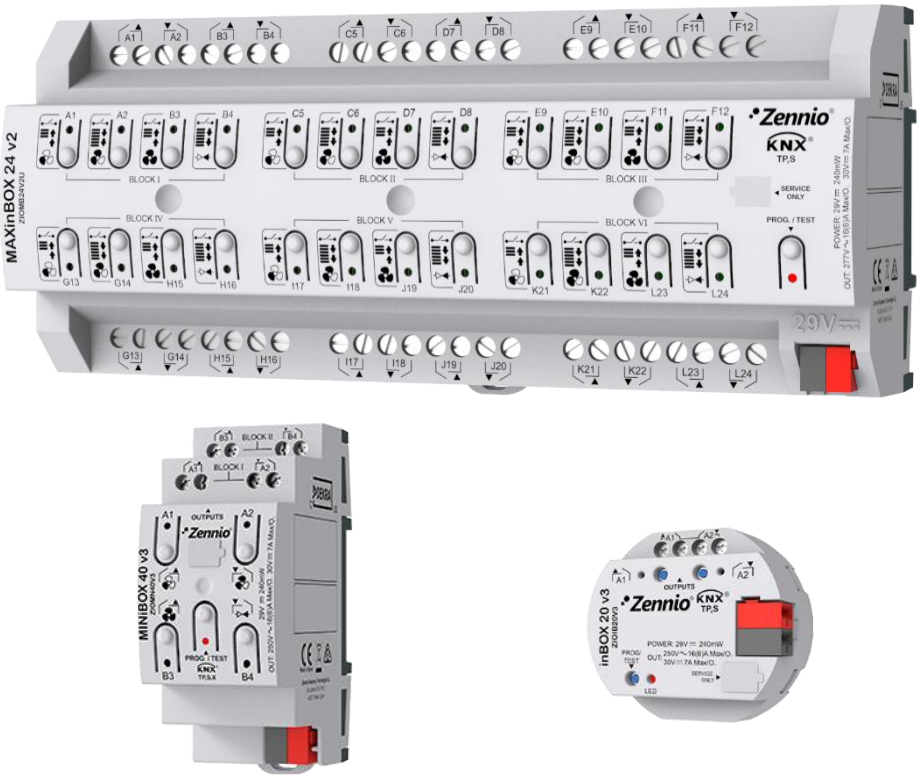



# Multifunction Output Actuators

## Output-only actuators

**ZIOMB24V2 / ZIOMB20 / ZIOMB16V4 / ZIOMB12 / ZIOMB8V4**
**ZIOMB24V2U / ZIOMB20U / ZIOMB16V4U / ZIOMB12U / ZIOMB8V4U**
**ZIOMN40V3 / ZIOMN20V2**
**ZIOIB20V3**

Application program versions: [2.2] [2.1]
User manual edition: [2.2]_a

www.zennio.com

USER MANUAL

# CONTENTS

Contents ..... 2
Document updates ..... 3
1 Introduction ..... 4
1.1 Multifunction Output Actuators ..... 4
1.2 Start-Up and Power Loss ..... 5
1.3 Comparative Table for the Different Actuators ..... 5
2 Configuration ..... 6
2.1 General ..... 6
2.2 Outputs ..... 8
2.2.1 Individual Binary Output ..... 8
2.2.2 Shutter Channel ..... 8
2.2.3 Fan Coil Modules ..... 8
2.3 Logic Functions ..... 8
2.4 Master Light ..... 9
2.5 Scene Timing ..... 12
2.6 Manual Control ..... 13
ANNEX I. Comparative table for devices ..... 18
ANNEX II. Communication objects ..... 19

# DOCUMENT UPDATES

| Version | Changes |
|---|---|
| [2.2]_a | Manual updated to include the latest references for the USA and Canadian markets.<br>Application program version 2.2 applies only to MAXinBOX 24 v2 / 20 / 16 v4 / 12 / 8 v4. |
| [2.1]_a | Unified manual for all Zennio Multifunction Output Actuators |
| | **Application Program changes:**<br>• Functional block updates |

# 1 INTRODUCTION

## 1.1 MULTIFUNCTION OUTPUT ACTUATORS

Multifunction output actuators from Zennio are versatile KNX secure actuators featuring a wide variety of functions:

- **Relay outputs**, configurable as:
  - Independent shutter channels (with or without slats),
  - Individual ON/OFF outputs,
  - Two-pipe fan coil modules where both fan speed control and valve control are via relays,
  - A combination of the above where supported. (For further information, see ANNEX I).
- **Customisable, multi-operation logic functions**.
- **Master light controls** for an easy, out-of-the-box control of a set of luminaires (or functionally equivalent devices) one of which acts as a general lamp and the others as secondary lamps.
- **Manual operation / supervision** of the relay outputs through the on-board pushbuttons and LEDs.
- **Heartbeat** or periodical 'still-alive' notification.
- **Scene-triggered action control,** with an optional delay in the execution.
- **KNX Secure**. For detailed information about the functionality and configuration of KNX secure, consult the specific user manual "KNX Secure Guide", available in the product section of the Zennio web portal (www.zennio.com).

## 1.2 START-UP AND POWER LOSS

During the start-up of the device, the Test/Prog. LED will flash blue colour for a few seconds before the device is ready. External orders will not be executed during this time, but afterwards.

Depending on the configuration, some specific actions will also be performed during the start-up. For example, it can be configured whether the output channels 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, the device will interrupt any pending actions and will save its state so it can be recovered once the power supply is restored.

For safety reasons, all **shutter channels** will be stopped (i.e., the relays will open) if a power loss takes place, while the individual outputs will switch to the specific state configured in ETS (if any).

## 1.3 COMPARATIVE TABLE FOR THE DIFFERENT ACTUATORS

Due to their common overall functionality, the following devices share this manual:

- **MAXinBOX 24 v2 / 20 / 16 v4 / 12 / 8 v4**
- **MINiBOX 40 v3 / 20 v2**
- **inBOX 20 v3**

However, they all have minor differences, which are shown in ANNEX I. Comparative table for devices.

# 2 CONFIGURATION

## 2.1 GENERAL

After importing the corresponding database in ETS and adding the device to the desired project topology, the configuration process is started by accessing the device parameters tab.

### ETS PARAMETERISATION

The only screen that can be parameterised by default is General. From this screen it is possible to activate/deactivate all the necessary functions.

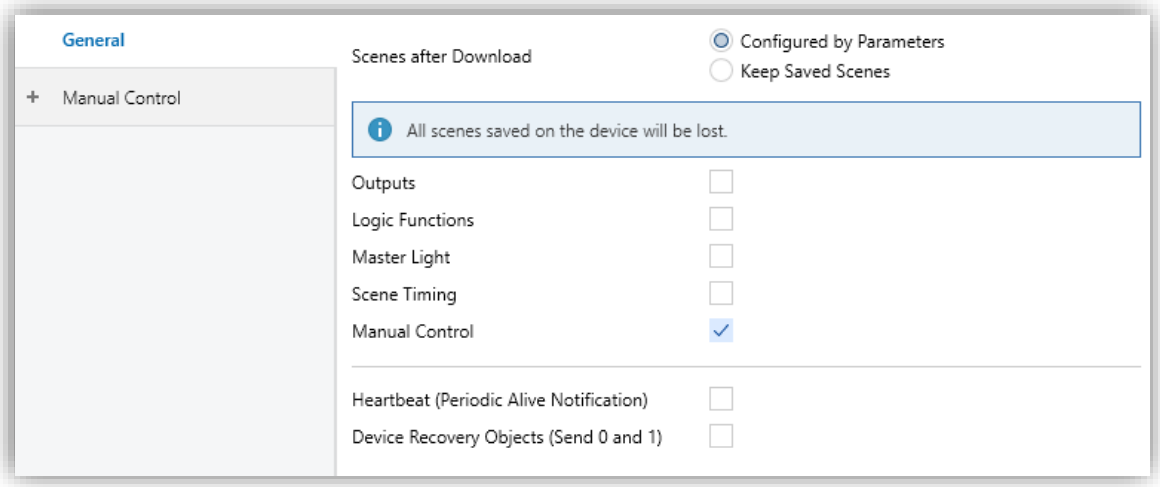



**Figure 1.** General tab

- **Scenes after Download** [*Configured by Parameters* / *Keep Saved Scenes*][1]: allows defining whether scene values are taken from the configured parameters or previously saved values are retained after a download.

