MINiBOX 20 v2

2-Output Multifunction Actuator with KNX Secure

ZIOMN20V2

Application programme version: [1.7]
User manual edition: [1.7]_a

CONTENTS

1. INTRODUCTION

1.1 MINiBOX 20 V2

MINiBOX 20 v2 from Zennio is a **KNX Secure**, versatile and small-size KNX **actuator** featuring a wide variety of functions and four relay outputs for a variety of applications, such as controlling shutter drives.

The most outstanding features are:

- **KNX bus** communication. In order to ensure secure communications between both devices, **group addresses configured with KNX security** must be used.

  For detailed information about the functionality and the configuration of the KNX Secure, please refer to the specific manual "**Security KNX**" available in the MINiBOX 20 v2 product section at the Zennio website (www.zennio.com).

- **2 relay outputs**, configurable as:

  - ➢ 1 shutter channels (with or without slats), or

  - ➢ Up to 2 individual ON/OFF outputs.

- **2 master light control instances** 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.

- **10 customisable, multi-operation logic functions**.

- **Scene-triggered action control**, with an optional delay in the execution.

- **Manual operation / supervision** of the 2 relay outputs through the on-board pushbuttons and LEDs.

- **Heartbeat** or periodic "still-alive" notification.

- **Relay Switches Counter.**

1.2 INSTALLATION

MINiBOX 20 v2 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. Output Status LED Indicator.

2. Manual Control Pushbutton.

3. KNX Connector.

4. Prog./Test LED.

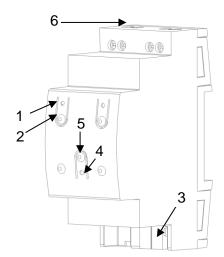**5.** Prog./Test Pushbutton.

6. Relay Outputs.

Figure 1. MINiBOX 20 v2. Elements.

The main elements of the device are described next.

- **Prog./Test Pushbutton (5)**: 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 (6)**: output ports for the insertion of the stripped cables of the systems being controlled by the actuator (see section 2.2). 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 Prog./Test LED will blink in blue colour for a few seconds before the device is ready. External orders will not be executed during this time, but will be executed 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 refer to the next sections of this document for further details.

On the other hand, when a bus power failure takes place, the device will interrupt any pending actions, and will save its state so it can be recovered once the power supply is restored. For safety reasons, all **shutter channels** will be stopped (i.e., the relays will open) if a power loss takes place, while the individual outputs will switch to the specific state configured in ETS (if any).

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 entering the Parameters tab of the device.

ETS PARAMETERISATION

From the General parameter screen it is possible to activate/deactivate all the required functionality.

Figure 2. General

- Once activated, **Outputs, Logic Functions, Master Light, Scene Timing** and **Manual control** additional tabs to the menu on the left will be included. These functions and their parameters will be explained in later sections of this document.

  The **Manual Control** function is enabled by default, and so is the corresponding configuration tab.

- **Heartbeat (Periodic Alive Notification)**: incorporates a one-bit object to the project ("**[Heartbeat] Object to Send '1'**") that will be sent periodically with a value of "1" to notify that the device is still working (still *alive*).
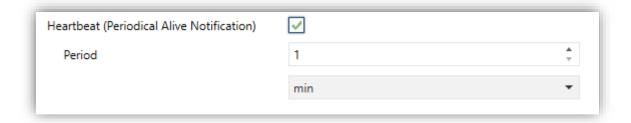
Figure 3. Heartbeat (Periodic Alive Notification).

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

- **Device Recovery Objects (Send 0 and 1):** [*disabled/enabled*]: this parameter lets the integrator activate two new communication objects ("**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** [*0…255*] to this sending.

Figure 4. Device Recovery Objects

Note: *after download or bus failure, the sending takes place with a delay of up to 6,35 seconds plus the parameterised delay, to prevent bus overload.*

- **Show Relay Switches Counter Objects** [*disabled/enabled*]: enables two communication objects to keep track of the number of switches performed by each of the relays ("**[Relay X] Number of Switches**") and the maximum number of switches carried out in a minute ("**[Relay X] Maximum Switches per Minute**").

