Passion for KNX innovation

ACTinBOX QUATRO

4 Output Actuator

ZN1IO-AB40

Application program version: [2.3]
User manual edition: [2.3]_a

www.zennio.com

USER MANUAL

Contents

DOCUMENT UPDATES

Version	Changes	Page(s)
[2.3]_a	Changes in the application program: • Improved management of the output order buffer.	-
	Example added about the “Multiply” parameter.	14
	General revision of texts and styles	-
[2.2]_a	Changes in the application program: • Improved compatibility with certain batch numbers of the device.	-
[2.1]_a	Changes in the application program: • Improvement in the behavior of the shutter channels. ➢ On the activation of the additional time when the shutter status is already 100%. ➢ On joint activations of the additional time of various shutter channels whose status is already 100%. • Optimization of the number of statuses referred to the individual outputs that are sent at the start-up.	-

1 INTRODUCTION

1.1 ACTINBOX QUATRO

ACTinBOX QUATRO is a KNX actuator that combines in only one device the following features:

- **4 multifunction binary outputs**, up to 10A each, configurable as:
  - Up to 2 shutter channels (with or without slats/lamellas).
  - Up to 4 individual outputs.
- 10x multi-operation **logical function** module, with an enable/disable communication object per function.
- **Possibility of manually operating** the actuator outputs, for testing purposes, by means of the Zennio infrared remote control.

The outputs and the logical function module work independently and can interact with each other as if they were two autonomous devices connected to the KNX bus.

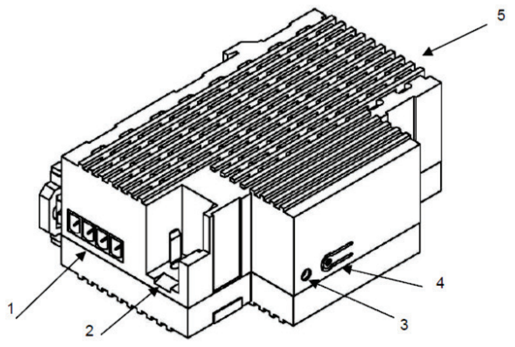
Figure 1 ACTinBOX QUATRO Actuator

1.2 INSTALLATION

ACTinBOX QUATRO connects to the KNX bus through the on-board KNX connector.

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

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



1 – Outputs (channel B)

2 – KNX connector

3 – Programming LED

4 – Programming button

5 – Outputs (channel A)

Figure 2 ACTinBOX QUATRO. Element Scheme

The functionality of the main elements of the actuator is described below:

- **Programming button (4)**: a short press on this button sets the actuator into the programming mode, making the associated LED (3) light in red.

  Note: *if this button is held while plugging the device into the KNX bus, ACTinBOX QUATRO goes into secure mode. The LED will blink in red every 0.5 seconds*.

  While ACTinBOX QUATRO is initializing, after a download or a bus power failure, the LED blinks in blue. During the time the actuator takes to initialize, every order received will be taken into account and will be executed once the initialization is completed.

- **Outputs (1 and 5)**: the screw terminal block (bundled with the original packaging) must be plugged here to allow the connection of the different systems to be controlled by the actuator. Wiring the terminal block may be carried out before inserting it into the corresponding slot.

To get detailed information about the technical features of ACTinBOX QUATRO, as well as on security and on the installation procedure, please refer to the actuator **Datasheet**, bundled with the original package of the device and also available at: http://www.zennio.com .

2 CONFIGURATION

2.1 INDIVIDUAL OUTPUTS

ACTinBOX QUATRO incorporates **4 relay outputs** that allow controlling different loads autonomously. Each output can be enabled or disabled independently and perform different functionalities.

Every individual output can be configured as **normally open** (the activation of the output makes the relay close) or **normally closed** (the deactivation of the output makes the relay open).

Besides the output type, ACTinBOX QUATRO allows the configuration of the following functionalities for the individual outputs:

- **Timers.** Permits a timed control over the outputs, being possible to set times for the switch-on and for the switch-off of the output.
- **Scenes.** Allows running and/or saving a specific action over the output/s where this function is enabled. The status of the outputs will vary depending on the action set for the parameterized scene.
- **Alarm.** Allows changing the status of the output where this function is enabled, being possible to configure the status the output will be set to, both on the alarm activation and on the alarm deactivation.

  Note: *the alarm behaves with priority over the other functionalities*.
- **Start-up configuration**: default or custom.

All these configuration options are explained in detail in section 3, ETS Parametrization.

2.2 SHUTTER CHANNELS

The output channels of ACTinBOX QUATRO allow controlling **up to 2** different shutter drives (or similar window/door automated systems).

Thus, it is possible to control the movement of the shutter within the domotic installation.

- **Basic control**: moving the shutter up/down
- **Precise control**: precise control of the shutter and the slats (for blinds featuring slats).

Each channel (A and B) consists of two consecutive individual outputs; i.e., Channel A is made of the individual outputs 1 and 2, while channel B consists of outputs 3 and 4. The first output of every channel sends orders to **raise** the shutter, whereas the second output sends orders to **lower** the shutter. Therefore, the cables of the shutter motor carrying out these actions should be properly connected to the corresponding output of the channel to perform the required action.

Table 1 shows the action carried out by the outputs of each channel:

Channel	Outputs	Action
A	1	Move up
	2	Move down
B	3	Move up
	4	Move down

Table 1 Shutter channels: actions of the outputs

Each channel can be configured as a **Shutter (No Slats)** or as a **Blind (with Slats)**.

Besides the shutter type, ACTinBOX QUATRO allows the configuration of the following functionalities for the shutter channels:

- **Times.** Sets the main times that define the movement of the shutter: the length of the rising movement, the length of the descending movement, a security time for making a pause in the movement of the motor when the direction changes, and an additional movement time when the shutter gets to its limit (top or bottom). For blinds with slats (or lamellas), it is also possible to configure a “secondary time” for the slat movement length and for the slat step time.
- **Status objects.** They report the current position of the shutter (and of the slats, if applicable).

- **Precise control.** Allows moving the shutter to a particular position (defined by parameter). Moreover, for blinds with slats, it is also possible to establish a particular position for the slats (value between 0% and 100%).
- **Scenes.** Allows running and/or saving a specific action over the channel/s where this function is enabled.
- **Alarms.** Two alarms are available for each shutter channel. The parameterized action will be executed when an alarm event is received.
- **Reverse movement.** Allows inverse shutter control.
- **Direct positioning.** Function that permits moving the shutter to a preset specific position by sending a 1-bit communication object.
- **Start-up configuration**. Default or custom.

All these options are explained in detail in section 3, ETS Parametrization.

2.3 TESTING THE OUTPUTS FROM THE IR REMOTE

The ACTinBOX QUATRO actuator incorporates a built-in infrared receiver, located next to the programming LED. This functionality allows testing the actuator outputs by making use of the Zennio infrared remote control (the one used to control the InZennio Z38i and ZAS touch panels. See Figure 3).

Note: *to control the outputs from the IR remote,* ***it is necessary to have the programming LED on*** *(lighting in red).*

The control of the outputs is carried out as follows:

- Press over the right button: closes the relay of the corresponding output.
- Press over the left button: opens the relay of the corresponding output.

Figure 3 shows a representation of the buttons of the IR remote that must be pressed depending on the output to be remotely controlled.

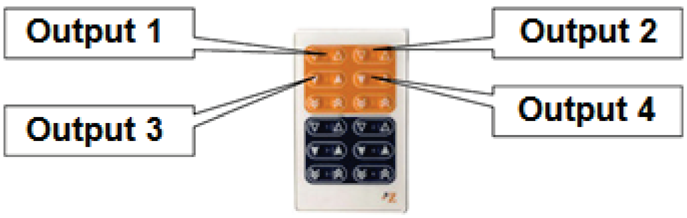


Figure 3 Controlling the outputs from the IR remote

The IR control of the device outputs is only provided for testing purposes. None of the parameterized functions (such as scenes, timers, etc.) is taken into account for opening / closing the relays remotely. No status objects will be sent to the bus either.

