MAXinBOX 66

Multifunction Actuator with 6 Outputs and 6 Inputs

ZN1IO-MB66

Application program version: [1.1]
User manual edition: [1.1]_a

www.zennio.com

USER MANUAL

CONTENTS

1 INTRODUCTION

1.1 MAXINBOX 66

MAXinBOX 66 from Zennio is a versatile KNX actuator featuring a wide variety of functions:

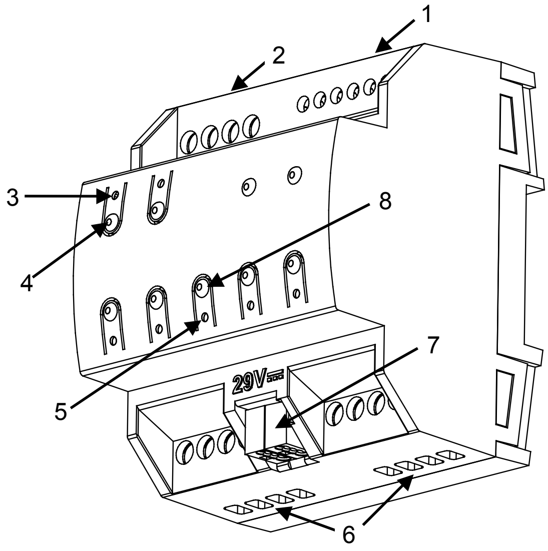
- **6 relay outputs**, configurable as:
  - Up to 3 independent shutter channels (with or without slats),
  - Up to 6 individual ON/OFF outputs,
  - A combination of the above.
- **6 multi-purpose inputs**, configurable as:
  - Temperature probes,
  - Binary inputs (i.e., pushbuttons, switches, sensors),
  - Motion detectors.
- **10 customisable, multi-operation logic functions**.
- **4 independent thermostats**.
- **Master light control** for an easy, out-of-the-box control of a set of luminaires (or functionally equivalent devices) one of which acts as a general lamp and the others as secondary lamps.
- **Manual operation / supervision** of the 6 relay outputs through the on-board pushbuttons and LEDs.

1.2 INSTALLATION

MAXinBOX 66 connects to the KNX bus through the on-board KNX connector.

Once the device is provided with power from the KNX bus, both the individual address and the associated application program may be downloaded.

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



1. Analogue/Digital Inputs.
2. Upper Outputs.
3. Output LED Indicator.
4. Manual Control Pushbutton.
5. Test/Prog. LED.
6. Lower Outputs.
7. KNX Bus Connector.
8. Test/Prog. Pushbutton.

Figure 1. MAXinBOX 66. Elements

The main elements of the device are described next.

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

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

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

- **Inputs (1)**: input ports for the insertion of the stripped cables of external elements such as switches / motion detectors / temperature probes, etc. One of the two cables of each element needs to be connected to one of the slots labelled “1” to “6”, while the other cable should be connected to the slot labelled as “C”. Note that all the external input devices share the “C” slot for one of the two cables. Please secure the connection by means of the on-board screws.

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

1.3 START-UP AND POWER LOSS

During the start-up of the device, the Test/Prog. LED will blink in blue colour for a few seconds before MAXinBOX 66 is ready. External orders will not be executed during this time, but afterwards.

Depending on the configuration, some specific actions will also be performed during the start-up. For example, the integrator can set whether the 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, MAXinBOX 66 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).

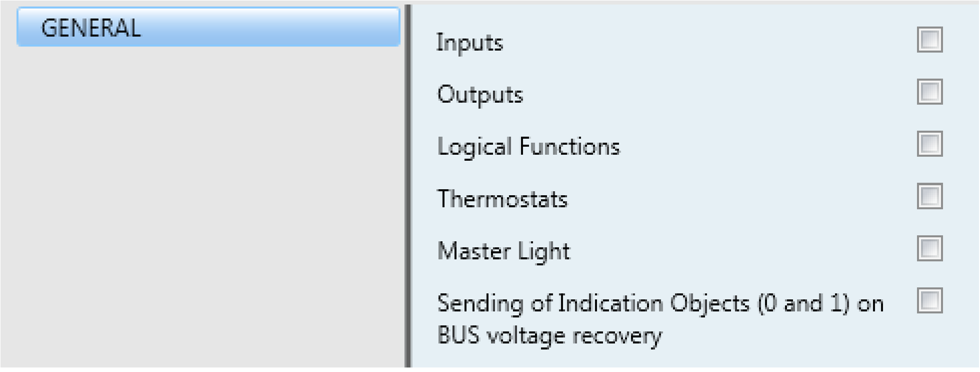
2 CONFIGURATION

2.1 GENERAL

After importing the corresponding database in ETS and adding the device into the topology of the desired project, the configuration process begins by right-clicking into the device and selecting *Edit parameters*.

ETS PARAMETERISATION

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



Inputs □
Outputs □
Logical Functions □
Thermostats □
Master Light □
Sending of Indication Objects (0 and 1) on BUS voltage recovery □

Figure 2. General screen

- Once activated, **Inputs, Outputs, Logical Functions, Thermostats** and **Master Light** bring additional tabs to the menu on the left. These functions and their parameters will be explained in later sections of this document.

- **Sending of Indication Objects (0 and 1) on Bus Voltage Recovery**: this parameter lets the integrator activate two new communication objects ("**Reset 0**" and "**Reset 1**"), which will be sent to the KNX bus with values "0" and "1" respectively whenever the device begins operation (for example, after a bus power failure). It is possible to parameterise a certain **delay** to this sending (0 to 255 seconds).

2.2 INPUTS

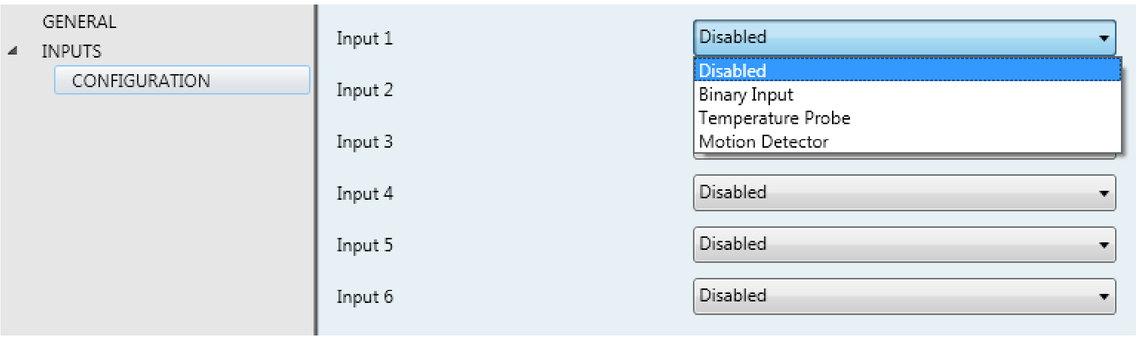
MAXinBOX 66 incorporates **6 analogue/digital inputs**, each configurable as a:

- **Binary Input**, for the connection of a pushbutton or a switch/sensor.
- **Temperature Probe**, to connect a temperature sensor (model ZN1AC-NTC68 S/E/F from Zennio).
- **Motion Detector**, to connect a motion detector (models ZN1IO-DETEC-P and ZN1IO-DETEC-X from Zennio).

  Important: *older models of the Zennio motion detector (e.g., ZN1IO-DETEC and ZN1IO-DETEC-N) will not work properly with MAXinBOX 66.*

ETS PARAMETERISATION

When **Inputs** has been activated in the General parameters screen, the following drop-lists will be available for the selection of the specific functions required.



Input 1 Disabled
Input 2
Input 3
Input 4 Disabled
Input 5 Disabled
Input 6 Disabled

Figure 3. Inputs - Configuration

All inputs are disabled by default. Depending on the function selected for each input, additional tabs will be included in the menu on the left.

2.2.1 BINARY INPUT

The following tasks are performed for binary inputs:

- Retrieving the **state** (1/0) of the input line and detecting **changes** (i.e., button presses, sensor changes, etc.).
- **Reporting** the KNX bus about the above states/changes and triggering the corresponding **actions**, depending on the case.
- Detecting **sabotage** (i.e., unexpected voltage levels on the line) on inputs configured as switch/sensor.

Every binary input needs itself to be configured as one of these two types: **push button** and **switch/sensor**.

2.2.1.1 PUSH BUTTON

The actions to be triggered on both, short and long presses (and even on the release of the push button) are independent and parameterisable. It is even possible to set **how long** a press must be to be considered as long.

These actions can consist in **sending the KNX bus** a binary value (0, 1 or an alternating value), a shutter control order, a dimmer control order, a scene run/save order or a constant numeric value (a 1-byte integer, a percentage value, a 2-byte integer or a 2-byte floating point).

A certain **delay** before sending the value to the bus can be configured in some cases. In the case of the binary values, a periodical re-sending can be also configured, which may be useful if intending to link such value to an alarm monitor or similar.

2.2.1.2 SWITCH/SENSOR

Binary values (configurable) will be sent to the bus whenever rising or falling edges are detected in the input line. In this case, it is the fact of switching from one state to the other one what matters, and not whether it is for a longer or shorter time. A simple example would be a sensor that commutes between two states depending on certain physical conditions (light, floods, weight…), being one of which undesired. The usual behaviour is that a certain binary value is sent to the KNX bus whenever the switch/sensor commutes from one state to the other one.

It is possible to introduce a certain **delay** prior to sending these values to the KNX bus – a delay for the “0” and a delay for the “1”, no matter which one is sent after which edge (rising / falling). Moreover, **periodically re-sending** the last value is possible by configuring the desired period.

Security checks can optionally be performed for inputs of the Switch/Sensor type, as long as an **end-of-line resistor** has been connected to the switch/sensor. The value of such resistor needs to be configured by parameter (the available values are 2.2 kΩ, 2.7 kΩ, 3.3 kΩ, 4.7 kΩ and 10 kΩ) as well as whether it has been connected in parallel or serially, which depends on the switch/sensor type (*normally open* or *normally closed*).

- If **normally open**, the line will remain at a low voltage level in the absence of the undesired condition. The occurrence of that situation, however, will cause a rising edge (the switch/sensor gets closed). This type of sensor requires connecting the end-of-line resistor **in parallel**.

- On the other hand, if **normally closed**, the line remains at a high voltage level until the occurrence of the undesired situation, which will cause a falling edge (the switch/sensor will open). This requires connecting the end-of-line resistor **in series**.

By means of this resistor, it will be possible to distinguish not only the two states of the switch/sensor, but also additional (unexpected) voltage levels (e.g., short-circuits and open circuits due to a **breakdown** or a **sabotage**), which will be reported to the bus through alarm objects.

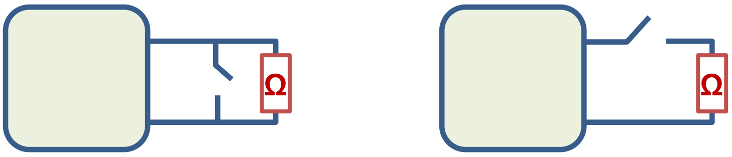


Figure 4. Left: normally open Sensor (parallel resistor). Right: Normally Closed Sensor (serial resistor).

2.2.1.3 LOCKING A BINARY INPUT

It is possible to **lock/unlock** each input independently by writing to the proper objects. While an input remains locked, the application will ignore further switches that may take place on the line, however the **periodical sending** of values, if parameterised, will not be interrupted (the last value will still be re-sent, even if the input switches the state). On the other hand, when the **unlock** event occurs:

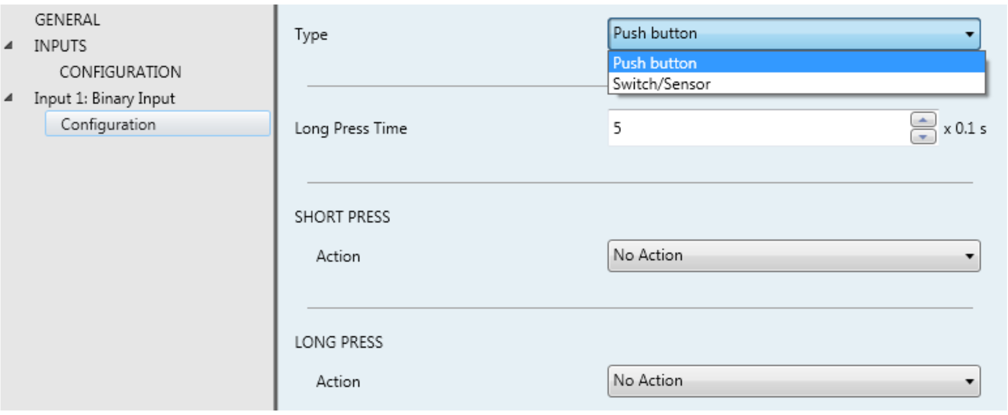
- **Push Button**: a fresh evaluation of the current state (high / low) of the line will be performed, as well as the corresponding action.
- **Switch/Sensor**: if specifically enabled in ETS, the current state of the line will be compared to that prior to the lock event. If they are different, it can be assumed that an edge (rising or falling) has taken place while the input remained locked, and therefore the associated action will be triggered.

