



MAXinBOX FC 0-10V FAN

**2x Two/Four-Pipe Fan Coil Controller
with 0-10 VDC Fan Control Signal**

ZCL-FC010F

Application Program Version: [1.8]

User Manual Version: [1.8]_a

CONTENTS

Contents	2
Document updates.....	3
1 Introduction	5
1.1 MAXinBOX FC 0-10V FAN	5
1.2 Installation.....	6
1.3 Start-up and Power Loss.....	7
2 Configuration	8
2.1 General.....	8
2.2 Inputs.....	10
2.2.1 Binary Input	10
2.2.2 Temperature Probe	10
2.2.3 Motion Detector	10
2.3 Binary Outputs.....	11
2.4 0-10V Analogue Outputs	12
2.5 Fan Coils.....	15
2.5.1 Fan Coil x.....	15
2.5.1.1 General	15
2.5.1.2 Advanced Settings	18
2.6 Logic Functions.....	23
2.7 Thermostats	23
2.8 Manual control.....	24
2.9 Master Light	28
Annex I: Independence of the modules	31
Annex II: Operation Examples	33
Annex III: Communication objects	35

DOCUMENT UPDATES

Version	Changes	Page(s)
[1.8]_a	Changes in the application program: <ul style="list-style-type: none"> Change of DPT objects [FCx] Manual Fan Control and [FCx] Manual Fan Control (Status) to 1.003 Enable. Internal optimisation. 	-
[1.7]_a	Changes in the application program: <ul style="list-style-type: none"> Optimisation of modules: Master Light, Binary Inputs, Logic Functions, Motion Detector, Thermostat and Temperature Probe. 	-
[1.6]_a	Changes in the application program: <ul style="list-style-type: none"> Internal optimisation. 	-
[1.5]_a	Changes in the application program: <ul style="list-style-type: none"> Optimisation of modules: Heartbeat, Master Light, Binary Inputs, Individual Digital Outputs, Logic Functions, Motion Detector, Thermostat and Temperature Probe. Support for long messages in ETS download. 	-
[1.4]_a	Changes in the application program: <ul style="list-style-type: none"> Heartbeat functionality added. Master Light functionality added. Fan speed manual control. Optimisation of the Fan Coil, Binary Inputs, Individual Digital Outputs, Analog Outputs, Logic Functions, Motion Detector and Thermostat and Temperature Probe modules. 	-
[1.3]_a	Changes in the application program: <ul style="list-style-type: none"> Optimisation of the motion detector module. 	-

[1.2]_a	Changes in the application program: <ul style="list-style-type: none">• Update of the motion detector function.	-
[1.1]_a	Changes in the application program: <ul style="list-style-type: none">• Optimisation of the timed actions management in the logic functions module.	-

1 INTRODUCTION

1.1 MAXinBOX FC 0-10V FAN

MAXinBOX FC 0-10V FAN from Zennio is a versatile KNX multi-function actuator that aims at fulfilling the climate control needs of KNX environments with integrated fan coil units where the fan speed is controlled through an **analogue 0-10 VDC signal**, while the gates of the water pipes need to be controlled through **binary outputs** (relays).

The actuator provides two analogue outputs and four binary outputs, all of which can be **enabled and configured independently**, which lets the integrator combine them as desired to control up to 2 two-pipe or four-pipe fan coil units.

At a glance, the most outstanding features of MAXinBOX FC 0-10V FAN are:

- **2 independent analogue 0-10 VDC outputs,**
- **4 independent binary (relay) outputs,**
- **2 independent fan coil functions,** to control up to two fan coil units by means of the above analogue and binary outputs.
- **4 multi-purpose inputs,** each of them configurable as:
 - Temperature probe,
 - Binary input (i.e., pushbuttons, switches, sensors),
 - Motion detector.
- **10 customisable, multi-operation logic functions.**
- **2 independent thermostats.**
- **Manual operation / supervision** of the relay outputs and the 0-10 VDC signals through the on-board pushbuttons and LEDs.

- **Master light control** for an easy, out-of-the-box control of a set of luminaires (or functionally equivalent devices) one of which acts as a general lamp and the others as secondary lamps.
- **Heartbeat** or periodical “still-alive” notification.

The MAXinBOX FC 0-10V FAN application program focuses on the control of fan coil drives that consist of two or four pipes (each with its own open/close valve) and a fan system whose speed is controlled through a 0 to 10 VDC analogue signal (more voltage means more speed).

1.2 INSTALLATION

MAXinBOX FC 0-10V FAN 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 may be downloaded.

This device does not need any additional external power since it is entirely powered through the KNX bus.

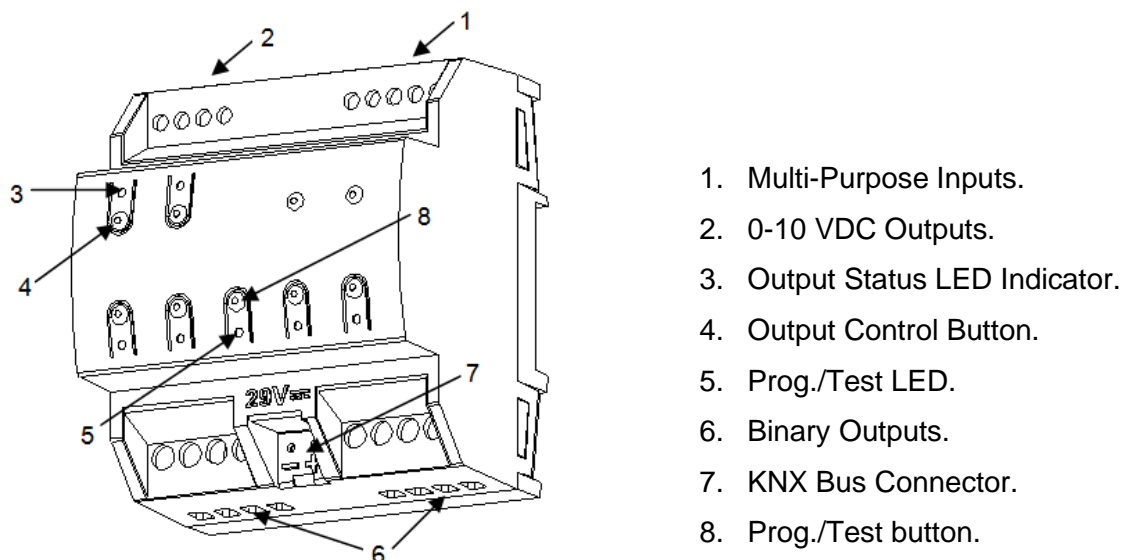


Figure 1 MAXinBOX FC 0-10V.

The main elements of the device are described next.

- **Test/Prog. pushbutton (8):** a short press on this button sets the device into the programming mode, making the associated LED (5) light in red.

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

- **Outputs (2) and (6):** output ports for the insertion of the stripped cables of the systems being controlled by the actuator (see sections 2.3 and 2.4). Please secure the connection by means of the on-board screws.

For detailed information about the technical features of the device and for safety instructions and on the installation process, please refer to the **Datasheet** bundled with the original packaging of the device. Also available at www.zennio.com.

1.3 START-UP AND POWER LOSS

During the start-up of the device, the Test/Prog. LED will blink in blue colour for a few seconds before MAXinBOX FC 0-10V FAN is ready. External orders will not be executed during this time, but afterwards.

Depending on the configuration, some specific actions will also be performed during the start-up. For example, the integrator can set whether the outputs should switch to a particular state and whether the device should send certain objects to the bus after the power recovery. Please consult the next sections of this document for further details.

On the other hand, when a bus power failure takes place, MAXinBOX FC 0-10V FAN will interrupt any pending actions and will save its state so it can be recovered once the power supply is restored.

2 CONFIGURATION

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.

2.1 GENERAL

The first parameterisable screen available by default is “General”. From this screen it is possible to activate/deactivate all the required functionality.

ETS PARAMETERISATION

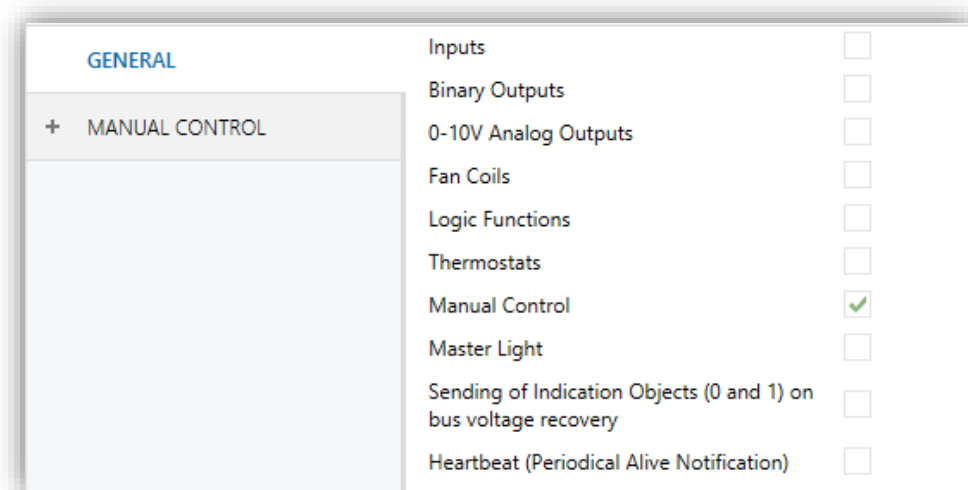
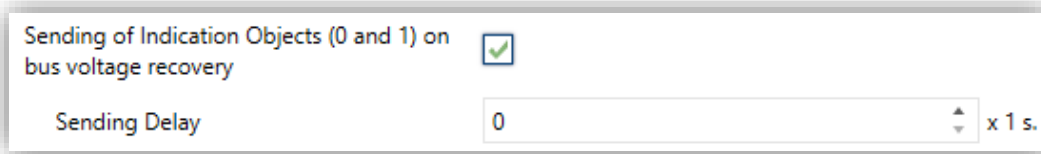


Figure 2 General screen.