3 ETS PARAMETRIZATION

To begin with the parameterization of the ACTinBOX QUATRO actuator it is necessary, once the ETS application is running, to import the database of the product (ACTinBOX QUATRO application program).

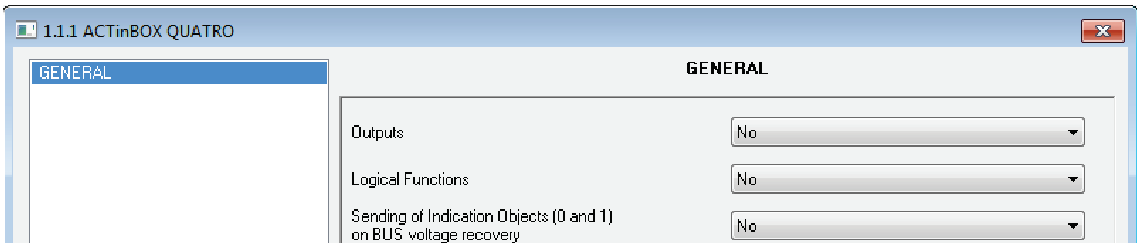
Next, the device must be added to the project where desired. Finally, right-click on the device and select "Edit parameters" to start with the configuration.

The following sections provide a detailed explanation about each of the different functionalities of the application in ETS.

3.1 DEFAULT CONFIGURATION

This section shows the default configuration the device parameterization starts from.

When the parameter edition is entered for the first time, the following window will come up:


1.1.1 ACTinBOX QUATRO

GENERAL

GENERAL

Outputs	No
Logical Functions	No
Sending of Indication Objects (0 and 1) on BUS voltage recovery	No

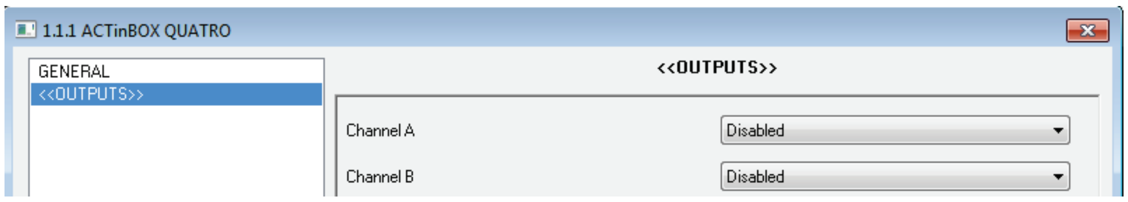
Figure 4 Configuration screen by default

As shown in Figure 4, outputs, inputs and logical functions are disabled by default, so there will be no communication objects available until the user enables some of the functions of the actuator.

If the parameter "**Sending of indication objects on bus voltage recovery**" is set to "Yest", two new 1-bit communication objects will show up ("Reset 0" and "Reset 1"), which will send to the KNX bus the values "0" and "1" after a bus power failure, in order to recover the communication with the rest of the devices in the installation. This sending may be immediate or after a configured delay (in seconds).

3.2 OUTPUTS

When "Outputs" is set to "Yes", a new tab will be added to the left-side panel, making it possible to configure the outputs of the device. This screen will look as follows:



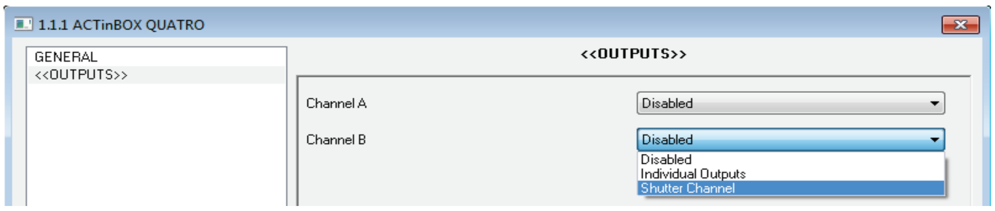
<<OUTPUTS>>

Channel A Disabled

Channel B Disabled

Figure 5 Configuring the outputs

The outputs are grouped into channels (Channel A and Channel B, both disabled by default), with two outputs each. A drop-down list permits configuring the behaviour of each channel, as **individual outputs** or **shutter channels**.



<<OUTPUTS>>

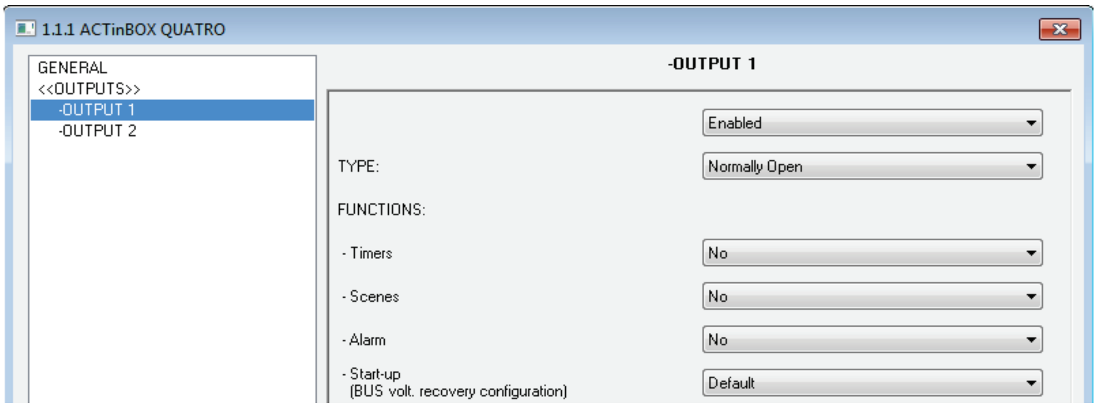
Channel A Disabled

Channel B Disabled

Figure 6 Configuring the channels

3.2.1 INDIVIDUAL OUTPUTS

Figure 7 represents an example of how a channel can be parameterized: in this case, channel A is parameterized as "individual outputs" (output 1 and 2).


1.1.1 ACTinBOX QUATRO

GENERAL
<<OUTPUTS>>
-OUTPUT 1
-OUTPUT 2

-OUTPUT 1

	Enabled
TYPE:	Normally Open
FUNCTIONS:	
- Timers	No
- Scenes	No
- Alarm	No
- Start-up (BUS volt. recovery configuration)	Default

Figure 7 Channel A configured as individual outputs

Once the output is enabled, the ETS topology will automatically display the following communication objects (1-bit each):

- **[OX] ON/OFF**: allows activating (ON) or deactivating (OFF) the corresponding output by sending the value "1" or "0", depending on the parameterized output type.
- **[OX] Status**: shows the current status of the output (active or inactive).
- **[OX] Block**: allows locking/unlocking the output (i.e., disabling/enabling its control) by sending the values "1" or "0" to the object, respectively.

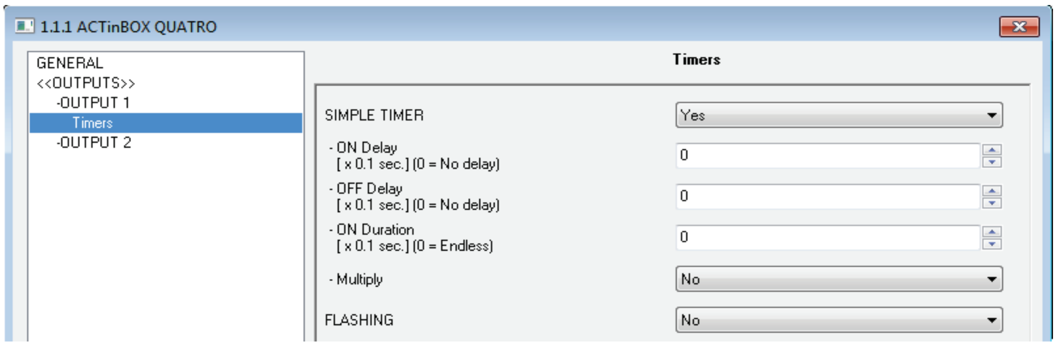
  <u>**Note**</u>: *only the Alarm function has a higher priority than the block function; if an alarm signal arrives when the output is locked, the output will be set to the status defined by the alarm function. When the alarm gets deactivated, the output returns to the lock status.*

The first thing to be parameterized is the type of each output of the channel:

- **Normally open**: the output will be considered as active (ON) when the relay stays closed and inactive (OFF) when the relay stays open.
- **Normally closed**: the output will be considered as active (ON) when the relay stays open and inactive (OFF) when the relay stays closed.