2.2.1.4 INITIAL STATES

Finally, regarding the **initial states**, it is important to keep in mind the following:

- The **lock** state is maintained after a bus power failure. However, a download from ETS sets every input to *unlocked*.
- On switches/sensors with security checks, the **alarm** conditions are evaluated at the start-up of the device, and the alarm objects updated when required.
- The **periodical sending** of values, if configured in ETS, will be resumed after a power loss.
- When the device recovers from a power loss, the **new state of the push buttons** will always be re-evaluated, while that of the **switches/sensors** will be compared the previous one only if configured by parameter (similar behaviour as for the lock function).
- **Toggle** binary objects (0, 1, 0, 1…) are always sent with value “1” on the very first time after a download from ETS. This applies to switches/sensors configured with action “Switching 0/1”.

ETS PARAMETERISATION



Type: Push button

Long Press Time: 5 x 0.1 s

SHORT PRESS

Action: No Action

LONG PRESS

Action: No Action

Figure 5. Binary Input - Configuration

When an input has been configured as Binary Input, the object "**[Ix] Input Lock**" turns visible (when it receives a "1", the input will become locked, while a "0" will unlock it).

Apart from that, a specific entry will become visible in the menu on the left of the parameters window. This entry will itself comprise a tab called **Configuration**, containing the following parameter:

- **Type**: sets whether the input is a "Push button" (default) or a "Switch/sensor".

Push Button

When **Type** is set to "Push button", the following parameters are available:

- **Long Press Time**: sets the minimum time (1 to 50 tenths of a second) the user press should last before it is considered a long press.

- **Short Press – Action**: defines the action to be triggered when a short press is detected. The available actions are:

  - Nothing

  - Sending of 0/1.

    - **Response**: sets whether to send (through object "**[Ix] Short Press B**") a value (B), which can be "0", "1" or an alternation of "1" and "0".

    - **Delay**: sets a delay between the detection of the button press and the actual sending of the response. The delay must be set in seconds (0-255), minutes (0-255) or hours (0-18).

- **Periodical Response Sending**: sets whether to re-send the response values periodically or not. The options are "Always", "Only for 0", "Only for 1" and "No" (default). The cycle time must be set in seconds (0-255), minutes (0-255) or hours (0-18).

➢ Shutter Control.

- **Response**: sets the particular order to be sent to the shutter actuator. The options are:
    - "Up": one "0" will be sent through "**[In] [Short Press] Move Up Shutter**",
    - "Down": one "1" will be sent through "**[In] [Short Press] Move Down Shutter**",
    - "Up/Down (switched)": values "1" and "0" will be sent alternatively through "**[Ix] [Short Press] Move Up/Down Shutter**". In this case an additional, writable object ("**[Ix] [Short Press] Shutter Status (input)**") will be available; it may be linked to the shutter status object from the shutter actuator to receive feedback about the current state of the shutter. This will avoid sending move-up orders if the shutter is already at 0% or move-down orders if it is already at 100%.
    - "Stop/Step Up": one "0" will be sent through "**[Ix] [Short Press] Stop/Step Up Shutter**",
    - "Stop/Step Down": one "1" will be sent through "**[Ix] [Short Press] Stop/Step Down Shutter**",
    - "Stop/Switched Step": values "1" and "0" will be sent (alternatively with every press) through "**[Ix] [Short Press] Stop/Step Shutter (switched)**".
- **Delay**: similar as for "Sending of 0/1".

➢ Dimmer Control.

- **Response**: sets the particular order to be sent to the light dimmer:

- "Light ON": one "1" will be sent through the "**[Ix] [Short Press] Dimmer ON**" binary object,
- "Light OFF": one "0" will be sent through the "**[Ix] [Short Press] Dimmer OFF**" binary object,
- "Light ON/OFF (switched)": values "1" and "0" will be sent alternatively through the "**[Ix] [Short Press] Dimmer ON/OFF**" binary object,
- "Brighter": on every even press, one 4-bit order will be sent (through "**[Ix] [Short Press] Brighter**") to increase the light level by a certain percentage, which needs to be configured through parameter "**Dimming Step**". On the other hand, on every odd press, an order to stop the dimming will be sent. The sequence is therefore Brighter → Stop → Brighter → Stop → etc.
- "Darker": analogous to the previous one, but for decreasing the light level,
- "Brighter/Darker (switched)": analogous to the above two, although the dimming orders will be in this case increase/decrease orders (alternatively). The sequence will be therefore Brighter → Stop → Darker → Stop → Brighter → etc. These orders will be sent through object "**[Ix] [Short Press] Brighter/Darker**". In this case there will be an additional, writable object ("**[Ix] [Short Press] Dimming Status (input)**" that may be linked to the dimming status object from the dimmer, in order to receive feedback about the current light level. This will avoid increase orders if the current level is 100% or decrease orders if it is already 0%

- **Delay**: similar as for "Sending of 0/1".

➢ Sending of a Scene.

- **Response**: sets the particular order to be sent to the bus. The options are "Run Scene" (an order to run a specific scene will be sent through "**[Ix] [Short Press] Run Scene**") and "Save Scene" (an order to save the current state as a specific scene will be sent through "**[Ix] [Short Press] Save Scene**").

- **Scene**: sets the desired scene number (1 to 64) for the above run/save orders.

- **Delay**: similar as for "Sending of 0/1".

➢ 1-Byte Constant (Integer):

- **Response**: sets a constant value to be sent to the bus (through "**[Ix] [Short Press] Constant Value (Integer)**"), in the range 0 to 255.

➢ 1-Byte Constant (Percentage):

- **Response**: sets a constant value to be sent to the bus (through "**[Ix] [Short Press] Constant Value (Percentage)**"), in the range 0 to 100.

➢ 2-Byte Constant (Integer):

- **Response**: sets a constant value to be sent to the bus (through "**[Ix] [Short Press] Constant Value (Integer)**"), in the range 0 to 65535.

➢ 2-Byte Constant (Float) :

- **Response**: sets a constant value to be sent to the bus (through "**[Ix] [Short Press] Constant Value (float)**"), in the range -671088.64 to 670760.96.

- **Long Press – Action**: defines the action to be triggered when a long press is detected. The available actions are totally analogous as for the short press.

Switch/Sensor

When **Type** has been set to "Switch/Sensor", the following parameters are also available:

- **Security**: selecting or unselecting this checkbox determines whether the input line includes a security end-of-line resistor so it is therefore possible to monitor sabotage or breakdown situations (which will be notified by periodically sending the value "1" through object "**[Ix] [Switch/Sensor] Alarm: Breakdown or Sabotage**"; once they are over, this object will send one "0"). When selected, two more parameters come up:

  - ➢ **Switch/Sensor Type**: sets whether the switch/sensor is normally open and therefore with a resistor connected in parallel ("N.O. (parallel resistor)") or normally closed and therefore with a resistor connected in series ("N.C. (serial resistor)").

  - ➢ **Resistor Value**: sets the value of the resistor, which should be one of the following: 2.2 kΩ, 2.7 kΩ, 3.3 kΩ, 4.7 kΩ and 10 kΩ.

- **Actions.**

  - ➢ **Rising Edge**: indicates what should be sent to the KNX bus when a rising edge takes place on the input line. The options are "No Action" (default), "0", "1" and "Switching 0/1" (i.e., the values "1" and "0" will alternate with every rising edge detected). These values are sent through object "**[Ix] [Switch/Sensor] Edge**".

  - ➢ **Falling Edge**: analogous to the above parameter. The response to falling edges will be sent through the same object ("**[Ix] [Switch/Sensor] Edge**").

- **Periodical Sending.**

  - ➢ **Sending of "0"**: sets every how much time (0 to 255 seconds, 0 to 255 minutes or 0 to 18 hours) the value "0" should be periodically sent once the corresponding edge has been detected. If no periodical re-sending is required, just leave this at 0.

  - ➢ **Sending of "1"**: analogous to the above one, but for the value "1".

- **Delay.**

  - ➢ **Sending of "0"**: sets a certain delay (0 to 255 seconds, 0 to 255 minutes or 0 to 18 hours) before sending the value "0" once the corresponding edge has been detected. For an immediate sending, just leave this at 0.

  - ➢ **Sending of "1"**: analogous to the above one, but for the value "1".

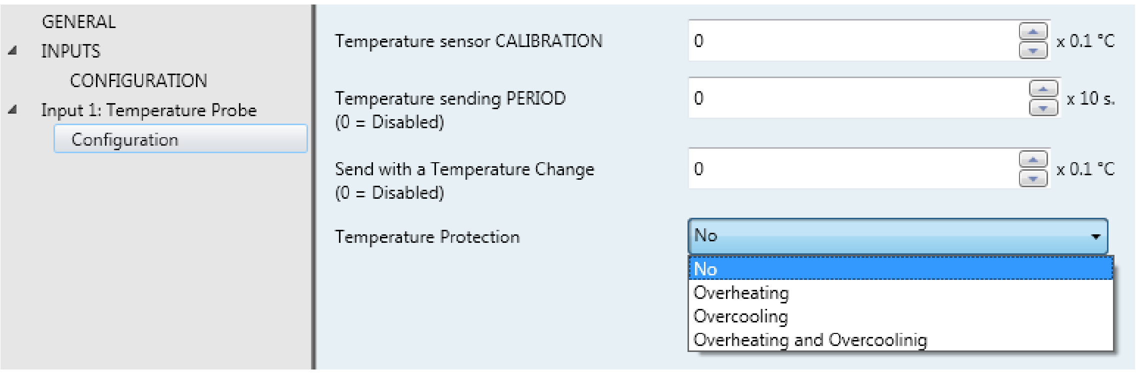
- **Evaluate the Input State after Unlock or Reset**: sets whether the state of the input line should be evaluated or not whenever it gets unlocked (through object "**[Ix] Input Lock**") or when the device recovers from a power loss, so this new state can be compared to the last known, making the device trigger the proper response in case the two are different.

- **Sending Status (0 and 1) on Bus Voltage Recovery**: sets if the status of the line (i.e., the corresponding action) should always be sent to the bus when the device recovers from a power loss, even if the status is the same as before the power loss.

2.2.2 TEMPERATURE PROBE

Connecting a temperature probe (such as models ZN1AC-NTC68 S / E / F from Zennio) to one of the inputs will make it possible to take temperature measurements and monitor them in MAXinBOX 66, which will be able to **send those values to the bus** as well as to report **events of high / low temperature** detection.

ETS PARAMETERISATION



Temperature sensor CALIBRATION: 0 x 0.1 °C

Temperature sending PERIOD (0 = Disabled): 0 x 10 s.

Send with a Temperature Change (0 = Disabled): 0 x 0.1 °C

Temperature Protection: No

Figure 6. Temperature Probe - Configuration

When an input has been configured as Temperature Probe, objects "**[Ix] Current Temperature**" (2 bytes) and "**[Ix] Probe Error**" (1 bit) become visible. The former will report the current value of the temperature (periodically or after a certain increment/decrement, according to the parameter configuration), while the latter will notify (by sending the value "1" periodically) about an unexpected reading in the input line due to a **breakdown** or an incorrect connection of the temperature probe.

Apart from that, a specific entry will become visible in the menu on the left of the parameter window. This entry will itself comprise a tab called **Configuration**, containing the following parameters:

- **Temperature Sensor Calibration**: defines an offset to be applied to the measurement received from the probe to correct deviations due to external factors. The offset should be in the range [-50, 50] tenths of a degree.

- **Temperature Sending Period**: sets every how much time the value of the current temperature should be sent to the bus (through "**[Ix] Current Temperature**"), in the range 0 – 255 tens of second. The value "0" leaves this periodical sending disabled.

- **Send with a Temperature Change**: defines a threshold (in the range 0 to 255 tenths of a degree) so that whenever a new reading of the current temperature is found to differ (from the last value sent to the bus) more than such threshold, an extra sending will take place.

- **Temperature Protection**: drop-down list with the following options:

  - No: no temperature protection is required.

  - Overheating: overheating protection is required. Two extra parameters will come up:

    - **Overheating Temp.**: maximum temperature permitted, in the range -30 to 125 ºC. Temperature readings greater than this will be considered overheat, and therefore a "1" will be periodically sent through object "**[Ix] Overheat**". Once the overheat is over, a "0" will be sent (once).