  **Note**: if "Keep Saved Scenes" option has been configured, but it is the first download of the device or a different version from the current one, the values configured by parameter will be adopted. If new scenes are added in successive downloads, it will be necessary to perform a download by checking the option "Configured by Parameters" to ensure the correct operation of these scenes.

[1] The default values of each parameter will be highlighted in blue in this document, as follows: [*default* / rest of options].

- **Outputs** [*disabled / enabled*]: enables or disables the "Outputs" tab on the left menu. See section 2.2 for more details.

- **Logic Functions** [*disabled / enabled*]: enables or disables the "Logic Functions" tab on the left menu. See section 2.3 for more details.

- **Master Light** [*disabled / enabled*]: enables or disables the "Master Light" tab on the left menu. See section 2.4 for more details.

- **Scene Timing** [*disabled / enabled*]: enables or disables the "Scene Timing" tab on the left menu. See section 2.5 for more details.

- **Manual Control** [*disabled / enabled*]: enables or disables the "Manual Control" tab on the left menu. See section 2.6 for more details.

- **Heartbeat (Periodic Alive Notification)** [*disabled / enabled*]**:** 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*).

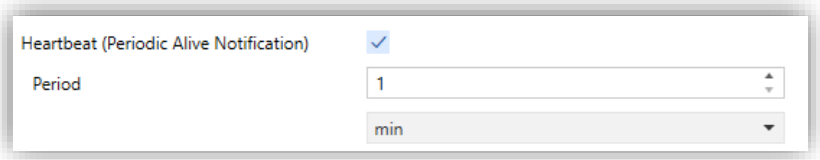



**Figure 2.** Heartbeat (Periodic Alive Notification)

**Note**: *the first sending after download or bus failure takes place with a delay of up to 255 seconds, to prevent bus overload. Subsequent transmissions follow the configured period.*

- **Device Recovery Objects (Send 0 and 1)** [*disabled / enabled*]: this parameter lets the integrator activate two new communication objects **("[Heartbeat] Device Recovery**"), which will be sent to the KNX bus with values "0" and "1" whenever the device begins operation (for example, after a bus power failure). It is possible to parameterise a certain **delay** [*0…255*] [*s*] to this sending.

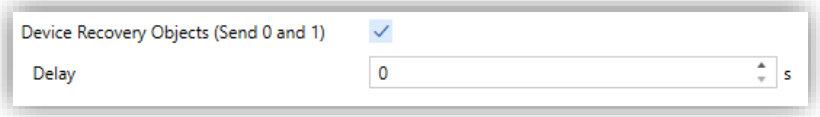



**Figure 3.** Device Recovery Objects

**Note**: *After download or bus failure, the sending takes place with a delay of up to 6.35 seconds plus the parameterised delay, to prevent bus overload.*

## 2.2 OUTPUTS

The devices incorporates **different numbers of relay outputs**, which can be consulted in ANNEX I. There are several possible configurations, which are explained below.

### 2.2.1 INDIVIDUAL BINARY OUTPUT

Independent control of a load. Between 2 and 24 outputs can be controlled, depending on the device.

Please refer to the "**Individual Binary Outputs**" user manual, available under the product section at www.zennio.com.

### 2.2.2 SHUTTER CHANNEL

Control of the motion of a blind, with or without slats. Between 1 and 12 independent blind channels can be controlled depending on the device being used, as each channel requires two relay outputs.

Please refer to the "**Shutter Channels**" user manual, available under the product section at www.zennio.com.

### 2.2.3 FAN COIL MODULES

Control of the fan and the valve of two-pipe fan coils. Up to 6 independent fan coil units can be controlled depending on the device, as shown in Table 1. The fan coil function is available on all devices except inBOX 20 v3 and MINiBOX 20 v2, as each unit requires 4 relay outputs.

Please refer to the 'Fan Coil Relays' user manual, available under the product section at www.zennio.com. Note that only two-pipe *fan coils* with On/Off valve can be controlled. References to four-pipe *fan coils* and 3-point valves do not apply to these devices.

## 2.3 LOGIC FUNCTIONS

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

Depending on the device being considered, **between 10 and 30 different and independent logical functions** can be implemented, which are fully customisable and consist of a **maximum of 4 consecutive operations for 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**" user manual available under the product section at the Zennio homepage (www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

## 2.4 MASTER LIGHT

The Master Light function brings the option to monitor the state of up to 30 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 a master command will consist of:

- A **general switch-off** order, if at least one of the up to thirty status objects is found to be on.
- A **courtesy switch-on** order, if none of the up to thirty 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 – the value to be sent to the KNX bus can be configured in each case: a shutter order, a thermostat setpoint or mode switch order, a constant value, a scene… Only the trigger object and the configured 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 the master pushbutton and make all the lamps turn off together.

On returning to the room and with all the lamps off, pressing 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.

In addition, 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. This makes it possible to, 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 (Figure 4) contains the following options:

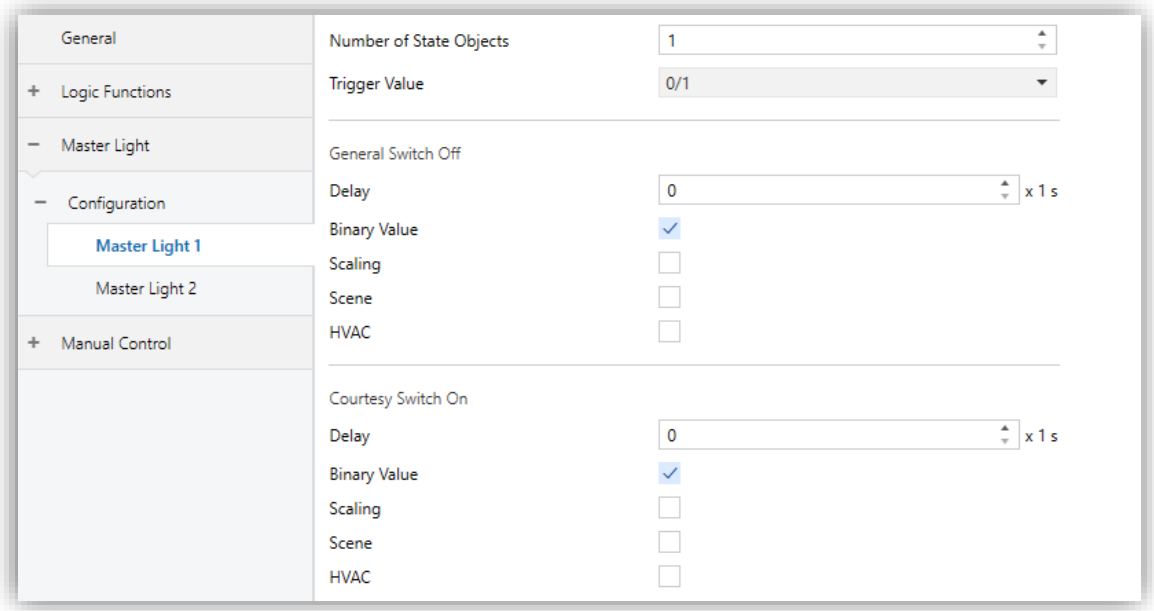



**Figure 4.** Master Light

- **Number of State Objects** [*1…30*]: defines the number of 1-bit status objects required. 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** [*0 / 1 / 0/1*]: sets the value that will trigger, when received through "**[ML] Trigger**", the master action (general switch-off or courtesy switch-on).