2.2 OUTPUTS

MINiBOX 20 v2 incorporates **2 relay outputs**, configurable as:

- **Individual binary outputs**, which allow an independent control of loads.

- **One shutter channel**, which allows controlling the motion of shutters or blinds.

For detailed information about the functionality and the configuration of the related parameters, please refer to the following specific manuals, all of them available in the MINiBOX 20 v2 product section at the Zennio homepage (www.zennio.com):

- **Individual outputs.**

- **Shutters channels.**

2.3 LOGIC FUNCTIONS

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

MINiBOX 20 v2 can implement **up to 10 different and independent functions**, each of them entirely customisable and consisting in **up to 4 consecutive operations each**.

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

Please refer to the "**Logic Functions**" user manual, available within the MINiBOX 20 v2 product section at the Zennio homepage, www.zennio.com, for detailed information about the functionality and the configuration of the related parameters.

2.4 MASTER LIGHT

MINiBOX 20 v2 implements **two Master Light** modules which can be enabled and configured independently.

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

Such master order will consist in:

- A **general switch-off** order, if at least one of the up to twelve status objects is found to be on.

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

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

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

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

ETS PARAMETERISATION

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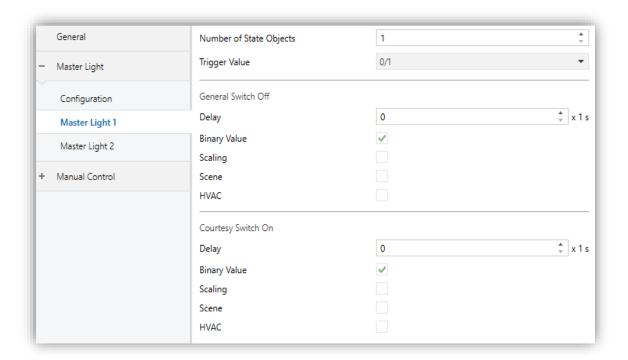

Figure 5. Master Light.

- **Number of State Objects** [*1…12*]: defines the number of 1-bit status objects required. These objects are called "**[ML] Status Object *n*.**"

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

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

- **General Switch-Off**.

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

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

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

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

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

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

2.5 SCENE TIMING

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

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

ETS PARAMETERISATION

Prior to setting the **scene timing**, it is necessary to have one or more scenes configured in some of the outputs. When entering the Configuration window under Scene Timing, all configured scenes will be listed, together with a few checkboxes to select which of them need to be temporised, as shown in Figure 6.

Figure 6. Scene timing

Enabling a certain scene number n brings a new tab with such name to the menu on the left, from which it is possible to configure the temporisation of that scene <u>for each of the outputs where it has been configured</u>.
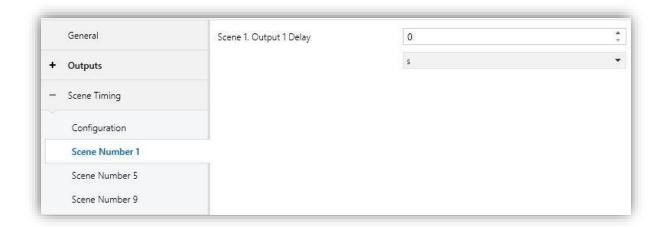
Figure 7. Configuration of the scene timing

Therefore, parameter "**Scene m. Z Delay**" defines the delay that will be applied to the action defined in Z for the execution of scene m (where Z may be a specific individual output or shutter channel). The range of this delay is 0 to 3600 seconds, 0 to 1440 minutes or 0 to 24 hours.

2.6 MANUAL CONTROL

MINiBOX 20 v2 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 can be accessed 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 **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 is 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.

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

- **Disabled output**: short and long presses will switch the state of the corresponding relay. In case this consists in closing the relay, then the remaining relays will open, for safety reasons.

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 delivered from factory with all the output channels configured as disabled outputs, and with both manual modes (Test Off and Test On) enabled.*

ETS PARAMETERISATION

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

Figure 8. Manual control screen

The only two parameters are:

- **Manual Control**: options are "Disabled", "Only Test Off Mode", "Only Test On Mode" and "Test Off Mode + Test On Mode" (default). Depending on the selection, the device will permit using the manual control under the Test Off, the Test On, or 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, the object "**Manual Control Lock**" turns visible, as well as two more parameters:

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

  - ➢ **Initialisation**: sets how the lock state of the manual control should remain after the device start-up (after an ETS download or a bus power failure): "Unlocked", "Locked" or "*Last Value*".

