

MAXinBOX FC 0-10V VALVE

Two/Four-Pipe Fan Coil Controller with 0-10 VDC Valve Control Signal

ZCL-FC010V

Application Programme Version: [1.4]

User Manual Version: [1.4]_a

www.zennio.com

CONTENTS

Contents	2
Document Updates	3
1 Introduction	4
1.1 MAXinBOX FC 0-10V VALVE.....	4
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 Individual Binary Outputs.....	11
2.4 Individual 0-10V Analogue Outputs	12
2.5 Fan Coil	15
2.5.1 Configuration	15
2.5.2 Valves.....	28
2.5.3 Fan	33
2.6 Logic Functions	44
2.7 Thermostats	45
2.8 Manual control	46
2.9 Master Light	50
ANNEX I: Fan Coil Priorities	53
ANNEX II: Communication objects	54

DOCUMENT UPDATES

Version	Changes	Page(s)
[1.4]_a	Changes in the application programme: <ul style="list-style-type: none"> Internal optimisation. 	-
[1.3]_a	Changes in the application programme: <ul style="list-style-type: none"> Optimisation of the modules: Heartbeat, Binary Inputs, Individual Digital Outputs, Logic Functions, Master Light, Motion Detector, Temperature Probe and Thermostat. Support for long messages in ETS download. 	-
[1.2]_a	Changes in the application programme: <ul style="list-style-type: none"> Heartbeat functionality added. Master Light functionality added. New parameter to change the polarity of the object to switch fan mode auto/manual. Optimisation of the Fan Coil, Binary Inputs, Individual Digital Outputs, Analog Outputs, Logic Functions, Motion Detector and Thermostat modules. 	-
[1.1]_a	Changes in the application programme: <ul style="list-style-type: none"> Optimisation of the motion detector module. 	-

1 INTRODUCTION

1.1 MAXinBOX FC 0-10V VALVE

MAXinBOX FC 0-10V VALVE 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 valves in the pipes are controlled through **analogue 0-10 VDC signals**, while the fan speed needs to be controlled through **binary outputs** (relays).

The actuator provides two analogue outputs and four binary outputs, which permits controlling **one fan coil drive consisting of two or four pipes** (each with its own valve) and a fan system with up to four speeds.

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

- **1 control module for two-pipe or four-pipe fan coil units** with either valves controlled through a 0-10V DC signal proportional to the opening level, or valves with their own KNX actuator (remote valve control).
- **2 analogue 0-10 VDC outputs** to control the valves of the fan coil pipes, or for an independent use.
- **4 binary (relay) outputs** to control up to four fan speeds, or for an independent use.
- **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.

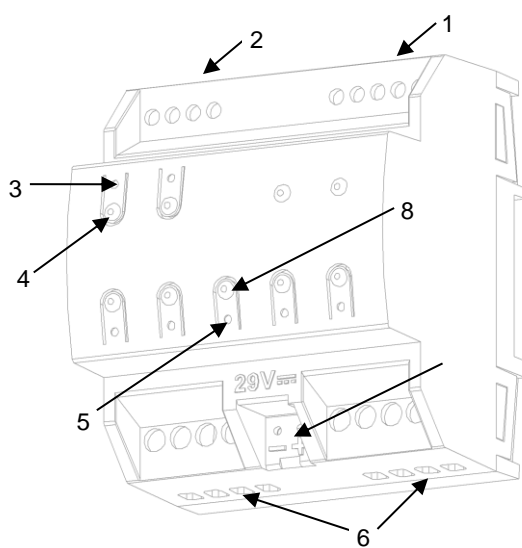
- **Master light control** for and easy, out-of-the-box control of a set of luminaires (or functionally equivalent devices) one of which acts as a general lamp and the others as secondary lamps.
- **Heartbeat** or periodical “still-alive” notification.

1.2 INSTALLATION

MAXinBOX FC 0-10V VALVE 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 programme may be downloaded.

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



1. Multi-Purpose Inputs.
2. 0-10 VDC Outputs.
3. Output Status LED Indicator.
4. Output Control Button.
5. Prog./Test LED.
6. Binary Outputs.
7. KNX Bus Connector.
8. Prog./Test button.

Figure 1 MAXinBOX FC 0-10V VALVE.

The main elements of the device are described next.

- **Prog./Test pushbutton (8):** a short press on this button sets the device into 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, 2.4 and 2.5). 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 and also available at www.zennio.com.

1.3 START-UP AND POWER LOSS

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

Depending on the configuration of the fan coil control module, 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 VALVE will interrupt any pending actions, and will save its state so it can be recovered once the power supply is restored. Moreover, all relay outputs will remain open, for safety reasons.

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

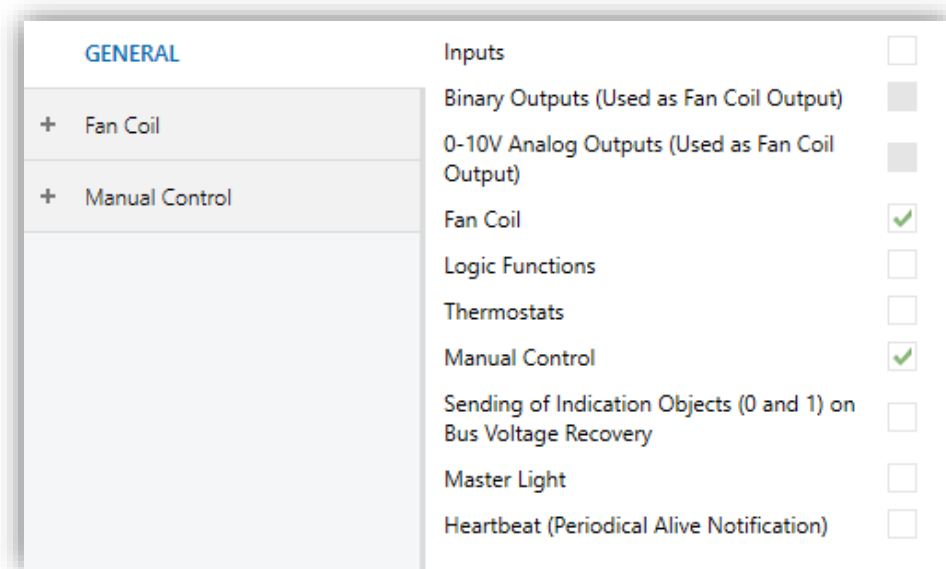


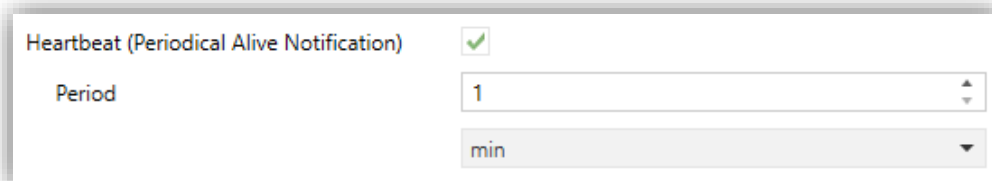
Figure 2 General screen.

- Once activated, **Inputs**, **Binary Outputs**, **Analogue Outputs**, **Fan Coil** (enabled by default), **Logical Functions**, **Thermostats** and **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.

Note: when the Fan Coil function gets enabled, the checkboxes to enable the Binary Outputs and the Analogue Outputs will become locked as these

outputs are required for the fan coil control. Nevertheless, they may become unlocked again depending on how the fan coil module is configured.

- **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).
- **Heartbeat (Periodical Alive Notification):** 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*).



Heartbeat (Periodical Alive Notification)	<input checked="" type="checkbox"/>
Period	1
	min

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.*

2.2 INPUTS

MAXinBOX FC 0-10V VALVE 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 VALVE 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 VALVE 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 VALVE. 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 VALVE product section at www.zennio.com, for detailed information about the functionality and the configuration of the related parameters.

2.3 INDIVIDUAL BINARY OUTPUTS

MAXinBOX FC 0-10V VALVE incorporates **four binary outputs**, each of which can be enabled and configured in parameters independently, **as long as they are not required by the fan coil control module** (see section 2.5) as their typical application in this device is controlling the different fan speeds of the fan coil (up to four).

Their parameterisation as independent binary outputs is similar to that of the individual relay outputs of any other MAXinBOX actuators. Note that in such case they all work **independently**, although it is possible to group them as required (for example, to switch an indicator light on when the fan is running) through joint group addresses, as well as to configure each of them as “normally open” or “normally closed”.

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

2.4 INDIVIDUAL 0-10V ANALOGUE OUTPUTS

MAXinBOX FC 0-10V VALVE incorporates **two analogue voltage outputs** that provide a signal with a voltage 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.

These outputs can be enabled and configured independently in parameters **as long as they are not required by the fan coil control module** (see section 2.5) as their typical application in this device is controlling the opening and closing of the valves in the fan coil pipes.

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.

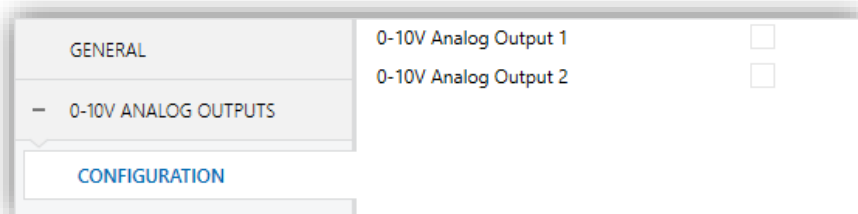


Figure 4 0-10V Analog Outputs - Configuration.

The two analogue outputs can be activated independently (provided that they are not in use by the fan coil module; see section 2.5) by marking the checkboxes. This also brings new tabs to the tree.