    - **Hysteresis**: *dead band* or threshold (in the range 1 to 200 tenths of a degree; the default is 20 tenths of a degree –i.e., two degrees–) around the overheat temperature defined above. This dead band prevents the device from sending the overheat alarm and no-alarm over and over when the current temperature is moving around the overheat limit (T): once the overheat alarm has been triggered, the no-alarm will not be sent until the current temperature is lower than that T minus the hysteresis. After that, if the current temperature reaches T again, the alarm will be re-sent.

  - Overcooling: overcooling protection is required. Two extra parameters (analogous to the above two) will come up:

    - **Overcooling Temp.**: minimum temperature permitted, in the range -30 to 125 ºC. Temperature readings lower than this will be considered overcool, and therefore a "1" will be periodically sent through object "**[Ix] Overcool**". After the overcool is over, a "0" will be sent (once).

    - **Hysteresis**: dead band or threshold (in the range 1 to 200 tenths of a degree; the default is 20 tenths of a degree) around the overcooling temperature. As for the overheat, once the alarm has been triggered, the no-alarm will not be sent until the current temperature is greater than T plus the hysteresis. After that, if the current temperature reaches T again, the alarm will be re-sent.

- Overheating and Overcooling: both overheating and overcooling protection are required. The following three parameters will come up:
  - **Overheating Temp.**
  - **Overcooling Temp.**
  - **Hysteresis**.

  The three of them are analogous to those already explained separately.

2.2.3 MOTION DETECTOR

It is possible to connect motion detectors to the input ports of MAXinBOX 66. This brings the actuator 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 bear in mind that the functionality and configuration of the motion detection function may differ slightly depending on each Zennio device, so please refer to the "**Zennio Motion Detector for MAXinBOX 66**" specific user manual (available at the Zennio homepage, www.zennio.com) for detailed information.

On the other hand, please note that only detectors with references **ZN1IO-DETEC-P** and **ZN1IO-DETEC-X** are compatible with MAXinBOX 66, as long as the corresponding micro-switch is set to the **Type B** mode.

Reading the **datasheet of the specific model** of the motion detector being connected is highly advisable.

Note: *detectors with references ZN1IO-DETEC and ZN1IO-DETEC-N are Type A, and therefore may report inexact measurements if connected to MAXinBOX 66.*

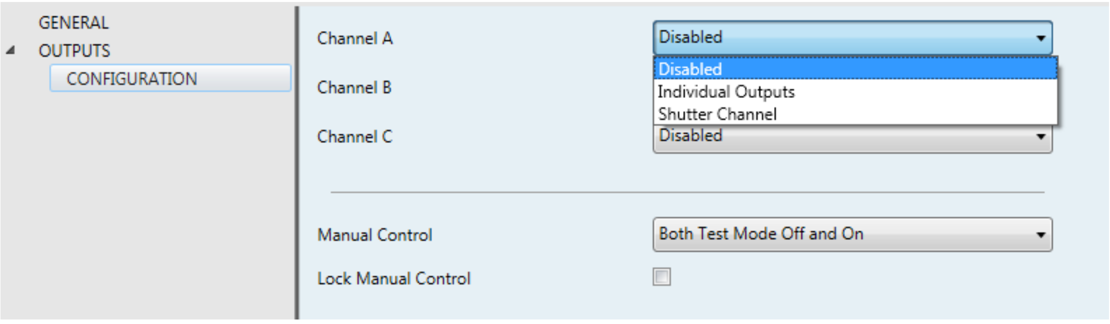
2.3 OUTPUTS

MAXinBOX 66 incorporates **6 relay outputs** that allow an independent control of up to **6 different loads**, or up to **3 shutter drives** (or equivalent window/door automated systems), named A, B and C. Each output can be enabled or disabled by parameter and perform different functions.

MAXinBOX 66 also features a set of on-board pushbuttons that bring the integrator the option to perform a **manual control** of the output channels, for example for testing purposes. How the outputs react to these buttons will depend on how the device has been parameterised, as explained in the later sections.

ETS PARAMETERISATION

When **Outputs** has been activated in the General parameter screen, the following screen will be available for the selection of the specific function required for each output channel.



Channel A: Disabled
Channel B
Channel C: Disabled
Manual Control: Both Test Mode Off and On
Lock Manual Control: □

Figure 7. Outputs - Configuration

All outputs are disabled by default. Depending on the function selected for each output, some additional tabs will be included in the menu on the left.

2.3.1 INDIVIDUAL OUTPUTS

Every individual output can be configured as **normally open** (i.e., switching the output on makes the relay close) or **normally closed** (i.e., the switching the output on makes the relay open).

Besides the output type, the following options can be configured for each individual output:

- **Timers**: permits performing timed actions over the outputs.

  On the first hand, the **Flashing** function consists in performing a continuous, timed on/off sequence when a specific trigger object is received.

  On the other hand, the **Simple Timer** function consists in performing a single, timed switch-on / switch-off when one "1" (or one "0") is received through a specific object ("**[Ox] Timer**"). Depending on whether the order is a switch-on or a switch-off, a certain delay ("**On Delay**" or "**Off Delay**") will apply. These delays are parameterisable. Moreover, in case it is a switch-on, an "**On Duration**" should be defined. This is the time the output will remain on once the On Delay has expired and therefore the output has finally switched on.

  - When enabled, the **Multiplication** function permits progressively multiplying the pre-set ON Delay, OFF Delay or ON Duration (depending on which of them is currently counting) as many times as the switch-on or switch-off orders are received through "**[Ox] Timer**". Different situations are distinguished:

    - **Without Multiplication**:

      - If the On delay count is already running, it will be re-started every time a new "1" is received through "**[OX] Timer**".

      - If the output has already been activated and the On Duration time is counting, it will be re-started whenever a new "1" is received.

      - If the Off delay count is already running, it will be re-started every time a new "0" is received.

- **With Multiplication**:
  - If the On delay count is already running and the value "1" is received several times through the "**[OX] Timer**" object, then the effective delay time will be "n" times the parameterised time, being "n" the number of times the value "1" is received (**five times** at most; further receptions will be ignored until the next trigger of the simple timer function).
  - While the On Duration time is counting after the switch-on of the output, if one "1" is received several times, then the effective duration will be "n" times the parameterised time, being "n" the number of times the value is received (as above, **five** at most).
  - If the Off delay count is already running and the value "0" is received several times, then the effective delay time will be "n" times the parameterised time, being "n" the number of times the value "0" is received (as above, **five** at most).

Note: *the Multiplication option is useful under parameterisations with no ON / OFF delays. Nevertheless, as already explained and as the following example shows, these delay times do also admit multiplication.*

The example shows how the multiplication works, assuming On and Off delays of 3 seconds, and an On duration of 5 seconds. "→0" and "→1" represent the arrivals of the values "0" and "1" through "**[OX] Timer**".

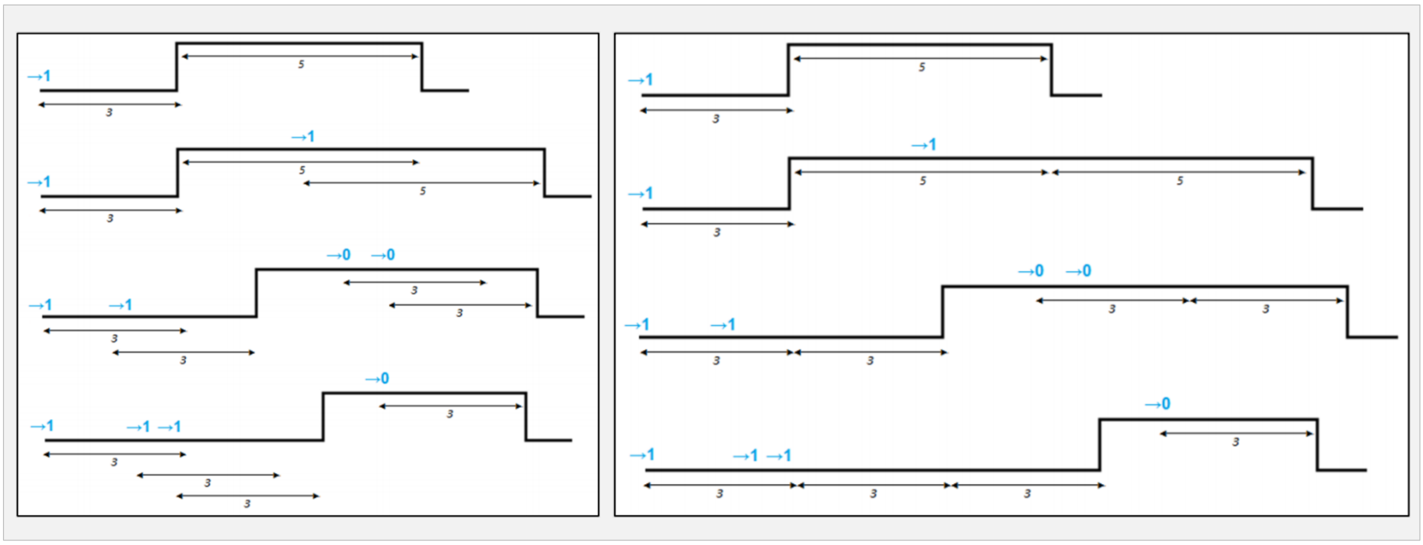


Figure 8. No multiplication (left) and Multiplication (right)

- **Scenes**: allows running a switch-on or a switch-off of the output on the reception of a scene object.

- **Alarms**: allows changing the state of the output on the reception of an alarm trigger from the KNX bus. It is possible to configure the state the output will be switched to, both on the alarm activation and on the alarm deactivation.

  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 actuator 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. The first case triggers the deactivation action as soon as the alarm object recovers its normal value. The second one, on the other hand, requires that an external acknowledgement (through another object) is received after the alarm object has recovered its normal value.

  Note: *alarms always prevail over any other function (e.g.: lock orders during the alarm state will be ignored; alarm orders during the lock state will not).*

- **Lock Action**: permits switching the output to a particular state (ON / OFF / No Change) upon the reception of the value "1" through a specific lock object (one per output).

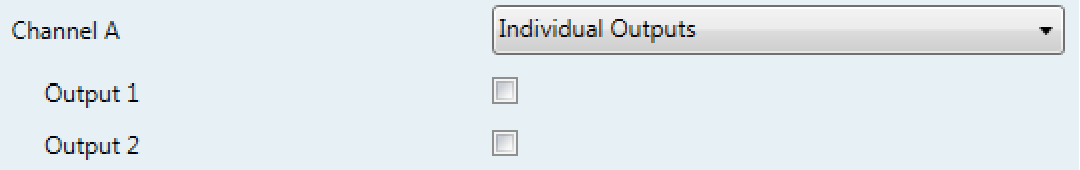
- **Shutdown Action**: brings the option to set the output to a particular state (ON / OFF / No Change) during the switch-off of the actuator (e.g., on a bus power loss).

- **Start-up configuration**: brings the option to set the output to a particular state at the start-up of the actuator.

  - Default configuration: off after an ETS download, and unchanged after a bus power failure.

  - Custom configuration: Off / On / Last, after either an ETS download or a bus power failure (last will be off on the very first start-up). The status object can also be sent to the bus (with a customisable delay).

ETS PARAMETERISATION



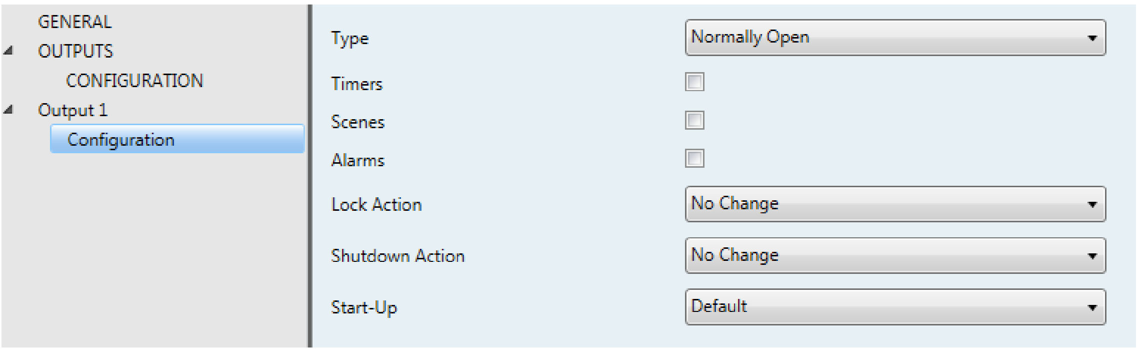
Channel A: Individual Outputs

Output 1 □

Output 2 □

Figure 9. Enabling each Output Separately

When a channel has been configured as Individual Outputs, two checkboxes will be unhidden for enabling or disabling independently the two relay outputs that constitute the channel.