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 programme itself.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	Name	Function
1	1 Bit		C - - T -	DPT_Trigger	0/1	[Heartbeat] Object to Send '1'	Sending of '1' Periodically
2	1 Bit		C - - T -	DPT_Trigger	0/1	[Heartbeat] Device Recovery	Send 0
3	1 Bit		C - - T -	DPT_Trigger	0/1	[Heartbeat] Device Recovery	Send 1
4	1 Bit	I	C - W - -	DPT_Enable	0/1	Lock Manual Control	0 = Lock; 1 = Unlock
	1 Bit	I	C - W - -	DPT_Enable	0/1	Lock Manual Control	0 = Unlock; 1 = Lock
5, 16	1 Byte	I	C - W - -	DPT_SceneControl	0-63; 128-191	[Ox] Scenes	0 – 63 (Execute 1 – 64); 128 – 191 (Save 1 – 64)
6, 17	1 Bit	I	C - W - -	DPT_BinaryValue	0/1	[Ox] On/Off	N.O. (0=Open Relay; 1=Close Relay)
	1 Bit	I	C - W - -	DPT_BinaryValue	0/1	[Ox] On/Off	N.C. (0=Close Relay; 1= Open Relay)
7, 18	1 Bit	O	C R - T -	DPT_BinaryValue	0/1	[Ox] On/Off (Status)	0=Output Off; 1=Output On
8, 19	1 Bit	I	C - W - -	DPT_Enable	0/1	[Ox] Lock	0=Unlock; 1=Lock
9, 20	1 Bit	I	C - W - -	DPT_Start	0/1	[Ox] Timer	0=Switch Off; 1=Switch On
10, 21	1 Bit	I	C - W - -	DPT_Start	0/1	[Ox] Flashing	0=Stop; 1=Start
11, 22	1 Bit	I	C - W - -	DPT_Alarm	0/1	[Ox] Alarm	0=Normal; 1=Alarm
	1 Bit	I	C - W - -	DPT_Alarm	0/1	[Ox] Alarm	0=Alarm; 1=Normal
12, 23	1 Bit	I	C - W - -	DPT_Ack	0/1	[Ox] Unfreeze Alarm	Alarm=0 + Unfreeze=1 => End Alarm
13, 24	1 Bit	O	C R - T -	DPT_State	0/1	[Ox] Warning Time (Status)	0=Normal; 1=Warning
14, 25	4 Bytes	I/O	C R W T -	DPT_LongDeltaTimeSec	-2147483648 - 2147483647	[Ox] Operating Time (s)	Time in Seconds
15, 26	2 Bytes	I/O	C R W T -	DPT_TimePeriodHrs	0 - 65535	[Ox] Operating Time (h)	Time in Hours
269	1 Byte	I	C - W - -	DPT_SceneControl	0-63; 128-191	[Shutter] Scenes	0 - 63 (Execute 1 - 64); 128 - 191 (Save 1 - 64)
270	1 Bit	I	C - W - -	DPT_UpDown	0/1	[Cx] Move	0 = Raise; 1 = Lower
271	1 Bit	I	C - W - -	DPT_Step	0/1	[Cx] Stop/Step	0 = Stop/StepUp; 1 = Stop/StepDown
	1 Bit	I	C - W - -	DPT_Trigger	0/1	[Cx] Stop	0 = Stop; 1 = Stop