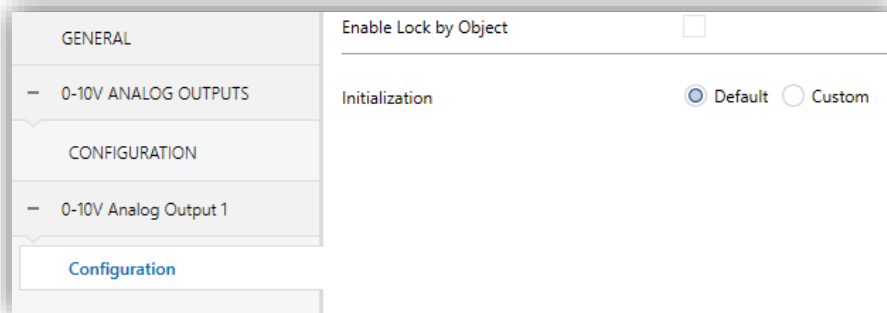


Figure 5 0-10V Analog Output X – Configuration.

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

- “[AOx] Output Value (Control)”: this object is provided to receive percentage values from the KNX bus, making the device generate a voltage output between 0 VDC and 10 VDC (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:

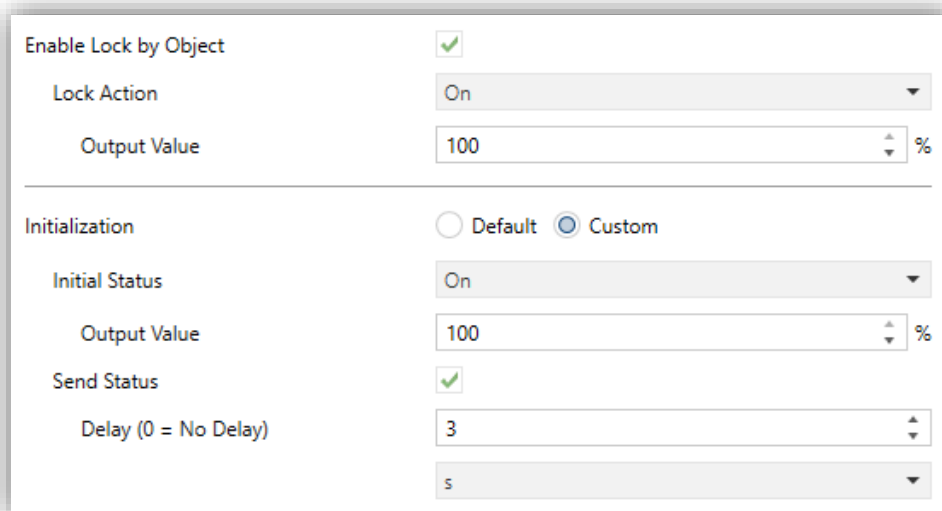


Figure 6 0-10V Analogue Output X – Configuration (detail).

- **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 COIL

MAXinBOX FC 0-10V VALVE incorporates **one fan coil** module that implements all the logic involved in the control of a two-pipe or four-pipe fan coil unit where the fan speed can be switched through **relay outputs** and the opening and closing of the pipe valves can be commanded through **a 0 to 10 VDC signal** (depending on the required degree of opening).

ETS PARAMETERISATION

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

The fan coil module requires setting some general parameters and some specific parameters related with the valves and the fan. All of them are described next.

2.5.1 CONFIGURATION

The general configuration of the Fan Coil module entails, on the first hand, setting whether the fan coil is composed of either:

- **Four pipes (two valves).**
- **Two pipes (one valve),**

Every two pipes make up a circuit where the water flow is controlled by means of a **valve**, whose position is managed through an analogue 0 to 10 volt DC signal.

Typically, the two water circuits of a **four-pipe fan coil unit** correspond to the **cooling** and **heating** functions (and to the cooling and heating valves, respectively), being therefore both modes available during the device operation.

On the other hand, the single water circuit of a **two-pipe fan coil unit** may be configured for:

- **Cooling**
- **Heating**
- **Cooling and Heating.**

No. of pipes	Output	Action
4	Valve Output 1	Cooling Valve
	Valve Output 2	Heating Valve
2	Valve Output 1	Heating Valve
		Cooling Valve
		Heating + Cooling Valve

Table 1 Actions performed by the analogue outputs (valves).

Also within the general configuration of the device, it is necessary to set the **valve control type**, as either **local** (the valves will be commanded by MAXinBOX FC 0-10V VALVE itself) or **remote** (the orders will be sent by MAXinBOX FC 0-10V VALVE bus to a separate valve actuator, through the KNX bus).

On the other hand, up to four binary outputs are available for the fan speed control, being possible to configure them as **commuting relays** (one specific relay for each fan speed) or as **accumulating relays** (the more relays closed, the higher the fan speed).

The integrator can also enable, in parameters, an **automatic air recirculation** in the cooling mode. This will make the fan remain always on while the current mode is cooling, even if the temperature setpoint has been reached (stopping the fan may cause an uncomfortable feeling). Note that orders with a higher priority (such as alarms configured to switch to fan speed 0) will still prevail over this option.

One parameter is also provided to enable the **ventilation mode**, being then possible to switch to that mode through a specific binary object or a scene. The ventilation mode consists in closing the valve(s) and leaving the fan on. However, the **fan speed** will depend on some factors:

- **If the current mode is cooling**, the fan speed will be automatically determined by the system according to the configuration, unless a specific speed has been manually ordered (i.e., the fan speed selection is in manual mode). See section 2.5.3 for details on the fan speed selection.
- **If the current mode is heating**, the fan will remain stopped, unless a specific speed has been manually ordered (i.e., the fan speed selection is in manual mode). See section 2.5.3 for details on the fan speed selection.

ETS PARAMETERISATION

Figure 7 Fan Coil – Configuration.

The Configuration screen (see Figure 7) is accessible by default after enabling the Fan Coil function. It contains the following parameters:

PIPE CONFIGURATION

- **Type of Fan Coil:** “2 pipes” or “4 pipes”. If “2 pipes”, it is necessary to define the available HVAC modes:

HVAC Mode: “Cooling”, “Heating” or “Cooling + Heating”. This parameter is only available for two-pipe fan coils (in the case of four-pipe fan coils, both modes are assumed to be available). If set to “Cooling + Heating”, a binary object (“**[FCV1] H/C Mode**”) will be provided to switch the current mode at any time, as well as the corresponding status object.

VALVE CONTROL CONFIGURATION

- **Valve Control Type:** “Local” (default option) or “Remote”.

FAN CONTROL CONFIGURATION

- **Number of Fan Speeds:** up to four fan speeds can be controlled, depending on the fan model. This determines the number of relays required (unneeded relay outputs may be configured as individual binary outputs; see section 2.3).
- **Type of Relays Control** (only available if the above is set to other than “1”):

- **“Switching”** (default setting): the activation of each speed requires closing one specific relay (and leaving the others open).

It is possible to set up a **Delay between Fan Speed Switching** (between 3 and 100 ds), which represents the time between the opening of the source speed relay and the closing of the target speed relay (so both stay open during such time).

- **“Accumulation”**: the speed will be proportional to the number of active outputs (i.e. closed relays), which therefore get triggered in sequence.

ADVANCED GENERAL CONFIGURATION

- **Fan Coil always On?:**

If this option is set to **“No”** (by default), the fan coil turns on/off when receiving a “1” or “0” through the object **“[FCV1] On/Off”**. The state of the fan coil can be obtained by reading the **“[FCV1] 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 **“[FCV1] On/Off”** and **“[FCV1] On/Off (Status)”** are not available.

- **Automatic Air Recirculation in Cooling Mode:** sets whether the fan should always remain on while in the Cooling mode or not.
- **Ventilation Mode:** sets whether the ventilation mode is enabled or not. If enabled, the communication objects **“[FCV1] Ventilation Mode”** (which triggers / stops the ventilation mode when it receives the values “1” and “0”, respectively) and **“[FCV1] Ventilation Mode (Status)”** will be incorporated to

the project topology. Ventilation mode can also activate through **scenes** (please refer to section 2.5.1.4).

- **Alarms:** enables or disables the alarm functions (see section 2.5.1.1).
- **Forced Position:** enables or disables the forced position function (see section 2.5.1.2).
- **Purge:** enables or disables the purge function (see section 2.5.1.3).
- **Scenes:** enables or disables the scene functions (see section 2.5.1.4).

There are some other objects that will be available or not depending on the above parameters:

Object	2 tubes			4 tubes
	Cooling	Heating	Both	
"[FCV1] H/C Mode"			√	√
"[FCV1] H/C Mode (Status)"			√	√
"[FCV1] Cooling Input Control Signal"	√		√	√
"[FCV1] Heating Input Control Signal"		√	√	√
"[FCV1] Valve (Status)"			√	
"[FCV1] Cooling Valve (Status)"	√			√
"[FCV1] Heating Valve (Status)"		√		√

Table 2 Fan Coil communication objects depending on the Type and Mode.

Note: the last three objects will change their names in case the valve control has been set to **remote** instead of **local**.

Local	Remote
"[FCV1] Valve (Status)"	"[FCV1] Valve Control Signal"
"[FCV1] Cooling Valve (Status)"	"[FCV1] Cooling Valve: Control Signal"
"[FCV1] Heating Valve (Status)"	"[FCV1] Heating Valve: Control Signal"

Table 3 Valve status / control objects.

Regarding the functionality of all these objects:

- The **input control signal** objects are provided for the reception of the percentage control variables from a thermostat (such as any of the internal thermostats of this device; see section 2.7).

Note: *the values received through “[FCV1] Cooling Input Control Signal” will only have effect if Cooling is the current mode (in such case, the values received through “[FCV1] Heating Input Control Signal” will be ignored, although they will be taken into account if the mode changes to Heating). The same happens for the opposite case.*

- Regarding the valve objects:
 - The **valve status objects** (only available under a **local** valve control) inform about the current positions (in terms of percentage) of the valves, and will be sent to the bus whenever the positions change.
 - The **valve control signal objects** (only available under a **remote** valve control) behave analogously, although in this case they should be linked to the input objects of the remote actuator, in order to run the valves.