- **General Switch-Off**.

  - **Delay** [*0…255*] [*x 1 s*]: defines a certain delay (once the trigger has been received) before the execution of the general switch-off.

  - **Binary Value** [*disabled / enabled*]: if checked, object "**[ML] General Switch-off: Binary Object**" will be enabled, which will send one "0" whenever the general switch-off takes place.

  - **Scaling** [*disabled / enabled*]: if checked, object "**[ML] General Switch-off: Scaling**" will be enabled, which will send a percentage value (configurable in **Value** [*0…100*]) whenever the general switch-off takes place.

  - **Scene** [*disabled / enabled*]: if checked, object "**[ML] General Switch-off: Scene**" will be enabled, which will send a scene run / save order (configurable in **Action** [*Run / Save*] and **Scene Number** [*1…64*]) whenever the general switch-off takes place.

  - **HVAC** [*disabled / enabled*]: if checked, object "**[ML] General Switch-off: HVAC mode**" will be enabled, which will send an HVAC thermostat mode value (configurable in **Value** [*Auto / Comfort / Standby / Economy / Building Protection*] whenever the general switch-off takes place.

  **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 scene execution commands 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)*.

## 2.5 SCENE TIMING

The scene timing allows delays to be applied to **output scenes of the outputs**. These delays are defined in parameters and can be applied to the execution of one or more scenes that may have been configured.

Please bear in mind that, as multiple delayed scenes can be configured for each individual output / shutter channel / fan coil module, if an execution command is received to execute one of them when a previous temporisation is still pending for that output / channel / module, such temporisation will be interrupted and only the delay and the action of the new scene will be executed.

### ETS PARAMETERISATION

Prior to setting the **scene timing**, it is necessary to have one or more scenes configured in some of the outputs. When entering the Configuration window under Scene Timing, all configured scenes will be listed, together with a few checkboxes to select which of them need to be temporised, as shown in Figure 5.

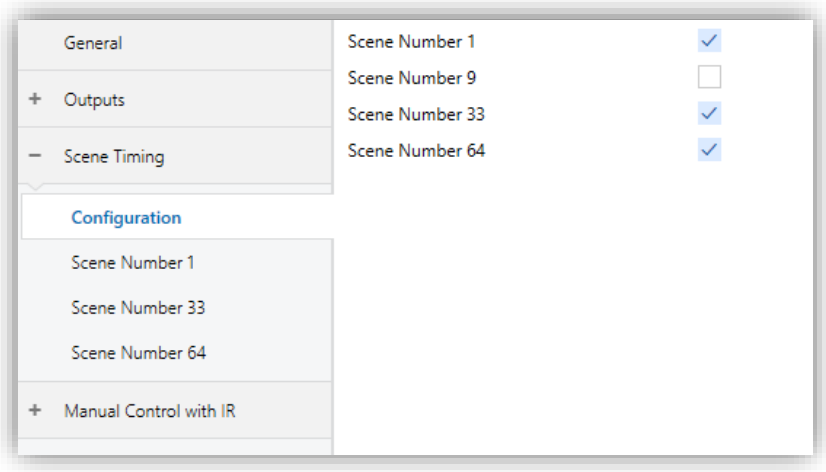



**Figure 5.** Scene timing

Enabling a certain scene number n brings a new tab with that name to the menu on the left, from which it is possible to configure the temporisation of that scene for each of the outputs where it has been configured.

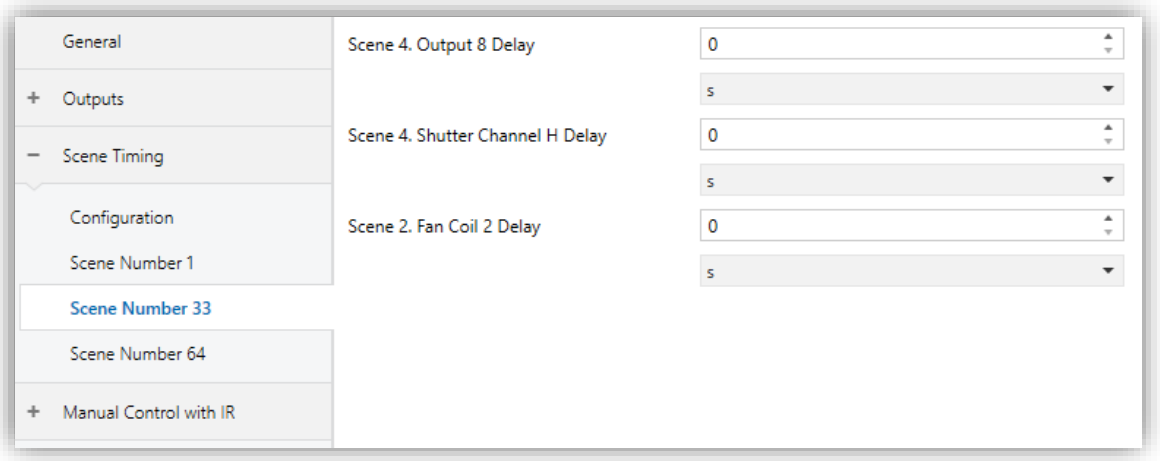



**Figure 6.** Configuration of Scene Timing

Therefore, parameter "**Scene m. Z Delay**" [*0…3600 [s] / 0…1440 [min] / 0…24 [h]*], defines the delay that will be applied to the action defined in Z for the execution of scene m (where Z may be a specific individual output, shutter channel or fan coil module).

**Note**: *In the configuration of a scene of an output / shutter channel / fan coil it is possible to parameterise several scenes with the same scene number. This means that several delay parameters associated with the same output appear in the configuration tab of the delays of that scene. With this parameterization, the behaviour will be as follows: the action and delay of the first scene parameterised with the same scene number will always prevail, where the highest priority scene is 1 (the first in the scene configuration tab) and the lowest priority is the last.*

## 2.6 MANUAL CONTROL

The device allows manually switching the state of its output relays 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, referred to as **Test On mode** (for testing purposes during the configuration of the device) and **Test Off mode** (for normal operation). Whether both, only one, or none of these modes should be accessible needs to be parameterised in ETS. Moreover, it is possible to enable a specific binary object for locking and unlocking the manual control at runtime.

**Notes**:

- *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 if a manual-control lock command is received from the KNX bus.*

## 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, depending on whether the output is configured as an individual output, a shutter channel or a fan-coil module.

- **Individual output**: a simple press (short or long) will make the output switch its on-off state, which will be reported to the KNX bus through the corresponding status object, if enabled.

- **Shutter Channel**: when the button is pressed, act on the output according to the length of the button press and to the current state.

  - A **long press** makes the shutter start moving (upwards or downwards, depending on the button being pressed). The LED will light in green until the end of the motion. If the button is pressed when the shutter is already at its upper or lower limit the top or bottom positions, nothing will happen (the LED will not light).