Type: Normally Open

Timers □

Scenes □

Alarms □

Lock Action: No Change

Shutdown Action: No Change

Start-Up: Default

Figure 10. Individual Output - Configuration

A specific parameter screen per individual output will be included in the menu on the left after such output has been enabled. This screen contains the following parameters:

- **Type**: should be set to “Normally Open” (default) or “Normally Closed” depending on the *natural* state of the relay.
- **Timers**: activates or deactivates the Timers function, which should be configured from a specific parameter screen (see below).
- **Scenes**: activates or deactivates the Scenes function, which should be configured from a specific parameter screen (see below).
- **Alarms**: activates or deactivates the Alarm function, which should be configured from a specific parameter screen (see below).

- **Lock Action**: sets whether to leave the relay as is ("No Change", default option), to switch it off ("Off") or to switch it on ("On") in the event of a lock order through object "**[Ox] Lock**" (1 = lock; 0 = unlock).

- **Shutdown Action**: sets whether to leave the relay as is ("No Change", default option), to switch it off ("Off") or to switch it on ("On") in the event of a shutdown of the device due for example to a bus power failure.

- **Start-Up**: sets whether to perform the default action ("Default") or a custom action ("Custom") over the output during the device start-up. The latter should be configured from a specific parameter screen (see below).

After setting an output to Individual Output, three more objects are shown by default:

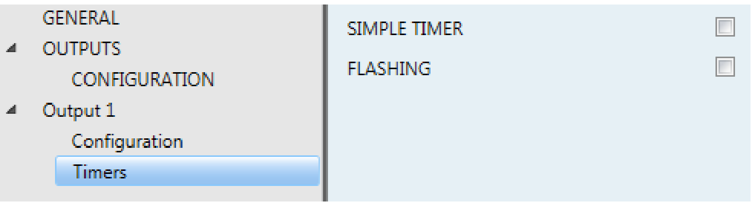
- **[Outputs] Scenes**: 1-byte object intended for the reception of scene values (0-63 for running scenes 1-64, and 128-191 for saving scenes 1-64) from the KNX bus. It becomes available after enabling the first individual output, as it is a sole object for every other individual output enabled in the actuator.

- **[Ox] On/Off**: 1-bit object for the reception of on/off orders from the bus. One "1" will make the output switch-on and one "0" will make it switch off. Depending on whether the output has been set to Normally Closed or to Normally Open, switching the output on will consist in opening or closing the relay (and the opposite action for the switch-off).

- **[Ox] On/Off Status**: 1-bit object which will report the current state of the individual output. It can be read anytime and will respond with a "1" or a "0" depending on whether the output is on or off, respectively.

Timers Tab

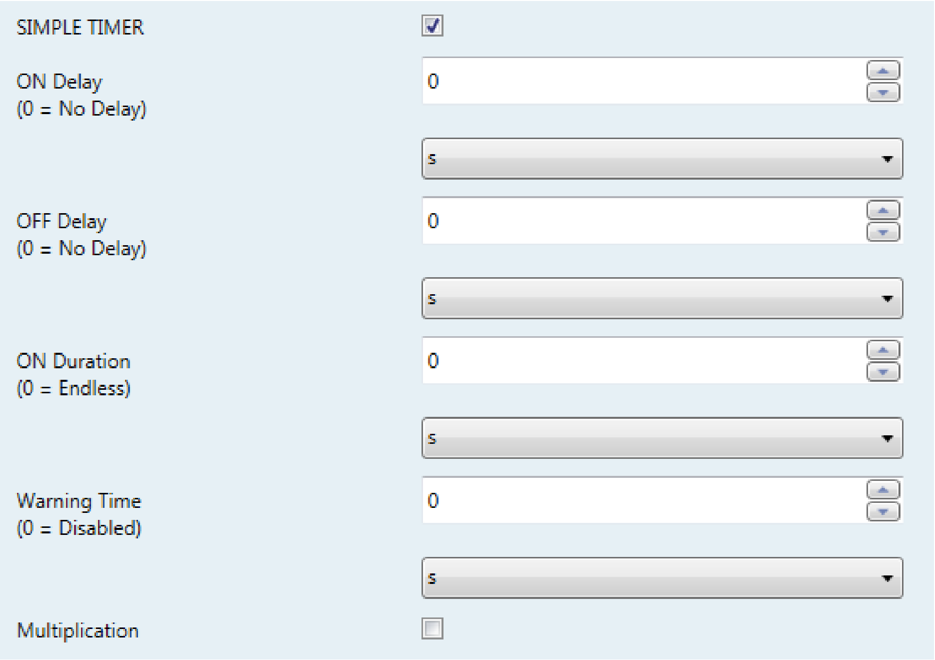
The two different timer functions can be enabled through their specific checkboxes:



SIMPLE TIMER □
FLASHING □

Figure 11. Individual Output - Timers

- **Simple Timer**: enables or disables the simple timer function, i.e., the execution of a timed switch-on or switch-off after the reception of one “1” or one “0” through “**[Ox] Simple Timer**”, according to the following parameters:



SIMPLE TIMER ☑

ON Delay (0 = No Delay): 0 s

OFF Delay (0 = No Delay): 0 s

ON Duration (0 = Endless): 0 s

Warning Time (0 = Disabled): 0 s

Multiplication □

Figure 12. Individual Output – Simple Timer

- **ON Delay**: sets whether the “on” action should be executed immediately (“0”; default option) after receiving one “1” through object “**[Ox] Timer**”, or if it should be delayed by a certain time (0 to 600 tenths of a second; 0 to 3600 seconds; 0 to 1440 minutes; 0 to 24 hours).

- **OFF Delay**: analogous to the above one, but referred only to the switch-off order performed when one “0” arrives through “**[Ox] Timer**”.

- **ON Duration**: sets how much time the output will remain on once the “ON delay” time count ends and the output gets actually switched on (0 to 600 tenths of a second; 0 to 3600 seconds; 0 to 1440 minutes; 0 to 24 hours). If set to zero, the output will not switch off after a certain time.

- **Warning Time**: sets an anticipation time (0 to 600 tenths of a second; 0 to 3600 seconds; 0 to 1440 minutes; 0 to 24 hours) prior to the switch-off action so that the output starts intermitting (two seconds time) to notify that the timer action is about to end. Moreover, the “**[Ox] Warning Time (Status)**” object will acquire the value “1” while this intermittence is running (and "0" at any other time). If set to zero, no warning intermittence will take place. Note that this time needs to be lower than the ON Duration time.

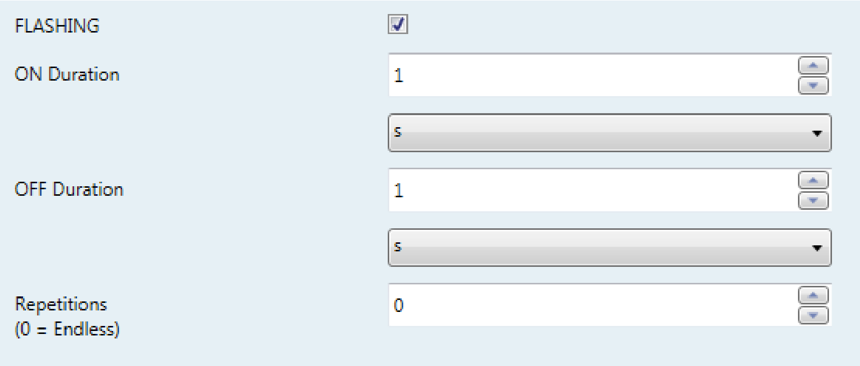
- **Multiplication**: enables or disables the multiplication function.

- **Flashing**: when enabled, object "**[Ox] Flashing**" will show up in the project. One "1" through this object triggers the intermittence of the output, while one "0" will stop it. This intermittence will be subject to the following parameters:

  - **ON Duration**: length of each "on" stage (5 to 600 tenths of a second; 1 to 3600 seconds; 1 to 1440 minutes; 1 to 24 hours).

  - **OFF Duration**: length of each "off" stage (5 to 600 tenths of a second; 1 to 3600 seconds; 1 to 1440 minutes; 1 to 24 hours).

  - **Repetitions**: number of times (0-255) the on-off sequence will take place. If set to "0", the intermittence will only stop when one "0" is received through "**[Ox] Flashing**".



FLASHING ☑

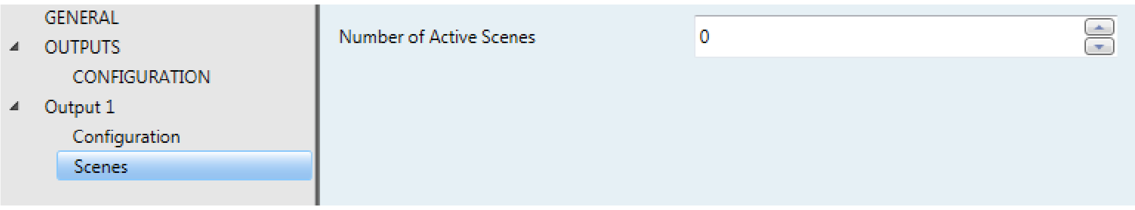
ON Duration: 1 s

OFF Duration: 1 s

Repetitions (0 = Endless): 0

Figure 13. Individual Output – Flashing

Scenes Tab



Number of Active Scenes: 0

Figure 14. Individual outputs. Scenes

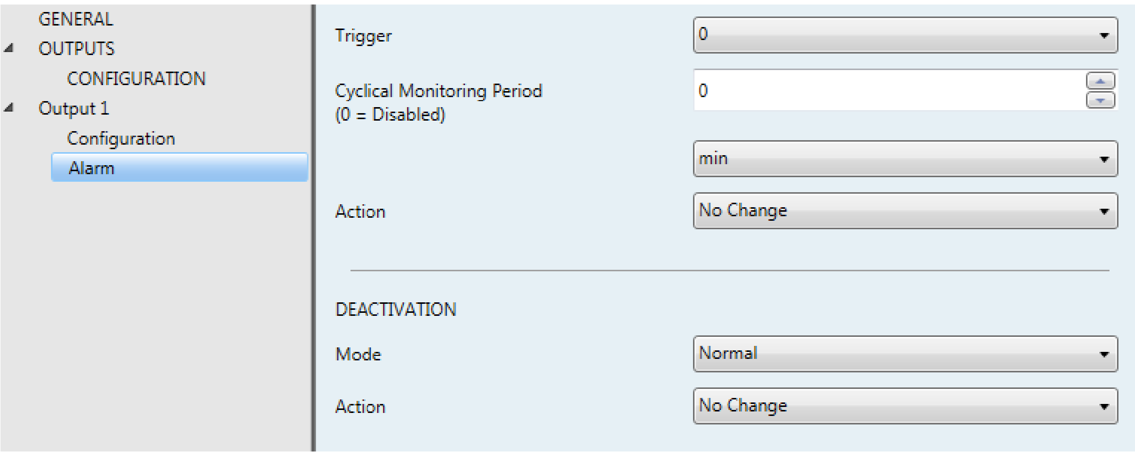
Up to five scenes can be configured. The number of scenes must be entered into **Number of Active Scenes** ("0", by default).

Note: *after typing the value, it is necessary to move the user focus to another part of the window in order to have it refreshed with the required parameters*.

Depending on the above number, the following two parameters will be shown multiple times (one per scene):

- **Scene Number**: sets the desired scene number, so that when that value is received (decreased by one, according to the KNX standard) through "**[Outputs] Scenes**", the action configured below will be triggered.
- **Action**: "Off" (default) or "On", depending on the state the output should acquire when the scene is triggered.

Alarms Tab



Trigger: 0

Cyclical Monitoring Period (0 = Disabled): 0 min

Action: No Change

DEACTIVATION

Mode: Normal

Action: No Change

Figure 15. Individual Outputs - Alarms

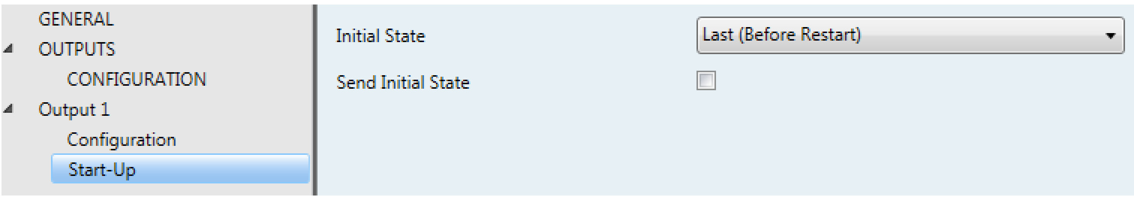
The specific Alarm configuration screen contains the following parameters:

- **Trigger**: sets the value ("0" –default– or "1") that when received from the KNX bus through object "**[Ox] Alarm**" will be interpreted by the actuator 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 5 to 600 tenths of a second; 1 to 3600 seconds; 1 to 1440 minutes; and 1 to 24 hours.