Important: *a greater value of the thermostat control variable means a higher effort of the climate system. This typically translates into a greater opening of the valve (after applying the configured restrictions), which under the **local** valve control causes a higher-voltage signal in the analogue output of MAXinBOX FC 0-10V VALVE.*

*For valves controlled through other voltage ranges or without a direct proportion between the valve opening level and the outputted voltage, it is advisable to use the **remote** valve control, and either:*

- *Pass the valve control signal object through a logic function (see section 2.6) and afterwards assign the result to any of the analogue outputs configured as independent outputs, or*
- *Send the valve control signal object to a specific remote actuator configured to generate the proper voltage level for the required opening level.*

2.5.1.1 ALARMS

The Alarms function allows changing the state of the fan coil unit on the reception of an alarm trigger from the KNX bus, being possible to configure both the target valve opening level and the target fan speed.

Cyclically monitoring the alarm trigger is also possible by defining a certain time period. The actuator will check that the alarm or no-alarm state is received at least once before the period expires (note: this check doesn't take place unless the object has been received at least once). In case the object stops being refreshed (i.e., the device does not receive an updated value anymore), the alarm action will be performed as well, for safety reasons.

Regarding the deactivation of the alarm, it is also possible to configure a simple deactivation or an **acknowledgement-demanding** deactivation.

- Under a simple deactivation, the device will leave the alarm state as soon as the alarm object recovers its normal value.
- Under an acknowledgement-demanding deactivation, the device will only leave the alarm state until an external acknowledgement (through a separate object) is received after the alarm object has recovered its normal value.

Note: *alarms always prevail over any other function (i.e., any orders during the alarm state will be ignored).*

MAXinBOX FC 0-10V VALVE incorporates **two independent alarm functions**.

Alarm 1 has a higher priority over alarm 2. If the fan coil module is under the “alarm 2” state and alarm 1 gets triggered, it will execute the action of alarm 1 and will wait till alarm 1 gets deactivated (switching then back to the “alarm 2” state, but without executing again the action of alarm 2). On the other hand, while it is in “alarm 1”, triggering alarm 2 will have no effect: it will remain in “alarm 1” until alarm 1 gets deactivated (in that moment the fan coil module will perform the action of alarm 2).

ETS PARAMETERISATION

After enabling “**Alarms**” in the Configuration screen (see section 2.5.1), a new tab will be incorporated into the tab tree on the left.

Figure 8 Fan Coil – Alarms.

While the parameters of alarm no. 1 are shown by default, those of alarm 2 require marking the corresponding checkbox.

- **Deactivation – Mode:** “Normal” (default) or “Frozen (Acknowledgement Needed)”. The second option enables a new 1-bit object, “[FCV1] Unfreeze Alarm”, which should be used for externally unfreezing the alarm once “[FCV1] Alarm” has received the no-alarm value (i.e., the inverse of the trigger value).

Note: *the acknowledgement should be sent necessarily after “[FCV1] Alarm” has acquired the no-alarm value. Sending it while the trigger value is still active will have no effect.*

- **Trigger:** sets the value (“1” –default– or “0”) that, when received from the KNX bus through object “[FCV1] Alarm” (or “[FCV1] Alarm 2” for alarm number 2), will be interpreted by the device as an alarm trigger and will therefore initiate the action configured below.

- **Cyclical Monitoring Period:** sets every how much time, at most, the alarm object should be updated from the bus after an initial reception has already taken place. If exceeded, and for safety reasons, the alarm action will be triggered as well. If this parameter is set to zero (default option), the cyclical monitoring function will remain disabled. Permitted values are 0 to 600 tenths of a second; 0 to 3600 seconds; 0 to 1440 minutes; and 0 to 24 hours.
- **Valve Position:** sets, in terms of percentage, the target opening level of the valve or valves, depending on whether the module has been configured as a four-pipe or a two-pipe fan coil (see section 2.5.1).
- **Fan Speed:** sets the target fan speed, depending on how many speeds have been parameterised (see section 2.5.1).

2.5.1.2 FORCED POSITION

It is possible to define a **forced position** for the Cooling mode and another forced position for the Heating mode.

Forced positions consist in a **pre-set configuration** for the valve opening level and for the fan speed, so that the fan coil unit adopts such configuration when the forced-position trigger object receives the value “1”.

- One forced-position trigger object is provided **for heating and one for cooling** (unless only one of the two modes has been enabled in parameters).
- In order to make the fan coil unit adopt the corresponding forced position, the object being triggered **needs to be the one that corresponds to the current mode** (heating / cooling). Otherwise, it will be ignored.

Once under the forced-position state, the device will only respond to alarms (section 2.5.1.1), purge orders (section 2.5.1.3), mode switchovers (heating / cooling) and orders to switch the fan coil module on/off.

The device leaves the forced-position state until the trigger object receives the value “0”.

ETS PARAMETERISATION

After enabling **Forced Position** in the Configuration screen (see section 2.5.1), a new tab will be incorporated into the tree on the left.

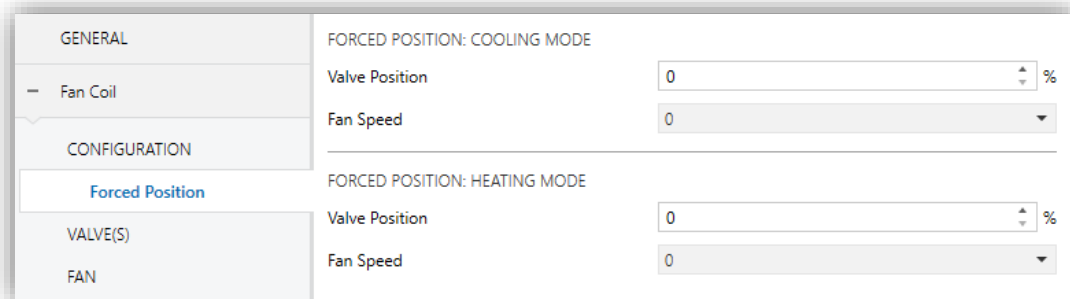


Figure 9 Fan Coil – Forced Position.

Moreover, objects “[FCV1] **Forced Position (Cooling)**”, “[FCV1] **Forced Position (Heating)**”, or both (depending on the parameterisation), will be incorporated to the project topology, together with their corresponding status objects.

Regarding the parameters:

- **Valve Position:** sets, in terms of percentage, the target opening level of the valve.
- **Fan Speed:** sets the target fan speed, depending on how many speeds have been parameterised (see section 2.5.1).

2.5.1.3 PURGE

The configuration of the valve allows **enabling a purge function** (one per valve), which consists in opening the selected valve and stopping the fan for a configurable time space.

The purge begins when one “1” is received through the corresponding object, and ends when the parameterised time elapses, when one “0” arrives, when a bus failure takes place, when the fan coil control module is turned off, or when the user enters the Test On mode (see section 2.8). To restart the process (from the beginning), the corresponding object should be written again with a “1”.

Notes:

- Any control order or any mode switch request received during the purge function **will be executed after the purge ends**.
- Any **alarm or switch-off** order will stop the purge process and will be executed. During the alarm or switch-off state, purge orders will be ignored.
- If no control orders are received, once the **purge** is over the fan coil control module **will recover the previous state**.

ETS PARAMETERISATION

After enabling **Purge** in the Configuration screen (see section 2.5.1), a new tab will be incorporated into the tree on the left.

Section	Parameter	Value	Unit
COOLING VALVE PURGE CONFIGURATION	Purge Duration	10	s
	Valve Position	100	%
HEATING VALVE PURGE CONFIGURATION	Purge Duration	10	s
	Valve Position	100	%

Figure 10. Fan coil - Purge.

Two new communication objects are also incorporated (together with their corresponding status objects).

- “[FCV1] Valve: Purge Heating Valve”,
- “[FCV1] Valve: Purge Cooling Valve”.

If only one valve is available, they will be replaced by a sole object: “[FCV1] Valve: Purge”.

Writing the value “1” to any of these objects triggers the purge of the corresponding valve, while value “0” interrupts it.

The following parameters are shown:

- **Purge Duration:** time lapse the valve will remain in the purge state once the function has been triggered (from 10 to 3600 seconds, where 10 seconds is the default value).
- **Valve Position:** valve opening level during the purge process (value from 0 to 100%, where 100% is the default value).

2.5.1.4 SCENES

It is possible to define up to **five scenes** so that, when the corresponding value is received from the bus, the module adopts a certain state, which must be defined in terms of the following options:

- **On / Off state** of the module.
- Activation / deactivation of the **Ventilation** mode.
- **Manual / Auto** mode for the fan speed selection.
- **Fan speed** (in case the above option is set to manual).
- Activation / deactivation of the **silent mode**.

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 of the fan coil module when the scene is executed.

For example, if a manual speed selection order is received during the valve purge process, the order will be buffered and executed after the completion of the purge, as it would happen in case of receiving the request through the analogous communication object.

On the other hand, if the scene in execution implies turning the fan coil module off, the response will be immediate, and the purge process will be aborted.

This module does not permit scene recording.

ETS PARAMETERISATION

Once Scenes has been enabled from the Configuration tab (see section 2.5.1), a new tab named “Scenes” will be available in the tab tree on the left, containing the following options:

Scene	Scene Number	Fan Coil State	Ventilation Mode (Mode Not Enabled)	Manual/Auto Mode	Silent Mode (Requires both Silent Mode and Automatic Mode Enabled)	Manual Fan Speed (Requires Manual Mode)
Scene 1	1	No Change	No Change	No Change	No Change	No Change
Scene 2						
Scene 3						
Scene 4						
Scene 5						

Figure 11. Fan Coil - Scenes.