- Once activated, **Inputs, Binary Outputs, 0-10V Analog Outputs, Fan Coils, Logical Functions, Thermostats, Manual Control** (enabled by default) and **Master Light** bring additional tabs to the menu on the left. These functions and their parameters will be explained in later sections of this document.
- **Sending of Indication Objects (0 and 1) on Bus Voltage Recovery**: this parameter lets the integrator activate two new communication objects (“**Reset 0**” and “**Reset 1**”), which will be sent to the KNX bus with values “0” and “1” respectively whenever the device begins operation (for example, after a bus

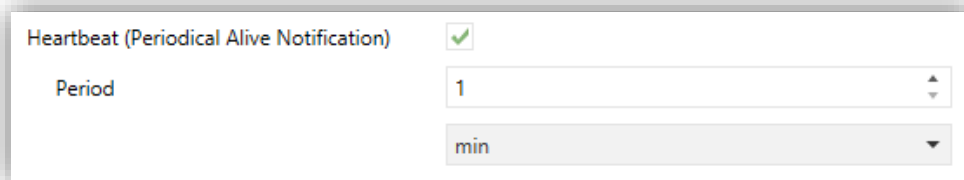
power failure). It is possible to parameterise a certain **delay** for this sending (0 to 255 seconds).



The screenshot shows a configuration window with the title "Sending of Indication Objects (0 and 1) on bus voltage recovery". There is a checked checkbox to the right of the title. Below the title, there is a label "Sending Delay" followed by a numeric input field containing the value "0" and a unit selector dropdown menu set to "x 1 s".

Figure 3. Sending of Indication objects on bus voltage recovery

- **Heartbeat (Periodical Alive Notification):** this parameter lets the integrator incorporate a one-bit object to the project (“[Heartbeat] Object to Send ‘1’”) that will be sent periodically with value “1” to notify that the device is still working (*still alive*).



The screenshot shows a configuration window with the title "Heartbeat (Periodical Alive Notification)". There is a checked checkbox to the right of the title. Below the title, there is a label "Period" followed by a numeric input field containing the value "1" and a unit selector dropdown menu set to "min".

Figure 4. 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.*

2.2 INPUTS

MAXinBOX FC 0-10V FAN incorporates **two analogue/digital inputs**, each configurable as a:

- **Binary Input**, for the connection of a pushbutton or a switch/sensor.
- **Temperature Probe**, to connect a temperature sensor from Zennio.
- **Motion Detector**, to connect a motion detector from Zennio.

2.2.1 BINARY INPUT

Please refer to the “**Binary Inputs**” user manual, available under the MAXinBOX FC 0-10V FAN product section at www.zennio.com.

2.2.2 TEMPERATURE PROBE

Please refer to the “**Temperature Probe**” user manual, available under the MAXinBOX FC 0-10V FAN product section at www.zennio.com.

Note: *This device does not include support for custom NTC probes.*

2.2.3 MOTION DETECTOR

It is possible to connect motion detectors from Zennio to the input ports of MAXinBOX FC 0-10V FAN. This brings the device with the possibility of monitoring motion and presence in the room, as well as the light level. Depending on the detection, different response actions can be parameterised.

Please refer to the “**Motion Detector**” user manual, available under the MAXinBOX FC 0-10V FAN product section at www.zennio.com, for detailed information about the functionality and the configuration of the related parameters.

2.3 BINARY OUTPUTS

MAXinBOX FC 0-10V FAN incorporates **four binary outputs**, each of which can be enabled and configured in parameters independently.

Although in this device they are offered with the aim of controlling the **gate valves** of the fan coil tubes (up to four), their parameterisation is similar to that of the individual relay outputs of any other MAXinBOX actuators. Note that they all work **independently**, although it is possible to group them as required (for example, to pen a gate valve when another one closes) through joint group addresses and to configure each of them as “normally open” or “normally closed”, to open / close its gate valve on the reception of a “0” or of a “1”.

Practical usage examples of the different outputs and modules all together are offered in “**Annex I: Independence of the modules**” of this user manual.

Please refer to the specific manual “**Binary Outputs in MAXinBOX FC 0-10V FAN**” (available at the Zennio homepage, www.zennio.com) for detailed information.

2.4 0-10V ANALOGUE OUTPUTS

MAXinBOX FC 0-10V FAN incorporates **two analogue voltage outputs** that provide a voltage signal in the range 0 to 10 VDC. This voltage will be in accordance to the percentage values that are received through a specific communication object.

Each analogue output can be enabled or disabled in parameters, and is intended to control the fan speed of one fan coil unit.

One on-board **LED indicator** is offered per output to show the status: it will remain off while the output signal is 0V, and on while it is 10V. Under intermediate values, it will blink with variable rates (according to the voltage).

ETS PARAMETERISATION

After enabling “**0-10V Analog Outputs**” in the General screen (see section 2.1), a new tab will be incorporated into the tree on the left.

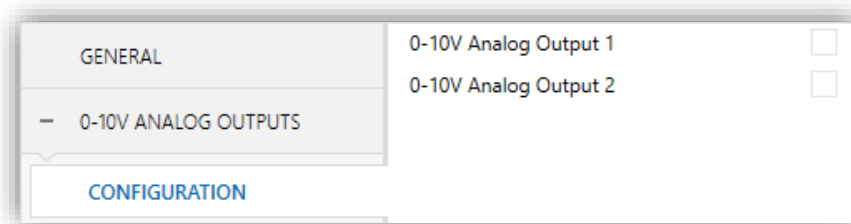


Figure 5 0-10V Analog Outputs - Configuration.

The two analogue outputs can be activated independently by marking the checkboxes. This also brings new tabs to the tree.

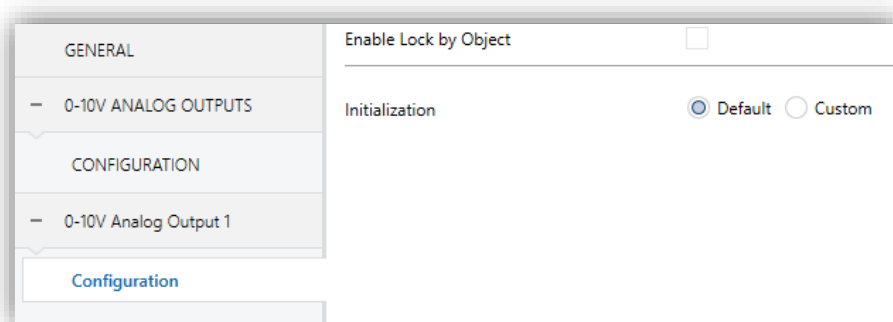


Figure 6 0-10V Analog Output X – Configuration.

After enabling any of them, two objects are included by default:

- “[AOx] Output Value (Control)”: this object is supposed to receive percentage values from the KNX bus, making the device generate a voltage output between 0 V and 10 V (proportional to the percentage value).
- “[AOx] Output Value (Status)”: status object that shows, in terms of percentage, the current value of the analogue output signal. This object is automatically sent to the bus after receiving a new voltage setpoint, and when the output status changes due to a lock order.

The following parameters can be configured from the corresponding tab:

Enable Lock by Object	<input checked="" type="checkbox"/>
Lock Action	On
Output Value	100 %
<hr/>	
Initialization	<input type="radio"/> Default <input checked="" type="radio"/> Custom
Initial Status	On
Output Value	100 %
Send Status	<input checked="" type="checkbox"/>
Delay (0 = No Delay)	3
	s

Figure 7 0-10V Analog Output X – Configuration (in detail).

- **Enable Lock by Object:** when this checkbox is marked, the one-bit object “[AOx] Lock” turns visible, as well as one more parameter:
 - **Lock Action:** sets the particular state (“No Change” / “On” / “Off”) the output will adopt upon the reception of the value “1” through the “[AOx] Lock” object. If set to “On”, the desired **Output Value** needs to be configured, in terms of percentage.

Note: while an output is locked, further voltage setpoints are ignored (the device will respond with the current status of the output).

- **Initialization:** brings the option to set the output to a particular state at the start-up of the actuator.
 - **Default:** off after an ETS download, and unchanged after a bus power failure.
 - **Custom:** this option brings two new parameters:
 - **Initial Status:** “Previous”, “On”, “Off”, after either an ETS download or a bus power failure (*last* will be *off* on the very first start-up). When selecting “On”, the desired **Output Value** needs to be configured, in terms of percentage.
 - **Send Status:** when marked, the status object will be sent to the bus with a customisable **Delay**, from 0 to 600 ds, 0 to 3600 s, 0 to 1440 min or 0 to 24 hours (3 seconds by default).

Note: *the lock status is preserved after a bus failure. In case of conflict between the state defined for the lock function and the state defined for the initialization, the **lock state prevails**. That is, if a bus failure occurs with an output locked, after the bus recovery the output will have the same value, independently of the Initial status defined in the Initialization function.*

2.5 FAN COILS

MAXinBOX FC 0-10V FAN incorporates **two independent fan coil** functions that implement all the logic involved in the control of up to two fan coil units.

ETS PARAMETERISATION

After enabling “**Fan Coils**” in the General screen (see section 2.1), a new tab will be incorporated into the tree on the left.

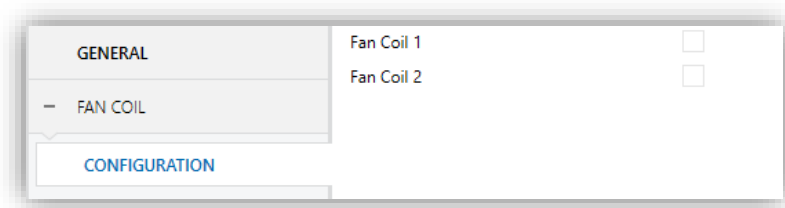


Figure 8 Fan Coils - Configuration.

The two fan coil functions can be activated independently by marking their checkboxes. This brings some more tabs to the tree.

2.5.1 FAN COIL x

Each fan coil function requires setting some general parameters and, optionally, some advanced parameters. All of them are described next.

2.5.1.1 GENERAL

Among other options regarding the fan coil operation modes and the values and delays involved in the opening and closing of the valves, the general parameters bring the option to set two offsets:

- The **minimum value of the control variable** that should activate the fan, i.e., the minimum percentage value that will be required to make the actuator actually operate the fan; values under that minimum will be ignored by the actuator. This is called “Offset 1” in Figure 9.
- The **minimum fan speed**, i.e., the minimum voltage (in percentage) that permits appreciating motion in the fan. This parameter is useful when the fan

is not able to effectively start motion with voltage values lower than a certain percentage. This is called “Offset 2” in Figure 9.