  - A **short press** will make the shutter drive stop (if in motion), as it normally does when a step/stop order is received from the KNX bus. If the shutter is

not moving, pressing the button does not cause any action, unless slats/lamellas have been parameterised – in such case, a step movement (up/down, depending on the button pressed) will take place. The status objects will be sent to the bus when applicable.

- **Fan Coil module**: the behaviour will depend on whether a fan-labelled or valve-labelled button is pressed:

  - **Fan**: for this type of buttons, it must be considered that there are two types of control for the fan speed:

    - **Switching control**: a short or long press will switch the relays to set the selected speed, unless it matches the current speed – in such case all the relays will be opened (speed 0). The associated LEDs will indicate the state of the fan speed control relays (on = relay closed; off = relay open).

    - **Accumulation control:** a short or long press switches to the selected speed, closing the relay associated with that speed, and the relays assigned to the lower speeds, unless it matches the current speed – in such case all the relays will be opened (speed 0). The associated LEDs will indicate the state of the fan speed control relays (on = relay closed; off = relay open).

    **Note**: *the behaviour of the relays will depend on the parameterisation, i.e., on the **number of fan speeds**, and on the **delay** between switches.*

  - **Valve**: a short or long press will switch the current status of the relay and therefore of the valve. The LED will show the state of the relay at all times (on = relay closed; off = relay open).

- **Disabled output**: outputs disabled by parameter will not react to button presses under the Test Off mode.

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

## Test On Mode

After entering the Test On mode, it will only be possible to control the outputs through the on-board pushbuttons. Orders received through communication objects will be ignored, regardless of the channel or output or the output they are addressed to.

Depending on whether the output has been parameterised as an individual output or as part of a shutter channel, the reactions to the button presses will differ.

- **Individual output**: short or long pressing the button will toggle the relay state of the relay.
- **Shutter channel**: pressing the button will make the shutter drive move upward or downward (depending on the button) until the button is released again, thus ignoring the position of the shutter and the parameterised times. For safety reasons, only one closed relay per shutter channel is allowed.

  **Note**: *after leaving the Test On mode, the status objects will recover the values they had prior to entering Test On. As the device is never aware of the actual position of the shutter (as the shutter drive does not provide any feedback), these values may not show the real position. This can be solved by performing a complete move-up or move-down order, or by calibrating the shutter position in the Test On mode until it matches the status objects*.

- **Fan Coil module**: the behaviour is similar to that of the Test Off mode, although in this case the three fan speeds are assumed to be available.
- **Disabled output**: short and long presses will switch the state of the corresponding relay. If the action closes the relay, then the remaining relays of its block will open, for safety reasons.

As described previously, if the device is in Test On mode, any command sent from the KNX bus to the actuator will not affect the outputs and no status objects will be sent (only periodically timed objects such as Heartbeat, logic functions or master light will continue to be sent to the bus) while Test ON mode is active. However, in the case of the "Alarm" and "Block" objects, although in Test ON mode the actions received by each object are not considered, the evaluation of their status is carried out when exiting this mode, so that any change in the alarm status or blocking of the outputs while Test ON mode is active is taken into account when exiting this mode and is updated with the last status detected.

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

## ETS PARAMETERISATION

After enabling 'Manual Control' on the General screen (see section 2.1), a new tab is added to the tree on the left.

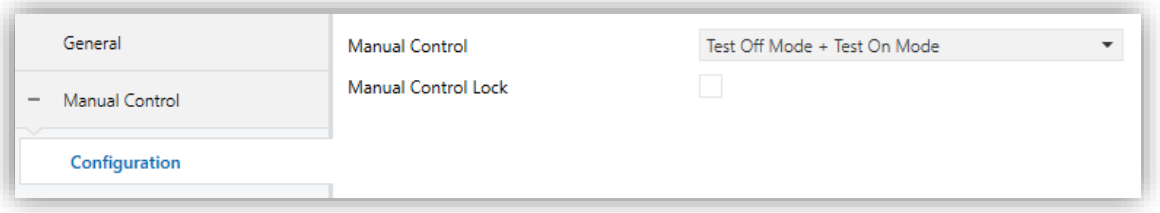



**Figure 7.** Manual Control

The only two parameters are:

- **Manual Control** [*Disabled / Only Test Off Mode / Only Test On Mode / Test Off Mode + Test On Mode*]. Depending on the selection, the device will permit using the manual control under the Test Off, the Test On, or both modes. Note that, as stated before, using the Test Off mode does not require any special action, while switching to the Test On mode does require long pressing the Prog/Test button.

- **Manual Control Lock** [*enabled / disabled*]: unless the above parameter has been "Disabled," the Lock Manual Control parameter provides an optional procedure for locking the manual control in runtime. When this checkbox is enabled, object "**Manual Control Lock**" becomes visible, as well as two more parameters:

  - **Value** [*0 = Lock; 1 = Unlock / 0 = Unlock; 1 = Lock*]: defines whether the manual control lock/unlock should take place respectively upon the reception (through the aforementioned object) of values "0" and "1", or the opposite.

  - **Initialization** [*Unlocked / Locked / Last Value*]: sets how the lock state of the manual control should remain after the device start-up (after an ETS download or a bus power failure). "Last Value" (default; on the very first start-up, this will be Unlocked).

# ANNEX I. COMPARATIVE TABLE FOR DEVICES

| Device / Functionality | inBOX 20 v3 | MINiBOX 20 v2 | MINiBOX 40 v3 | MAXinBOX 8 v4 | MAXinBOX 12 | MAXinBOX 16 v4 | MAXinBOX 20 | MAXinBOX 24 v2 |
|---|---|---|---|---|---|---|---|---|
| **KNX Secure** | ✕ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✕ |
| **Fan Coil Relays** | ✕ | ✕ | 1 | 2 | 3 | 4 | 5 | 6 |
| **Logic Functions** | 10 | 10 | 10 | 20 | 30 | 20 | 30 | 30 |
| **Master Light** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **Shutter Channels** | 1 | 1 | 2 | 4 | 6 | 8 | 10 | 12 |
| **Individual Outputs** | 2 | 2 | 4 | 8 | 12 | 16 | 20 | 24 |

**Table 1.** Comparative table of the different actuators covered by this manual

# ANNEX II. COMMUNICATION OBJECTS

- **"Functional range"** shows the values that, irrespective 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 under either the KNX standard or the application program itself.

- Please note that **the complete object table corresponds to MAXinBOX 24 v2**, as this is the device with the highest number of instances. The other devices will not have some communication objects because they have fewer outputs or do not have certain functionalities. For more information, see Table 1.