- **Action**: "No Change" (default), "Off", "On" or "Flashing". Selecting the latter brings some more parameters:
  - "**ON Duration**", "**OFF Duration**" and "**Number of Repetitions**": all of them are analogous to those in the Flashing function, within Timers (see preceding parameters).
- **Deactivation – Mode**: "Normal" (default) or "Frozen (Acknowledgement Needed)". The second option enables a new 1-bit object, "**[Ox] Unfreeze Alarm**", which should be used for externally unfreezing the alarm once "**[Ox] Alarm**" has received the no-alarm value (i.e., the inverse of the trigger value).

  Note: *the acknowledgement should be sent necessarily after "**[Ox] Alarm**" has acquired the no-alarm value. Sending it while the trigger value is still active will have no effect.*
- **Deactivation – Action**: sets the state the output should acquire once the alarm has been deactivated (and acknowledged, if required). It may be: "No Change" (default), "Off", "On" or "Last (Before Alarm)".

Start-Up Tab



Initial State: Last (Before Restart)

Send Initial State: □

Figure 16. Individual Outputs - Start-Up

Contains the following parameters:

- **Initial State**: sets the state the output should acquire at the start-up of the actuator. It can be: "Last (Before Restart)" (default; during the very first start-up, the output will remain off), "Off" or "On".
- **Send Initial State**: sets whether the status object of the output should be sent to the KNX bus (in order to update other KNX devices) after the start-up of the actuator. It is possible to impose a delay (0 to 600 tenths of a second; 0 to 3600 seconds; 0 to 1440 minutes; 0 to 24 hours) to ensure that this value is sent once the other devices are ready to receive it.

2.3.2 SHUTTER CHANNELS

Every shutter channel is capable of controlling the motion of one blind in the domotic system, by means of two complementary functionalities:

- **Basic control** (simple up/down orders).
- **Precise position control** of the shutter and of the slats (if any).

Each shutter channel (A, B, etc.) consists of two consecutive relay outputs (i.e., channel A is formed by outputs 1 and 2; channel B is formed by outputs 3 and 4; etc.). The first output of every channel will send electric pulses to raise the shutter, whereas the second output will send pulses to lower the shutter. The cables from the motor of the shutter drive should be connected to the actuator in accordance with the above.

Channel	Outputs	Action
A	Output 1	Move up
	Output 2	Move down
B	Output 3	Move up
	Output 4	Move down
(…)	(…)	(…)

Table 1. Shutter Channel. Actions of the Outputs.

Each shutter channel can be configured as "**Shutter (No Slats) / Awning**" or as "**Blinds (With Slats)**".

Besides the shutter type, it is possible to configure the following for the shutter channels:

- **Times**: times that define the motion of the shutter: the length of the rising course, the length of the lowering course and a safety reversion delay to prevent mechanical and electrical issues due to a sudden direction change when the shutter was already in motion. Moreover, it is possible to set an additional end-of-stroke time to be applied once the shutter gets to its limit (top or bottom).

  For blinds with **slats**, a spin-around time for the entire slat course and a step time for brief step movements need to be configured as well.

Regarding the slats, it is possible to set whether the actuator should *force* the slats to maintain their position after the shutter moves up or down. Note that due to the fact that both the shutter and the slats move together, maintaining the position of the slats may sometimes require performing a certain correction of the position of the shutter.

- **Status Objects**: they report the current position of the shutter and that of the slats (if existing). 0% means shutter totally up (or slats totally open, with the light passing through), while 100% means shutter totally down (or slats totally closed).

- **Precise Control Objects**: they bring the option to raise/lower the shutter or the slats to the exact desired position, in terms of percentage.

- **Scenes**: allows setting the shutter (or the slats) to a certain position on the reception of a scene object.

- **Alarms**: allows moving the shutter (or the slats) to a pre-defined position on the reception of an alarm trigger from the KNX bus. Two alarms (i.e., two independent trigger objects with independent target positions) are provided per shutter channel. It is possible to configure not only the position the shutter will be set to on the alarm **activation** but also on the **deactivation**.

  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 does not take place if the object has never been received yet). In case the object stops being refreshed (i.e., the actuator 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.

  - ➢ The first case triggers the deactivation action as soon as the alarm object recovers its normal value.

  - ➢ The second one, on the other hand, requires that an external acknowledgement (through another object) is received after the alarm object has recovered its normal value.

<u>**Note**</u>:

- ***Alarms always prevail over any other function*** *(e.g.: lock orders during the alarm state will be ignored; alarm orders during the lock state will not).*

- ***Alarm 1 has a higher priority than alarm 2****. If the channel is in "alarm 2" 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 "alarm 2", but without executing again the action of alarm 2). On the other hand, while one channel is in "alarm 1", triggering alarm 2 will have no effect: the channel will remain in "alarm 1" until alarm 1 gets deactivated (in that moment the channel will perform the action of alarm 2).*

- **Reverse movement**: this makes it possible to control a shutter with the usual orders inverted. While the default shutter control object reacts to a "1" and a "0" respectively raising and lowering the shutter, a specific object will be enabled to control it the other way round ("1" for lowering and "0" for raising).

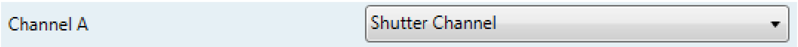
- **Direct Positioning**: permits moving the shutter (and the slats, if existing) to a pre-set, specific position by means of a 1-bit communication object. Two Direct Positioning functions are implemented per shutter channel, each with the option of *learning* new target positions in runtime (i.e., overwriting the one defined by parameter) by simply sending one "1" to the specific "save" object.

- **Start-up Configuration**: brings the option to set the shutter to a particular position during the start-up of the actuator.

  - <u>Default configuration</u>: this will leave the shutter as is. Please note that on the very first start-up (after an ETS download), the actuator will assume the shutter is fully raised (at 0%).

  - <u>Custom configuration</u>: Current Position / Up / Down / Specific Position, both after an ETS download and a bus power failure. Optionally, the status objects can be sent to the bus during after a customisable delay.

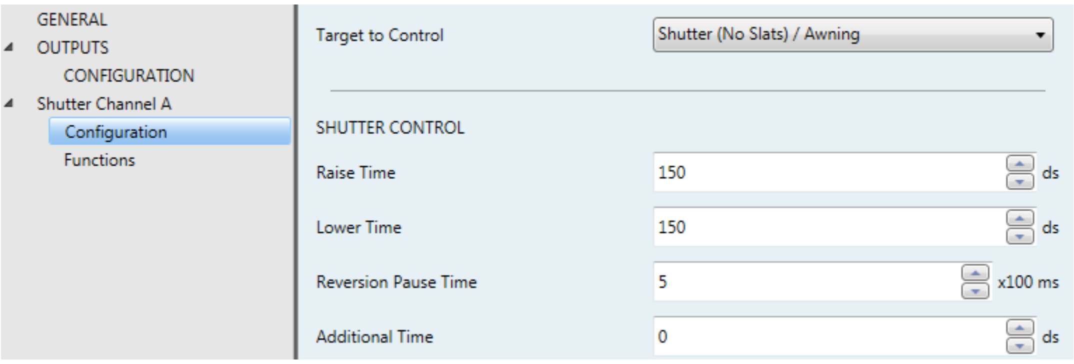
ETS PARAMETERISATION



Channel A: Shutter Channel

Figure 17. Enabling a Shutter Channel

Configuration tab



Target to Control: Shutter (No Slats) / Awning

SHUTTER CONTROL

Raise Time: 150 ds

Lower Time: 150 ds

Reversion Pause Time: 5 x100 ms

Additional Time: 0 ds

Figure 18. Shutter Channel - Configuration

This screen contains the following parameters:

- **Target to Control**: defines the shutter channel as "Shutter (No Slats) / Awning" (default) or "Blinds (With Slats)".

- **Raise Time**: total time the shutter needs to move to the top (position = 0%) from the bottom (position = 100%). The allowed range is 5 to 30,000 seconds.

- **Lower Time**: total time the shutter needs for the inverse journey (0% to 100%). The allowed range is 5 to 30,000 seconds.

- **Reversion Pause Time**: sets the time (5 to 255 hundred milliseconds) the actuator will wait if it was already in motion and an order to switch the direction arrives.

- **Additional Time**: sets an end-of-stroke time (0 to 255 tenths of a second; by default, 0) in order to guarantee the shutter gets to the top or to the bottom.

In case of configuring the shutter as "Blinds (With Slats)", other parameters show up:

- **Spin-Around Time**: total time the slats take for the entire travel between "totally open" (0%, with the light passing through) and "totally closed" (100%). Valid values are 1 to 255 tenths of a second.

- **Step Time**: length of the pulse to be sent to the drive when an up/down step order is received. Valid values are 1 to 255 hundred milliseconds.

- **Recover Position After Shutter Completes Motion**: sets whether to try (or not) to preserve the position of the slats after moving the shutter.

- **Position at Bottom**: defines the position the slats should move to once the shutter itself gets to the bottom position (100%, completely down).

The following objects are visible by default for every shutter channel:

- **[Shutter] Scenes**: 1-byte object for the reception from the KNX bus of scene orders, the response to which needs to be configured from the Scenes specific tab of each shutter channel, as explained later.

- **[Cx] Move**: 1-bit object for the reception from the KNX bus of the shutter move-up ("0") and move-down ("1") orders.

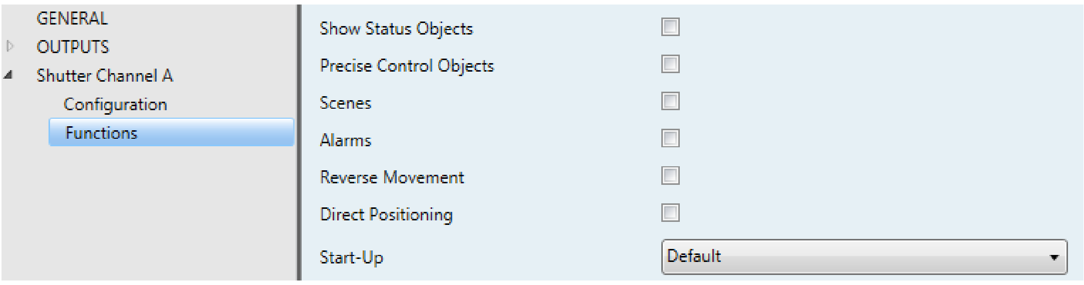
- **[Cx] Stop**: 1-bit object for the reception from the KNX bus of the shutter stop orders ("0" or "1"), which will interrupt any move-up or move-down order in execution. If the shutter has been configured as "Blinds (With Slats)", the name of the object will be "**[Cx] Stop/Step**" and it will implement exactly the same function (interrupt a move-up or move-down order in execution), plus the step control function: if the shutter is still, one "0" will be interpreted as a step-up order, and one "1" will be interpreted as a step-down order.

  Note: *successive step orders received before the end of the step movement will reset the step time counter.*

- **[Cx] Lock**: 1-bit object for externally locking ("1") or unlocking ("0") the shutter. When the lock trigger is received, the actuator will interrupt any action being performed and will ignore further orders received from the bus until the unlock trigger is received.

  Note: *lock orders are ignored if the shutter alarm is active (although the alarm state itself also implies that the shutter does not respond to external orders).*

Functions tab

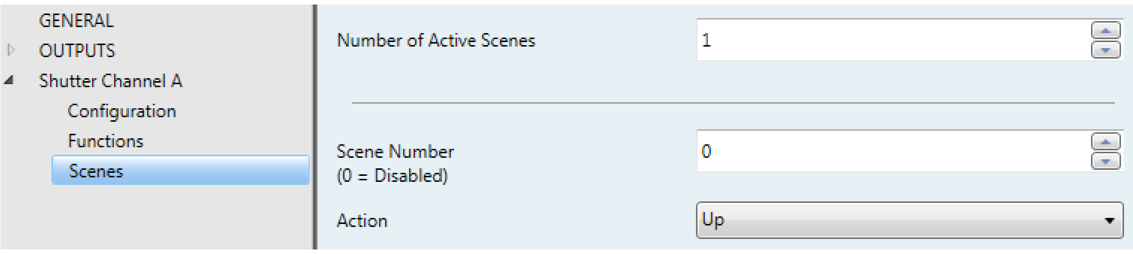


Show Status Objects □
Precise Control Objects □
Scenes □
Alarms □
Reverse Movement □
Direct Positioning □
Start-Up: Default

Figure 19. Shutter Channel - Functions

This screen lets the integrator enable/disable a variety of additional functions related to the shutter channel control.