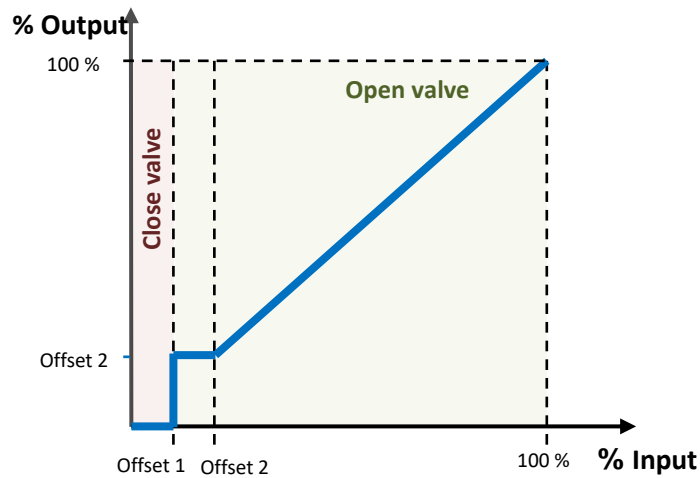


Figure 9 Output percentage value depending on the input setpoint and the offsets.

ETS PARAMETERISATION

The Configuration screen (see Figure 10) is accessible by default after enabling the Fan Coil function. It contains the following parameters:

GENERAL	Type of Fan Coil	<input checked="" type="radio"/> 2 pipes <input type="radio"/> 4 pipes
- FAN COIL	Mode	Heating
CONFIGURATION	Value Sent to Open/Close Valve	<input type="radio"/> 0 = Open Valve; 1 = Close Valve <input checked="" type="radio"/> 0 = Close Valve; 1 = Open Valve
- Fan Coil 1	Offsets Applied to the Control (set both to 0 to disable)	
Configuration	Minimum Control Variable Value that Activates the Fan	10 %
	Minimum Fan Speed (it must be higher or equal than the minimum control)	20 %
	HEATING	
	Delay for Fan Activation when Valve Opens (0 = No Delay)	0 s
	Delay for Fan Deactivation when Valve Closes (0 = No Delay)	0 s
	Advanced Settings	<input type="checkbox"/>

Figure 10 Fan Coil X – Configuration.

- **Type of Fan Coil:** sets the type of the fan coil. The options are: “2 pipes” and “4 pipes”. Under “2 pipes”, it is necessary to set the operating Mode:
- **Mode:** sets the operating mode of the fan coil system. The options are: “Cooling”, “Heating” or “Both”. This parameter is only available for two-pipe fan coils (in the case of four-pipe fan coils, both modes are always available).

Depending on the value of Type and Mode and according to Table 1, either one set of objects or another will be available.

Object	2 tubes			4 tubes
	Cooling	Heating	Both	
[FCx] Mode and Mode (Status)			x	x
[FCx] Control Variable (Cooling)	x		x	x
[FCx] Control Variable (Heating)		x	x	x
[FCx] Valve Control	x	x	x	
[FCx] Valve Control (Cooling)				x
[FCx] Valve Control (Heating)				x

Table 1 Fan Coil communication objects depending on the Type and Mode.

Note: the values received through “[FCx] Control Variable (Cooling)” will only have effect if Cooling is the current mode, while the values received through “[FCx] Control Variable (Heating)” will not be applied, although the will be taken into account when the mode changes to Heating. The same happens for the opposite case.

Please refer to “**Annex II: Operation Examples**” for some operation examples where these objects are received in different scenarios.

- **Value Sent to Open/Close Valve:** sets the values required to open and close the valve. The options are: “0 = Close Valve; 1 = Open Valve” (by default) and “0 = Open Valve; 1 = Close Valve”.

The following parameters define two offset values applied to fan speed control:

- **Minimum Control Variable Value that Activates the Fan:** from 0 to 100% (10% by default).

- **Minimum Fan Speed:** from 0 to 100% (20% by default). This value must be higher or equal than the Minimum Control.

Note: *both offsets must be lower than the Maximum Fan Speed parameter (see 2.5.1.2).*

Depending on the mode, the following delays must be configured:

- **Delay for Fan Activation when Valve Opens:** time to wait since the valve opens until the fan can be turned on. This delay must be configured for both Cooling and Heating. The allowed values are: 0 to 600 ds, 0 to 3600 s, 0 to 1440 min or 0 to 24 hours.
- **Delay for Fan Deactivation when Valve Closes:** time to wait since the valve closes until the fan should be turned off. This delay must be configured for both Cooling and Heating. The allowed values are: 0 to 600 ds, 0 to 3600 s, 0 to 1440 min or 0 to 24 hours.
- **Minimum Delay** for the mode change: minimum time the device should wait between the order to close the valve (under the old mode) and the order to open the valve (in the new mode). The allowed values are: 0 to 600 ds, 0 to 3600 s, 0 to 1440 min or 0 to 24 hours

This parameter is only visible when the two modes are available (i.e., with a two-pipe fan coil with both modes, or with a four-pipe fan coil).

2.5.1.2 ADVANCED SETTINGS

Among other options, in the advanced settings can be set if the **fan coil is always on** or not.

Moreover, it is possible to define a **maximum fan speed level**, which is set in terms of percentage. If set, any fan regulation order greater than this value will lead the output to the maximum, as shown in Figure 11.

One more advanced feature is the option to **keep the fan in motion after the valve is closed** (i.e., when the setpoint is under the minimum value that activates the fan, or Offset 1). This option is applied only in the cooling mode. When enabled, the desired constant fan speed must be set by parameter (this fan speed can be changed at runtime when required). Figure 12 illustrates this behaviour.

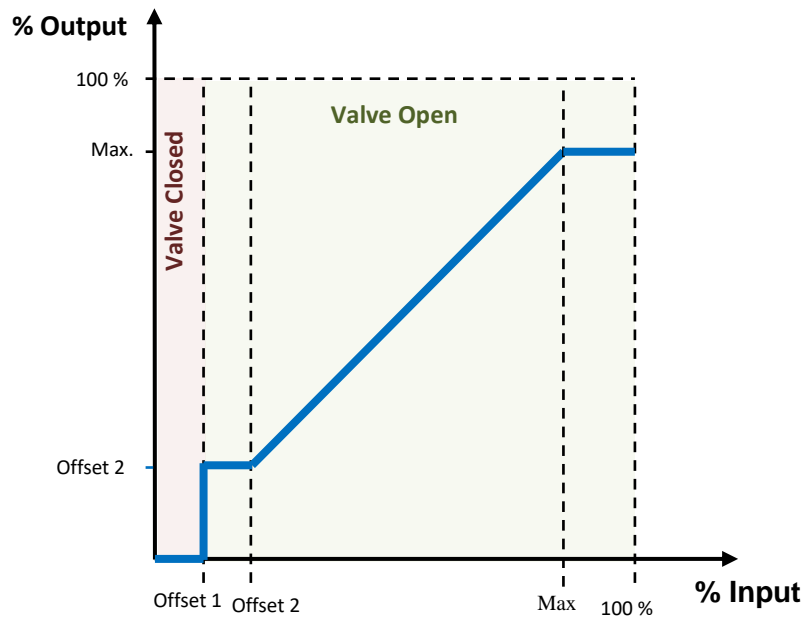


Figure 11 Output percentage depending on the input setpoint, the two offsets and the maximum.

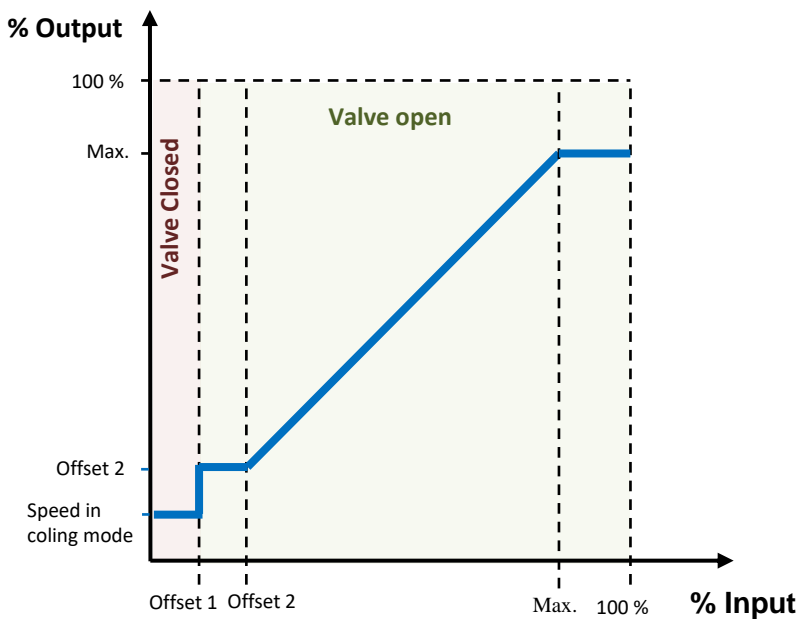


Figure 12 Output percentage depending on the input setpoint, the two offsets and the fan speed with the valve closed.

It is possible to define a **forced position** in parameters, which allows setting (on request through the corresponding object) the fan speed to a pre-set value (with no action on the valves). Under the forced position mode:

- The **valves** remain in the state they were before the forced position order.

- Any control order is **ignored**, but taken into account after leaving this mode.
- The forced position resumes after a **power failure**.
- An **Off order** will be performed as usual, and leads to leaving the forced position mode.
- Upon the reception of **On order**, the fan control order will be sent again, as well as the On status, but there will be no action on the valves.
- The pre-set output value for this mode does not depend on the **offsets**, being up to the integrator whether to set a value greater than Offset 1.
- **Mode switch orders** are not executed until the forced position mode is left, although they are responded by sending the mode status object.
- If the **fan coil module is off**, forced position orders are ignored.

Finally, it is possible to **enable fan speed manual control** during the normal operation and when the valve is closed in cooling mode. There are the following objects related to this functionality: enabling / disabling manual control object and its status and the manual control variable object (which will be applied to the fan speed).

The manual control orders will be ignored if:

- The fan coil is off.
- The fan coil is on but the valve is closed.

There is an exception in this case: if the option to **keep the fan activated in cooling mode when valve closes** is enabled, the manual control orders will be applied, instead of the value set by parameter (see Figure 12, "Speed in cooling mode").

- Force position is active.
- The control orders are sent during any delay.

Furthermore, it is possible to establish the **duration of manual control**, i.e., the time to reactivate the automatic mode. This time is reset when a new control order is received, when turning the fan coil off and on or after a bus failure.

ETS PARAMETERISATION

After enabling “**Advanced Settings**” in the Configuration screen of the fan coil function, a new tab will be incorporated into the tree on the left.