| Number | Size | I/O | Flags | Data type (DPT) | Functional Range | Name | Function |
|---|---|---|---|---|---|---|---|
| 1 | 1 Bit | O | C R - T - | DPT_Trigger | 0/1 | [Heartbeat] Object to Send '1' | Sending of '1' Periodically |
| 2 | 1 Bit | O | C R - T - | DPT_Trigger | 0/1 | [Heartbeat] Device Recovery | Send 0 |
| 3 | 1 Bit | O | C R - T - | DPT_Trigger | 0/1 | [Heartbeat] Device Recovery | Send 1 |
| 4 | 1 Bit | I | C - W - - | DPT_Enable | 0/1 | Manual Control Lock | 0 = Lock; 1 = Unlock |
| | 1 Bit | I | C - W - - | DPT_Enable | 0/1 | Manual Control Lock | 0 = Unlock; 1 = Lock |
| 5, 19, 33, 47, 61, 75, 89, 103, 117, 131, 145, 159, 173, 187, 201, 215, 229, 243, 257, 271, 285, 299, 313, 327 | 1 Byte | I | C - W - - | DPT_SceneControl | 0-63; 128-191 | [Ox][x] Scenes | 0 – 63 (Execute 1 – 64); 128 – 191 (Save 1 – 64) |
| 6, 20, 34, 48, 62, 76, 90, 104, 118, 132, 146, 160, 174, 188, 202, 216, 230, 244, 258, 272, 286, 300, 314, 328 | 1 Bit | I | C - W - - | DPT_BinaryValue | 0/1 | [Ox][x] On/Off | N.O. (0 = Open Relay; 1 = Close Relay) |
| | 1 Bit | I | C - W - - | DPT_BinaryValue | 0/1 | [Ox][x] On/Off | N.C. (0 = Close Relay; 1 = Open Relay) |
| | 1 Bit | I | C - W - - | DPT_BinaryValue | 0/1 | [Ox] On/Off | N.O. (0 = Open Relay; 1 = Close Relay) |
| 7, 21, 35, 49, 63, 77, 91, 105, 119, 133, 147, 161, 175, 189, 203, 217, 231, 245, 259, 273, 287, 301, 315, 329 | 1 Bit | O | C R - T - | DPT_BinaryValue | 0/1 | [Ox][x] On/Off (Status) | 0 = Output Off; 1 = Output On |
| | 1 Bit | O | C R - T - | DPT_BinaryValue | 0/1 | [Ox] On/Off (Status) | 0 = Output Off; 1 = Output On |
| 8, 22, 36, 50, 64, 78, 92, 106, 120, 134, 148, 162, 176, 190, 204, 218, 232, 246, 260, 274, 288, 302, 316, 330 | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [Ox][x] Switched Control | 0/1 = On/Off Depending on the Last Status |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 9, 23, 37, 51, 65, 79, 93, 107, 121, 135, 149, 163, 177, 191, 205, 219, 233, 247, 261, 275, 289, 303, 317, 331 | 1 Bit | I | **C - W - -** | DPT_Enable | 0/1 | [Ox][x] Lock | 0 = Unlock; 1 = Lock |
| 10, 24, 38, 52, 66, 80, 94, 108, 122, 136, 150, 164, 178, 192, 206, 220, 234, 248, 262, 276, 290, 304, 318, 332 | 1 Bit | O | **C R - T -** | DPT_Enable | 0/1 | [Ox][x] Lock (Status) | 0 = Unlock; 1 = Lock |
| 11, 25, 39, 53, 67, 81, 95, 109, 123, 137, 151, 165, 179, 193, 207, 221, 235, 249, 263, 277, 291, 305, 319, 333 | 1 Bit | I | **C - W - -** | DPT_Start | 0/1 | [Ox][x] Timer | 0 = Switch Off; 1 = Switch On |
| 12, 26, 40, 54, 68, 82, 96, 110, 124, 138, 152, 166, 180, 194, 208, 222, 236, 250, 264, 278, 292, 306, 320, 334 | 2 Bytes | I | **C - W - -** | DPT_TimePeriodSec | 0 - 65535 | [Ox][x] On Duration Time (s) | 0 - 3600 s |
| | 2 Bytes | I | **C - W - -** | DPT_TimePeriod100Msec | | [Ox][x] On Duration Time (ds) | 0 - 600 ds |
| | 2 Bytes | I | **C - W - -** | DPT_TimePeriodMin | 0 - 65535 | [Ox][x] On Duration Time (min) | 0 - 1440 min |
| | 2 Bytes | I | **C - W - -** | DPT_TimePeriodHrs | 0 - 65535 | [Ox][x] On Duration Time (h) | 0 - 24 h |
| 13, 27, 41, 55, 69, 83, 97, 111, 125, 139, 153, 167, 181, 195, 209, 223, 237, 251, 265, 279, 293, 307, 321, 335 | 1 Bit | O | **C R - T -** | DPT_State | 0/1 | [Ox][x] Warning Countdown (Status) | 0 = Normal; 1 = Warning |
| 14, 28, 42, 56, 70, 84, 98, 112, 126, 140, 154, 168, 182, 196, 210, 224, 238, 252, 266, 280, 294, 308, 322, 336 | 1 Bit | I | **C - W - -** | DPT_Start | 0/1 | [Ox][x] Flashing | 0 = Stop; 1 = Start |
| 15, 29, 43, 57, 71, 85, 99, 113, 127, 141, 155, 169, 183, 197, 211, 225, 239, 253, 267, 281, 295, 309, 323, 337 | 1 Bit | I | **C - W - -** | DPT_Alarm | 0/1 | [Ox][x] Alarm | 0 = No Alarm; 1 = Alarm |
| | 1 Bit | I | **C - W - -** | DPT_Alarm | 0/1 | [Ox][x] Alarm | 0 = Alarm; 1 = No Alarm |
| 16, 30, 44, 58, 72, 86, 100, 114, 128, 142, 156, 170, 184, 198, 212, 226, 240, 254, 268, 282, 296, 310, 324, 338 | 1 Bit | O | **C R - T -** | DPT_Alarm | 0/1 | [Ox][x] Alarm (Status) | 0 = No Alarm; 1 = Alarm |
| | 1 Bit | O | **C R - T -** | DPT_Alarm | 0/1 | [Ox][x] Alarm (Status) | 0 = Alarm; 1 = No Alarm |
| 17, 31, 45, 59, 73, 87, 101, 115, 129, 143, 157, 171, 185, 199, 213, 227, 241, 255, 269, 283, 297, 311, 325, 339 | 1 Bit | I | **C - W - -** | DPT_Ack | 0/1 | [Ox][x] Unfreeze Alarm | Alarm = 0 + Unfreeze = 1 => End Alarm |
| 18, 32, 46, 60, 74, 88, 102, 116, 130, 144, 158, | 4 Bytes | I/O | **C R W T -** | DPT_LongDeltaTimeSec | -2147483648 - 2147483647 | [Ox][x] Operating Time (s) | Time in Seconds |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 172, 186, 200, 214, 228, 242, 256, 270, 284, 298, 312, 326, 340 | 4 Bytes | I/O | C R W T - | 1.xxx | 0/1 | [Ox][x] Operating Time (h) | Time in Hours |
| 341 | 1 Byte | I | C - W - - | DPT_SceneControl | 0-63; 128-191 | [Shutter] Scenes | 0 - 63 (Execute 1 - 64); 128 - 191 (Save 1 - 64) |
| 342 | 2 Bytes | I | C - W T U | DPT_Value_Temp | -273.00º - 670433.28º | [Shutter] Outdoor Temperature Input | -30°C ... 60°C |
| 343 | 1 Byte | I | C - W T U | DPT_Angle | | [Shutter] Azimuth | Azimuth External Reference [0° ... 360°] |
| 344 | 2 Bytes | I | C - W T U | DPT_Rotation_Angle | | [Shutter] Elevation | Elevation External Reference [-90° ... 90°] |
| 345, 386, 427, 468, 509, 550, 591, 632, 673, 714, 755, 796 | 1 Bit | I | C - W - - | DPT_UpDown | 0/1 | [Cx] Shutter - Move Control | 0 = Up; 1 = Down |
| 346, 387, 428, 469, 510, 551, 592, 633, 674, 715, 756, 797 | 1 Bit | I | C - W - - | DPT_Step | 0/1 | [Cx] Shutter - Stop/Step Control | 0 = Stop/Step Up; 1 = Stop/Step Down |
| | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [Cx] Shutter - Stop Control | 0/1 = Stop |
| 347, 388, 429, 470, 511, 552, 593, 634, 675, 716, 757, 798 | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [Cx] Shutter - Switched Control | 0/1 = Up, Down or Stop, Depending on the Last Move |
| 348, 389, 430, 471, 512, 553, 594, 635, 676, 717, 758, 799 | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [Cx] Shutter - Switched Control Up | 0/1 = Up or Stop, Depending on the Last Move |