Next, the list of functions available for each output:

- **Timers**: allow performing a timed control of the outputs, both through a <u>simple timer</u> and/or through an <u>intermittent sequence</u> (*flashing*).


1.1.1 ACTinBOX QUATRO

GENERAL
<<OUTPUTS>>
-OUTPUT 1
Timers
-OUTPUT 2

Timers

SIMPLE TIMER	Yes
- ON Delay [x 0.1 sec.] (0 = No delay)	0
- OFF Delay [x 0.1 sec.] (0 = No delay)	0
- ON Duration [x 0.1 sec.] (0 = Endless)	0
- Multiply	No
FLASHING	No

Figure 8 Timers screen. Simple timer

➢ **Simple timer**: allows switching the output on and off on the reception of the values "1" and "0" through the "[OX] Timer" object. These switches are subject to a set of parameterizable times:

- **On Delay**: time the actuator will wait before switching on the output once the ON order has been received (through the "[OX] Timer" object). The value "0" will cause an immediate response.
- **Off delay**: time the actuator will wait before switching off the output once the OFF order has been received (through the "[OX] Timer" object). The value "0" will cause an immediate response.
- **On Duration**: time the output remains ON before switching back to the OFF status. A "0" in this field means the output will remain permanently ON.

These parameters apply as follows:

  - When ACTinBOX QUATRO receives a "1" through the "[OX] Timer" communication object, an ON order is sent to the output after waiting for the "On Delay" time. The output will stay switch off again after the "On Duration" time (if other than 0).
  - When ACTinBOX QUATRO receives a "0" through the "[OX] Timer" communication object, an OFF order is sent to the output after waiting for the "Off Delay" time.

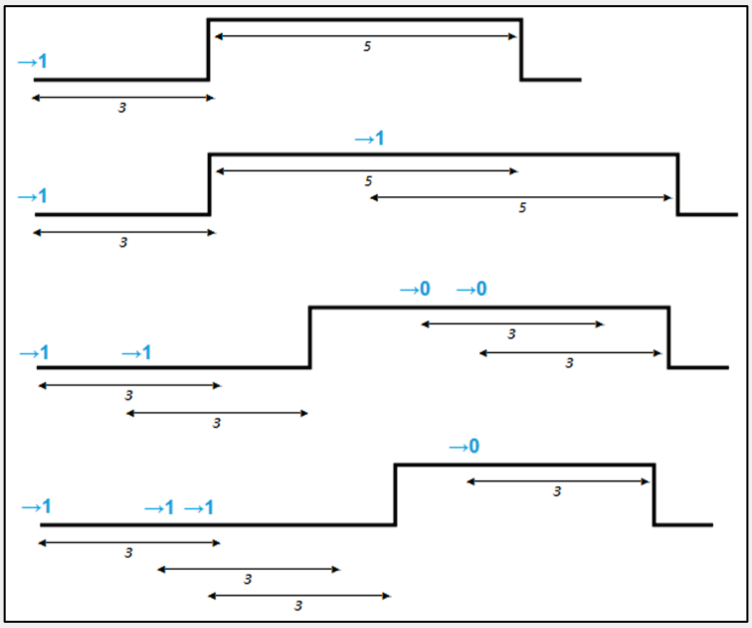
- **Multiply**: allows progressively increasing (multiplying), in runtime, the On Duration time or the On/Off delays of the output. Two situations are distinguished:

- **No multiply**:
  - If the On delay count is already running, it will be reset every time a new “1” is received through the “[OX] Timer” object.
  - If the output has already been activated and the On Duration time is counting, it will be reset whenever a new “1” is received.
  - If the Off delay count is already running, it will be reset every time a new “0” is received.
- **Multiply**:
  - If the On delay count is already running and the value “1” is received several times through the “[OX] Timer” object, then the actual delay time will be “n” times the parameterized time, being “n” the number of times the value “1” is received .
  - If the output has already been activated and while the On Duration time is counting the value “1” is received several times, then the actual duration will be “n” times the parameterized time, being “n” the number of times the value “1” is received.
  - If the Off delay count is already running and the value “0” is received several times, then the actual delay time will be “n” times the parameterized time, being “n” the number of times the value “0” is received.

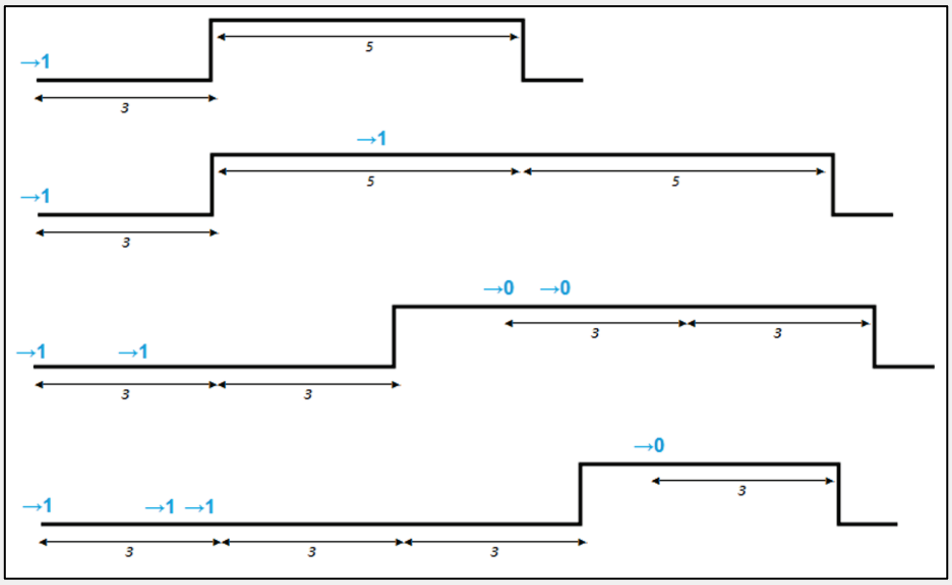
<u>Note</u>: *the Multiply option may result particularly useful under parameterizations with no ON and OFF delays. Nevertheless, as already explained and as the following example shows, these delay times, if parameterized with a value other than 0, do also admit multiplication.*

***Example**: the following is parameterized: On Delay = 3 seconds; Off Delay = 3 seconds, On Duration = 5 seconds. The graphs bellow reflect some possible situations if the values "0" or "1" are received from the (which is represented as →0 and →1), respectively for the cases of having the "multiply" option enabled and disabled.*

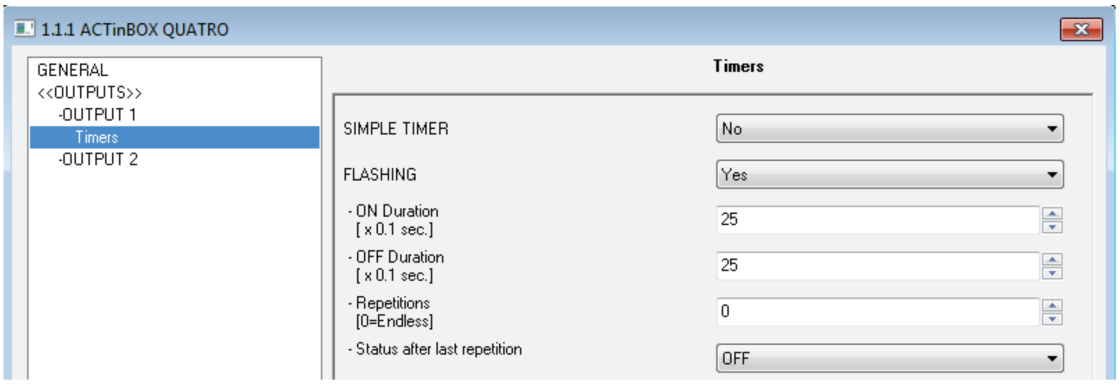
<u>With no multiplication</u>:



<u>With multiplication</u>:



- **Flashing** (Figure 9). This function allows the execution of alternating ON-OFF sequences when needed. It is possible to parameterize the ON and an OFF time length, as well as the number of repetitions (the value "0" makes the sequence endless until a "0" is received through the "[OX] Flashing" communication object). It is also possible to define the final status (ON or OFF) of the output after the last repetition.