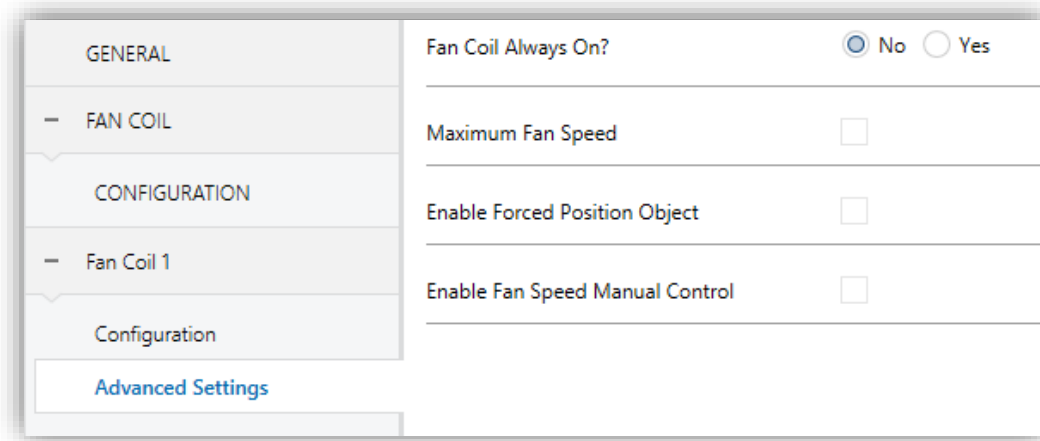


Figure 13 Fan Coil X – Advanced Settings.

The following parameters can be configured:

- **Fan Coil Always On?:** sets whether the fan coil will always be on or not.
 - If this option is set to “No” (by default), the fan coil turns on/off when receiving a “1” or “0” through the object “[FCx] On/Off”, respectively. The state of the fan coil can be obtained by reading the “[FCx] On/Off (Status)” communication object any time.
 - If this option is set to “Yes”, the Fan Coil remains always on, waiting for regulation orders. The objects “[FCx] On/Off” and “[FCx] On/Off (Status)” are not available.

Please refer to section “Annex II: Operation Examples” for some operation examples where these objects are received in different scenarios.

- **Maximum Fan Speed:** when enabled, sets a maximum value for the fan speed through the following parameter:
 - **Maximum:** from 0 to 100% (by default).

- **Enable Forced Position Object:** when enabled, a new communication object (“**[FCx] Forced Position**”) becomes available. When it receives the value “1” and the system is on, the system enters the forced position mode, and the fan speed is set to the value set by the following parameter:
 - **Cooling/Heating Fan Speed:** from 0 (by default) to 100%.
- **Enable Fan Speed Manual Control:** three new objects will be available; the enabling object and his status: “**[FCx] Manual Fan Control**”, “**[FCx] Manual Fan Control (State)**” and “**[FCx] Manual Fan Control Variable**” for the control orders from 0 to 100%.

The manual control is activated when sending enable through the object “**[FCx] Manual Fan Control**” or when sending a control order through the object “**[FCx] Manual Fan Control Variable**”. The manual control is deactivated when sending disable through the object “**[FCx] Manual Fan Control**” but also when the mode switches or when turning off the fan coil.

- **Return to Automatic Mode after a Period of Time:** when enabled, the following parameter appears:
 - **Duration of Manual Control:** time from the last manual control order to reactivate the automatic mode. Range: from 1 to 1440 minutes, from 1 to 24 hours.
- **Keep Fan Activated in Cooling Mode when Valve Closes:** if enabled, the fan speed will change to a certain parameterisable value whenever the control order received is under Offset 1 (Minimum Control Variable Value that Activates the Fan), instead of being set to 0%. A new communication object is added to change this fan speed at runtime (“**[FCx] Fan Speed when Valve Closed (Cooling)**”).
 - **Fan Speed:** from 0 to 100% (20% by default).

Note: *this option is only available for cooling mode, so it should had been selected in general Configuration (see section 2.5.1.1), por tanto debe haberse seleccionado dicho modo en la pestaña de Configuración general (ver sección 2.5.1.1).*

2.6 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.

MAXinBOX FC 0-10V FAN can implement **up to 20 different and independent functions**, each of them entirely customisable and consisting in **up to 4 consecutive operations each one**.

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 “**Logic Functions**” specific manual available under the MAXinBOX FC 0-10V FAN product section at the Zennio homepage, www.zennio.com for detailed information.

2.7 THERMOSTATS

MAXinBOX FC 0-10V FAN implements **two Zennio Thermostats** which can be independently enabled and fully customised.

Please refer to the “**Zennio Thermostat**” specific manual available under the MAXinBOX FC 0-10V FAN product section at the Zennio homepage, www.zennio.com for detailed information.

2.8 MANUAL CONTROL

MAXinBOX FC 0-10V FAN allows manually switching the state of its binary and 0-10V analogue outputs through the respective pushbuttons on the top of the device. A specific pushbutton is therefore available per output.

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 should be accessible can 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 outputs 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 output will behave as if an order had been received through the corresponding communication object, so it has no effect if the output is locked or under alarm status. The status objects of the different functions will still be sent in the usual way.

The action performed depends on the output type and, in the case of 0-10V analogue outputs, on the press type. For both outputs types, the presses have no effect if the output is disabled in parameters.

- **Binary output:** a simple press (short or long) will make the output switch its on-off state, with independence of the state of the other binary outputs. This will be reported to the KNX bus through the corresponding status object, if enabled.
- **0-10V analogue output:** the action depends on the type of press:
 - **Short press:** it is equivalent to a 0% or 100% regulation order through the “[AOx] Output Value (Control)”, causing a 0V or 10V output signal. If the current state is greater than 0% the regulation order will be 0% (0V), while if the current state is 0% the regulation will be 100% (10V).
 - **Long press:** if the button is long-pressed, the actuator will start increasing or decreasing the fan speed progressively as long as the button is being held. The direction of the regulation with every new long press will be downwards (i.e., a decrease of the fan speed), unless the current state is 0% (in such case, the regulation will consist in an increase). The updated state object will be sent once the button is released, or when the maximum / minimum value is reached. The complete regulation cycle (0% to 100% or vice versa) takes 10 seconds.
- **Disabled output:** outputs disabled in parameters will not react to button presses under the Test Off mode.

Regarding the lock, timer, alarm 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 outputs through the on-board pushbuttons. Orders received through communication objects will be ignored, with independence of the output they are addressed to.

Depending on the type of output, binary or analogue, the reactions to the button presses will differ.

- **Binary output:** short or long-pressing the button will commute the on-off state of the relay.
- **0-10V analogue output:** the behaviour is the same as described for the Test Off Mode, except that the fan status objects do not change (they will be updated when leaving the Test On Mode).
- **Disabled output:** under the Test On mode, disabled outputs will behave as if they were enabled, thus, as described above (depending on whether they are binary or analogue outputs).

The lock, timer, alarm and scene functions will not work while the device is under the Test On mode, but will be taken into account when leaving the Test On Mode. Status objects will not be sent to the bus, either.

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

ETS PARAMETERISATION

The **Manual Control** is configured from a specific tab which can be enabled from the General screen (see section 2.1).

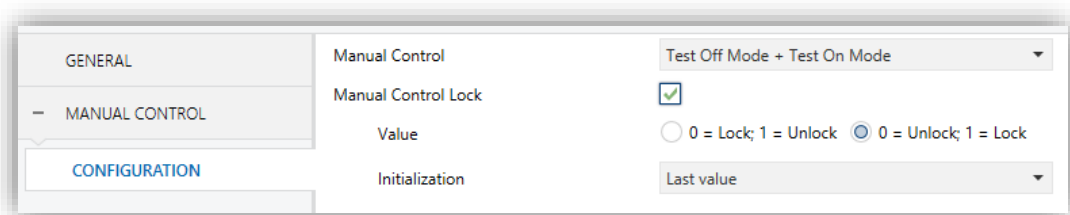


Figure 14 Manual Control.

The only two parameters are:

- **Manual Control:** options are “Disabled”, “Only with Test Off Mode”, “Only with Test On Mode” and “Test Off Mode + Test On Mode” (default). Depending on the selection, the device will permit using the manual control under the Test Off, the Test On, or either of the two 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:** unless the above parameter has been “Disabled”, enabling the Lock Manual Control parameter will provide a runtime procedure for locking the manual control. When this checkbox is enabled, object “**Manual Control Lock**” turns visible, as well as two more parameters:
 - **Value:** defines whether the lock/unlock of the manual control should take place respectively upon the reception (through the aforementioned object) of values “0” and “1”, or the opposite.
 - **Initialization:** sets how the manual control should remain after the device start-up (after an ETS download or a bus power failure): “Unlocked”, “Locked” or, by default, “Last Value” (on the very first start-up, this will be Unlocked).

2.9 MASTER LIGHT

The Master Light function brings the option to monitor the state of up to 12 light sources (or even more, if the Master Light controls from multiple Zennio devices are linked together) or of any other elements whose state is transmitted through a binary object and, depending on those states, perform a **master order** every time a certain trigger signal (again, a binary value) is received through a specific object.

Such master order will consist in:

- A **general switch-off** order, if at least one of the up to twelve status objects is found to be on.
- A **courtesy switch-on** order, if none of the up to twelve status objects is found to be on.

Note that the above switch-off and switch-on orders are not necessarily a binary value being sent to the bus – it is up to the integrator the decision of what to send to the KNX bus in both cases: a shutter order, a thermostat setpoint or mode switch order, a constant value, a scene... Only the trigger object and the twelve status objects are required to be binary (on/off).

The most typical scenario for this Master Light control would be a hotel room with a master pushbutton next to the door. When leaving the room, the guest will have the possibility of pressing on the master pushbutton and make all the lamps turn off together. Afterwards, back on the room and with all the lamps off, pressing on the same master pushbutton will only make a particular lamp turn on (e.g., the closest lamp to the door) – this is the courtesy switch-on.

Besides, it is possible to concatenate two or more Master Light modules by means of a specific communication object which represents the general state of the light sources of each module. Thereby, it is possible to expand the number of light sources by considering the general state of one module as an additional light source for another.

ETS PARAMETERISATION

Once the Master Light function has been enabled, a specific tab will be included in the menu on the left. This new parameter screen () contains the following options:

- **Number of State Objects:** defines the number of 1-bit status objects required. The minimum (and default) value is “1”, and the maximum is “12”. These objects are called “[ML] Status Object *n*”.