| 349, 390, 431, 472, 513, 554, 595, 636, 677, 718, 759, 800 | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [Cx] Shutter - Switched Control Down | 0/1 = Down or Stop, Depending on the Last Move |
| 350, 391, 432, 473, 514, 555, 596, 637, 678, 719, 760, 801 | 1 Bit | I | C - W - - | DPT_Enable | 0/1 | [Cx] Shutter - Lock | 0 = Unlock; 1 = Lock |
| 351, 392, 433, 474, 515, 556, 597, 638, 679, 720, 761, 802 | 1 Byte | I | C - W - - | DPT_Scaling | 0% - 100% | [Cx] Shutter - Percentage Control | 0% = Top; 100% = Bottom |
| 352, 393, 434, 475, 516, 557, 598, 639, 680, 721, 762, 803 | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [Cx] Shutter - Percentage Status | 0% = Top; 100% = Bottom |
| 353, 394, 435, 476, 517, 558, 599, 640, 681, 722, 763, 804 | 1 Byte | I | C - W - - | DPT_Scaling | 0% - 100% | [Cx] Shutter - Slats Percentage Control | 0% = Open; 100% = Closed |
| 354, 395, 436, 477, 518, 559, 600, 641, 682, 723, 764, 805 | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [Cx] Shutter - Slats Percentage Status | 0% = Open; 100% = Closed |
| 355, 396, 437, 478, 519, 560, 601, 642, 683, 724, 765, 806 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Rising Relay Status | 0 = Open; 1 = Closed |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 356, 397, 438, 479, 520, 561, 602, 643, 684, 725, 766, 807 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Lowering Relay Status | 0 = Open; 1 = Closed |
| 357, 398, 439, 480, 521, 562, 603, 644, 685, 726, 767, 808 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Move Status | 0 = Stopped; 1 = Moving |
| 358, 399, 440, 481, 522, 563, 604, 645, 686, 727, 768, 809 | 1 Bit | O | C R - T - | DPT_UpDown | 0/1 | [Cx] Shutter - Move Direction Status | 0 = Upward; 1 = Downward |
| 359, 400, 441, 482, 523, 564, 605, 646, 687, 728, 769, 810 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Totally Up Status | 0 = Other Positions; 1 = Up |
| 360, 401, 442, 483, 524, 565, 606, 647, 688, 729, 770, 811 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Totally Down Status | 0 = Other Positions; 1 = Down |
| 361, 402, 443, 484, 525, 566, 607, 648, 689, 730, 771, 812 | 1 Bit | I | C - W - - | DPT_Switch | 0/1 | [Cx] Shutter - Auto: On/Off | 0 = On; 1 = Off |
| | 1 Bit | I | C - W - - | DPT_Switch | 0/1 | [Cx] Shutter - Auto: On/Off | 0 = Off; 1 = On |
| 362, 403, 444, 485, 526, 567, 608, 649, 690, 731, 772, 813 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Auto: On/Off Status | 0 = On; 1 = Off |
| | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [Cx] Shutter - Auto: On/Off Status | 0 = Off; 1 = On |
| 363, 404, 445, 486, 527, 568, 609, 650, 691, 732, 773, 814 | 1 Bit | I | C - W - - | DPT_UpDown | 0/1 | [Cx] Shutter - Auto: Move Control | 0 = Up; 1 = Down |
| 364, 405, 446, 487, 528, 569, 610, 651, 692, 733, 774, 815 | 1 Bit | I | C - W - - | DPT_Step | 0/1 | [Cx] Shutter - Auto: Stop/Step Control | 0 = Stop/Step Up; 1 = Stop/Step Down |
| | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [Cx] Shutter - Auto: Stop Control | 0/1 = Stop |
| 365, 406, 447, 488, 529, 570, 611, 652, 693, 734, 775, 816 | 1 Byte | I | C - W - - | DPT_Scaling | 0% - 100% | [Cx] Shutter - Auto: Percentage Control | 0% = Top; 100% = Bottom |
| 366, 407, 448, 489, 530, 571, 612, 653, 694, 735, 776, 817 | 1 Byte | I | C - W - - | DPT_Scaling | 0% - 100% | [Cx] Shutter - Auto: Slats Percentage Control | 0% = Open; 100% = Closed |
| 367, 408, 449, 490, 531, 572, 613, 654, 695, 736, 777, 818 | 1 Bit | I | C - W T U | DPT_Scene_AB | 0/1 | [Cx] Shutter - Sunshine/Shadow Input | 0 = Sunshine; 1 = Shadow |
| | 1 Bit | I | C - W T U | DPT_Scene_AB | 0/1 | [Cx] Shutter - Sunshine/Shadow Input | 0 = Shadow; 1 = Sunshine |
| 368, 409, 450, 491, 532, 573, 614, 655, 696, 737, 778, 819 | 2 Bytes | I | C - W T U | DPT_Value_Lux | | [Cx] Shutter - Sunshine/Shadow Input | 1 Lux ... 100.000 Lux |
| 369, 410, 451, 492, 533, 574, 615, 656, 697, 738, 779, 820 | 1 Bit | I | C - W T U | DPT_Heat_Cool | 0/1 | [Cx] Shutter - Cooling/Heating | 0 = Cooling; 1 = Heating |
| | 1 Bit | I | C - W T U | DPT_Heat_Cool | 0/1 | [Cx] Shutter - Cooling/Heating | 0 = Heating; 1 = Cooling |
| | 1 Bit | I | C - W T U | DPT_Occupancy | 0/1 | [Cx] Shutter - Presence/No Presence | 0 = No Presence; 1 = Presence |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 370, 411, 452, 493, 534, 575, 616, 657, 698, 739, 780, 821 | 1 Bit | I | C - W T U | DPT_Occupancy | 0/1 | [Cx] Shutter - Presence/No Presence | 0 = Presence; 1 = No Presence |
| 371, 372, 373, 374, 375, 376, 412, 413, 414, 415, 416, 417, 453, 454, 455, 456, 457, 458, 494, 495, 496, 497, 498, 499, 535, 536, 537, 538, 539, 540, 576, 577, 578, 579, 580, 581, 617, 618, 619, 620, 621, 622, 658, 659, 660, 661, 662, 663, 699, 700, 701, 702, 703, 704, 740, 741, 742, 743, 744, 745, 781, 782, 783, 784, 785, 786, 822, 823, 824, 825, 826, 827 | 1 Bit | I | C - W - - | DPT_Alarm | 0/1 | [Cx] Shutter - x | 0 = No Alarm; 1 = Alarm |
| | 1 Bit | I | C - W - - | DPT_Alarm | 0/1 | [Cx] Shutter - x | 0 = Alarm; 1 = No Alarm |
| 377, 418, 459, 500, 541, 582, 623, 664, 705, 746, 787, 828 | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [Cx] Shutter - Unfreeze Alarm | Alarm1 = Alarm2 = ... = Alarm6 = No Alarm + Unfreeze (1) => End Alarm |
| 378, 419, 460, 501, 542, 583, 624, 665, 706, 747, 788, 829 | 1 Bit | O | C R - T - | DPT_Alarm | 0/1 | [Cx] Shutter - General Alarm Status | 0 = No Alarm; 1 = Alarm |
| | 1 Bit | O | C R - T - | DPT_Alarm | 0/1 | [Cx] Shutter - General Alarm Status | 0 = Alarm; 1 = No Alarm |
| 379, 420, 461, 502, 543, 584, 625, 666, 707, 748, 789, 830 | 1 Bit | I | C - W - - | DPT_Scene_AB | 0/1 | [Cx] Shutter - Move Control (Reversed) | 0 = Down; 1 = Up |
| 380, 421, 462, 503, 544, 585, 626, 667, 708, 749, 790, 831 | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [Cx] Shutter - Direct Positioning 1 | 0 = No Action; 1 = Go to Position |