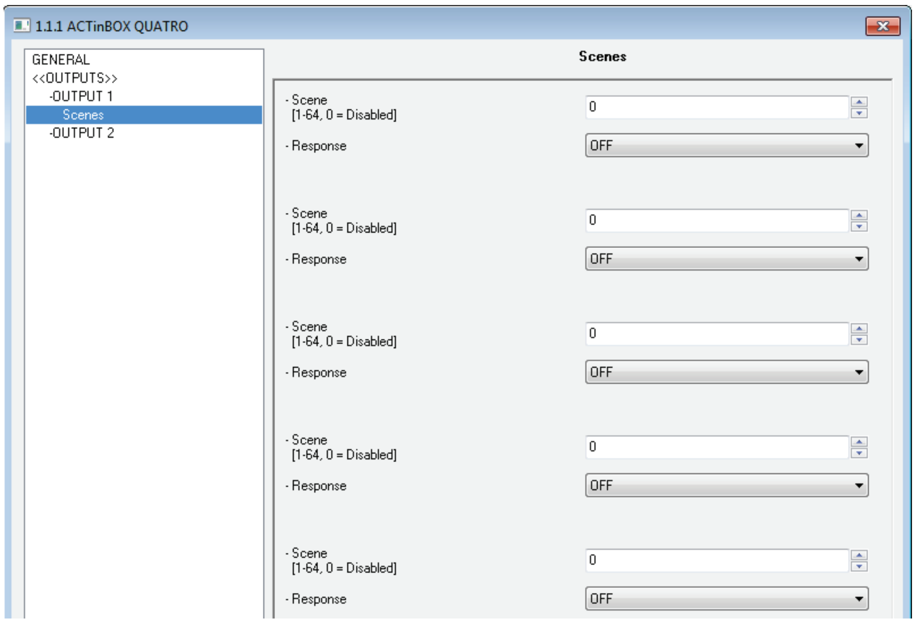


SIMPLE TIMER: No
FLASHING: Yes
- ON Duration [x 0.1 sec.]: 25
- OFF Duration [x 0.1 sec.]: 25
- Repetitions [0=Endless]: 0
- Status after last repetition: OFF

Figure 9 Flashing

It is **important** to keep in mind that ACTinBOX QUATRO allows parameterizing both a simple timer and an intermitting sequence for the same output.

- **Scenes**: scenes consist of a synchronized activation of the devices in the domotic installation, so that different predefined atmospheres can be generated by simply sending a scene value over the bus.



- Scene [1-64, 0 = Disabled]: 0
- Response: OFF
- Scene [1-64, 0 = Disabled]: 0
- Response: OFF
- Scene [1-64, 0 = Disabled]: 0
- Response: OFF
- Scene [1-64, 0 = Disabled]: 0
- Response: OFF
- Scene [1-64, 0 = Disabled]: 0
- Response: OFF

Figure 10 Scenes

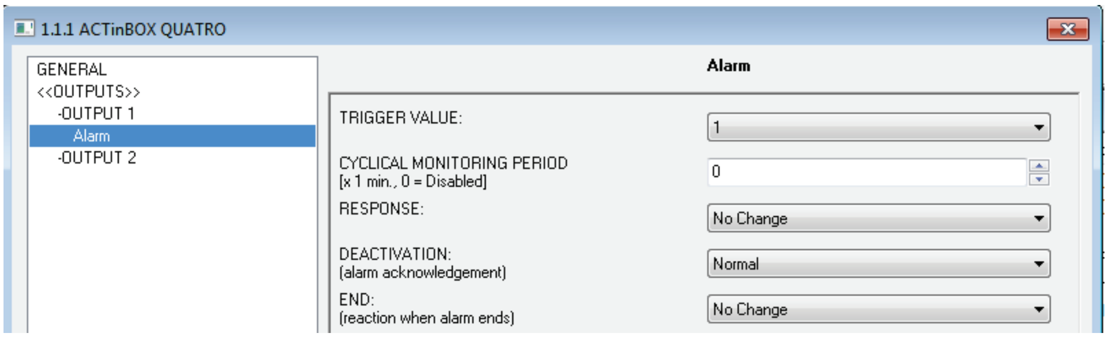
There is a 1-byte communication object associated with Scenes for the individual outputs: "Scenes (Individual Outputs)", which shows when the "Outputs" tab in ETS is enabled, even if the outputs are disabled.

In the case of the individual outputs, scenes allow linking a numerical value (between 1 and 64, where 0 means that the option is disabled) to an output status (OFF or ON). Thus, when the predefined scene value is received through the Scenes object, the parameterized action for the output (a switch-off or a switch-on) will be performed; so it will be possible to create different environments in the installation.

Besides running scenes, it is possible to **learn** (save) scenes, taking into account that the associated numerical values for learning scenes are in the range 128-191.

ACTinBOX QUATRO allows running and/or learning **up to 5 different scenes** for each output.

- **Alarm**: for each output it is possible to configure an alarm, which, once activated, will have **priority** over the rest of the orders that the actuator may receive, i.e., any order received while the alarm is active will be ignored until the alarm becomes inactive.



TRIGGER VALUE: 1

CYCLICAL MONITORING PERIOD [x 1 min., 0 = Disabled]: 0

RESPONSE: No Change

DEACTIVATION: (alarm acknowledgement): Normal

END: (reaction when alarm ends): No Change

Figure 11 Alarm

The following parameters can be configured for alarms:

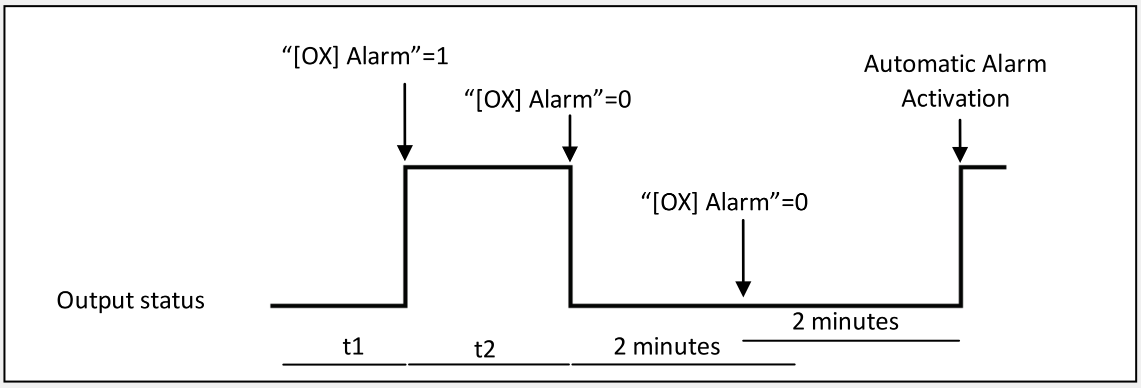
- **Trigger value**: sets the value that will trigger the alarm status. It can be "1" or "0". The alarm status will be activated when the value set in this field is sent to the “[OX] Alarm” object, and it will be deactivated when the opposite value is sent to the aforementioned communication object.