In addition, the general status object (“[ML] General status”) will always be available in the project topology. It will be sent to the bus with a value of “1” whenever there is at least one of the above state objects with such value. Otherwise (i.e., if none of them has a value of “1”), it will be sent with a value of “0”.

- **Trigger Value:** sets the value (“0”, “1” or “0/1”, being the latter the default option) that will trigger, when received through “[ML] Trigger”, the master action (the general switch-off or the courtesy switch-on).

- **General Switch-Off.**

- **Delay:** defines a certain delay (once the trigger has been received) before the execution of the general switch-off. The allowed range is 0 to 255 seconds.
- **Binary Value:** if checked, object “[ML] General Switch-off: Binary Object” will be enabled, which will send one “0” whenever the general switch-off takes off.
- **Scaling:** if checked, object “[ML] General Switch-off: Scaling” will be enabled, which will send a percentage value (configurable in “Value”) whenever the general switch-off takes off.
- **Scene:** if checked, object “[ML] General Switch-off: Scene” will be enabled, which will send a scene run / save order (configurable in “Action” and “Scene Number”) whenever the general switch-off takes off
- **HVAC:** if checked, object “[ML] General Switch-off: HVAC mode” will be enabled, which will send an HVAC thermostat mode value

(configurable in “**Value**”, being the options “Auto”, “Comfort”, “Standby”, “Economy” and “Building Protection”) whenever the general switch-off takes off

Note: *the above options are not mutually exclusive; it is possible to send values of different nature together.*

● **Courtesy Switch-On:**

The parameters available here are entirely analogous to those already mentioned for General Switch-Off. However, in this case the names of the objects start with “[ML] **Courtesy Switch-On (...)**”. On the other hand, sending **scene save orders** is not possible for the courtesy switch-on (only orders to play scenes are allowed).

Note: *object “[ML] **Courtesy Switch-On: Binary Object**” sends the value “1” (when the courtesy switch-on takes place), in contrast to object “[ML] **General Switch-Off: Binary Object**”, which sends the value “0” (during the general switch-off, as explained above).*

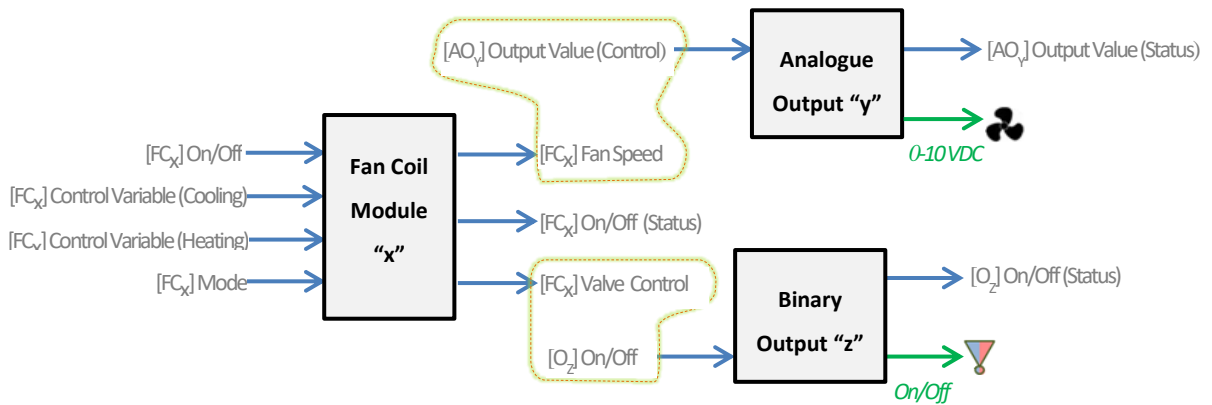
GENERAL	Number of State Objects	1
MASTER LIGHT	Trigger Value	0/1
CONFIGURATION	General Switch Off	
	Delay	0 x 1 s.
	Binary Value	<input checked="" type="checkbox"/>
	Scaling	<input type="checkbox"/>
	Scene	<input type="checkbox"/>
	HVAC	<input type="checkbox"/>
	Courtesy Switch On	
	Delay	0 x 1 s.
	Binary Value	<input checked="" type="checkbox"/>
	Scaling	<input type="checkbox"/>
	Scene	<input type="checkbox"/>
	HVAC	<input type="checkbox"/>

Figure 15 Master Light.

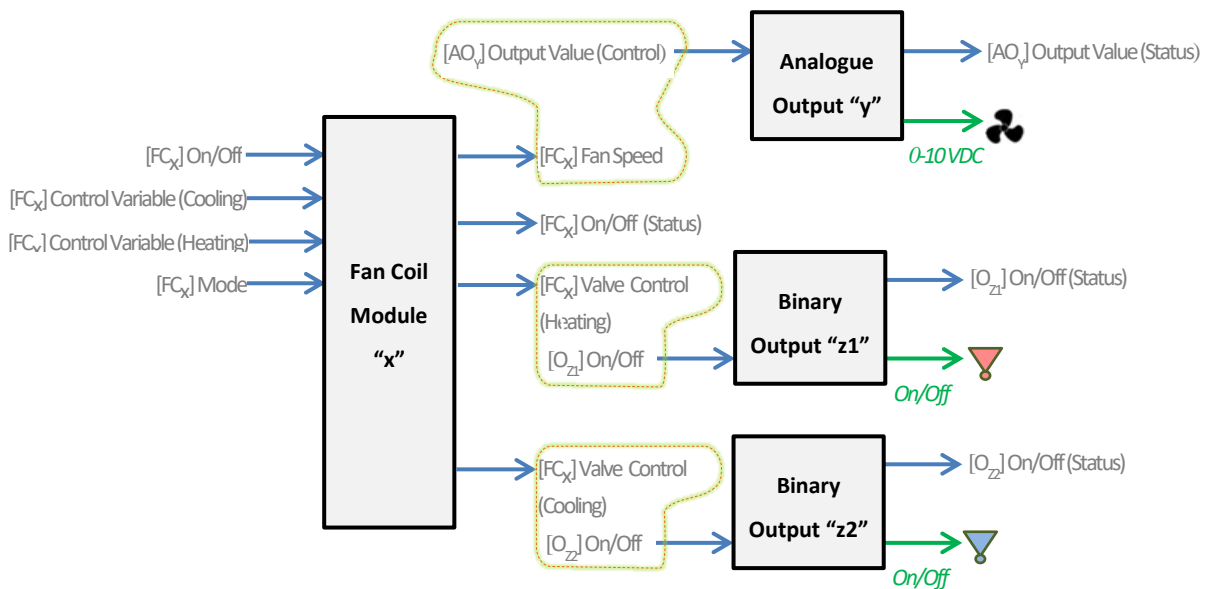
ANNEX I: INDEPENDENCE OF THE MODULES

The following figures illustrate how the communication objects and group addresses of the fan coil and the outputs modules should be linked depending on the system to control.

Two-Pipe Fan Coil System



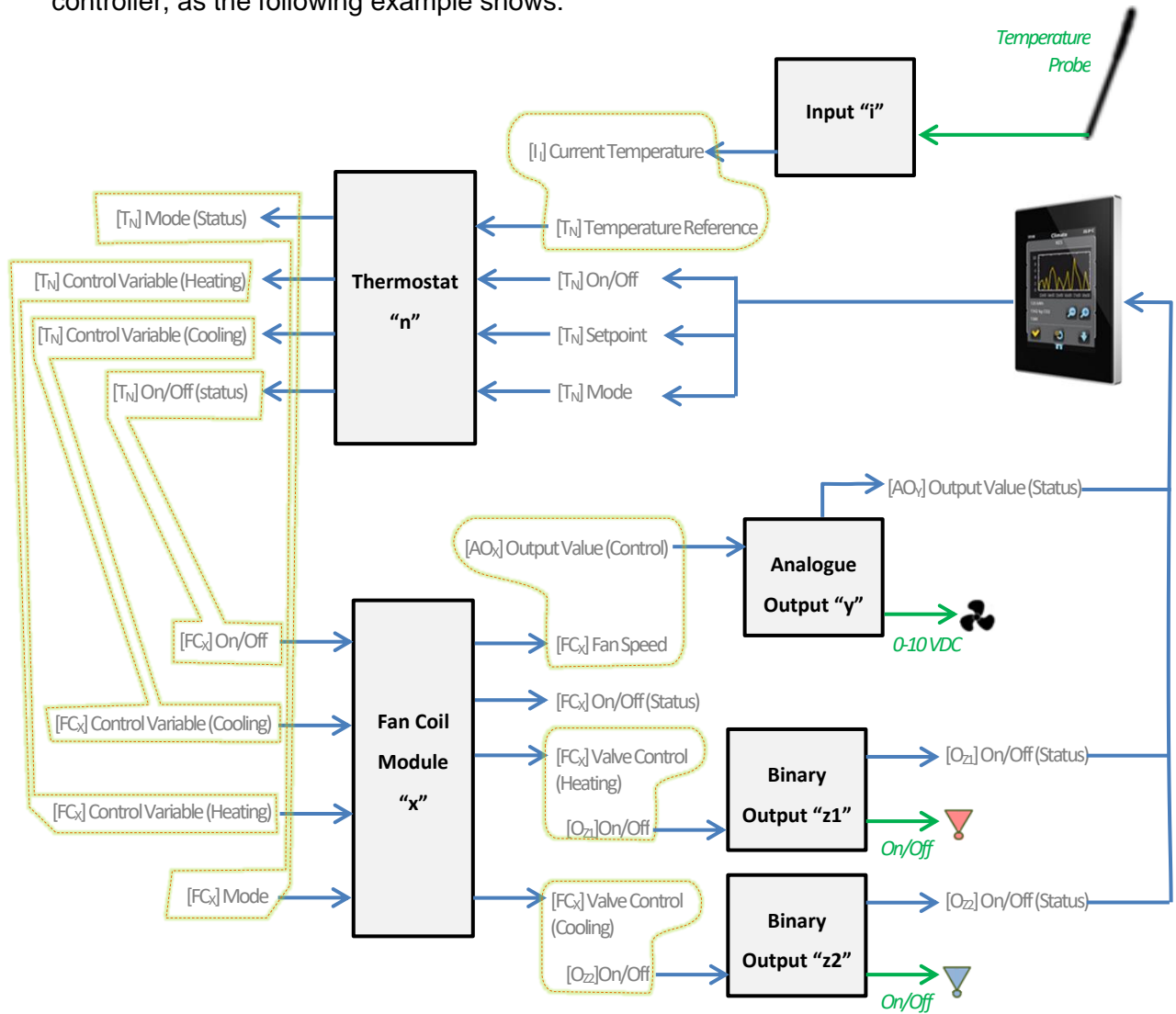
Four-Pipe Fan Coil System



It is worth emphasising that **the two fan coil modules are completely independent of the analogue and binary outputs modules**. It is up to the integrator whether to control the fan coil module together with the built-in outputs modules (by grouping their

respective communication objects) or not. This makes the device more versatile and capable of controlling fan coils systems with their own valve / fan actuators by linking the objects of the fan coil module to the objects of the external actuators.