● **Scene “n”**: enables or disables scene “n”, which should be then configured through the following additional parameters:

- **Scene number**: sets the value (1-64) that, when received through object “[FCV] Scenes”, will trigger the adoption of the configuration defined next.
- **Fan Coil State**: may be set to “Turn On”, “Turn Off” or “No Change” (default value).
- **Ventilation Mode**: may be set to “Activate”, “Deactivate” or “No Change” (default value).
- **Manual/Auto Mode**: may be set to “Automatic”, “Manual” or “No Change” (default value).
- **Silent Mode**: may be set to “Activate”, “Deactivate” or “No Change” (default value).
- **Manual Fan Speed**: may be set to any of the fan speeds available (see section 2.5.1). Note that this option is only customisable if **Manual/Auto Mode** is set to “Manual”. Otherwise, it will remain as “No Change”.

Note: depending on the settings of the fan coil module, some of these options may not be customisable.

2.5.2 VALVES

The valve control of the fan coil module entails a **scaling algorithm** of the input control signals (or **ICS**), i.e., the percentage control variables for heating and for cooling that are received from the thermostat, which itself may be external or any of the thermostats incorporated in MAXinBOX FC 0-10V VALVE.

The result of the scaling algorithm and of the application of any other required restrictions makes up the output control signals (or **OCS**), i.e., the percentage control values (or 0-10V signals) that will govern the valves.

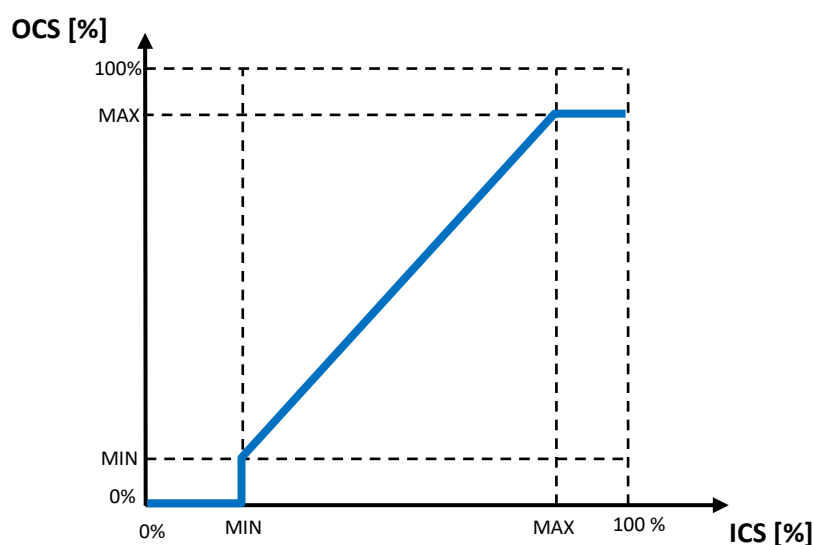


Figure 12. Fan Coil – Control signal scaling.

This scaling algorithm can be widely customised by the integrator, being possible to define for each of the modes / valves:

- An **effective input range** (i.e., the minimum and maximum values of the ICS),
- An **effective output range** (i.e., the minimum and maximum values of the OCS),

In addition, the valve control can be customised through the following options:

- A **minimum spacing** between successive OCS orders,
- A delay in the **mode switchovers**,
- The **minimum valve opening** level.

ETS PARAMETERISATION

GENERAL	ICS = Input Control Signal (from Thermostat) OCS = Output Control Signal (to Actuator)	
— Fan Coil		
CONFIGURATION		
VALVE(S)		
FAN		
CONTROL SIGNAL SCALING		
Cooling Valve		
Minimum of the ICS	0	%
Maximum of the ICS	100	%
Minimum of the OCS	0	%
Maximum of the OCS	100	%
Heating Valve		
Minimum of the ICS	0	%
Maximum of the ICS	100	%
Minimum of the OCS	0	%
Maximum of the OCS	100	%
CONTROL SIGNAL CHANGE LIMITS		
Cooling Valve		
Minimum Change Between Two Successive OCS Orders	5	%
Minimum Time Between Two Successive OCS Orders	10	
	<input checked="" type="radio"/> s <input type="radio"/> min	
Heating Valve		
Minimum Change Between Two Successive OCS Orders	5	%
Minimum Time Between Two Successive OCS Orders	10	
	<input checked="" type="radio"/> s <input type="radio"/> min	
MODE SWITCH DELAYS:		
Heating to Cooling Switch Delay	5	
	<input checked="" type="radio"/> s <input type="radio"/> min	
Cooling to Heating Switch Delay	5	
	<input checked="" type="radio"/> s <input type="radio"/> min	
ADVANCED VALVE CONFIGURATION		
Minimum Valve Opening	<input type="checkbox"/>	

Figure 13. Fan Coil – Valve(s).

The Valve(s) screen (see Figure 13) is accessible by default after enabling the fan coil module. It contains the following parameters:

CONTROL SIGNAL SCALING

- **Minimum of the ICS (from Thermostat):** sets the minimum value of the input signal, in terms of percentage, **that is desired** to make the valve open. For lower values, the valve will be left closed. The default value is "0%".
- **Maximum of the ICS (from Thermostat):** sets the maximum value of the input signal, in terms of percentage, **that is desired** to make the valve keep opening. Input values greater than this will be ignored as the valve is considered already open. The default value is "100%".
- **Minimum of the OCS (to Actuator):** sets the minimum voltage or value of the output signal, in terms of percentage, **that causes** the valve begins to open (lower voltages have no effect on the valve). This is valve-specific. The default value is "0%".
- **Maximum of the OCS (to Actuator):** sets the maximum voltage or value of the output signal, in terms of percentage, **that causes** the valve keeps opening (greater voltages have no effect on the valve). This is valve-specific. The default value is "100%".

The above four parameters are available **for each valve** (in case of having two valves).

Notes:

- *Please ensure to set minimum values that are lower than the maximum values.*
- *This scaling does not apply to other functions with a higher priority, such as **Alarms and Purge** (please see ANNEX I: Fan Coil Priorities).*

CONTROL SIGNAL CHANGE LIMITS

- **Minimum Change Between Two Successive OCS Orders:** sets the minimum change of the OCS (in terms of percentage, between 0% and 10%) that should make the device actually operate the valve (local control) or update the valve control objects (remote control). This prevents continuous operation of the valve in case the OCS changes frequently.

- **Minimum Time Between Two Successive OCS Orders:** sets a minimum time space (from 0 to 3600 seconds or from 0 to 1440 minutes) between consecutive updates of the valve position, either by operating on the valve (local control) or by sending the valve control objects (remote control). This prevents continuous operation of the valve in case the OCS changes frequently.

The above two parameters are available **for each valve** (in case of having two valves).

MODE SWITCH DELAYS

These delays apply to the mode switchovers (heating to cooling, and vice versa), and consist in waiting a certain time **before opening the valve of the target mode** after closing the valve of the source mode.

- **Heating to Cooling Switch Delay:** from 0 to 3600 seconds and from 0 to 1440 minutes.
- **Cooling to Heating Switch Delay:** from 0 to 3600 seconds and from 0 to 1440 minutes.

ADVANCED VALVE CONFIGURATION

- **Minimum Valve Opening:** enables or disables a new tab to configure the minimum valve opening level. See section 2.5.2.1.

2.5.2.1 MINIMUM VALVE OPENING

Due to a variety of reasons (type of installation, safety, extreme temperatures...), some valves may require **imposing a minimum opening level**, to ensure they always remain open at least at that level.

MAXinBOX FC 0-10V VALVE permits defining a minimum opening level (i.e., a particular position) for each of the valves, so the valves never remain at lower positions, no matter what the OCS value is.

This function also entails defining a **threshold**. In such case:

- When **the current ICS is lower than this threshold**, the valve will remain completely closed.

- While **the ICS falls between the above threshold and the minimum ICS** (see section 2.5.2), the valve will remain at the minimum opening level.
- When the **ICS is greater than the minimum ICS**, then the valve will adopt the position determined by the usual valve control.

A **threshold of 0%** will guarantee that the valve is never completely closed. An additional option is provided to indicate whether this is also required **while the fan coil module is off**.

Note:

- In case it is set to a value **greater than the minimum OCS** (see section 2.5.2), then the latter will be considered as the effective minimum opening.

ETS PARAMETERISATION

The Minimum Position screen becomes available after enabling the corresponding option in the Valve(s) screen (see section 2.5.2):

GENERAL	COOLING MODE	
- Fan Coil	Minimum Cooling Mode Opening	0 %
CONFIGURATION	Activation Mode	<input checked="" type="radio"/> Threshold <input type="radio"/> Even With Fan Coil Off
VALVE(S)	Threshold	0 %
Minimum Posi...		
FAN	HEATING MODE	
	Minimum Heating Mode Opening	0 %
	Activation Mode	<input checked="" type="radio"/> Threshold <input type="radio"/> Even With Fan Coil Off
	Threshold	0 %

Figure 14. Fan Coil – Minimum Opening.

- **Minimum Cooling/Heating Mode Opening:** sets the minimum opening level (in terms of percentage) required by the valve, to ensure that it never switches to a lower position.
- **Activation mode:** sets whether this restriction should apply even when the fan coil module is off ("Even with Fan Coil Off") or not ("Threshold").

The option "Threshold" implies configuring the following parameter:

- **Threshold:** sets the value of the ICS threshold, in terms of percentage. If no threshold is required, please set it to "0%".

The above configuration is available **for each valve** (in case of having two valves).

2.5.3 FAN

The fan speed control of the fan coil module may be **manual** (the fan speed will be controlled externally; for instance, by the user) or **automatic**.

Under the automatic fan speed control, the fan speed will be controlled by the module itself. This is done according to **up to four threshold values** (depending on the number of available fan speeds), set by the integrator. These threshold values define the range of the ICS that will correspond to each available fan speed. In other words, the integrator must define **which fan speed** should be adopted while in the automatic mode **depending on the current value of the ICS**. A **hysteresis** can also be defined, to prevent continuous fan speed switchovers in case the ICS keeps moving around one of the threshold values.