272	1 Bit	I	C - W - -	DPT_Trigger	0/1	[Cx] Switched Control	0, 1 = Up, Down or Stop, Depending on the Last Move
273	1 Bit	I	C - W - -	DPT_Enable	0/1	[Cx] Lock	0 = Unlock; 1 = Lock
274	1 Byte	I	C - W - -	DPT_Scaling	0% - 100%	[Cx] Shutter Positioning	0% = Top; 100% = Bottom
275	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Cx] Shutter Position (Status)	0% = Top; 100% = Bottom
276	1 Byte	I	C - W - -	DPT_Scaling	0% - 100%	[Cx] Slats Positioning	0% = Open; 100% = Closed
277	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[Cx] Slats Position (Status)	0% = Open; 100% = Closed
278	1 Bit	O	C R - T -	DPT_Switch	0/1	[Cx] Rising Relay (Status)	0 = Open; 1 = Closed
279	1 Bit	O	C R - T -	DPT_Switch	0/1	[Cx] Lowering Relay (Status)	0 = Open; 1 = Closed
280	1 Bit	O	C R - T -	DPT_Switch	0/1	[Cx] Movement (Status)	0 = Stopped; 1 = Moving
281	1 Bit	O	C R - T -	DPT_UpDown	0/1	[Cx] Movement Direction (Status)	0 = Upward; 1 = Downward
282	1 Bit	I	C - W - -	DPT_Switch	0/1	[Cx] Auto: On/Off	0 = On; 1 = Off
	1 Bit	I	C - W - -	DPT_Switch	0/1	[Cx] Auto: On/Off	0 = Off; 1 = On
283	1 Bit	O	C R - T -	DPT_Switch	0/1	[Cx] Auto: On/Off (Status)	0 = On; 1 = Off
	1 Bit	O	C R - T -	DPT_Switch	0/1	[Cx] Auto: On/Off (Status)	0 = Off; 1 = On
284	1 Bit	I	C - W - -	DPT_UpDown	0/1	[Cx] Auto: Move	0 = Raise; 1 = Lower
285	1 Bit	I	C - W - -	DPT_Step	0/1	[Cx] Auto: Stop/Step	0 = Stop/StepUp; 1 = Stop/StepDown
	1 Bit	I	C - W - -	DPT_Trigger	0/1	[Cx] Auto: Stop	0 = Stop; 1 = Stop
286	1 Byte	I	C - W - -	DPT_Scaling	0% - 100%	[Cx] Auto: Shutter Positioning	0% = Top; 100% = Bottom
287	1 Byte	I	C - W - -	DPT_Scaling	0% - 100%	[Cx] Auto: Slats Positioning	0% = Open; 100% = Closed
288	1 Bit	I	C - W T U	DPT_Scene_AB	0/1	[Cx] Sunshine/Shadow	0 = Sunshine; 1 = Shadow
	1 Bit	I	C - W T U	DPT_Scene_AB	0/1	[Cx] Sunshine/Shadow	0 = Shadow; 1 = Sunshine
289	1 Bit	I	C - W T U	DPT_Heat_Cool	0/1	[Cx] Cooling/Heating	0 = Heating; 1 = Cooling
	1 Bit	I	C - W T U	DPT_Heat_Cool	0/1	[Cx] Cooling/Heating	0 = Cooling; 1 = Heating
290	1 Bit	I	C - W T U	DPT_Occupancy	0/1	[Cx] Presence/No Presence	0 = Presence; 1 = No Presence
	1 Bit	I	C - W T U	DPT_Occupancy	0/1	[Cx] Presence/No Presence	0 = No Presence; 1 = Presence
291, 292	1 Bit	I	C - W - -	DPT_Alarm	0/1	[Cx] Alarm x	0 = No Alarm; 1 = Alarm
	1 Bit	I	C - W - -	DPT_Alarm	0/1	[Cx] Alarm x	0 = Alarm; 1 = No Alarm
293	1 Bit	I	C - W - -	DPT_Ack	0/1	[Cx] Unfreeze Alarm	Alarm1 = Alarm2 = No Alarm + Unfreeze (1) => End Alarm
294	1 Bit	I	C - W - -	DPT_Scene_AB	0/1	[Cx] Move (Reversed)	0 = Lower; 1 = Raise
295	1 Bit	I	C - W - -	DPT_Ack	0/1	[Cx] Direct Positioning 1	0 = No Action; 1 = Go to Position
296	1 Bit	I	C - W - -	DPT_Ack	0/1	[Cx] Direct Positioning 2	0 = No Action; 1 = Go to Position
297	1 Bit	I	C - W - -	DPT_Ack	0/1	[Cx] Direct Positioning 1 (Save)	0 = No Action; 1 = Save Current Position
298	1 Bit	I	C - W - -	DPT_Ack	0/1	[Cx] Direct Positioning 2 (Save)	0 = No Action; 1 = Save Current Position