The same applies to the built-in **thermostat** module, which can be optionally linked to the other internal modules and commanded from an external touchscreen or room controller, as the following example shows:



ANNEX II: OPERATION EXAMPLES

The device operation varies notoriously depending on the current state and the parameterisation. The following diagrams show the expected behaviour in different situations.

On/Off Orders

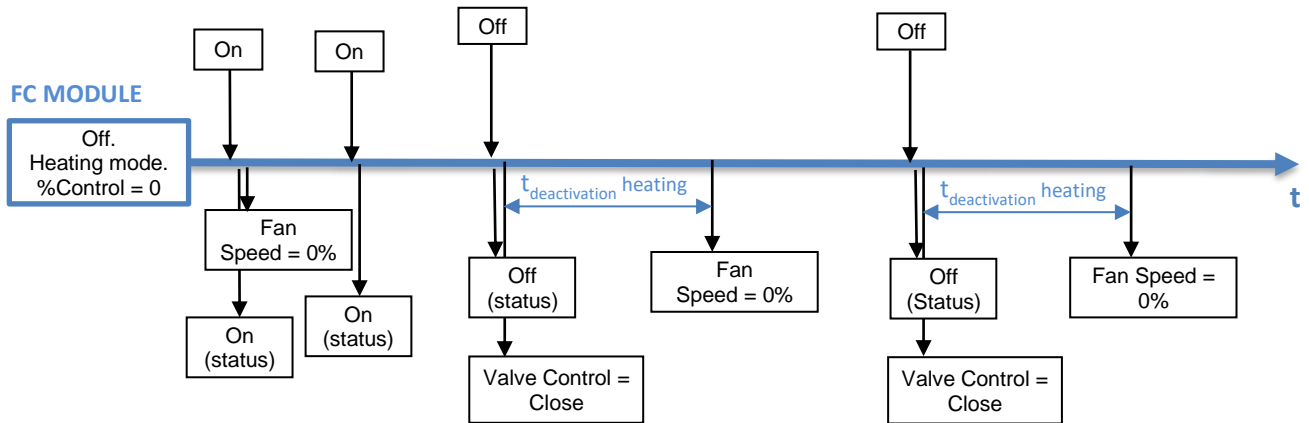


Figure 16 On/Off: 2 pipes (any mode), initial value = 0%.

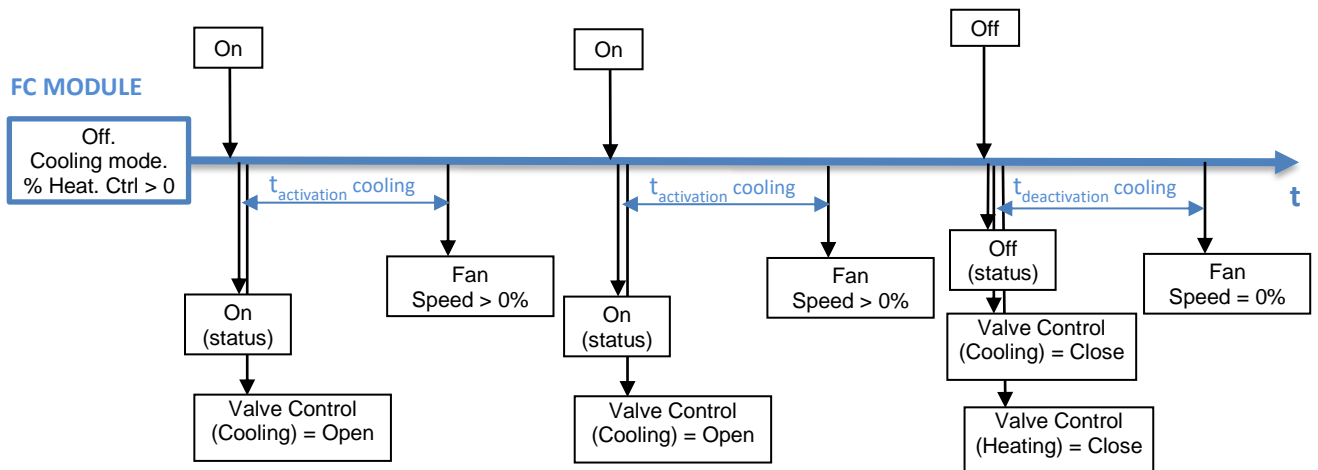


Figure 17 On/Off: 4 pipes, initial value > 0%.

Mode orders

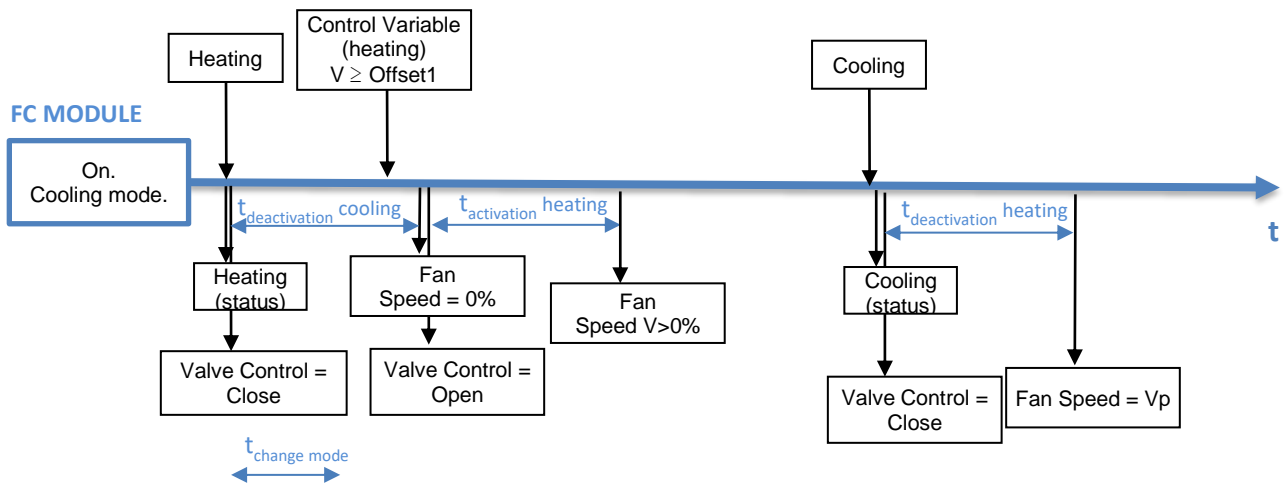


Figure 18 Change mode when Fan Coil is On: 2 pipes (“Both”).
 $t_{\text{change}} < t_{\text{deactivation}}$. Fan Speed= V_p when valve closes.

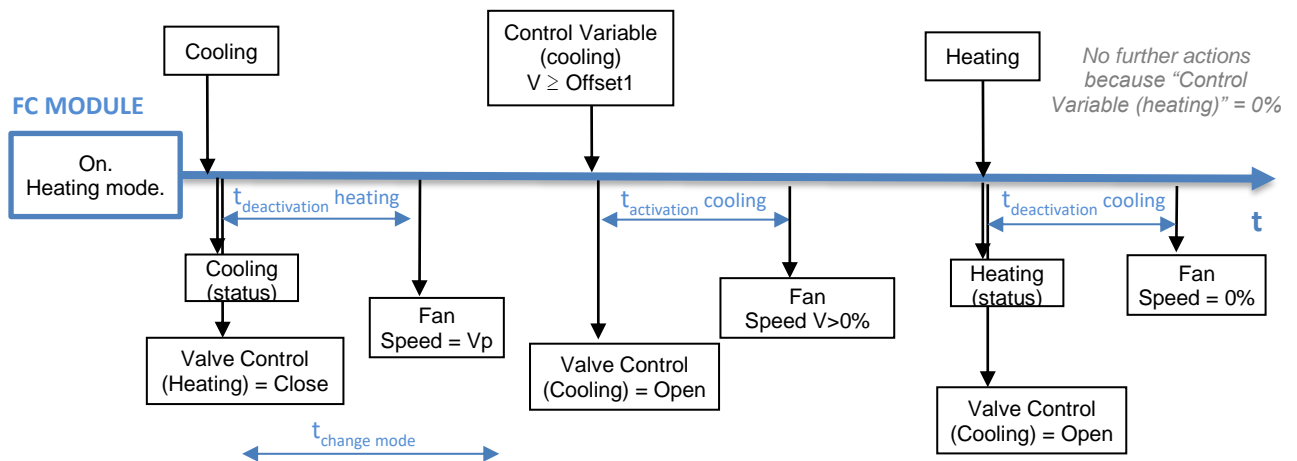


Figure 19 Change mode when Fan Coil is On: 4 pipes.
 $t_{\text{change}} > t_{\text{deactivation}}$. Fan Speed= V_p when valve closes.