- **Show Status Objects**: enables or disables two 1-byte objects ("**[Cx] Shutter Position (Status)**" and "**[Cx] Slats Position (Status)**") that will reflect, respectively, the instant position of the shutter and of the slats (if available) in terms of percentage.
  - **Periodic Notification While Moving (1s)**: sets whether the status objects should be sent (updated) every one second while the shutter or the slats are moving, or just at the end of the motion.
- **Precise Control Objects**: enables or disables two 1-byte objects ("**[Cx] Shutter Positioning**" and "**[Cx] Slats Positioning**") that will accept precise position orders (in terms of percentage) from the KNX bus for the shutter and the slats respectively.
- **Scenes**: activates or deactivates the Scenes function, which should be configured from a specific parameter screen:



Number of Active Scenes: 1

Scene Number (0 = Disabled): 0

Action: Up

Figure 20. Shutter Channel - Scenes

Up to five scenes can be configured. The number of scenes must be entered into **Number of Active Scenes** ("0", by default).

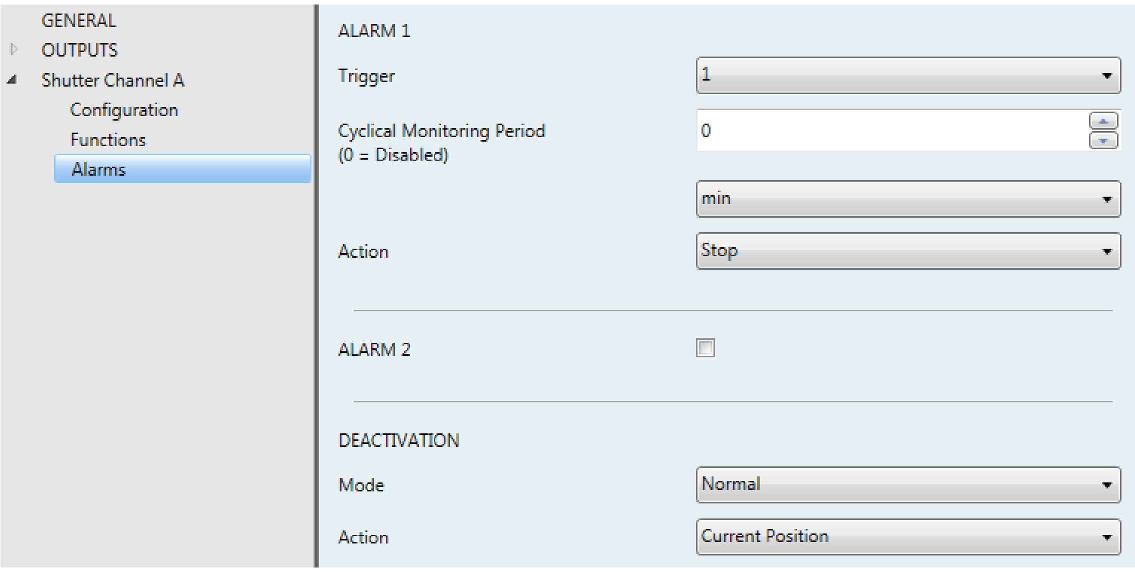
Note: *after typing the value, it is necessary to move the user focus to another part of the window in order to have it refreshed with the required parameters.*

Depending on the above number, the following two parameters may become visible multiple times (one per scene):

- **Scene Number**: sets the desired scene number, so that when that value is received (decreased by one, according to the KNX standard) through "**[Shutter] Scenes**", the action configured below will be triggered.

- **Action**: “Up” (default), “Down” or “Specific Position”, respectively depending on whether the shutter should move to the top position, to the bottom position or to an intermediate position. The latter permits independently configuring a position for the shutter itself, for the slats, or for both.

- **Alarms**: activates or deactivates the Alarms function, which should be configured from a specific parameter screen.



ALARM 1

Trigger: 1

Cyclical Monitoring Period (0 = Disabled): 0 min

Action: Stop

ALARM 2 □

DEACTIVATION

Mode: Normal

Action: Current Position

Figure 21. Shutter Channel - Alarms

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

- **Trigger**: sets the value (“1” –default– or “0”) that, when received from the KNX bus through object “**[Cx] Alarm**” (or “**[Cx] Alarm 2**” for alarm no. 2), will be interpreted by the actuator 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 5 to 600 tenths of a second; 1 to 3600 seconds; 1 to 1440 minutes; and 1 to 24 hours.

- **Action**: "<u>Stop</u>" (default), "<u>Up</u>", "<u>Down</u>" or "<u>Specific Position</u>". Selecting the latter brings one or two more parameters:

  - "**Shutter Position**" and, if applicable, "**Slats Position**": they define, in terms of percentage, the specific position the shutter and the slats will move to when the alarm is triggered.

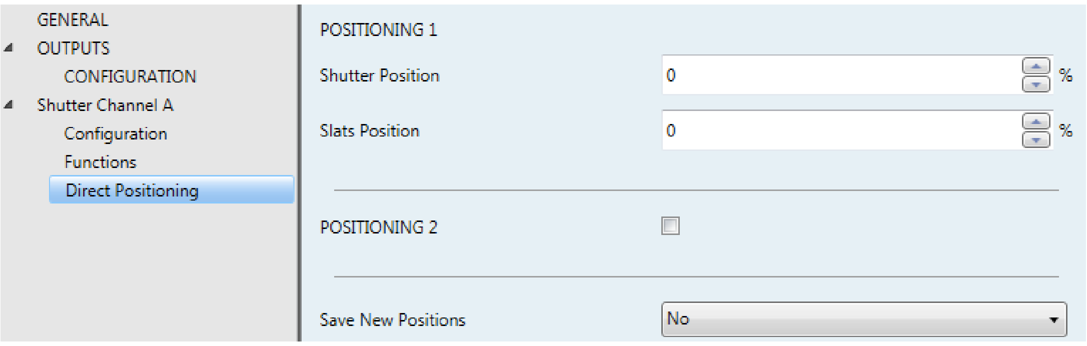
- **Deactivation – Mode**: "<u>Normal</u>" (default) or "<u>Frozen (Acknowledgement Needed)</u>". The second option enables a new 1-bit object, "**[Ox] Unfreeze Alarm**", which should be used for externally unfreezing the alarm once "**[Ox] Alarm**" has received the no-alarm value (i.e., the inverse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

  <u>Note</u>: *the acknowledgement should be sent necessarily after "**[Ox] Alarm**" has acquired the no-alarm value. Sending it while the trigger value is still active will have no effect*.

- **Deactivation – Action**: sets the state the output should acquire once the alarm has been deactivated (and acknowledged, if required). It may be: "<u>No Change</u>" (default), "<u>Off</u>", "<u>On</u>" or "<u>Last (Before Alarm)</u>".

- **Reverse Movement**: enables or disables the "**[Cx] Move (Reserved)**" communication object, which is equivalent to "**[Cx] Move**" except for the fact that one "0" will move the shutter down and one "1" will move it up.

- **Direct Positioning**: activates or deactivates the Direct Positioning function, which should be configured from a specific parameter screen.



POSITIONING 1

Shutter Position 0 %

Slats Position 0 %

POSITIONING 2 □

Save New Positions No

Figure 22. Shutter Channel - Direct Positioning

While the parameters of direct positioning no. 1 are shown by default, those of no. 2 are only shown after marking the corresponding checkbox. These parameters are:

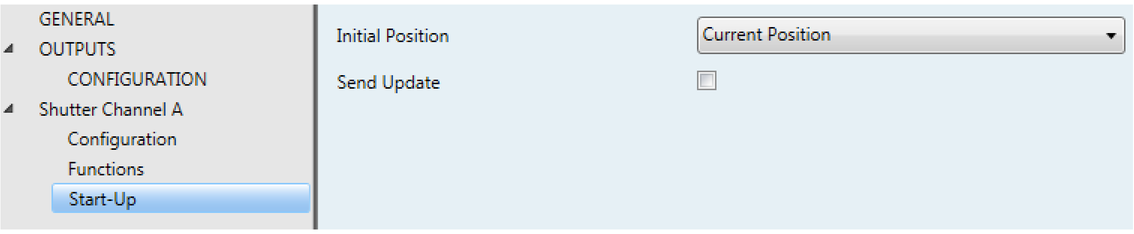
- **Shutter Position**: sets the position (in terms of percentage) the shutter will move to when one "1" is received through "**[Cx] Direct Positioning**" (or through "**[Cx] Direct Positioning 2**" for no. 2).

- **Slats Position** (only for shutter channels with slats): analogous to the above parameter, but for the slats.

With independence of the number of direct positioning functions enabled (one or two), the following parameter will also be shown:

- **Save New Positions**: if enabled, one new 1-bit object (or two, if Direct Positioning no. 2 is also enabled) will be added to the project topology: "**[Cx] Direct Positioning (Save)**" (and "**[Cx] Direct Positioning 2 (Save)**" for no. 2). When it receives one "1" from the KNX bus, the associated direct position will be overwritten with the current position of the shutter (and of the slats, if existing). Further triggers of the direct positioning functions will therefore take the shutter/slats to this new position.

- **Start-Up**: sets whether to perform the default action ("Default") or a custom action ("Custom") during the device start-up. The latter should be configured from a specific parameter screen:



Initial Position: Current Position

Send Update: □

Figure 23. Shutter Channel - Start-Up

This screen contains the following parameters:

- **Initial Position**: sets the position the shutter should acquire at the start-up of the actuator. It may be: "Current Position" (i.e., left as is; default option), "Up", "Down" or "Specific Position". The latter brings two more parameters:

- “**Shutter Position**” and, if applicable, “**Slats Position**”: they define, in terms of percentage, the specific position the shutter and the slats will move to when the actuator starts up.

- **Send Update**: sets whether the status objects of the shutter should be sent to the KNX bus (in order to update other KNX devices) after the start-up of the actuator. It is possible to impose a delay (0 to 600 tenths of a second; 0 to 3600 seconds; 0 to 1440 minutes; 0 to 24 hours) to ensure they are sent once the other devices are ready to receive them.

2.3.3 MANUAL CONTROL

MAXinBOX 66 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, 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 needs to be parameterised in ETS. Moreover, it is possible to enable a specific binary object for locking and unlocking the manual control in runtime.

Note:

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

Test Off Mode

Under the Test Off Mode, the 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 or as a shutter channel.

- **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, the device will act over 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 gets pressed being the shutter already at 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. In case of not being the shutter in motion, pressing the button does not cause any action, unless slats/lamellas have been parameterized – 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 corresponding.

- **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, with independence of the channel 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 commute the on-off 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.

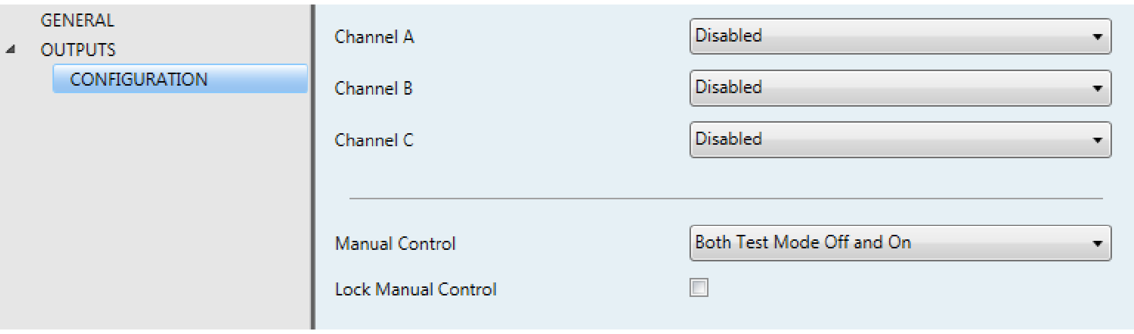
- **Disabled output**: under the Test On mode, short and long presses will cause the same effect for disabled outputs as for individual outputs (i.e., the relay will switch its state).

The lock, timer, alarm and scene functions will not work while the device is under the Test On mode. Status objects will not be sent to the bus, either.

Important: *the device is factory delivered with all the output channels configured as shutters (without slats), and with both manual modes (Test Off and Test On) enabled.*

ETS PARAMETERISATION

The **Manual Control** is configured from the Configuration tab itself under Outputs.



Channel A: Disabled

Channel B: Disabled

Channel C: Disabled

Manual Control: Both Test Mode Off and On

Lock Manual Control: □

Figure 24. Manual Control

The only two parameters are:

- **Manual Control**: options are "Disabled", "Only Test Mode Off", "Only Test Mode On" and "Both Test Mode Off and On" (default). 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.

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

- **Value**: 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**: sets how the manual control should remain after the device start-up (after an ETS download or a bus power failure): “Unlocked”, “Locked” or “Last Value” (default; on the very first start-up, this will be Unlocked).

2.4 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 66 can implement **up to 10 different and independent functions**, each of them entirely customisable and consisting in **up to 5 consecutive operations each**.