The integrator can configure if the user (from a control interface) can **switch from one mode to another** or, on the contrary, if only one of the two modes will be available.

When both, the manual and the automatic control modes are permitted, a communication object will let **switching from one control mode to the other**, while the corresponding status object will reflect the current mode.

In addition, the integrator can configure:

- Whether the reception of manual fan speed orders while in the automatic mode should **make the fan control switch to manual mode** or not.
- A period of time so that, **in the absence of manual speed orders** during such time, the fan speed control switches to automatic.
- Whether the fan speed control should switch to automatic whenever the **heating / cooling mode changes**.

Apart from the fan speed control modes, this configuration involves the following options:

- The **minimum fan speed** permitted in manual mode while the valve is open. This must be configured for each of the valves. Please note that, to prevent damage to the installation, **speed 0 (fan off) is never available in manual mode while the valve is open**. This ensures that the fan remains in motion whenever the valve stays open.
- **Delays**. Please refer to section 2.5.3.1.
- **Silent mode**. Please refer to section 2.5.3.2.
- **Custom Start**. Please refer to section 2.5.3.3.

ETS PARAMETERISATION

The Fan screen (see Figure 15) is accessible by default after enabling the Fan Coil module. It contains the following parameters:

SPEED STATUS OBJECTS

- **Individual Speed Objects (1 Bit)**: enables/disables the “[FCV1] Fan: Speed x (Status)” binary objects, where “x” runs from 0 to 4 (or to the number of fan speed; see section 2.5.1). These objects show the value “1” when the speed level they refer to is active and “0” when not.
- **Enumeration Object (1 Byte)**: enables/disables the “[FCV1] Fan: Enumeration Speed (Status)” one-byte object, which will show values from 0 to 4 according to the current fan speed.
- **Percentage Object (1 Byte)**: enables/disables the “[FCV1] Fan: Percentage Speed (Status)” one-byte object, which will show values from 0% to 100% according to the current fan speed and to the KNX standard.

Available speeds	Speed	Percentage
One speed	0	0%
	1	1% – 100%
Two speeds	0	0%
	1	0.4% – 50.2%
	2	50.4% – 100%
Three speeds	0	0%
	1	0.4% – 33.3%
	2	33.7% – 66.6%
	3	67% - 100%

Table 4. Manual control through the percentage object

SPEED CONTROL CONFIGURATION

- **Fan Control Mode:** sets if the fan speed will be controllable in “Automatic”, “Manual” or “Automatic+Manual” mode.

In the latter case, the initial mode after the device start-up is automatic. Moreover, the one-bit object “[FCV1] Fan: Manual/Automatic” will permit switching the mode any time (value “0” for automatic and value “1” for manual), while “[FCV1] Fan: Manual/Automatic (Status)” will inform the bus whenever the mode changes.

- **Period of Time to Return to Auto Mode:** sets the time (from 0 to 1440 minutes or from 0 to 24 hours) after which, if no manual speed orders have been received, the fan control mode will switch back to automatic. The value “0” leaves this function disabled.

- **Switch to Manual Mode After Receiving a Manual Control Order:** sets whether the arrival of a manual fan speed order while in the automatic mode should switch the fan control to manual or not.

Note: *this parameter remains hidden unless at least one speed control object has been enabled. Please, read below.*

- **Return to Auto Mode After Cooling/Heating Mode Switch:** sets whether the arrival of a cooling/heating mode switch order should activate the automatic fan speed control, in case of being in manual mode.
- **Automatic/Manual Switch Object:** sets which value will activate automatic control and which manual control, when it is received through the object “[FCVn] Fan: Manual/Automatic”. The options are: “0 = Automatic; 1 = Manual” (default) or “0 = Manual; 1 = Automatic”.

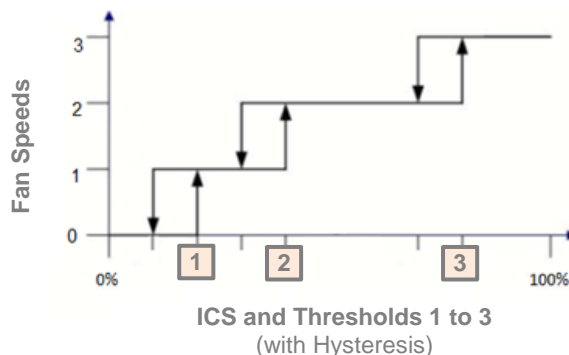
GENERAL	SPEED STATUS OBJECTS	
Fan Coil	Individual Speed Objects (1 Bit)	<input type="checkbox"/>
	Enumeration Object (1 Byte)	<input type="checkbox"/>
	Percentage Object (1 Byte)	<input type="checkbox"/>
CONFIGURATION	SPEED CONTROL CONFIGURATION	
VALVE(S)	Fan Control Mode	Automatic + Manual
FAN	Period of Time to Return to Auto Mode (0 = Disable)	0
		s
	Return to Auto Mode After Cooling/Heating Mode Switch	<input checked="" type="checkbox"/>
	Automatic/Manual Switch Object	<input checked="" type="radio"/> 0 = Automatic; 1 = Manual <input type="radio"/> 0 = Manual; 1 = Automatic
	SPEED THRESHOLDS (AUTO MODE)	
	Speed 1 Threshold	1 %
	Speed 2 Threshold	26 %
	Speed 3 Threshold	51 %
	Speed 4 Threshold	76 %
	Hysteresis	5 %
Note: "Speed 1 Threshold" Only for Heating Mode (for Cooling Mode = 0%)		
SPEED CONTROL OBJECTS (MANUAL MODE)		
	Individual Control (1 Bit)	<input type="checkbox"/>
	Enumeration Control (1 Byte)	<input type="checkbox"/>
	Percentage Control (1 Byte)	<input type="checkbox"/>
	Step Control (1 Bit)	<input type="checkbox"/>
MINIMUM SPEED (MANUAL MODE)		
	Minimum Manual Speed With Open Valve in Cooling Mode	1
	Minimum Manual Speed With Open Valve in Heating Mode	1
ADVANCED FAN CONFIGURATION		
	Delays	<input type="checkbox"/>
	Silent Mode	<input type="checkbox"/>
	Custom Start	<input type="checkbox"/>

Figure 15. Fan Coil – Fan Configuration.

SPEED THRESHOLDS (AUTO MODE)

- **Speed X Threshold:** sets the value of the ICS (in terms of percentage) that, being in the fan speed number X-1, will trigger speed number X while in the

automatic mode. In other words, the successive thresholds define different ranges for the ICS, each one corresponding to a particular fan speed.



- **Hysteresis:** sets a hysteresis (from 0% to 5%) for the upper thresholds. This helps avoiding continuous speed changes in case the ICS keeps moving around one of the threshold values.

SPEED CONTROL OBJECTS (MANUAL MODE)

Regarding the Manual Mode, the integrator must select the type(s) of the communication object(s) that will permit commuting the fan speed:

- **Individual Control (1 Bit)**, which enables one binary object (“[FCV1] Manual Fan: Speed X”) per fan speed. This object will activate the corresponding speed on the reception of the value “1”.
- **Enumeration Control (1 byte)**, which enables a one-byte object (“[FCV1] Manual Fan: Percentage Control”) that will switch the fan speed upon the arrival of the proper integer value (0, 1, 2, 3, 4).

Note: to prevent damage to the installation, under the manual fan control, speed 0 (fan off) will not be available while the valve is open.

- **Percentage Control (1 Byte)**, which enables a one-byte object (“[FCV1] Manual Fan: Percentage Control”) that will switch the fan speed upon the arrival of the proper percentage value according to the KNX standard (see Table 4).
- **Step Control (1 Bit)**, which enables a one-bit object (“[FCV1] Manual Fan: Step Control”) for increasing (value “1”) or decreasing (value “0”) the current

speed level sequentially. This sequence can be either **cyclical** (a further step once reaching the maximum level activates the minimum level again) or not.



Figure 16. Non-cyclical fan step control (with three fan speeds).

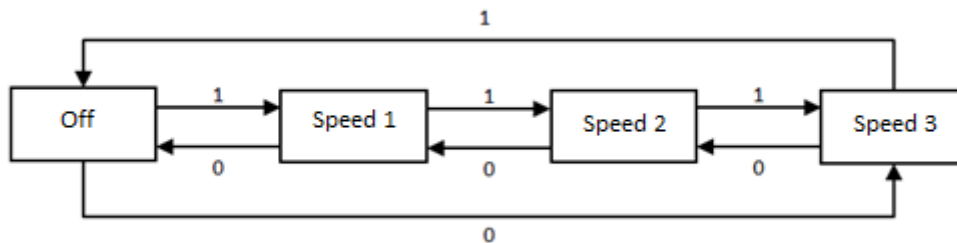


Figure 17. Cyclical fan step control. (with three fan speeds).

If desired, the above sequence can incorporate as an additional state **the automatic speed mode**, so the user can enter this mode through the same fan speed control. In such case, the following options are possible:

- Under a cyclical control: the automatic speed control will be an intermediate state between the two ends of the sequence.
- Under a non-cyclical control: an addition parameter (**Auto Step Position**) will let the integrator define the specific position of the automatic speed control within the sequence.
 - “Auto ↔ Speed 0 ... Speed Max ↔ Auto”: the automatic mode will correspond to the initial and final steps of the sequence.

- “Auto ↔ Speed 0 ... Speed Max”: the automatic mode will correspond to the first step of the sequence.
- “Speed 0 ... Speed Max ↔ Auto”: the automatic mode will be the final step of the sequence.

MINIMUM SPEED (MANUAL MODE)

- **Minimum Manual Speed With Open Valve in Cooling Mode:** sets a minimum fan speed to ensure the fan never adopts lower speeds while the cooling valve is open and the speed control remains manual. Note that speed 0 is not permitted in any case while the valve is open.
- **Minimum Manual Speed With Open Valve in Heating Mode:** analogous to the above parameter, but for the heating valve.