ANNEX III: 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	Reset 0	Voltage Recovery -> Sending of 0
2	1 Bit		C - - T -	DPT_Trigger	0/1	Reset 1	Voltage Recovery -> Sending of 1
3	1 Bit	I	C - W - -	DPT_Enable	0/1	Lock Manual Control	0 = Lock; 1 = Unlock
	1 Bit	I	C - W - -	DPT_Enable	0/1	Lock Manual Control	0 = Unlock; 1 = Lock
4	1 Byte	I	C - W - -	DPT_SceneControl	0-63; 128-191	[Thermostat] Scene Input	Scene Value
5, 35	2 Bytes	I	C - W - -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Temperature Source 1	External Sensor Temperature
6, 36	2 Bytes	I	C - W - -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Temperature Source 2	External Sensor Temperature
7, 37	2 Bytes	O	C R - T -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Effective Temperature	Effective Control Temperature
8, 38	1 Byte	I	C - W - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode	1-byte HVAC Mode
9, 39	1 Bit	I	C - W - -	DPT_Ack	0/1	[Tx] Special Mode: Comfort	0 = Nothing; 1 = Trigger
	1 Bit	I	C - W - -	DPT_Switch	0/1	[Tx] Special Mode: Comfort	0 = Off; 1 = On
10, 40	1 Bit	I	C - W - -	DPT_Switch	0/1	[Tx] Special Mode: Standby	0 = Off; 1 = On
	1 Bit	I	C - W - -	DPT_Ack	0/1	[Tx] Special Mode: Standby	0 = Nothing; 1 = Trigger
11, 41	1 Bit	I	C - W - -	DPT_Switch	0/1	[Tx] Special Mode: Economy	0 = Off; 1 = On
	1 Bit	I	C - W - -	DPT_Ack	0/1	[Tx] Special Mode: Economy	0 = Nothing; 1 = Trigger
12, 42	1 Bit	I	C - W - -	DPT_Switch	0/1	[Tx] Special Mode: Protection	0 = Off; 1 = On
	1 Bit	I	C - W - -	DPT_Ack	0/1	[Tx] Special Mode: Protection	0 = Nothing; 1 = Trigger
13, 43	1 Bit	I	C - W - -	DPT_Window_Door	0/1	[Tx] Window Status (Input)	0 = Closed; 1 = Open
14, 44	1 Bit	I	C - W - -	DPT_Trigger	0/1	[Tx] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
15, 45	1 Byte	O	C R - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode Status	1-byte HVAC Mode
16, 46	2 Bytes	I	C - W - -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Setpoint	Thermostat Setpoint Input
	2 Bytes	I	C - W - -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Basic Setpoint	Reference Setpoint

17, 47	1 Bit	I	C - W - -	DPT_Step	0/1	[Tx] Setpoint Step	0 = -0.5°C; 1 = +0.5°C
18, 48	2 Bytes	I	C - W - -	DPT_Value_Tempd	-671088.64° - 670433.28°	[Tx] Setpoint Offset	Float Offset Value
19, 49	2 Bytes	O	C R - T -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Setpoint Status	Current Setpoint
20, 50	2 Bytes	O	C R - T -	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Basic Setpoint Status	Current Basic Setpoint
21, 51	2 Bytes	O	C R - T -	DPT_Value_Tempd	-671088.64° - 670433.28°	[Tx] Setpoint Offset Status	Current Setpoint Offset
22, 52	1 Bit	I	C - W - -	DPT_Reset	0/1	[Tx] Offset Reset	Reset offset
	1 Bit	I	C - W - -	DPT_Reset	0/1	[Tx] Setpoint Reset	Reset Setpoint to Default
23, 53	1 Bit	I	C - W - -	DPT_Heat_Cool	0/1	[Tx] Mode	0 = Cool; 1 = Heat
24, 54	1 Bit	O	C R - T -	DPT_Heat_Cool	0/1	[Tx] Mode Status	0 = Cool; 1 = Heat
25, 55	1 Bit	I	C - W - -	DPT_Switch	0/1	[Tx] On/Off	0 = Off; 1 = On
26, 56	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] On/Off Status	0 = Off; 1 = On
27, 57	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Tx] Control Variable (Cool)	PI Control (Continuous)
28, 58	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Tx] Control Variable (Heat)	PI Control (Continuous)
29, 59	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] Control Variable (Cool)	PI Control (PWM)
	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] Control Variable (Cool)	2-Point Control
30, 60	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] Control Variable (Heat)	2-Point Control
	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] Control Variable (Heat)	PI Control (PWM)
31, 61	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] Additional Cool	Temp >= (Setpoint+Band) => "1"
32, 62	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] Additional Heat	Temp <= (Setpoint-Band) => "1"
33, 63	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
34, 64	1 Bit	O	C R - T -	DPT_Switch	0/1	[Tx] PI State (Heat)	0 = PI signal 0%; 1 = PI signal greater than 0%
65, 69, 73, 77	2 Bytes	O	C R - T -	DPT_Value_Temp	-273.00° - 670433.28°	[Ix] Current Temperature	Temperature Sensor Value
66, 70, 74, 78	1 Bit	O	C R - T -	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm; 1 = Alarm
67, 71, 75, 79	1 Bit	O	C R - T -	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm; 1 = Alarm
68, 72, 76, 80	1 Bit	O	C R - T -	DPT_Alarm	0/1	[Ix] Probe Error	0 = No Alarm; 1 = Alarm
81	1 Byte	I	C - W - -	DPT_SceneNumber	0 - 63	[Motion Detector] Scene Input	Scene Value
82	1 Byte	O	C - - T -	DPT_SceneControl	0-63; 128-191	[Motion Detector] Scene Output	Scene Value
83, 112, 141, 170	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
84, 113, 142, 171	1 Bit	O	C R - T -	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
85, 114, 143, 172	1 Bit	O	C R - T -	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
86, 115, 144, 173	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
87, 116, 145, 174	1 Byte	O	C R - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection

88, 117, 146, 175	1 Bit	O	C R - T -	DPT_Switch	0/1	[Ix] Presence State (Binary)	Binary Value
	1 Bit	O	C R - T -	DPT_Start	0/1	[Ix] Presence: Slave Output	1 = Motion Detected
89, 118, 147, 176	1 Bit	I	C - W - -	DPT_Window_Door	0/1	[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
90, 119, 148, 177	1 Bit	I	C - W - -	DPT_Start	0/1	[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
91, 120, 149, 178	2 Bytes	I	C - W - -	DPT_TimePeriodSec	0 - 65535	[Ix] Presence: Waiting Time	0-65535 s.
92, 121, 150, 179	2 Bytes	I	C - W - -	DPT_TimePeriodSec	0 - 65535	[Ix] Presence: Listening Time	1-65535 s.
93, 122, 151, 180	1 Bit	I	C - W - -	DPT_Enable	0/1	[Ix] Presence: Enable	According to parameters
94, 123, 152, 181	1 Bit	I	C - W - -	DPT_DayNight	0/1	[Ix] Presence: Day/Night	According to parameters
95, 124, 153, 182	1 Bit	O	C R - T -	DPT_Occupancy	0/1	[Ix] Presence: Occupancy State	0 = Not Occupied; 1 = Occupied
96, 125, 154, 183	1 Bit	I	C - W - -	DPT_Start	0/1	[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by an external sensor
97, 102, 107, 126, 131, 136, 155, 160, 165, 184, 189, 194	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Ix] [Cx] Detection State (Scaling)	0-100%
98, 103, 108, 127, 132, 137, 156, 161, 166, 185, 190, 195	1 Byte	O	C R - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] [Cx] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
99, 104, 109, 128, 133, 138, 157, 162, 167, 186, 191, 196	1 Bit	O	C R - T -	DPT_Switch	0/1	[Ix] [Cx] Detection State (Binary)	Binary Value
100, 105, 110, 129, 134, 139, 158, 163, 168, 187, 192, 197	1 Bit	I	C - W - -	DPT_Enable	0/1	[Ix] [Cx] Enable Channel	According to parameters
101, 106, 111, 130, 135, 140, 159, 164, 169, 188, 193, 198	1 Bit	I	C - W - -	DPT_Switch	0/1	[Ix] [Cx] Force State	0 = No Detection; 1 = Detection
199, 210, 221, 232	1 Byte	I	C - W - -	DPT_SceneControl	0-63; 128-191	[Ox] Scenes	0 - 63 (Execute 1 - 64); 128 - 191 (Save 1 - 64)
200, 211, 222, 233	1 Bit	I	C - W - -	DPT_BinaryValue	0/1	[Ox] On/Off	N.O. (0=Open Relay; 1=Close Relay)
	1 Bit	I	C - W - -	DPT_BinaryValue	0/1	[Ox] On/Off	N.C. (0=Close Relay; 1= Open Relay)
201, 212, 223, 234	1 Bit	O	C R - T -	DPT_BinaryValue	0/1	[Ox] On/Off (Status)	0=Output Off; 1=Output On
202, 213, 224, 235	1 Bit	I	C - W - -	DPT_Enable	0/1	[Ox] Lock	0=Unlock; 1=Lock
203, 214, 225, 236	1 Bit	I	C - W - -	DPT_Start	0/1	[Ox] Timer	0=Switch Off; 1=Switch On
204, 215, 226, 237	1 Bit	I	C - W - -	DPT_Start	0/1	[Ox] Flashing	0=Stop; 1=Start
205, 216, 227, 238	1 Bit	I	C - W - -	DPT_Alarm	0/1	[Ox] Alarm	0=Alarm; 1=Normal
	1 Bit	I	C - W - -	DPT_Alarm	0/1	[Ox] Alarm	0=Normal; 1=Alarm

206, 217, 228, 239	1 Bit	I	C - W - -	DPT_Ack	0/1	[Ox] Unfreeze Alarm	Alarm=0 + Unfreeze=1 => End Alarm
207, 218, 229, 240	1 Bit	O	CR - T -	DPT_State	0/1	[Ox] Warning Time (Status)	0=Normal; 1=Warning
208, 219, 230, 241	4 Bytes	I/O	CRWT -	DPT_LongDeltaTimeSec	-2147483648 - 2147483647	[Ox] Operating Time (s)	Time in Seconds
209, 220, 231, 242	2 Bytes	I/O	CRWT -	DPT_TimePeriodHrs	0 - 65535	[Ox] Operating Time (h)	Time in Hours
243, 249, 255, 261	1 Bit	I	C - W - -	DPT_Enable	0/1	[Ix] Input Lock	0 = Unlock; 1 = Lock
244, 250, 256, 262	4 Bit		C - - T -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0xD (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Darker	Decrease Brightness
	1 Bit	I/O	CRWT -	DPT_Switch	0/1	[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
	4 Bit		C - - T -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0xD (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter	Increase Brightness
	1 Bit		C - - T -	DPT_Switch	0/1	[Ix] [Short Press] 0	Sending of 0
	1 Bit		C - - T -	DPT_Switch	0/1	[Ix] [Short Press] 1	Sending of 1
	1 Bit	I	C - WT -	DPT_Switch	0/1	[Ix] [Short Press] 0/1 Switching	Switching 0/1
	1 Bit		C - - T -	DPT_UpDown	0/1	[Ix] [Short Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		C - - T -	DPT_UpDown	0/1	[Ix] [Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C - - T -	DPT_UpDown	0/1	[Ix] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C - - T -	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Byte		C - - T -	DPT_Value_1_Ucount	0 - 255	[Ix] [Short Press] Constant Value (Integer)	0 - 255
	1 Bit		C - - T -	DPT_Switch	0/1	[Ix] [Short Press] Light Off	Sending of 0 (Off)
4 Bit		C - - T -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0xD (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter/Darker	Switch Bright/Dark	