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

Please refer to the specific "**Logic Functions module for MAXinBOX 66**" user manual (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

2.5 THERMOSTATS

MAXinBOX 66 implements **four Zennio thermostats** which can be enabled and configured independently.

Please refer to the specific "**Zennio Thermostat for MAXinBOX 66**" user manual (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

2.6 MASTER LIGHT

The Master Light function brings the option to monitor the state of up to 12 light sources –or any other functionally-similar element 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.

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 25) 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***".

- **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

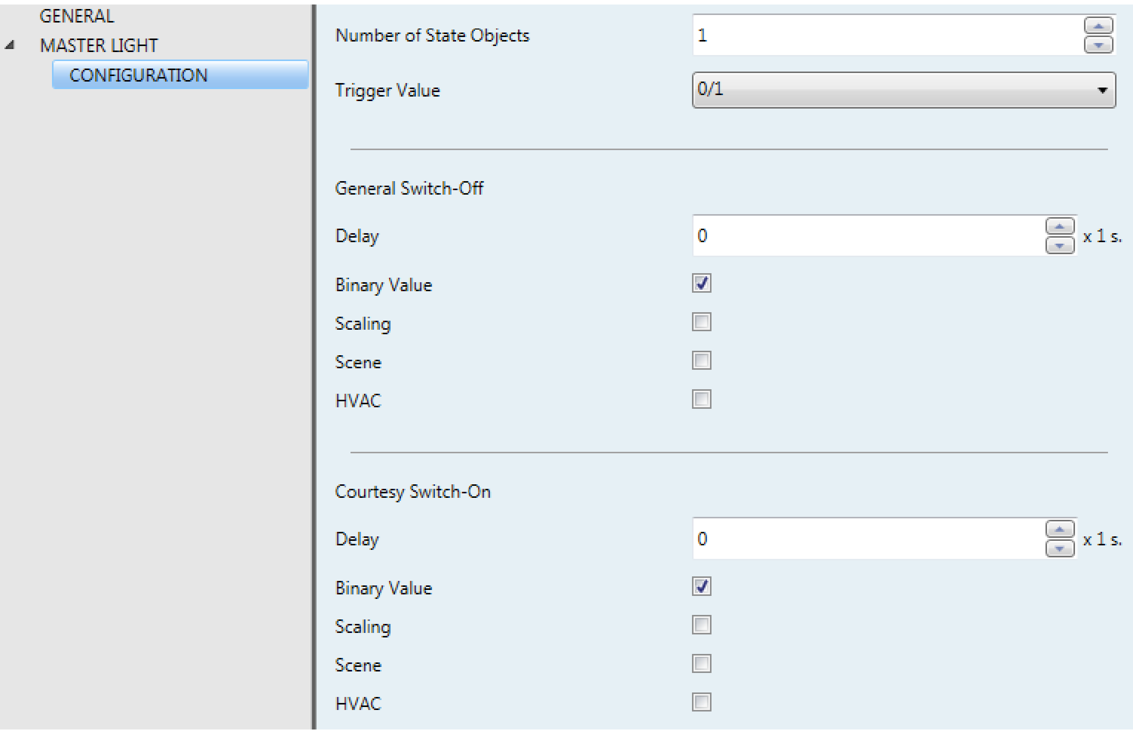
  Note: *the above options are not mutually exclusive; it is possible to send values of different nature together*.

- **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)*.



Number of State Objects: 1

Trigger Value: 0/1

General Switch-Off

Delay: 0 x 1 s.

Binary Value ☑

Scaling □

Scene □

HVAC □

Courtesy Switch-On

Delay: 0 x 1 s.

Binary Value ☑

Scaling □

Scene □

HVAC □

Figure 25. Master Light

ANNEX I. COMMUNICATION OBJECTS

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

- “**1st boot**” shows the cases where an object is assigned a certain value by the application program after a device download or a full reset. In case the value of such assignment can be parameterised, √ is shown in column “**P**”. Objects showing a hyphen (-) are not assigned a particular value and therefore can be assumed to be initialised with the value “0”, or with the corresponding updated value in case they depend on an external element (sensors, etc.). Moreover, if the object is sent (or is there an option to send it) to the bus (write or read requests) after a download or a device reset from ETS, the marks **(W)** or **(R)** will be shown, respectively for transmissions or read requests.

- “**Reboot**” shows the cases where an object is assigned a certain value by the application program after a bus power failure. In case the value of such assignment can be parameterised, √ is shown in column “**P**”. Objects showing a hyphen (-) are not assigned a particular value and therefore can be assumed to maintain their previous value after the failure, or with the corresponding updated value in case they depend on external elements. Moreover, if the object is sent (or is there an option to send it) to the bus (write or read requests) after a bus failure, the marks **(W)** or **(R)** will be shown, respectively for transmissions or read requests.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	1st Boot	P	Reboot	P	Name	Function
1	1 Bit		C T - - -	DPT_Trigger	0/1	0 W		0 W		Reset 0	Voltage Recovery -> Sending of 0
2	1 Bit		C T - - -	DPT_Trigger	0/1	1 W		1 W		Reset 1	Voltage Recovery -> Sending of 1
3	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		Manual Control Lock	0 = Lock; 1 = Unlock
	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		Manual Control Lock	0 = Unlock; 1 = Lock
4	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-		[Thermostat] Scene Input	Scene Value
5, 35, 65, 95	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	25,00		-		[Tx] Temperature Source 1	External Sensor Temperature
6, 36, 66, 96	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	25,00		-		[Tx] Temperature Source 2	External Sensor Temperature
7, 37, 67, 97	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	-		-		[Tx] Effective Temperature	Effective Control Temperature
8,38, 68, 98	1 Byte	E	C - - W -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	-	√	-		[Tx] Special Mode	1-byte HVAC Mode
9, 39, 69, 99	1 Bit	E	C - - W -	DPT_Trigger	0/1	-	√	-		[Tx] Special Mode: comfort	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	√	-		[Tx] Special Mode: comfort	0 = Off; 1 = On
10, 40, 70,	1 Bit	E	C - - W -	DPT_Trigger	0/1	-	√	-		[Tx] Special Mode: standby	0 = Nothing; 1 = Trigger

100	1 Bit	E	C - - W -	DPT_Switch	0/1	-	√	-		[Tx] Special Mode: standby	0 = Off; 1 = On
11, 41, 71, 101	1 Bit	E	C - - W -	DPT_Trigger	0/1	1	√	-		[Tx] Special Mode: economy	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	1	√	-		[Tx] Special Mode: economy	0 = Off; 1 = On
12, 42, 72, 102	1 Bit	E	C - - W -	DPT_Trigger	0/1	-	√	-		[Tx] Special Mode: protection	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	√	-		[Tx] Special Mode: protection	0 = Off; 1 = On
13, 43, 73, 103	1 Bit	E	C - - W -	DPT_Window_Door	0/1	-		-		[Tx] Window Status (input)	0 = Closed; 1 = Open
14, 44, 74, 104	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Tx] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
15, 45, 75, 105	1 Byte	S	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	3 W	√	- W		[Tx] Special Mode Status	1-byte HVAC Mode
16, 46, 76, 106	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	-		-		[Tx] Setpoint	Thermostat Setpoint Input
	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	-		-		[Tx] Basic Setpoint	Reference Setpoint
17, 47, 77, 107	1 Bit	E	C - - W -	DPT_Step	0/1	-		-		[Tx] Setpoint Step	0 = -0.5ºC; 1 = +0.5ºC
18, 48, 78, 108	2 Bytes	E	C - - W -	DPT_Value_Tempd	-670760.00 - 670760.00	-		-		[Tx] Setpoint Offset	Float Offset Value
19, 49, 79, 109	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	- W	√	- W		[Tx] Setpoint Status	Current Setpoint
20, 50, 80, 110	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	- W	√	- W		[Tx] Basic Setpoint Status	Current Basic Setpoint
21, 51, 81, 111	2 Bytes	S	C T R - -	DPT_Value_Tempd	-670760.00 - 670760.00	- W	√	- W		[Tx] Setpoint Offset Status	Current Setpoint Offset
22, 52, 82, 112	1 Bit	E	C - - W -	DPT_Reset	0/1	-		-		[Tx] Setpoint Reset	Reset Setpoint to Default
	1 Bit	E	C - - W -	DPT_Reset	0/1	-		-		[Tx] Offset Reset	Reset offset
23, 53, 83, 113	1 Bit	E	C - - W -	DPT_Heat_Cool	0/1	-		-		[Tx] Mode	0 = Cool; 1 = Heat
24, 54, 84, 114	1 Bit	S	C T R - -	DPT_Heat_Cool	0/1	- W	√	- W		[Tx] Mode Status	0 = Cool; 1 = Heat
25, 55, 85, 115	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[Tx] On/Off	0 = Off; 1 = On
26, 56, 86, 116	1 Bit	S	C T R - -	DPT_Switch	0/1	- W	√	- W	√	[Tx] On/Off Status	0 = Off; 1 = On
27, 57, 87, 117	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[Tx] Control Variable (Cool)	PI Control (Continuous)
28, 58, 88, 118	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[Tx] Control Variable (Heat)	PI Control (Continuous)
29, 59, 89,	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] Control Variable (Cool)	2-Point Control

119	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] Control Variable (Cool)	PI Control (PWM)
30, 60, 90, 120	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] Control Variable (Heat)	2-Point Control
	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] Control Variable (Heat)	PI Control (PWM)
31, 61, 91, 121	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] Additional Cool	Temp >= (Setpoint+Band) => "1"
32, 62, 92, 122	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] Additional Heat	Temp <= (Setpoint-Band) => "1"
33, 63, 93, 123	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
34, 64, 94, 124	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Tx] PI State (Heat)	0 = PI signal 0%; 1 = PI signal greater than 0%
125, 129, 133, 137, 141, 145	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	-		-		[Ix] Current Temperature	Temperature sensor value
126, 130, 134, 138, 142, 146	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[Ix] Overcooling	0 = No Alarm;1 = Alarm
127, 131, 135, 139, 143, 147	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[Ix] Overheating	0 = No Alarm;1 = Alarm
128, 132, 136, 140, 144, 148	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[Ix] Probe Error	0 = No Alarm;1 = Alarm
149	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-		[Motion Sensor] Scene Input	Scene Value
150	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[Motion Sensor] Scene Output	Scene Value
151, 175, 199, 223, 247, 271	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[Ix] Luminosity	0-100%
152, 176, 200, 224, 248, 272	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
153, 177, 201, 225, 249, 273	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
154, 178, 202, 226, 250, 274	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[Ix] Presence State (Scaling)	0-100%
155, 179, 203, 227, 251, 275	1 Byte	S	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	-		-		[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection

156, 180, 204, 228, 252, 278	1 Bit	S	C T R - -	DPT_Occupancy	0/1	-		-		[Ix] Presence State (Binary)	Binary Value
	1 Bit	S	C T R - -	DPT_Trigger	0/1	-		-		[Ix] Presence: Slave Output	1 = Motion Detected
157, 181, 205, 229, 253, 279	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
158, 182, 206, 230, 254, 280	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
159, 183, 207, 231, 255, 281	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by an external sensor
160, 165, 170, 184, 189, 194, 208, 213, 218, 232, 237, 242, 256, 261, 266, 280, 285, 290	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[Ix][Cy] Detection State (Scaling)	0-100%
161, 166, 171, 185, 190, 195, 209, 214, 219, 233, 238, 243, 257, 262, 267, 281, 286, 291	1 Byte	S	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	-		-		[Ix][Cy] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
162, 167, 172, 186, 191, 196, 210, 215, 220, 234, 239, 244, 258, 263, 268, 282, 287, 292	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-		[Ix][Cy] Detection State (Binary)	Binary Value
163, 168, 173, 187, 192, 197, 211, 216, 221, 235,	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[Ix][Cy] Channel Lock	According to parameters