ADVANCED FAN CONFIGURATION

- **Delays:** enables the “Delays” function. See section 2.5.3.1.
 - **Silent Mode:** enables the “Silent Mode” function. See section 2.5.3.2.
- Note:** *this function can only be enabled if the automatic fan speed control is available.*
- **Custom Start:** enables the “Custom Start” function. See section 2.5.3.3.

2.5.3.1 DELAYS

It is possible to parameterise a series of delays over the fan operation:

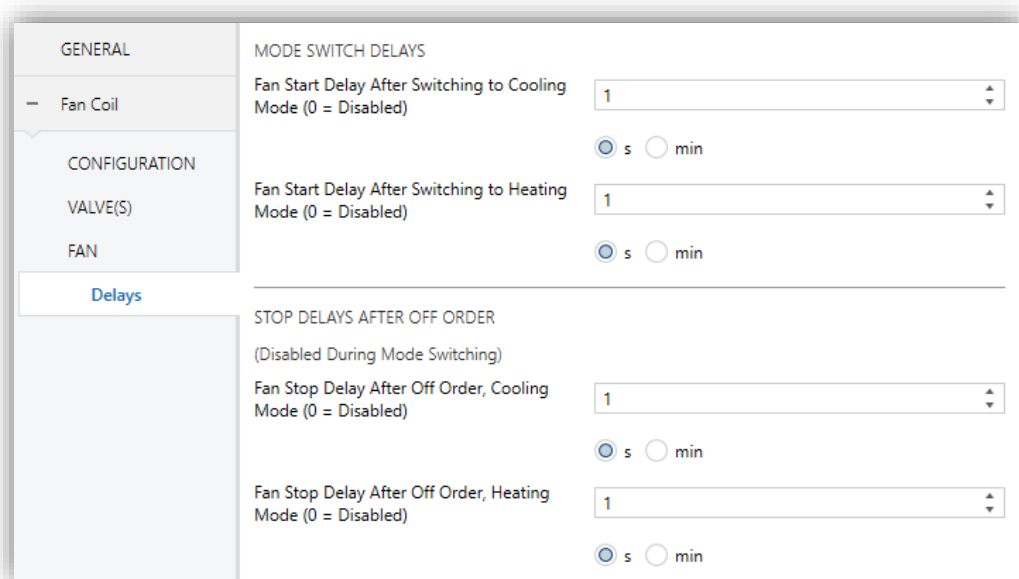
- **Mode Switch Delays:** these delays will apply to the fan start-up after a heating/cooling mode switchover, which can be useful to improve the comfort of the user (immediate ventilation after the mode switchover may be undesired).

Note that the mode switchover (i.e., opening one valve while closing the other valve) can itself be delayed (see section 2.5.2). In such case, the total time the fan will wait before responding to fan speed control orders will be the result of summing up both delays.

- **Stop Delays:** in case the fan coil module has not been configured to remain always on, these delays will apply to the switch-off of the fan after the module itself is turned off.

In both cases, it is possible to configure different delays for each mode (cooling / heating).

ETS PARAMETERISATION



GENERAL	MODE SWITCH DELAYS	
Fan Coil	Fan Start Delay After Switching to Cooling Mode (0 = Disabled)	
	1	
	<input checked="" type="radio"/> s <input type="radio"/> min	
	Fan Start Delay After Switching to Heating Mode (0 = Disabled)	
CONFIGURATION	1	
	<input checked="" type="radio"/> s <input type="radio"/> min	
	STOP DELAYS AFTER OFF ORDER (Disabled During Mode Switching)	
	Fan Stop Delay After Off Order, Cooling Mode (0 = Disabled)	
VALVE(S)	1	
	<input checked="" type="radio"/> s <input type="radio"/> min	
	Fan Stop Delay After Off Order, Heating Mode (0 = Disabled)	
	1	
FAN	<input checked="" type="radio"/> s <input type="radio"/> min	

Figure 18. Fan Coil – Delays.

The Delays screen is accessible after enabling the Delays function (see section 2.5.3). It contains the following parameters:

- **Fan Start Delay After Switching to Cooling Mode:** sets a delay (from 0 to 3600 seconds and from 0 to 1440 minutes) after commuting from heating to cooling and before the fan can be switched on.
- **Fan Star Delay After Switching to Heating Mode:** analogous to the above parameter, but after switching from cooling to heating.
- **Fan Stop Delay After Off Order, Cooling Mode:** sets a delay (from 0 to 3600 seconds and from 0 to 1440 minutes) after turning off the fan coil module in the cooling mode, and before the fan is switched off. In other

words, this sets an additional ventilation time after the fan coil module is turned off while in the cooling mode.

- **Fan Stop Delay After Off Order, Heating Mode:** analogous to the above parameter, but for the heating mode.

2.5.3.2 SILENT MODE

The Silent Mode function permits restricting the maximum fan speed **during the automatic fan speed** control. This guarantees that **the fan noise level remains low**.

To that end, one specific binary object is provided to permit entering / leaving the silent mode when desired, as well as the corresponding status object. Note that **entering the silent mode implies switching to the automatic fan speed control**, in case the manual speed control is active.

The fan coil module will also leave the silent mode if an alarm or an order with a **higher priority** (a switch-off, a purge, or a heating/cooling mode switch). Please refer to *ANNEX I: Fan Coil Priorities*.

On the other hand, **entering the manual fan speed control when the silent mode is active** will make the device ignore the latter, although it will be taken into account again after returning to the automatic fan speed control.

Note: *the silent mode function cannot be enabled in parameters if the automatic fan speed control is not available.*

ETS PARAMETERISATION

The Silent Mode screen is accessible after enabling such function (see section 2.5.3). It contains the following parameter:

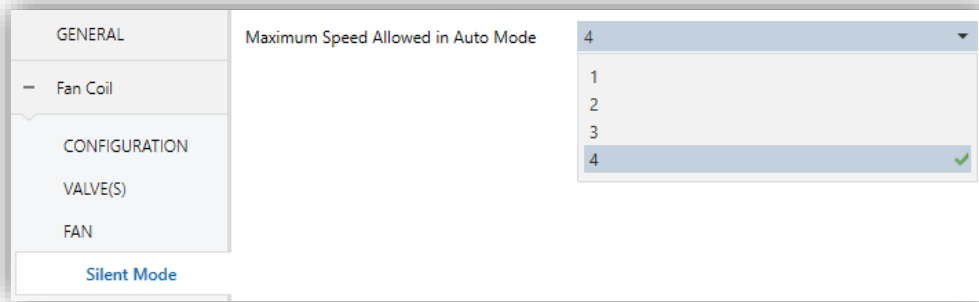


Figure 19. Fan Coil – Silent Mode

- **Maximum Speed Allowed in Auto Mode:** sets the maximum speed permitted for the fan while the silent mode remains active (as long as the fan speed control remains automatic).

The silent mode can be entered or left by sending the values “1” and “0” (respectively) to the binary object “[FCV1] Silent Mode”. On the other hand, “[FCV1] Silent Mode (status)” will show whether this mode is currently active or not.

2.5.3.3 CUSTOM START

In certain systems, it may be a requirement that, whenever the fan is turned on after being idle, **it remains at a certain speed for some instants**, to ensure enough torque prior to normal operation (i.e., prior to switching to the fan speed determined by the fan coil module or by the user).

MAXinBOX FC 0-10V VALVE provides the custom start function to meet such requirement.

ETS PARAMETERISATION

The Custom Start screen is accessible after enabling such function (see section 2.5.3). It contains the following parameters:

The screenshot shows a software interface for configuring a Fan Coil. On the left is a sidebar with a tree view containing 'GENERAL', 'Fan Coil', 'CONFIGURATION', 'VALVE(S)', and 'FAN'. 'Fan Coil' is currently selected. The main panel displays two settings: 'Custom Start Speed' with a dropdown menu showing the value '4', and 'Custom Start Duration' with a numeric input field showing '3' and a unit selector set to 's'. At the bottom of the main panel is a blue button labeled 'Custom Start'.

Figure 20. Fan Coil – Custom Start

- **Custom Start Speed:** sets the speed the fan must remain at, during the start-up.
- **Custom Start Duration:** sets the time (from 0 to 3600 seconds) the fan must remain at the above speed, prior to resuming normal operation.

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 VALVE 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 VALVE product section at the Zennio homepage, www.zennio.com for detailed information.

2.7 THERMOSTATS

MAXinBOX FC 0-10V VALVE 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 VALVE product section at the Zennio homepage, www.zennio.com for detailed information.

2.8 MANUAL CONTROL

MAXinBOX FC 0-10V VALVE 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.

● **Relay outputs:**

- If the output has been configured as an **individual 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.
- If the output has been configured as **part of the fan coil module** (i.e., the output is required for the fan speed control), pressing on the pushbuttons will switch the fan speeds according to the parameterisation (relay switching / relay accumulation, minimum speed, etc.) and the current state of the valve.

- **0-10V analogue outputs:** analogue outputs configured as part of the fan coil module (i.e., the output is required for a local valve control) are not controllable manually.

On analogue outputs specifically configured as individual analogue outputs, the action will depend 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: 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 decreasing output signal), 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 any other secondary 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:** the behaviour is analogous as in the Test Off mode, although the status objects will not change.
- **0-10V analogue output:** the behaviour is the same as described for the Test Off Mode, although the status objects will not change.
- **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).

Any other secondary 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).