Cyclical monitoring period (minutes): defines, for the case of having a periodically-sent alarm object (“1” or “0”, as corresponding each time), the maximum permitted time without receiving the “no alarm” value (“[OX] Alarm” = value contrary to the parameterized trigger value) before the actuator automatically assumes the alarm status, foreseeing the possibility of a failure in the transmitting device. If for whatever reason the monitoring period expires, ACTinBOX QUATRO will trigger the parameterized action (unless it does not imply a change in the output status). This will not happen while the “no alarm” value keeps being sent before each monitoring period expires. The cyclical monitoring can be disabled by simply setting a 0 under this filed.

The following example is provided for a better comprehension:

<u>Example</u>:

Suppose that a cyclical monitoring period of 2 minutes is configured. The trigger value is “1” and the reaction of the actuator when the alarm is activated consists in switching on the output, while when the alarm is deactivated, the output will be switched off. Assuming that the output is off, the alarm becomes active (“[OX] Alarm=1”), so the output is switched on. While the alarm is not deactivated, any action over the output will be ignored. After some time (t2), the alarm is deactivated (“[OX] Alarm=0”), which makes the output switch from ON to OFF. Before the parameterized cyclical monitoring period (2 minutes) ends, a new alarm deactivation order arrives, so the time count starts again. After 2 minutes without receiving further values through the alarm object, the alarm will be automatically activated, making the output status switch on. As before, any action over the output will be ignored until the alarm is deactivated. See the following figure.



- **Response**: sets the response (status) of the actuator output when the alarm is activated:

  - No change.
  - ON.
  - OFF.
  - Flashing: 3 drop-down lists are shown to configure the ON Duration, the OFF Duration and the number of repetitions of the sequence.

- **Deactivation**: two different procedures are provided to deactivate an active alarm:

  - Normal: depending on what was parameterized under “Trigger Value”, the alarm will be deactivated as soon as the actuator receives a “0” or a “1” through the alarm object.
  - Frozen: this method requires a normal deactivation, plus the reception of the value “1” over the “[OX] Unfreeze Alarm” object (if the latter does not occur, the alarm status will still be active). This method makes the channel output remain locked (even when the alarm situation is over) until it is externally (or manually) enabled.

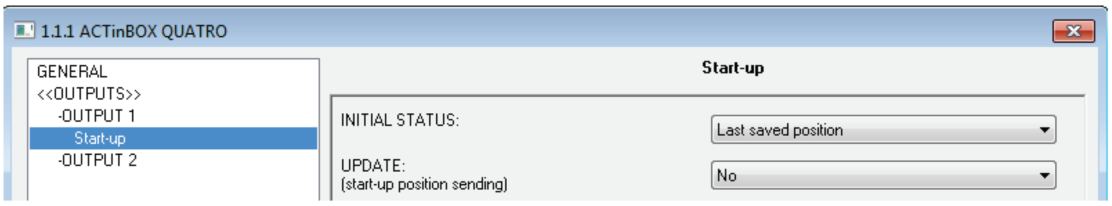
- **End (reaction when alarm ends)**: this parameter sets the status that will be adopted by the output once the alarm finishes:

  - No change
  - ON
  - OFF
  - Last (the output returns to the status it had before the alarm)

- **Start-up configuration**: defines the state (ON/OFF) that will be adopted by the outputs when the device recovers from a bus power failure, or after an ETS download. A default or a custom configuration may be selected.

  Selecting the default configuration will make the output stay off after a partial or complete ETS download, while after a bus power failure, the status of the output will be the same it had before the failure (ON or OFF).

If "custom" is chosen, ETS will show the following window:



INITIAL STATUS: Last saved position

UPDATE: (start-up position sending) No

Figure 12 Start-up configuration: custom.

Where the following parameters can be configured:

- **Initial status**: this field defines the initial status of the output after a bus power failure or after a download. The following options are available: <u>last saved position</u> (i.e., the status of the output before the bus power failure), <u>ON</u> or <u>OFF</u>.

- **Update**: enabling this field ("Yes") will make the actuator send the output status to the bus (through the corresponding communication object) after the start-up as a feedback for the rest of the devices in the installation. A parameterizable delay can also be applied to this sending. If the delay is set to 0, the status will be sent immediately.

The initial status is always sent through the object "[OX] Status".

3.2.2 SHUTTER CHANNEL

ACTinBOX QUATRO also allows configuring its outputs as shutter channels, making it possible to control **up to 2 separate shutters** in an installation.

When a channel is configured in ETS as a shutter channel, a 1-bit communication object ("[CX] Block") related to each enabled channel becomes visible. This object allows locking the channel outputs (i.e., disabling the control over them through both the ON/OFF and the timed control objects) when it is received with value "1". Moreover, if the shutter is already moving at the moment of being locked, the movement will be interrupted and any order over it will be ignored. The channel outputs will become unlocked when a "0" is sent through this object.

<u>Note</u>: *only the "alarm" function has a higher priority than the "block" function. If an alarm signal arrives while the channel is locked, the shutter will adopt the position* *configured for the alarm. Once the alarm is deactivated, the shutter will recover the position defined for the locked status.*

The first parameter to be configured is the type of the shutter:

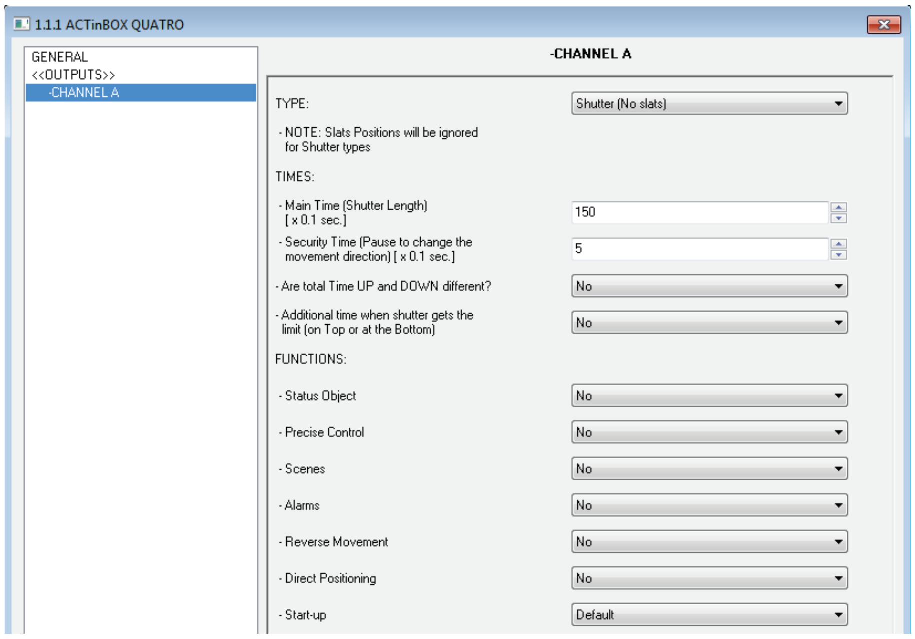
- **Shutter (no slats)**: typical shutter with a simple movement up/down. If this type is selected, two communication objects are enabled: "[CX] Move" and "[CX] Stop" to move the shutter up/down and to stop it, respectively.

  In addition, the following remark is shown for this type: "Slats positions will be ignored for Shutter types", which means that function parameters related to slat positions (%) will not be taken into account.

- **Blind (with slats)**: special slat shutters where both the shutter itself and the slats/lamellas are controlled with a single drive. ACTinBOX QUATRO allows controlling both movements: the slat rotation (which permits varying the level of the incident light) and the movement (up/down) of the blind. Two communication objects are enabled for this purpose: "[CX] Move" (which will send the orders to move the blind up/down) and "[CX] Stop/Step". If the device receives a "0" or a "1" through the latter when the blind is already in motion, the movement will be interrupted, while if the blind was not in motion, receiving a "0" through this communication object will make the slats rotate upwards, while receiving a "1" will make them rotate downwards. These step movements are useful to correct both the slat and the blind position together.

  To get detailed information about blinds with slats and their ETS configuration, please refer to **ANNEX I. Slats precise control**.

The next figures show the screens that contain the parameterizable options for channels configured as shutters (no slats) or as blinds (with slats).