	1 Byte		C - - T -	DPT_Scaling	0% - 100%	[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		C - - T -	9.xxx	-671088.64 - 670433.28	[Ix] [Short Press] Constant Value (Float)	Float Value
	1 Byte		C - - T -	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Run Scene	Sending of 0 - 63
	1 Bit	I	C - W T -	DPT_Switch	0/1	[Ix] [Short Press] Light On/Off	Switching 0/1
	1 Bit		C - - T -	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		C - - T -	DPT_Switch	0/1	[Ix] [Short Press] Light On	Sending of 1 (On)
	2 Bytes		C - - T -	DPT_Value_2_Ucount	0 - 65535	[Ix] [Short Press] Constant Value (Integer)	0 - 65535
	1 Bit		C - - T -	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	1 Byte		C - - T -	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Save Scene	Sending of 128 - 191
245, 251, 257, 263	1 Byte	I	C - W - -	DPT_Scaling	0% - 100%	[Ix] [Short Press] Shutter Status (Input)	0% = Top; 100% = Bottom
	1 Byte	I	C - W - -	DPT_Scaling	0% - 100%	[Ix] [Short Press] Dimming Status (Input)	0% - 100%
246, 252, 258, 264	4 Bit		C - - T -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0xD (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		C - - T -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0xD (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Darker	Long Pr. -> Darker; Release -> Stop
	4 Bit		C - - T -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0xD (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter/Darker	Long Pr. -> Brighter/Darker; Release -> Stop

	1 Bit		C--T-	DPT_Switch	0/1	[Ix] [Long Press] Light On	Sending of 1 (On)
	1 Bit		C--T-	DPT_Switch	0/1	[Ix] [Long Press] Light Off	Sending of 0 (Off)
	1 Bit	I	C-WT-	DPT_Switch	0/1	[Ix] [Long Press] Light On/Off	Switching 0/1
	1 Byte		C--T-	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Run Scene	Sending of 0 - 63
	1 Byte		C--T-	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Save Scene	Sending of 128 - 191
	1 Bit	O	CR-T-	DPT_Alarm	0/1	[Ix] [Switch/Sensor] Alarm: Breakdown or Sabotage	1 = Alarm; 0 = No Alarm
	2 Bytes		C--T-	9.xxx	-671088.64 - 670433.28	[Ix] [Long Press] Constant Value (Float)	Float Value
	2 Bytes		C--T-	DPT_Value_2_Ucount	0 - 65535	[Ix] [Long Press] Constant Value (Integer)	0 - 65535
	1 Bit		C--T-	DPT_Switch	0/1	[Ix] [Long Press] 1	Sending of 1
	1 Byte		C--T-	DPT_Value_1_Ucount	0 - 255	[Ix] [Long Press] Constant Value (Integer)	0 - 255
	1 Byte		C--T-	DPT_Scaling	0% - 100%	[Ix] [Long Press] Constant Value (Percentage)	0% - 100%
	1 Bit		C--T-	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	1 Bit		C--T-	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		C--T-	DPT_UpDown	0/1	[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C--T-	DPT_Switch	0/1	[Ix] [Long Press] 0	Sending of 0
	1 Bit		C--T-	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit	I	C-WT-	DPT_Switch	0/1	[Ix] [Long Press] 0/1 Switching	Switching 0/1
1 Bit		C--T-	DPT_UpDown	0/1	[Ix] [Long Press] Move Up Shutter	Sending of 0 (Up)	
1 Bit		C--T-	DPT_UpDown	0/1	[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)	
247, 253, 259, 265	1 Bit		C--T-	DPT_Trigger	0/1	[Ix] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
248, 254, 260, 266	1 Byte	I	C-W--	DPT_Scaling	0% - 100%	[Ix] [Long Press] Shutter Status (Input)	0% = Top; 100% = Bottom
	1 Byte	I	C-W--	DPT_Scaling	0% - 100%	[Ix] [Long Press] Dimming Status (Input)	0% - 100%
267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298	1 Bit	I	C-W--	DPT_Bool	0/1	[LF] (1-Bit) Data Entry x	Binary Data Entry (0/1)

299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314	1 Byte	I	C - W - -	DPT_Value_1_Ucount	0 - 255	[LF] (1-Byte) Data Entry x	1-Byte Data Entry (0-255)
315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330	2 Bytes	I	C - W - -	DPT_Value_2_Ucount	0 - 65535	[LF] (2-Byte) Data Entry x	2-Byte Data Entry
331, 332, 333, 334, 335, 336, 337, 338	4 Bytes	I	C - W - -	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] (4-Byte) Data Entry x	4-Byte Data Entry
339, 340, 341, 342, 343, 344, 345, 346, 347, 348	1 Bit	O	CR - T -	DPT_Bool	0/1	[LF] Function x - Result	(1-Bit) Boolean
	1 Byte	O	CR - T -	DPT_Value_1_Ucount	0 - 255	[LF] Function x - Result	(1-Byte) Unsigned
	2 Bytes	O	CR - T -	DPT_Value_2_Ucount	0 - 65535	[LF] Function x - Result	(2-Byte) Unsigned
	4 Bytes	O	CR - T -	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] Function x - Result	(4-Byte) Signed
	1 Byte	O	CR - T -	DPT_Scaling	0% - 100%	[LF] Function x - Result	(1-Byte) Percentage
	2 Bytes	O	CR - T -	9.xxx	-671088.64 - 670433.28	[LF] Function x - Result	(2-Byte) Float
	2 Bytes	O	CR - T -	DPT_Value_2_Count	-32768 - 32767	[LF] Function x - Result	(2-Byte) Signed
349, 352	1 Byte	I	C - W - U	DPT_Scaling	0% - 100%	[AOx] Output Value (Control)	0 - 100 %
350, 353	1 Byte	O	CR - T -	DPT_Scaling	0% - 100%	[AOx] Output Value (Status)	0 - 100 %
351, 354	1 Bit	I	C - W - U	DPT_Enable	0/1	[AOx] Lock	0 = Unlock; 1 = Lock
355, 370	1 Bit	I	C - W - U	DPT_Switch	0/1	[FCx] On/Off	0 = Off; 1 = On
356, 371	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] On/Off (Status)	0 = Off; 1 = On
357, 372	1 Bit	I	C - W - U	DPT_Heat_Cool	0/1	[FCx] Mode	0 = Cool; 1 = Heat
358, 373	1 Bit	O	CR - T -	DPT_Heat_Cool	0/1	[FCx] Mode (Status)	0 = Cool; 1 = Heat
359, 374	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] Valve Control	0 = Open Valve; 1 = Close Valve
	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] Valve Control (Cooling)	0 = Close Valve; 1 = Open Valve
	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] Valve Control	0 = Close Valve; 1 = Open Valve
	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] Valve Control (Cooling)	0 = Open Valve; 1 = Close Valve
360, 375	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] Valve Control (Heating)	0 = Close Valve; 1 = Open Valve
	1 Bit	O	CR - T -	DPT_Switch	0/1	[FCx] Valve Control (Heating)	0 = Open Valve; 1 = Close Valve
361, 376	1 Bit	I/O	CRWTU	DPT_Enable	0/1	[FCx] Forced Position	0 = Disable; 1 = Enable
362, 377	1 Bit	I	C - W - U	DPT_Enable	0/1	[FCx] Manual Fan Control	0 = Disable; 1 = Enable
363, 378	1 Bit	O	CR - T -	DPT_Enable	0/1	[FCx] Manual Fan Control (Status)	0 = Disabled; 1 = Enabled
364, 379	1 Byte	I	C - W - U	DPT_Scaling	0% - 100%	[FCx] Control Variable (Cooling)	0 - 100 %
365, 380	1 Byte	I	C - W - U	DPT_Scaling	0% - 100%	[FCx] Control Variable (Heating)	0 - 100 %
366, 381	1 Byte	O	CR - T -	DPT_Scaling	0% - 100%	[FCx] Fan Speed	0 - 100 %
367, 382	1 Byte	I	C - W - U	DPT_Scaling	0% - 100%	[FCx] Fan Speed when Valve Closed (Cooling)	0 - 100 %

368, 383	1 Byte	I	C - W - U	DPT_Scaling	0% - 100%	[FCx] Manual Fan Control Variable	0 - 100 %
369, 384	2 Bytes	I/O	CRWTU	DPT_TimePeriodHrs	0 - 65535	[FCx] Duration of Manual Control	0 = Endless; 1 - 24 h
	2 Bytes	I/O	CRWTU	DPT_TimePeriodMin	0 - 65535	[FCx] Duration of Manual Control	0 = Endless; 1 - 1440 min
385	1 Bit		C - - T -	DPT_Trigger	0/1	[Heartbeat] Object to Send '1'	Sending of '1' Periodically
386	1 Bit	I	C - W - -	DPT_Ack	0/1	[ML] Trigger	0 = Nothing; 1 = Trigger the Master Light Function
	1 Bit	I	C - W - -	DPT_Ack	0/1	[ML] Trigger	1 = Nothing; 0 = Trigger the Master Light Function
	1 Bit	I	C - W - -	DPT_Trigger	0/1	[ML] Trigger	Trigger the Master Light Function
387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398	1 Bit	I	C - W - -	DPT_Switch	0/1	[ML] Status Object x	Binary Status
399	1 Bit	O	CR - T -	DPT_Switch	0/1	[ML] General Status	Binary Status
400	1 Bit		C - - T -	DPT_Switch	0/1	[ML] General Switch Off: Binary Object	Switch Off Sending
401	1 Byte		C - - T -	DPT_Scaling	0% - 100%	[ML] General Switch Off: Scaling	0-100%
402	1 Byte		C - - T -	DPT_SceneControl	0-63; 128-191	[ML] General Switch Off: Scene	Scene Sending
403	1 Byte		C - - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[ML] General Switch Off: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection
404	1 Bit		C - - T -	DPT_Switch	0/1	[ML] Courtesy Switch On: Binary Object	Switch On Sending
405	1 Byte		C - - T -	DPT_Scaling	0% - 100%	[ML] Courtesy Switch On: Scaling	0-100%
406	1 Byte		C - - T -	DPT_SceneNumber	0 - 63	[ML] Courtesy Switch On: Scene	Scene Sending
407	1 Byte		C - - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[ML] Courtesy Switch On: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection

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