240, 245, 259, 264, 269, 283, 288, 293											
164, 169, 174, 188, 193, 198, 212, 217, 222, 236, 241, 246, 260, 265, 270, 284, 289, 294	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[Ix][Cy] Force State	0 = No Detection; 1 = Detection
295	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-		[Shutter] Scenes	0 - 63 (Execute 1 - 64); 128 - 191 (Save 1 - 64)
296, 311, 326	1 Bit	E	C - - W -	DPT_UpDown	0/1	-		-		[Cα] Move	0=Raise; 1=Lower
297, 312, 327	1 Bit	E	C - - W -	DPT_Step	0/1	-		-		[Cα] Stop/Step	0=Stop/StepUp; 1=Stop/StepDown
	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Cα] Stop	0=Stop; 1=Stop
298, 313, 328	1 Bit	E	C - - W -	DPT_Enable	0/1	-		-		[Cα] Lock	0=Unlock; 1=Lock
299, 314, 329	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	- W	√	- W	√	[Cα] Shutter Position (Status)	0%=Top; 100%=Botton
300, 315, 330	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	- W	√	- W	√	[Cα] Slats Position (Status)	0%=Open; 100%=Closed
301, 316, 331	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-		-		[Cα] Shutter Positioning	0%=Top; 100%=Bottom
302, 317, 332	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-		-		[Cα] Slats Positioning	0%=Open; 100%=Closed
303, 318, 333	1 Bit	E	C - - W -	DPT_Alarm	0/1	-		-		[Cα] Alarm	0=No Alarm; 1=Alarm
	1 Bit	E	C - - W -	DPT_Alarm	0/1	1		-		[Cα] Alarm	0=Alarm; 1=No Alarm
304, 319, 334	1 Bit	E	C - - W -	DPT_Alarm	0/1	-		-		[Cα] Alarm 2	0=No Alarm; 1=Alarm
	1 Bit	E	C - - W -	DPT_Alarm	0/1	1		-		[Cα] Alarm 2	0=Alarm; 1=No Alarm
305, 320, 335	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Cα] Unfreeze Alarm	Alarm=0 + Unfreeze=1 => End Alarm
306, 321, 336	1 Bit	E	C - - W -	DPT_UpDown	0/1	-		-		[Cα] Move (Reversed)	0=Lower Shutter; 1=Raise Shutter
307, 322, 337	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Cα] Direct Positioning	0=No Action; 1=Go to Position
308, 323, 338	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Cα] Direct Positioning 2	0=No Action; 1=Go to Position

309, 324, 339	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Cα] Direct Positioning (Save)	0=No Action; 1=Save Current Position
310, 325, 340	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Cα] Direct Positioning 2 (Save)	0=No Action; 1=Save Current Position
341	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-		[Outputs] Scenes	0 – 63 (Execute 1 – 64); 128 – 191 (Save 1 – 64)
342, 350, 358, 366, 374, 382	1 Bit	E	C - - W -	DPT_BinaryValue	0/1	-		-		[Ox] On/Off	N.O. (0=Open Relay; 1=Close Relay)
	1 Bit	E	C - - W -	DPT_BinaryValue	0/1	-		-		[Ox] On/Off	N.C. (0=Close Relay; 1= Open Relay)
343, 351, 359, 367, 375, 383	1 Bit	S	C T R - -	DPT_BinaryValue	0/1	- W	√	- W	√	[Ox] On/Off (Status)	0=Output Off; 1=Output On
344, 352, 360, 368, 376, 384	1 Bit	E	C - - W -	DPT_Enable	0/1	-		-		[Ox] Lock	0=Unlock; 1=Lock
345, 353, 361, 369, 377, 385	1 Bit	E	C - - W -	DPT_Start	0/1	-		-		[Ox] Timer	0=Switch Off; 1=Switch On
346, 354, 362, 370, 378, 386	1 Bit	E	C - - W -	DPT_Start	0/1	-		-		[Ox] Flashing	0=Stop; 1=Start
347, 355, 363, 371, 379, 387	1 Bit	E	C - - W -	DPT_Alarm	0/1	-		-		[Ox] Alarm	0=Normal; 1=Alarm
	1 Bit	E	C - - W -	DPT_Alarm	0/1	-		-		[Ox] Alarm	0=Alarm; 1=Normal
348, 356, 364, 372, 380, 388	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[Ox] Unfreeze Alarm	Alarm=0 + Unfreeze=1 => End Alarm
349, 357, 365, 373, 381, 389	1 Bit	S	C T R - -	DPT_Trigger	0/1	-		-		[Ox] Warning Time (Status)	0=Normal; 1=Warning
390, 396, 402, 408, 414, 420	1 Bit	E	C - - W -	DPT_Switch	0/1					[Ix] Input Lock	1 = Locked; 0 = Unlocked
391, 397, 403, 409, 415, 421	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Short Press] 0	Sending of 0
	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Short Press] 1	Sending of 1
	1 Bit	E	C T - W -	DPT_Switch	0/1	- W		- W		[Ix] [Short Press] 0/1 Switching	Switching 0/1
	1 Bit		C T - - -	DPT_UpDown	0/1	- W		- W		[Ix] [Short Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		C T - - -	DPT_UpDown	0/1	- W		- W		[Ix] [Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C T - - -	DPT_UpDown	0/1	- W		- W		[Ix] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C T - - -	DPT_Step	0/1	- W		- W		[Ix] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		C T - - -	DPT_Step	0/1	- W		- W		[Ix] [Short Press] Stop/Step Down	Sending of 1 (Stop/Step down)

										Shutter	
	1 Bit		C T - - -	DPT_Step	0/1	- W		- W		[Ix] [Short Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) 0x2 (Dec. by 50%) 0x3 (Dec. by 25%) 0x4 (Dec. by 12%) 0x5 (Dec. by 6%) 0x6 (Dec. by 3%) 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) 0xA (Inc. by 50%) 0xB (Inc. by 25%) 0xC (Inc. by 12%) 0xD (Inc. by 6%) 0xE (Inc. by 3%) 0xF (Inc. by 1%)	- W		- W		[Ix] [Short Press] Brighter	Increase Brightness
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) (…) 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) (…) 0xF (Inc. by 1%)	- W		- W		[Ix] [Short Press] Darker	Decrease Brightness
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) (…) 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) (…) 0xF (Inc. by 1%)	- W		- W		[Ix] [Short Press] Brighter/Darker	Switch Bright/Dark
	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Short Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Short Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	E	C T - W -	DPT_Switch	0/1	- W		- W		[Ix] [Short Press] Dimmer ON/OFF	Switching 0/1
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	- W		- W		[Ix] [Short Press] Run Scene	Sending of 0 - 63
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	- W		- W		[Ix] [Short Press] Save Scene	Sending of 128 - 191
	1 Bit	E/S	C T R W -	DPT_Switch	0/1	- W		- W		[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		C T - - -	DPT_Value_1_Ucount	0 - 255	- W		- W		[Ix] [Short Press] Constant Value (Integer)	0 - 255

	1 Byte		C T - - -	DPT_Scaling	0% - 100%	- W		- W		[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		C T - - -	DPT_Value_2_Ucount	0 - 65535	- W		- W		[Ix] [Short Press] Constant Value (Integer)	0 - 65535
	2 Bytes		C T - - -	9.xxx	-671088.64 - 670760.96	- W		- W		[Ix] [Short Press] Constant Value (float)	Float value
392, 398, 404, 410, 416, 422	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	- W		- W		[Ix] [Short Press] Shutter Status (input)	0% = Top; 100% = Bottom
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	- W		- W		[Ix] [Short Press] Dimming Status (input)	0% - 100%
393, 399, 405, 411, 417, 423	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Long Press] 0	Sending of 0
	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Long Press] 1	Sending of 1
	1 Bit	E	C T - W -	DPT_Switch	0/1	- W		- W		[Ix] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		C T - - -	DPT_UpDown	0/1	- W		- W		[Ix] [Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		C T - - -	DPT_UpDown	0/1	- W		- W		[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C T - - -	DPT_UpDown	0/1	- W		- W		[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C T - - -	DPT_Step	0/1	- W		- W		[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		C T - - -	DPT_Step	0/1	- W		- W		[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		C T - - -	DPT_Step	0/1	- W		- W		[Ix] [Long Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) 0x2 (Dec. by 50%) 0x3 (Dec. by 25%) 0x4 (Dec. by 12%) 0x5 (Dec. by 6%) 0x6 (Dec. by 3%) 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) 0xA (Inc. by 50%) 0xB (Inc. by 25%) 0xC (Inc. by 12%) 0xD (Inc. by 6%) 0xE (Inc. by 3%) 0xF (Inc. by 1%)	- W		- W		[Ix] [Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) (…) 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) (…)	- W		- W		[Ix] [Long Press] Darker	Long Pr. -> Darker; Release -> Stop

					0xF (Inc. by 1%)						
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) (…) 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) (…) 0xF (Inc. by 1%)	- W		- W		[Ix] [Long Press] Brighter/Darker	Long Pr. -> Brighter/Darker; Release -> Stop
	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Long Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		C T - - -	DPT_Switch	0/1	- W		- W		[Ix] [Long Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	E	C T - W -	DPT_Switch	0/1	- W		- W		[Ix] [Long Press] Dimmer ON/OFF	Switching 0/1
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	- W		- W		[Ix] [Long Press] Run Scene	Sending of 0 - 63
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	- W		- W		[Ix] [Long Press] Save Scene	Sending of 128 - 191
	1 Bit	S	C T R - -	DPT_Alarm	0/1	- W		- W		[Ix] [Switch/Sensor] Alarm: Breakdown or sabotage	1 = Alarm; 0 = No Alarm
	2 Bytes		C T - - -	9.xxx	-671088.64 - 670760.96	- W		- W		[Ix] [Long Press] Constant Value (float)	Float value
	2 Bytes		C T - - -	DPT_Value_2_Ucount	0 - 65535	- W		- W		[Ix] [Long Press] Constant Value (Integer)	0 - 65535
	1 Byte		C T - - -	DPT_Scaling	0% - 100%	- W		- W		[Ix] [Long Press] Constant Value (Percentage)	0% - 100%
	1 Byte		C T - - -	DPT_Value_1_Ucount	0 - 255	- W		- W		[Ix] [Long Press] Constant Value (Integer)	0 - 255
394, 400, 406, 412, 418, 424	1 Bit		C T - - -	DPT_Trigger	0/1	- W		- W		[Ix] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
395, 401, 407, 413, 419, 425	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	50% W		- W		[Ix] [Long Press] Dimming Status (input)	0% - 100%
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	50% W		- W		[Ix] [Long Press] Shutter Status (input)	0% = Top; 100% = Bottom
426	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[ML] Trigger	Trigger the Master Light Function
427	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 1	Binary Status
428	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 2	Binary Status
429	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 3	Binary Status
430	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 4	Binary Status
431	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 5	Binary Status
432	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 6	Binary Status
433	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 7	Binary Status
434	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 8	Binary Status
435	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 9	Binary Status

436	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 10	Binary Status
437	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 11	Binary Status
438	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[ML] Status Object 12	Binary Status
439	1 Bit		C T - - -	DPT_Switch	0/1	-		-		[ML] General Switch-Off: Binary Object	Off Sending
440	1 Byte		C T - - -	DPT_Scaling	0% - 100%	-		-		[ML] General Switch-Off: Scaling	0-100%
441	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[ML] General Switch-Off: Scene	Scene Sending
442	1 Byte		C T - - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	-		-		[ML] General Switch-Off: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection
443	1 Bit		C T - - -	DPT_Switch	0/1	-		-		[ML] Courtesy Switch-On: Binary Object	On Sending
444	1 Byte		C T - - -	DPT_Scaling	0% - 100%	-		-		[ML] Courtesy Switch-On: Scaling	0-100%
445	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[ML] Courtesy Switch-On: Scene	Scene Sending
446	1 Byte		C T - - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	-		-		[ML] Courtesy Switch-On: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection
447-478	1 Bit	E	C - - W -	DPT_Bool	0/1	-		-		[LF] (1 bit) Data Entry x	Binary Data Entry (0/1)
479-494	1 Byte	E	C - - W -	DPT_Value_1_Ucount	0 - 255	-		-		[LF] (1 byte) Data Entry x	1 byte Data Entry (0-255)
495-510	2 Bytes	E	C - - W -	DPT_Value_2_Ucount	0 - 65535	-		-		[LF] (2 bytes) Data Entry x	2 bytes Data Entry
				DPT_Value_2_Count	-32768 - 32767						
				DPT_Value_Temp	-273.00 - 670760.00						
511-518	4 Bytes	E	C - - W -	DPT_Value_4_Count	-2147483648 - 2147483647	-		-		[LF] (4 bytes) Data Entry x	4 bytes Data Entry
519-528	1 Bit	S	C T R - -	DPT_Bool	0/1	-		-		[LF] Function x - Result	(1 bit) Boolean
	1 Byte	S	C T R - -	DPT_Value_1_Ucount	0 - 255	-		-		[LF] Function x - Result	(1 byte) Unsigned
	2 Bytes	S	C T R - -	DPT_Value_2_Ucount	0 - 65535	-		-		[LF] Function x - Result	(2 bytes) Unsigned
	4 Bytes	S	C T R - -	DPT_Value_4_Count	-2147483648 - 2147483647	-		-		[LF] Function x - Result	(4 bytes) Signed
	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[LF] Function x - Result	(1 byte) Percentage
	2 Bytes	S	C T R - -	DPT_Value_2_Count	-32768 - 32767	-		-		[LF] Function x - Result	(2 bytes) Signed
	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	-		-		[LF] Function x - Result	(2 bytes) Float

Join and send us your inquiries
about Zennio devices:

http://zennioenglish.zendesk.com

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

Tel. +34 925 232 002.
Fax. +34 925 337 310.
www.zennio.com
info@zennio.com