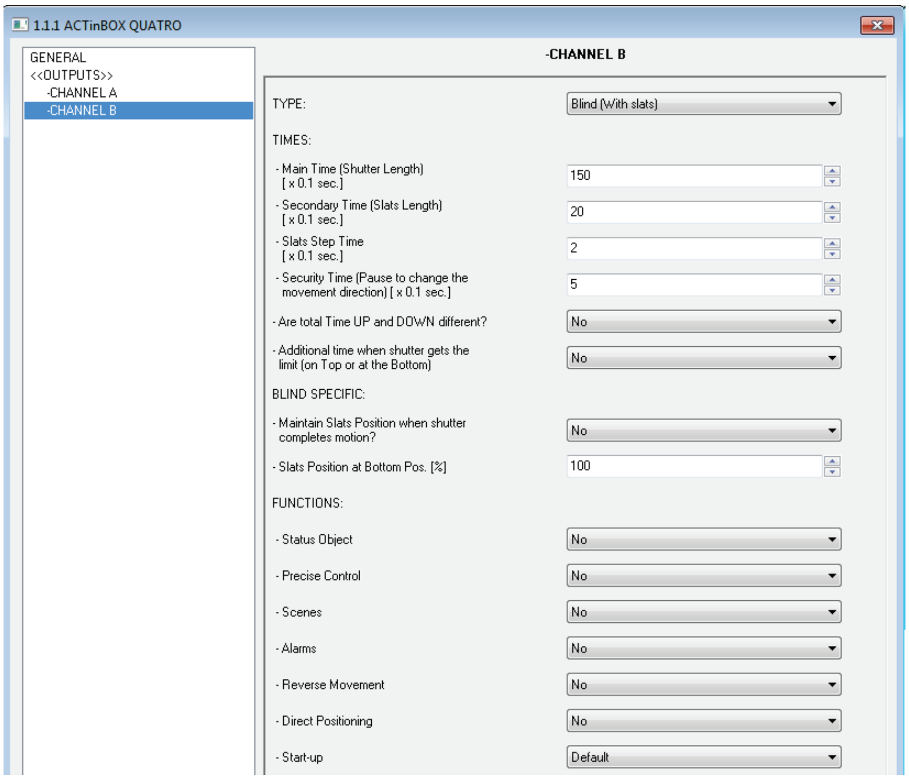

1.1.1 ACTinBOX QUATRO

GENERAL
<<OUTPUTS>>
-CHANNEL A

-CHANNEL A

TYPE:	Shutter (No slats)
- NOTE: Slats Positions will be ignored for Shutter types	
TIMES:	
- Main Time (Shutter Length) [x 0.1 sec.]	150
- Security Time (Pause to change the movement direction) [x 0.1 sec.]	5
- Are total Time UP and DOWN different?	No
- Additional time when shutter gets the limit (on Top or at the Bottom)	No
FUNCTIONS:	
- Status Object	No
- Precise Control	No
- Scenes	No
- Alarms	No
- Reverse Movement	No
- Direct Positioning	No
- Start-up	Default

Figure 13 Channel A configured as a shutter (no slats)


1.1.1 ACTinBOX QUATRO

GENERAL
<<OUTPUTS>>
-CHANNEL A
-CHANNEL B

-CHANNEL B

TYPE:	Blind (With slats)
TIMES:	
- Main Time (Shutter Length) [x 0.1 sec.]	150
- Secondary Time (Slats Length) [x 0.1 sec.]	20
- Slats Step Time [x 0.1 sec.]	2
- Security Time (Pause to change the movement direction) [x 0.1 sec.]	5
- Are total Time UP and DOWN different?	No
- Additional time when shutter gets the limit (on Top or at the Bottom)	No
BLIND SPECIFIC:	
- Maintain Slats Position when shutter completes motion?	No
- Slats Position at Bottom Pos. [%]	100
FUNCTIONS:	
- Status Object	No
- Precise Control	No
- Scenes	No
- Alarms	No
- Reverse Movement	No
- Direct Positioning	No
- Start-up	Default

Figure 14 Channel B configured as a blind (with slats)

Apart from the shutter type, it is necessary to configure some specific functions, which are common to both types:

- **Times**: this section allows setting the times (in tenths of seconds) that define to the shutter movement.

- **Main time (Shutter length)**: this is the time the shutter needs to move all the way down or up. In case the shutter does not take similar times for both directions, parameter "Are total time up and down different?" should be set to "Yes". In such case, this Main time will define the time required to move the shutter all the way down, while a secondary parameter ("Total Time Up") will become available to define the time required for the inverse movement.

  It is not necessary to periodically calibrate this parameter, since the exact shutter position is stored by ACTinBOX QUATRO at any time, even after power failures.

- **Security time (pause to change the movement direction)**: this is the time reserved by the actuator, as a preventive measure to protect the shutter drive, when the movement switches from one direction to the inverse. If the device receives an order to lower the shutter while it is being raised, ACTinBOX QUATRO will temporarily interrupt the movement (security time) and will afterwards lower the shutter. It is advisable to set a value not lower than 5 tenths of a second (the default value).

- **Are total time Up and Down different?** Heavy shutters may require different times to move up and to move down. When this happens, this field should be set to "Yes", making the already mentioned "Main time (Shutter length)" parameter be interpreted as the time required to lower the shutter, while a secondary parameter will be provided to define the time required to raise it.

- **Additional time when the shutter gets the limit**: this parameter is useful to prevent small maladjustments and to ensure that the shutter always gets to the lowest or highest position. This is done by defining an extra time to keep the drive moving once the rising/lowering times end. This parameter is not visible by default; however it is advisable to give it a value to guarantee a proper behaviour in the shutter.

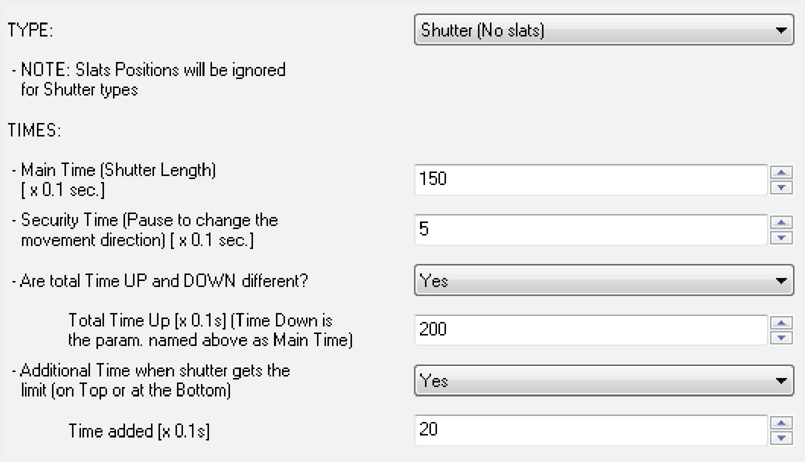
If the shutter is configured as a blind with slats (Figure 1), some more specific parameters appear. All of them are explained in detail in **ANNEX I. Slats precise control**.

Note: *after programming the device from ETS from ETS and unless a custom start-up configuration has been parameterized, ACTinBOX QUATRO will assume the shutter as completely raised, so any raising order will be ignored.*

The following is an example of a possible shutter configuration.

Example:

Channel A (shutter without slats) takes 15 seconds to be lowered and 20 to be raised. Additional times for direction changes (5 tenths of a second) and at the end of the movement (2 seconds) are also required. The required parameterization is as follows:



TYPE:	Shutter (No slats)
- NOTE: Slats Positions will be ignored for Shutter types	
TIMES:	
- Main Time (Shutter Length) [x 0.1 sec.]	150
- Security Time (Pause to change the movement direction) [x 0.1 sec.]	5
- Are total Time UP and DOWN different?	Yes
Total Time Up [x 0.1s] (Time Down is the param. named above as Main Time)	200
- Additional Time when shutter gets the limit (on Top or at the Bottom)	Yes
Time added [x 0.1s]	20

The following parameters add functionality or special features to both shutter types (with or without slats):

- **Status object**: this function provides a 1-byte communication object ("[CX] Current Shutter Position") which will show, at any time, the exact position of the shutter in percentage (%). This object throws the value 0 (or 0%) when the shutter in completely up and 255 (or 100%) when it is completely down. The remaining values represent intermediate positions.