| 381, 422, 463, 504, 545, 586, 627, 668, 709, 750, 791, 832 | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [Cx] Shutter - Direct Positioning 2 | 0 = No Action; 1 = Go to Position |
| 382, 423, 464, 505, 546, 587, 628, 669, 710, 751, 792, 833 | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [Cx] Shutter - Direct Positioning 1 (Save) | 0 = No Action; 1 = Save Current Position |
| 383, 424, 465, 506, 547, 588, 629, 670, 711, 752, 793, 834 | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [Cx] Shutter - Direct Positioning 2 (Save) | 0 = No Action; 1 = Save Current Position |
| 384, 425, 466, 507, 548, 589, 630, 671, 712, 753, 794, 835 | 1 Bit | O | C R - T - | DPT_BinaryValue | 0/1 | [Cx] Shutter - External Contact - Stop Movement | 0 = Open Relay; 1 = Close Relay |
| 385, 426, 467, 508, 549, 590, 631, 672, 713, 754, 795, 836 | 1 Bit | I | C - W - - | DPT_Start | 0/1 | [Cx] Shutter - Start/Stop Rise and Fall Times Measurement | 0 = Stop; 1 = Start |
| | 1 Bit | I | C - W - - | DPT_Start | 0/1 | [Cx] Shutter - Start/Stop Rise and Fall Times Measurement | 0 = Start; 1 = Stop |
| 837 | 1 Byte | I | C - W - - | DPT_SceneControl | 0-63; 128-191 | [Fan Coil] Scenes | 0 - 63 (Execute 1 - 64); 128 - 191 (Save 1 - 64) |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 838, 871, 904, 937, 970, 1003 | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] On/Off | 0 = Off; 1 = On |
| 839, 872, 905, 938, 971, 1004 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] On/Off (Status) | 0 = Off; 1 = On |
| 840, 873, 906, 939, 972, 1005 | 1 Bit | I | C - W - U | DPT_Heat_Cool | 0/1 | [FCx] Mode | 0 = Cool; 1 = Heat |
| 841, 874, 907, 940, 973, 1006 | 1 Bit | O | C R - T - | DPT_Heat_Cool | 0/1 | [FCx] Mode (Status) | 0 = Cool; 1 = Heat |
| 842, 875, 908, 941, 974, 1007 | 1 Bit | I | C - W - U | DPT_Enable | 0/1 | [FCx] Fan: Manual/Automatic | 0 = Automatic; 1 = Manual |
| | 1 Bit | I | C - W - U | DPT_Enable | 0/1 | [FCx] Fan: Manual/Automatic | 0 = Manual; 1 = Automatic |
| 843, 876, 909, 942, 975, 1008 | 1 Bit | O | C R - T - | DPT_Enable | 0/1 | [FCx] Fan: Manual/Automatic (Status) | 0 = Automatic; 1 = Manual |
| | 1 Bit | O | C R - T - | DPT_Enable | 0/1 | [FCx] Fan: Manual/Automatic (Status) | 0 = Manual; 1 = Automatic |
| 844, 877, 910, 943, 976, 1009 | 1 Bit | I | C - W - U | DPT_Step | 0/1 | [FCx] Manual Fan: Step Control | 0 = Down; 1 = Up |
| 845, 878, 911, 944, 977, 1010 | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] Manual Fan: Speed 0 | 0 = Off; 1 = On |
| 846, 879, 912, 945, 978, 1011 | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] Manual Fan: Speed 1 | 0 = Off; 1 = On |
| 847, 880, 913, 946, 979, 1012 | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] Manual Fan: Speed 2 | 0 = Off; 1 = On |
| 848, 881, 914, 947, 980, 1013 | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] Manual Fan: Speed 3 | 0 = Off; 1 = On |
| 849, 882, 915, 948, 981, 1014 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Fan: Speed 0 (Status) | 0 = Off; 1 = On |
| 850, 883, 916, 949, 982, 1015 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Fan: Speed 1 (Status) | 0 = Off; 1 = On |
| 851, 884, 917, 950, 983, 1016 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Fan: Speed 2 (Status) | 0 = Off; 1 = On |
| 852, 885, 918, 951, 984, 1017 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Fan: Speed 3 (Status) | 0 = Off; 1 = On |
| 853, 886, 919, 952, 985, 1018 | 1 Byte | I | C - W - U | DPT_Fan_Stage | 0 - 255 | [FCx] Manual Fan: Enumeration Control | S0 = 0; S1 = 1; S2 = 2; S3 = 3 |
| | 1 Byte | I | C - W - U | DPT_Fan_Stage | 0 - 255 | [FCx] Manual Fan: Enumeration Control | S0 = 0; S1 = 1; S2 = 2 |
| | 1 Byte | I | C - W - U | DPT_Fan_Stage | 0 - 255 | [FCx] Manual Fan: Enumeration Control | S0 = 0; S1 = 1 |
| 854, 887, 920, 953, 986, 1019 | 1 Byte | O | C R - T - | DPT_Fan_Stage | 0 - 255 | [FCx] Fan: Speed Enumeration (Status) | S0 = 0; S1 = 1; S2 = 2; S3 = 3 |
| | 1 Byte | O | C R - T - | DPT_Fan_Stage | 0 - 255 | [FCx] Fan: Speed Enumeration (Status) | S0 = 0; S1 = 1; S2 = 2 |
| | 1 Byte | O | C R - T - | DPT_Fan_Stage | 0 - 255 | [FCx] Fan: Speed Enumeration (Status) | S0 = 0; S1 = 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 855, 888, 921, 954, 987, 1020 | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Manual Fan: Percentage Control | S0 = 0%; S1 = 0,4-33,3%; S2 = 33,7-66,7%; S3 = 67,1-100% |
| | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Manual Fan: Percentage Control | S0 = 0%; S1 = 1-50%; S2 = 51-100% |
| | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Manual Fan: Percentage Control | S0 = 0%; S1 = 1-100% |
| 856, 889, 922, 955, 988, 1021 | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [FCx] Fan: Speed Percentage (Status) | S0 = 0%; S1 = 33,3%; S2 = 66,6%; S3 = 100% |
| | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [FCx] Fan: Speed Percentage (Status) | S0 = 0%; S1 = 1-50%; S2 = 51-100% |
| | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [FCx] Fan: Speed Percentage (Status) | S0 = 0%; S1 = 1-100% |
| 857, 890, 923, 956, 989, 1022 | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Cooling Fan: Continuous Control | 0 - 100% |
| | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Cooling Valve: PI Control (Continuous) | 0 - 100% |
| 858, 891, 924, 957, 990, 1023 | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Heating Fan: Continuous Control | 0 - 100% |
| | 1 Byte | I | C - W - U | DPT_Scaling | 0% - 100% | [FCx] Heating Valve: PI Control (Continuous) | 0 - 100% |
| 859, 892, 925, 958, 991, 1024 | 1 Bit | I | C - W - U | DPT_OpenClose | 0/1 | [FCx] Cooling Valve: Control Variable (1 bit) | 0 = Open Valve; 1 = Close Valve |
| | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] Cooling Valve: Control Variable (1 bit) | 0 = Close Valve; 1 = Open Valve |
| 860, 893, 926, 959, 992, 1025 | 1 Bit | I | C - W - U | DPT_OpenClose | 0/1 | [FCx] Heating Valve: Control Variable (1 bit) | 0 = Open Valve; 1 = Close Valve |
| | 1 Bit | I | C - W - U | DPT_Switch | 0/1 | [FCx] Heating Valve: Control Variable (1 bit) | 0 = Close Valve; 1 = Open Valve |