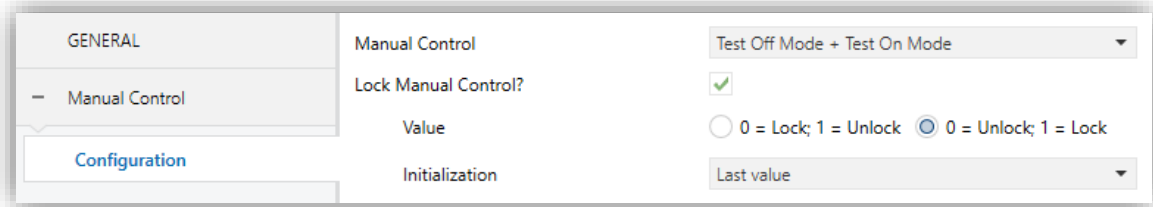


Figure 21 Manual Control.

The only two parameters are:

- **Manual Control:** options are “Disabled”, “Only Test Mode Off”, “Only Test Mode On” 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.
- **Lock Manual Control:** 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).*

The screenshot displays the configuration window for a Master Light. It features a sidebar with 'GENERAL' and 'Master Light' tabs, and a main area with a 'Configuration' sub-tab. The configuration is divided into two sections: 'General Switch Off' and 'Courtesy Switch On'. Each section includes a 'Delay' field (set to 0 with a 'x 1 s.' multiplier), a 'Binary Value' checkbox (checked with a green checkmark), and three unchecked checkboxes for 'Scaling', 'Scene', and 'HVAC'. The 'Number of State Objects' is set to 1, and the 'Trigger Value' is set to 0/1.

Section	Parameter	Value
General Switch Off	Number of State Objects	1
	Trigger Value	0/1
	Delay	0 x 1 s.
	Binary Value	<input checked="" type="checkbox"/>
	Scaling	<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 22 Master Light.

ANNEX I: FAN COIL PRIORITIES

The following diagram shows the priority (ordered from higher to lower) of the different functions or orders in MAXinBOX FC 0-10V Valve.

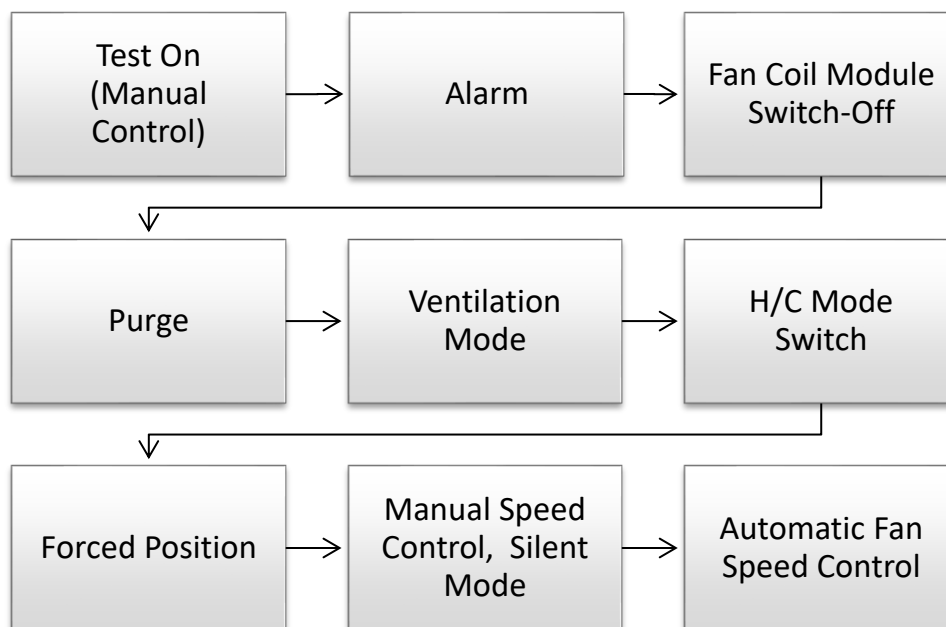


Figure 23 Fan Coil – Priorities.

ANNEX II: 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 programme 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 T R - -	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_Ack	0/1	[Tx] Special Mode: Standby	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Special Mode: Standby	0 = Off; 1 = On
11, 41	1 Bit	I	C - - W -	DPT_Ack	0/1	[Tx] Special Mode: Economy	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Special Mode: Economy	0 = Off; 1 = On
12, 42	1 Bit	I	C - - W -	DPT_Ack	0/1	[Tx] Special Mode: Protection	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Special Mode: Protection	0 = Off; 1 = On
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_Ack	0/1	[Tx] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
15, 45	1 Byte	O	C T R - -	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	CTR--	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Setpoint Status	Current Setpoint
20, 50	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00° - 670433.28°	[Tx] Basic Setpoint Status	Current Basic Setpoint
21, 51	2 Bytes	O	CTR--	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] Setpoint Reset	Reset Setpoint to Default
	1 Bit	I	C--W-	DPT_Reset	0/1	[Tx] Offset Reset	Reset offset
23, 53	1 Bit	I	C--W-	DPT_Heat_Cool	0/1	[Tx] Mode	0 = Cool; 1 = Heat
24, 54	1 Bit	O	CTR--	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	CTR--	DPT_Switch	0/1	[Tx] On/Off Status	0 = Off; 1 = On
27, 57	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] Control Variable (Cool)	PI Control (Continuous)
28, 58	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] Control Variable (Heat)	PI Control (Continuous)
29, 59	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Cool)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Cool)	PI Control (PWM)
30, 60	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Heat)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Heat)	PI Control (PWM)
31, 61	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Additional Cool	Temp >= (Setpoint+Band) => "1"
32, 62	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Additional Heat	Temp <= (Setpoint-Band) => "1"
33, 63	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
34, 64	1 Bit	O	CTR--	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	CTR--	DPT_Value_Temp	-273.00° - 670433.28°	[Ix] Current Temperature	Temperature Sensor Value
66, 70, 74, 78	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm; 1 = Alarm
67, 71, 75, 79	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm; 1 = Alarm
68, 72, 76, 80	1 Bit	O	CTR--	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		CT---	DPT_SceneControl	0-63; 128-191	[Motion Detector] Scene Output	Scene Value
83, 112, 141, 170	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
84, 113, 142, 171	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
85, 114, 143, 172	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
86, 115, 144, 173	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
87, 116, 145, 174	1 Byte	O	CTR--	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	CTR--	DPT_Occupancy	0/1	[Ix] Presence State (Binary)	Binary Value
	1 Bit	O	CTR--	DPT_Ack	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_Ack	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	1 - 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	CTR - -	DPT_Occupancy	0/1	[Ix] Presence: Occupancy State	0 = Not Occupied; 1 = Occupied
96, 125, 154, 183	1 Bit	I	C - - W -	DPT_Ack	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	CTR - -	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	CTR - -	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	CTR - -	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	CTR - -	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=Normal; 1=Alarm
	1 Bit	I	C - - W -	DPT_Alarm	0/1	[Ox] Alarm	0=Alarm; 1=Normal
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	CTR - -	DPT_State	0/1	[Ox] Warning Time (Status)	0=Normal; 1=Warning
208, 219, 230, 241	4 Bytes	I/O	CTR W -	DPT_LongDeltaTimeSec	0 - 2147483647	[Ox] Operating Time (s)	Time in Seconds
209, 220, 231, 242	2 Bytes	I/O	CTR W -	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	1 Bit		CT - - -	DPT_Switch	0/1	[Ix] [Short Press] 0	Sending of 0
	1 Bit		CT - - -	DPT_Switch	0/1	[Ix] [Short Press] 1	Sending of 1
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Ix] [Short Press] 0/1 Switching	Switching 0/1

1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Up Shutter	Sending of 0 (Up)
1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Down Shutter	Sending of 1 (Down)
1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
4 Bit		CT---	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
4 Bit		CT---	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
4 Bit		CT---	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 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] Light On	Sending of 1 (On)
1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] Light Off	Sending of 0 (Off)
1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Short Press] Light On/Off	Switching 0/1
1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Run Scene	Sending of 0 - 63
1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Save Scene	Sending of 128 - 191
1 Bit	I/O	CTRW-	DPT_Switch	0/1	[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix] [Short Press] Constant Value (Integer)	0 - 255
1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix] [Short Press] Constant Value (Integer)	0 - 65535
2 Bytes		CT---	9.xxx	-671088.64 - 670433.28	[Ix] [Short Press] Constant Value (Float)	Float Value

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	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	4 Bit		CT---	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		CT---	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		CT---	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		CT---	DPT_Switch	0/1	[Ix] [Long Press] Light On	Sending of 1 (On)
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] Light Off	Sending of 0 (Off)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Long Press] Light On/Off	Switching 0/1
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Run Scene	Sending of 0 - 63
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Save Scene	Sending of 128 - 191
	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] [Switch/Sensor] Alarm: Breakdown or	1 = Alarm; 0 = No Alarm

						Sabotage	
	2 Bytes		CT---	9.xxx	-671088.64 - 670433.28	[Ix] [Long Press] Constant Value (Float)	Float Value
	2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix] [Long Press] Constant Value (Integer)	0 - 65535
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix] [Long Press] Constant Value (Percentage)	0% - 100%
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix] [Long Press] Constant Value (Integer)	0 - 255
247, 253, 259, 265	1 Bit		CT---	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] Dimming Status (Input)	0% - 100%
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix] [Long Press] Shutter Status (Input)	0% = Top; 100% = Bottom
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
				DPT_Value_2_Count	-32768 - 32767		
				DPT_Value_Tempo	-273.00 - 670433.28		
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	CTR--	DPT_Bool	0/1	[LF] Function x - Result	(1-Bit) Boolean
	1 Byte	O	CTR--	DPT_Value_1_Ucount	0 - 255	[LF] Function x - Result	(1-Byte) Unsigned
	2 Bytes	O	CTR--	DPT_Value_2_Ucount	0 - 65535	[LF] Function x - Result	(2-Byte) Unsigned
	4 Bytes	O	CTR--	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] Function x - Result	(4-Byte) Signed
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[LF] Function x - Result	(1-Byte) Percentage
	2 Bytes	O	CTR--	DPT_Value_2_Count	-32768 - 32767	[LF] Function x - Result	(2-Byte) Signed
	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00° - 670433.28°	[LF] Function x - Result	(2-Byte) Float
349, 352	1 Byte	I	C--WU	DPT_Scaling	0% - 100%	[AOx] Output Value (Control)	0 - 100 %
350, 353	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[AOx] Output Value (Status)	0 - 100 %
351, 354	1 Bit	I	C--WU	DPT_Enable	0/1	[AOx] Lock	0 = Unlock; 1 = Lock
355	1 Byte	I/O	CTR W-	DPT_SceneControl	0-63; 128-191	[FCVx] Scenes	0 - 63 (Execute 1 - 64)
356	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] On/Off	0 = Off; 1 = On
357	1 Bit	O	CTR--	DPT_Switch	0/1	[FCVx] On/Off (Status)	0 = Off; 1 = On
358	1 Bit	I	C--W-	DPT_Heat_Cool	0/1	[FCVx] H/C Mode	0 = Cool; 1 = Heat

