Seconds
24, 43, 62, 81	2 Bytes	I/O	<b>CRWT -</b>	DPT_TimePeriodHrs	0 - 65535	[CHx] Operating Time (h)	Time in Hours
82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104	4 Bytes	O	<b>CR - T -</b>	DPT_Value_4_Ucount	0 - 4294967295	[Relay x] Number of Switches	Number of Switches
83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105	2 Bytes	O	<b>CR - T -</b>	DPT_Value_2_Ucount	0 - 65535	[Relay x] Maximum Switches per Minute	Maximum Switches per Minute
106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137	1 Bit	I	<b>C - W - -</b>	DPT_Bool	0/1	[LF] (1-Bit) Data Entry x	Binary Data Entry (0/1)
138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153	1 Byte	I	<b>C - W - -</b>	DPT_Value_1_Ucount	0 - 255	[LF] (1-Byte) Data Entry x	1-Byte Data Entry (0-255)
154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169	2 Bytes	I	<b>C - W - -</b>	DPT_Value_2_Ucount	0 - 65535	[LF] (2-Byte) Data Entry x	2-Byte Data Entry
170, 171, 172, 173, 174, 175, 176, 177	4 Bytes	I	<b>C - W - -</b>	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] (4-Byte) Data Entry x	4-Byte Data Entry
178, 179, 180, 181, 182, 183, 184, 185, 186, 187	1 Bit	O	<b>CR - T -</b>	DPT_Bool	0/1	[LF] Function x - Result	(1-Bit) Boolean
	1 Byte	O	<b>CR - T -</b>	DPT_Value_1_Ucount	0 - 255	[LF] Function x - Result	(1-Byte) Unsigned
	2 Bytes	O	<b>CR - T -</b>	DPT_Value_2_Ucount	0 - 65535	[LF] Function x - Result	(2-Byte) Unsigned

	4 Bytes	O	<b>CR-T-</b>	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] Function x - Result	(4-Byte) Signed
	1 Byte	O	<b>CR-T-</b>	DPT_Scaling	0% - 100%	[LF] Function x - Result	(1-Byte) Percentage
	2 Bytes	O	<b>CR-T-</b>	DPT_Value_2_Count	-32768 - 32767	[LF] Function x - Result	(2-Byte) Signed
	2 Bytes	O	<b>CR-T-</b>	9.xxx	-671088.64 - 670433.28	[LF] Function x - Result	(2-Byte) Float





Join and send us your inquiries  
about Zennio devices:

<https://support.zennio.com>

**Zennio Avance y Tecnología S.L.**  
C/ Río Jarama, 132. Nave P-8.11  
45007 Toledo (Spain).

*Tel. +34 925 232 002.*

*www.zennio.com*  
*info@zennio.com*