| 861, 894, 927, 960, 993, 1026 | 1 Bit | O | C R - T - | DPT_OpenClose | 0/1 | [FCx] Cooling Valve (Status) | 0 = Open; 1 = Closed |
| | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Cooling Valve (Status) | 0 = Closed; 1 = Open |
| | 1 Bit | O | C R - T - | DPT_OpenClose | 0/1 | [FCx] Valve (Status) | 0 = Open; 1 = Closed |
| | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Valve (Status) | 0 = Closed; 1 = Open |
| 862, 895, 928, 961, 994, 1027 | 1 Bit | O | C R - T - | DPT_OpenClose | 0/1 | [FCx] Heating Valve (Status) | 0 = Open; 1 = Closed |
| | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Heating Valve (Status) | 0 = Closed; 1 = Open |
| 863, 896, 929, 962, 995, 1028 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Cooling Valve: Anti-Seize Protection (Status) | 0 = Not Active; 1 = Active |
| | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Valve: Anti-Seize Protection (Status) | 0 = Not Active; 1 = Active |
| 864, 897, 930, 963, 996, 1029 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [FCx] Heating Valve: Anti-Seize Protection (Status) | 0 = Not Active; 1 = Active |
| 865, 898, 931, 964, 997, 1030 | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [FCx] Valve (Status) | 0 - 100% |
| | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [FCx] Cooling Valve (Status) | 0 - 100% |
| 866, 899, 932, 965, 998, 1031 | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [FCx] Heating Valve (Status) | 0 - 100% |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 867, 900, 933, 966, 999, 1032 | 1 Bit | O | C R - T - | DPT_Bool | 0/1 | [FCx] Control Value - Error | 0 = No Error; 1 = Error |
| 868, 901, 934, 967, 1000, 1033 | 2 Bytes | I | C - W - U | DPT_Value_Temp | -273.00º - 670433.28º | [FCx] Ambient Temperature | Ambient Temperature |
| 869, 902, 935, 968, 1001, 1034 | 2 Bytes | I | C - W - U | DPT_Value_Temp | -273.00º - 670433.28º | [FCx] Setpoint Temperature | Setpoint Temperature |
| 870, 903, 936, 969, 1002, 1035 | 2 Bytes | I/O | C R W T U | DPT_TimePeriodMin | 0 - 65535 | [FCx] Duration of Manual Control | 0 = Endless; 1 - 1440 min |
| | 2 Bytes | I/O | C R W T U | DPT_TimePeriodHrs | 0 - 65535 | [FCx] Duration of Manual Control | 0 = Endless; 1 - 24 h |
| 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099 | 1 Bit | I | C - W - - | DPT_Bool | 0/1 | [LF] (1-Bit) Data Entry x | Binary Data Entry (0/1) |
| 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1130, 1131 | 1 Byte | I | C - W - - | DPT_Value_1_Ucount | 0 - 255 | [LF] (1-Byte) Data Entry x | 1-Byte Data Entry (0-255) |
| 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163 | 2 Bytes | I | C - W - - | DPT_Value_2_Ucount | 0 - 65535 | [LF] (2-Byte) Data Entry x | 2-Byte Data Entry |
| | 2 Bytes | I | C - W - - | DPT_Value_2_Count | -32768 - 32767 | [LF] (2-Byte) Data Entry x | 2-Byte Data Entry |
| | 2 Bytes | I | C - W - - | 9.xxx | -671088.64 - 670433.28 | [LF] (2-Byte) Data Entry x | 2-Byte Data Entry |
| 1164, 1165, 1166, 1167, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179 | 4 Bytes | I | C - W - - | DPT_Value_4_Count | -2147483648 - 2147483647 | [LF] (4-Byte) Data Entry x | 4-Byte Data Entry |
| 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, | 1 Bit | O | C R - T - | DPT_Bool | 0/1 | [LF] Function x - Result | (1-Bit) Boolean |
| | 1 Byte | O | C R - T - | DPT_Value_1_Ucount | 0 - 255 | [LF] Function x - Result | (1-Byte) Unsigned |
| | 2 Bytes | O | C R - T - | DPT_Value_2_Ucount | 0 - 65535 | [LF] Function x - Result | (2-Byte) Unsigned |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209 | 4 Bytes | O | C R - T - | DPT_Value_4_Count | -2147483648 - 2147483647 | [LF] Function x - Result | (4-Byte) Signed |
| | 1 Byte | O | C R - T - | DPT_Scaling | 0% - 100% | [LF] Function x - Result | (1-Byte) Percentage |
| | 2 Bytes | O | C R - T - | DPT_Value_2_Count | -32768 - 32767 | [LF] Function x - Result | (2-Byte) Signed |
| | 2 Bytes | O | C R - T - | 9.xxx | -671088.64 - 670433.28 | [LF] Function x - Result | (2-Byte) Float |
| 1210, 1250 | 1 Bit | I | C - W - - | DPT_Trigger | 0/1 | [MLx] Trigger | Trigger the Master Light Function |
| | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [MLx] Trigger | 0 = Nothing; 1 = Trigger the Master Light Function |
| | 1 Bit | I | C - W - - | DPT_Ack | 0/1 | [MLx] Trigger | 1 = Nothing; 0 = Trigger the Master Light Function |
| 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1280 | 1 Bit | I | C - W - - | DPT_Switch | 0/1 | [MLx] Status Object x | Binary Status |
| 1241, 1281 | 1 Bit | O | C R - T - | DPT_Switch | 0/1 | [MLx] General Status | Binary Status |
| 1242, 1282 | 1 Bit | O | C - - T - | DPT_Switch | 0/1 | [MLx] General Switch Off: Binary Object | Switch Off Sending |
| 1243, 1283 | 1 Byte | O | C - - T - | DPT_Scaling | 0% - 100% | [MLx] General Switch Off: Scaling | 0-100% |
| 1244, 1284 | 1 Byte | O | C - - T - | DPT_SceneControl | 0-63; 128-191 | [MLx] General Switch Off: Scene | Scene Sending |
| 1245, 1285 | 1 Byte | O | C - - T - | DPT_HVACMode | 1=Comfort<br>2=Standby<br>3=Economy<br>4=Building Protection | [MLx] General Switch Off: HVAC mode | Auto, Comfort, Standby, Economy, Building Protection |
| 1246, 1286 | 1 Bit | O | C - - T - | DPT_Switch | 0/1 | [MLx] Courtesy Switch On: Binary Object | Switch On Sending |
| 1247, 1287 | 1 Byte | O | C - - T - | DPT_Scaling | 0% - 100% | [MLx] Courtesy Switch On: Scaling | 0-100% |
| 1248, 1288 | 1 Byte | O | C - - T - | DPT_SceneNumber | 0 - 63 | [MLx] Courtesy Switch On: Scene | Scene Sending |
| 1249, 1289 | 1 Byte | O | C - - T - | DPT_HVACMode | 1=Comfort<br>2=Standby<br>3=Economy<br>4=Building Protection | [MLx] Courtesy Switch On: HVAC mode | Auto, Comfort, Standby, Economy, Building Protection |

**Join and send us your inquiries about Zennio devices:**

**https://support.zennio.com**

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

*Tel. +34 925 232 002.*

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