359	1 Bit	O	CTR--	DPT_Heat_Cool	0/1	[FCVx] H/C Mode (Status)	0 = Cool; 1 = Heat
360	1 Bit	I	C--W-	DPT_Enable	0/1	[FCVx] Silent Mode	0 = Disable; 1 = Enable
361	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Silent Mode (Status)	0 = Inactive; 1 = Active
362	1 Bit	I	C--W-	DPT_Enable	0/1	[FCVx] Ventilation Mode	0 = Disable; 1 = Enable
363	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Ventilation Mode (Status)	0 = Inactive; 1 = Active
364	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Fan: Manual/Automatic	0 = Automatic; 1 = Manual
	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Fan: Manual/Automatic	0 = Manual; 1 = Automatic
365	1 Bit	O	CTR--	DPT_Switch	0/1	[FCVx] Fan: Manual/Automatic (Status)	0 = Automatic; 1 = Manual
	1 Bit	O	CTR--	DPT_Switch	0/1	[FCVx] Fan: Manual/Automatic (Status)	0 = Manual; 1 = Automatic
366	1 Bit	I	C--W-	DPT_Step	0/1	[FCVx] Manual Fan: Step Control	0 = Down; 1 = Up
367	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Manual Fan: Speed 0	0 = Off; 1 = On
368	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Manual Fan: Speed 1	0 = Off; 1 = On
369	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Manual Fan: Speed 2	0 = Off; 1 = On
370	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Manual Fan: Speed 3	0 = Off; 1 = On
371	1 Bit	I	C--W-	DPT_Switch	0/1	[FCVx] Manual Fan: Speed 4	0 = Off; 1 = On
372	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Fan: Speed 0 (Status)	0 = Inactive; 1 = Active
373	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Fan: Speed 1 (Status)	0 = Inactive; 1 = Active
374	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Fan: Speed 2 (Status)	0 = Inactive; 1 = Active
375	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Fan: Speed 3 (Status)	0 = Inactive; 1 = Active
376	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Fan: Speed 4 (Status)	0 = Inactive; 1 = Active
377	1 Bit	I	C--W-	DPT_Ack	0/1	[FCVx] Unfreeze Alarm	Alarm = Inactive + Unfreeze = 1 => End Alarm
378	1 Bit	I	C--W-	DPT_Alarm	0/1	[FCVx] Alarm 1: Trigger	0 = Active; 1 = Inactive
	1 Bit	I	C--W-	DPT_Alarm	0/1	[FCVx] Alarm 1: Trigger	0 = Inactive; 1 = Active
379	1 Bit	I	C--W-	DPT_Alarm	0/1	[FCVx] Alarm 2: Trigger	0 = Active; 1 = Inactive
	1 Bit	I	C--W-	DPT_Alarm	0/1	[FCVx] Alarm 2: Trigger	0 = Inactive; 1 = Active
380	1 Byte	I	C--W-	DPT_Value_1_Ucount	0 - 255	[FCVx] Manual Fan: Enumeration Control	0 = S0; 1 = S1
	1 Byte	I	C--W-	DPT_Value_1_Ucount	0 - 255	[FCVx] Manual Fan: Enumeration Control	0 = S0; 1 = S1; 2 = S2
	1 Byte	I	C--W-	DPT_Value_1_Ucount	0 - 255	[FCVx] Manual Fan: Enumeration Control	0 = S0; 1 = S1; 2 = S2; 3 = S3
	1 Byte	I	C--W-	DPT_Value_1_Ucount	0 - 255	[FCVx] Manual Fan: Enumeration Control	0 = S0; 1 = S1; 2 = S2; 3 = S3; 4 = S4
381	1 Byte	O	CTR--	DPT_Value_1_Ucount	0 - 255	[FCVx] Fan: Enumeration Speed (Status)	0 = S0; 1 = S1
	1 Byte	O	CTR--	DPT_Value_1_Ucount	0 - 255	[FCVx] Fan: Enumeration Speed (Status)	0 = S0; 1 = S1; 2 = S2
	1 Byte	O	CTR--	DPT_Value_1_Ucount	0 - 255	[FCVx] Fan: Enumeration Speed (Status)	0 = S0; 1 = S1; 2 = S2; 3 = S3
	1 Byte	O	CTR--	DPT_Value_1_Ucount	0 - 255	[FCVx] Fan: Enumeration Speed (Status)	0 = S0; 1 = S1; 2 = S2; 3 = S3; 4 = S4
382	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[FCVx] Manual Fan: Percentage Control	[0]% = S0; [0.4...100]% = S1
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[FCVx] Manual Fan: Percentage Control	[0]% = S0; [0.4...50.2]% = S1; [50.6...100]% = S2
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[FCVx] Manual Fan: Percentage Control	[0]% = S0; [0.4...33.3]% = S1; [33.7...66.7]% = S2; [67.1...100]% = S3

	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[FCVx] Manual Fan: Percentage Control	[0]% = S0; [0.4...25.1]% = S1; [25.5...50.2]% = S2; [50.6...75.3]% = S3; [75.5...100]% = S4
383	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Fan: Percentage Speed (Status)	S0 = 0%; S1 = 100%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Fan: Percentage Speed (Status)	S0 = 0%; S1 = 50.2%; S2 = 100%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Fan: Percentage Speed (Status)	S0 = 0%; S1 = 33.3%; S2 = 66.7%; S3 = 100%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Fan: Percentage Speed (Status)	S0 = 0%; S1 = 25.1%; S2 = 50.2%; S3 = 75.3%; S4 = 100%
384	2 Bytes	I/O	CTRW-	DPT_TimePeriodSec	0 - 65535	[FCVx] Manual Fan: Duration of Manual Override	0 = Endless; 1 - 3600 s
	2 Bytes	I/O	CTRW-	DPT_TimePeriodMin	0 - 65535	[FCVx] Manual Fan: Duration of Manual Override	0 = Endless; 1 - 1440 min
	2 Bytes	I/O	CTRW-	DPT_TimePeriodHrs	0 - 65535	[FCVx] Manual Fan: Duration of Manual Override	0 = Endless; 1 - 24 h
385	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[FCVx] Cooling Input Control Signal	[0...100]%
386	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[FCVx] Heating Input Control Signal	[0...100]%
387	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Cooling Valve (Status)	[0...100]%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Cooling Valve: Control Signal (Status)	[0...100]%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Valve (Status)	[0...100]%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Valve: Control Signal	[0...100]%
388	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Heating Valve (Status)	[0...100]%
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[FCVx] Heating Valve: Control Signal (Status)	[0...100]%
389	1 Bit	I	C--W-	DPT_Start	0/1	[FCVx] Valve: Purge Cooling Valve	0 = Stop; 1 = Start
389, 391	1 Bit	I	C--W-	DPT_Start	0/1	[FCVx] Valve: Purge	0 = Stop; 1 = Start
390	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Valve: Purge Cooling Valve (Status)	0 = Inactive; 1 = Active
390, 392	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Valve: Purge (Status)	0 = Inactive; 1 = Active
391	1 Bit	I	C--W-	DPT_Start	0/1	[FCVx] Valve: Purge Heating Valve	0 = Stop; 1 = Start
392	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Valve: Purge Heating Valve (Status)	0 = Inactive; 1 = Active
393	1 Bit	I	C--W-	DPT_Enable	0/1	[FCVx] Forced Position (Cooling)	0 = Disable; 1 = Enable
394	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Forced Position (Cooling) (Status)	0 = Inactive; 1 = Active
395	1 Bit	I	C--W-	DPT_Enable	0/1	[FCVx] Forced Position (Heating)	0 = Disable; 1 = Enable
396	1 Bit	O	CTR--	DPT_State	0/1	[FCVx] Forced Position (Heating) (Status)	0 = Inactive; 1 = Active
397	1 Bit		CT---	DPT_Trigger	0/1	[Heartbeat] Object to Send '1'	Sending of '1' Periodically
398	1 Bit	I	C--W-	DPT_Switch	0/1	[ML] Trigger	Trigger the Master Light Function
399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410	1 Bit	I	C--W-	DPT_Switch	0/1	[ML] Status Object x	Binary Status
411	1 Bit	O	CTR--	DPT_Switch	0/1	[ML] General Status	Binary Status
412	1 Bit		CT---	DPT_Switch	0/1	[ML] General Switch Off: Binary Object	Switch Off Sending
413	1 Byte		CT---	DPT_Scaling	0% - 100%	[ML] General Switch Off: Scaling	0-100%

414	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[ML] General Switch Off: Scene	Scene Sending
415	1 Byte		CT---	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[ML] General Switch Off: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection
416	1 Bit		CT---	DPT_Switch	0/1	[ML] Courtesy Switch On: Binary Object	Switch On Sending
417	1 Byte		CT---	DPT_Scaling	0% - 100%	[ML] Courtesy Switch On: Scaling	0-100%
418	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[ML] Courtesy Switch On: Scene	Scene Sending
419	1 Byte		CT---	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:

<http://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



RoHS