  It is possible to set, by parameter, whether the shutter position will be sent every second or not. This is done by enabling or disabling “Send current shutter position every second while moving?”, which appears once the “Status Object” parameter has been enabled.

  For Blinds with slats, the 1-byte object "[CX] Current Slats Position" is also provided. It will show the value 0 (0%) when the slats are completely "up" and the value 255 (100%) when the slats are completely "down".

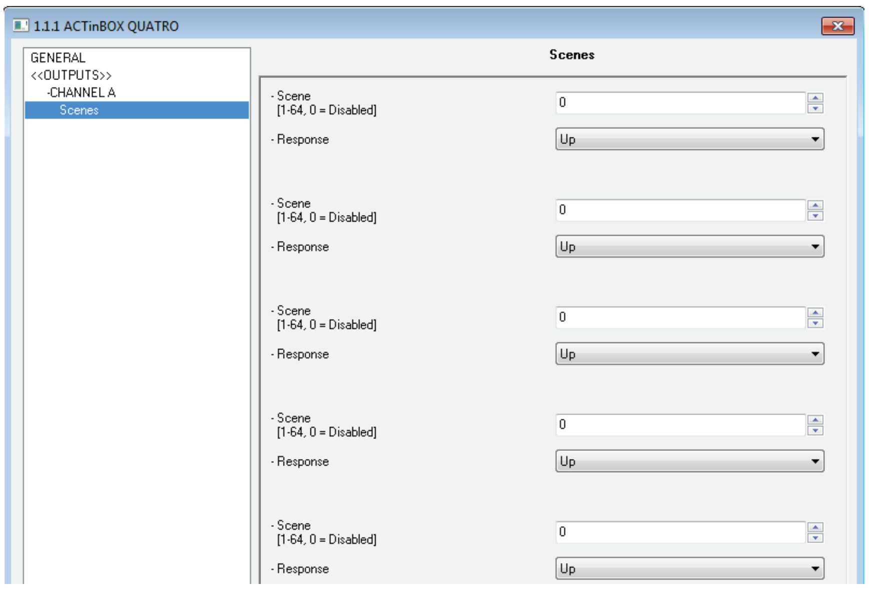
- **Precise control**: this function makes it possible to actually move the shutter to any position via a 1-byte communication object ("[CX] Shutter Positioning", in percentage). Every time ACTinBOX QUATRO receives a new percentage value through this object (e.g. 50%), the shutter moves to the corresponding position (e.g., the central position).

  For blinds with slats, the actuator also implements the 1-byte object "[CX] Slats Positioning" for setting the desired position (in percentage) for the slats.

- **Scenes**: this function makes it possible to use scenes to control the shutters. It allows choosing precise positions where the shutter will be moved to upon the reception of certain scene numbers through the 1 byte object “Scenes (Shutter Channels)”.

  Apart from running scenes, it is possible to **learn** (save) scenes, taking into account that the values that should be received to learn a scene must be in the range 128-191 (values 0-63 are reserved for running scenes).

  Up to 5 scenes can be run and/or leant, for each shutter channel.


1.1.1 ACTinBOX QUATRO

GENERAL
<<OUTPUTS>>
-CHANNEL A
Scenes

Scenes

- Scene [1-64, 0 = Disabled]	0
- Response	Up
- Scene [1-64, 0 = Disabled]	0
- Response	Up
- Scene [1-64, 0 = Disabled]	0
- Response	Up
- Scene [1-64, 0 = Disabled]	0
- Response	Up
- Scene [1-64, 0 = Disabled]	0
- Response	Up

Figure 15 Scenes

The “Scene” parameter indicates the scene value the shutter will react to. If this value is 0, the corresponding option is disabled.

“Response” sets the precise position where the shutter will be moved to when the above scene number arrives from the bus. The shutter can be moved completely up, completely down, or to a specific intermediate position (in percentage, from 0% to 100%). In case of choosing the latter, parameter "Shutter specific position?" will become visible to allow the definition of a specific shutter position (through the "Select Shutter Position [%]"parameter, after selecting "Yes") or to make the shutter stay as is.

For blinds with slats it is also possible to configure a specific position for the slats (via the parameter "Select Slats Position [%]") when the configured scene number is received, or to make them maintain their current position.

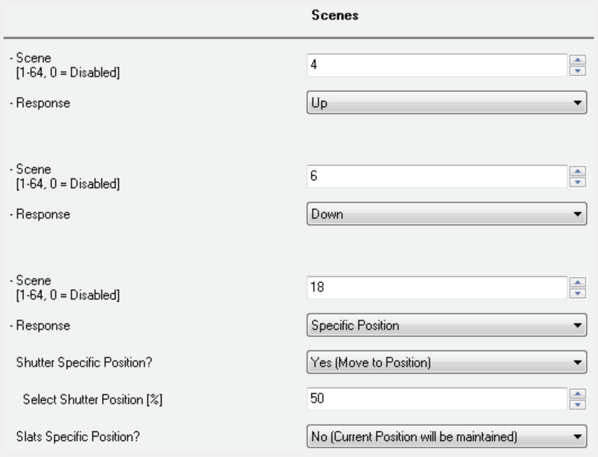
The following example illustrates the configuration process.

Example:

Consider a facility where 3 scenes will be distinguished (values "4", "6" and "18") and where ACTinBOX QUATRO will be used to move a blind with slats to precise positions on the reception of these scene values:

- *Scene 4: shutter up*
- *Scene 6: shutter down*
- *Scene 18: central position (50%). The slats maintain their current position.*

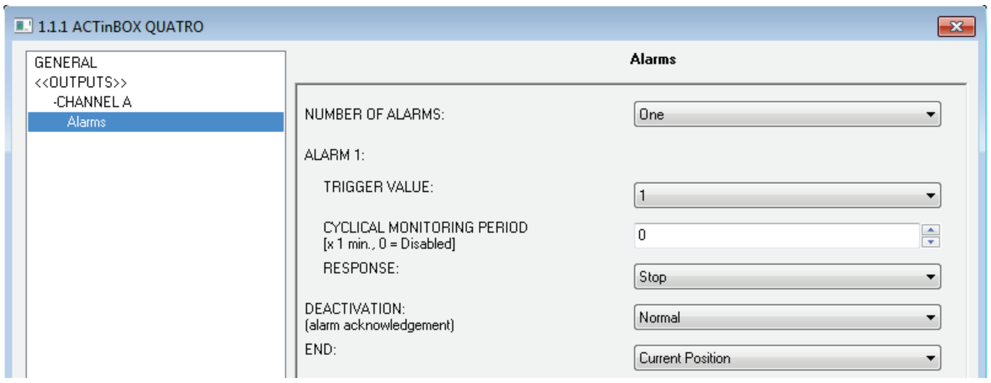
The associated parameterization should be as follows.



Scenes

- Scene [1-64, 0 = Disabled]	4
- Response	Up
- Scene [1-64, 0 = Disabled]	6
- Response	Down
- Scene [1-64, 0 = Disabled]	18
- Response	Specific Position
Shutter Specific Position?	Yes (Move to Position)
Select Shutter Position [%]	50
Slats Specific Position?	No (Current Position will be maintained)

- **Alarms**: ACTinBOX QUATRO allows configuring up to 2 alarms for each shutter channel. This function is designed for cases in which the actuator must react to an alarm situation. Having two alarms configured allows carrying out different reactions to two external events.



NUMBER OF ALARMS: One

ALARM 1:

TRIGGER VALUE: 1

CYCLICAL MONITORING PERIOD [x 1 min., 0 = Disabled]: 0

RESPONSE: Stop

DEACTIVATION: (alarm acknowledgement): Normal

END: Current Position

Figure 16 Alarms

The following parameters can be configured:

- **Number of alarms**: sets whether to use one or two alarms. Both of them can be independently managed through their corresponding communication objects ("[CX] Alarm" for Alarm 1 and "[CX] Alarm 2" for the second one).