817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880	1 Bit	I	C - W - -	DPT_Bool	0/1	[LF] (1-Bit) Data Entry x	Binary Data Entry (0/1)
881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912	1 Byte	I	C - W - -	DPT_Value_1_Ucount	0 - 255	[LF] (1-Byte) Data Entry x	1-Byte Data Entry (0-255)
913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944	2 Bytes	I	C - W - -	DPT_Value_2_Ucount	0 - 65535	[LF] (2-Byte) Data Entry x	2-Byte Data Entry
945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960	4 Bytes	I	C - W - -	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] (4-Byte) Data Entry x	4-Byte Data Entry
961, 962, 963, 964, 965, 966, 967, 968, 969, 970	1 Bit	O	C R - T -	DPT_Bool	0/1	[LF] Function x - Result	(1-Bit) Boolean
	1 Byte	O	C R - T -	DPT_Value_1_Ucount	0 - 255	[LF] Function x - Result	(1-Byte) Unsigned
	2 Bytes	O	C R - T -	DPT_Value_2_Ucount	0 - 65535	[LF] Function x - Result	(2-Byte) Unsigned
	4 Bytes	O	C R - T -	DPT_Value_4_Count	-2147483648 - 2147483647	[LF] Function x - Result	(4-Byte) Signed
	1 Byte	O	C R - T -	DPT_Scaling	0% - 100%	[LF] Function x - Result	(1-Byte) Percentage
	2 Bytes	O	C R - T -	DPT_Value_2_Count	-32768 - 32767	[LF] Function x - Result	(2-Byte) Signed
	2 Bytes	O	C R - T -	9.xxx	-671088.64 - 670433.28	[LF] Function x - Result	(2-Byte) Float
991, 993	4 Bytes	O	C R - T -	DPT_Value_4_Ucount	0 - 4294967295	[Relay x] Number of Switches	Number of Switches
992, 994	2 Bytes	O	C R - T -	DPT_Value_2_Ucount	0 - 65535	[Relay x] Maximum Switches per Minute	Maximum Switches per Minute
1039, 1061	1 Bit	I	C - W - -	DPT_Trigger	0/1	[MLx] Trigger	Trigger the Master Light Function
	1 Bit	I	C - W - -	DPT_Ack	0/1	[MLx] Trigger	1 = Nothing; 0 = Trigger the Master Light Function
	1 Bit	I	C - W - -	DPT_Ack	0/1	[MLx] Trigger	0 = Nothing; 1 = Trigger the Master Light Function

1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073	1 Bit	I	C - W - -	DPT_Switch	0/1	[MLx] Status Object x	Binary Status
1052, 1074	1 Bit	O	C R - T -	DPT_Switch	0/1	[MLx] General Status	Binary Status
1053, 1075	1 Bit		C - - T -	DPT_Switch	0/1	[MLx] General Switch Off: Binary Object	Switch Off Sending
1054, 1076	1 Byte		C - - T -	DPT_Scaling	0% - 100%	[MLx] General Switch Off: Scaling	0-100%
1055, 1077	1 Byte		C - - T -	DPT_SceneControl	0-63; 128-191	[MLx] General Switch Off: Scene	Scene Sending
1056, 1078	1 Byte		C - - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[MLx] General Switch Off: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection
1057, 1079	1 Bit		C - - T -	DPT_Switch	0/1	[MLx] Courtesy Switch On: Binary Object	Switch On Sending
1058, 1080	1 Byte		C - - T -	DPT_Scaling	0% - 100%	[MLx] Courtesy Switch On: Scaling	0-100%
1059, 1081	1 Byte		C - - T -	DPT_SceneNumber	0 - 63	[MLx] Courtesy Switch On: Scene	Scene Sending
1060, 1082	1 Byte		C - - T -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[MLx] Courtesy Switch On: HVAC mode	Auto, Comfort, Standby, Economy, Building Protection