  It is important to keep in mind that Alarm 1 has a higher priority than Alarm 2. This means that if a channel is under the Alarm 2 status and Alarm 1 occurs, then the shutter will switch to the Alarm 1 status and only switch back to the Alarm 2 status once Alarm 1 finishes. Analogously, when the channel is under the Alarm 1 status and Alarm 2 occurs, Alarm 1 prevails.

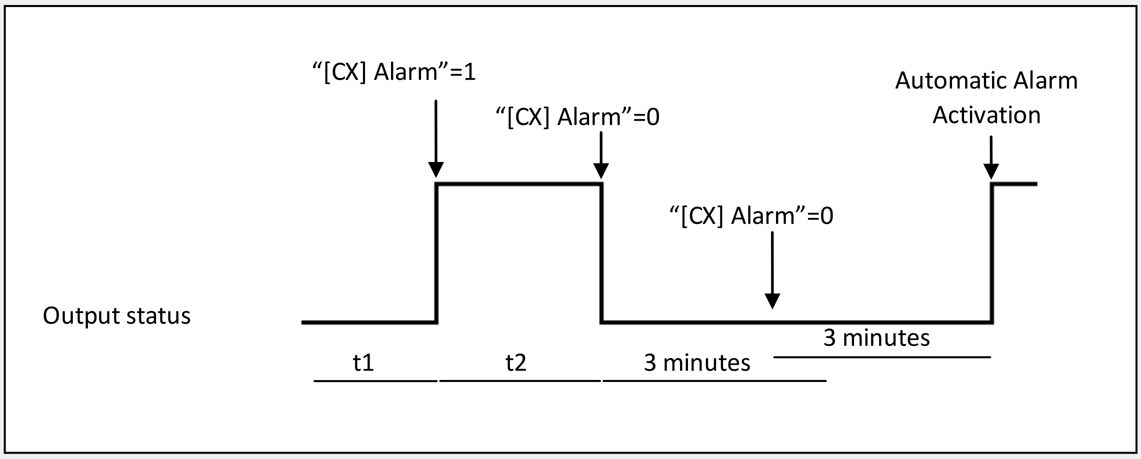
- **Trigger value**: sets the value ("1" or "0") that will trigger the alarm status. This value is intended to be received through object "[CX] Alarm" (or "[CX] Alarm2").

- **Cyclical monitoring period (minutes)**: defines, for the case of having a periodically-sent alarm object ("1" or "0", as corresponding each time), the maximum permitted time without receiving the "no alarm" value ("[OX] Alarm" = value contrary to the parameterized trigger value) before the actuator automatically assumes the alarm status, foreseeing the possibility of a failure in the transmitting device. If for whatever reason the monitoring period expires, ACTinBOX QUATRO will trigger the parameterized action (unless it does not imply a change in the output status). This will not happen while the "no alarm" value keeps being sent before each monitoring period expires. The cyclical monitoring can be disabled by simply setting a 0 under this filed.

The following example illustrates the above behaviour:

Example:

Suppose that a cyclical monitoring period of 3 minutes is configured. The trigger value is "1" and the reaction of the actuator when the alarm is activated consists in moving the shutter all the way up, while when the alarm is deactivated, the shutter will be raised. Assuming that the shutter has been lowered, when the alarm becomes active the actuator will begin raising the shutter. While the alarm is not deactivated, any action over the shutter channel will be ignored by the actuator. After some time (t2), one "0" arrives through the alarm object, which makes the alarm turn off and the shutter be raised. Before the parameterized cyclical monitoring period (3 minutes) ends, a new alarm deactivation order arrives, so the time count starts again. After 3 minutes without receiving further values through the alarm object, the alarm will be automatically activated, making the shutter move up again. As before, any action over the shutter will be ignored until the alarm is deactivated. See the following figure



➢ **Response**: sets the response order to be sent to the shutter when the alarm is activated:

- Stop.
- Up.
- Down.
- Specific position.

When the latter is chosen, a new drop-down list is shown, letting the integrator set a value for this specific position, in the range 0% (completely up) to 100% (completely down).

For blinds with slats, an option called "Select Slats position [%]" will also become visible, making it possible to set a position for the slats between 0% (totally open or "up") and 100% (totally closed or "down").

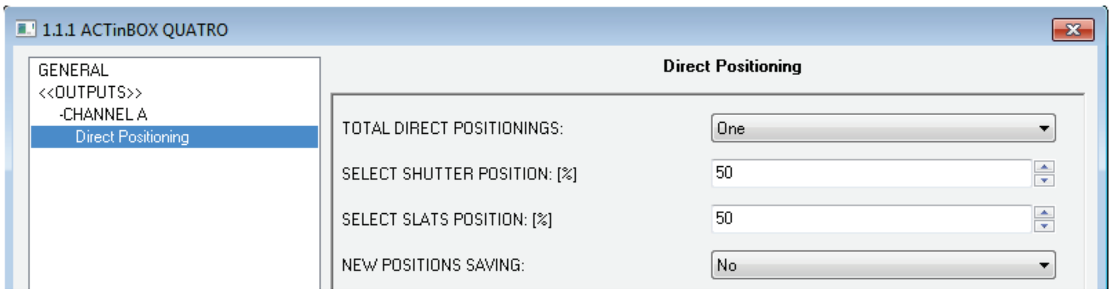
- **Deactivation**: two different procedures are provided to deactivate an active alarm:
  - Normal: depending on what was parameterized under “Trigger Value”, the alarm will be deactivated as soon as the actuator receives a “0” or a “1” through the corresponding alarm object.
  - Frozen: this method requires a normal deactivation, plus the reception of the value “1” over the corresponding *unfreeze alarm* object (if the latter does not occur, the alarm status will still be active). This method makes the channel output remain locked (even when the alarm situation is over) until it is externally (or manually) enabled.
- **End**: this parameter sets the desired shutter position once the alarm becomes inactive:
  - Current position.
  - Up.
  - Down.
  - Last position (before the alarm).

- **Reverse movement**: this function makes it possible to control a shutter with the usual orders inverted (ACTinBOX QUATRO normally raises the shutter when the value "0" is received through the object "[CX] Move" and stops it with value "1"). Therefore, if this function is enabled, ACTinBOX QUATRO will also raise the shutter when the value "1" is received through object "[CX] Reverse Movement", and will lower it on the reception of the value “0”.

  This reverse control is compatible with the usual control, since “[CX] Move” and “[CX] Reverse Movement” are provided as separate objects for normal and reverse control, respectively.

Reverse control becomes particularly useful when a centralised OFF order is sent over the installation to turn the lights off and to lower the shutters at the same time. In this case, the value "0" can be sent to the light ON/OFF objects and to the shutter "Reverse movement" objects.

- **Direct positioning**: this function allows moving the shutter to a preset position by a simple 1-bit order (through "[CX] Direct Positioning" and "[CX] Direct Positioning 2"). When a "1" is received through one of these objects, the shutter will be moved to the parameterized position. When a "0" is received, no action is performed.



TOTAL DIRECT POSITIONINGS:	One
SELECT SHUTTER POSITION: [%]	50
SELECT SLATS POSITION: [%]	50
NEW POSITIONS SAVING:	No

Figure 17 Direct positioning

It is possible to configure the following parameters:

- **Total direct positionings**: one or two direct positioning functions can be enabled.

- **Select shutter position [%]**: sets the exact position to move the shutter to (0%=Top; 100%=Bottom).

  For blinds with slats, an option called "Select Slats position [%]" will also become visible, making it possible to set a position for the slats when the value "1" is received through the corresponding positioning object.

  If two direct positioning controls are configured, two boxes will be available here: "Select Shutter Position 1" and "Select Shutter Position 2" (as well as "Select Slats position 1" and "Select Slats position 2" for blinds with slats).

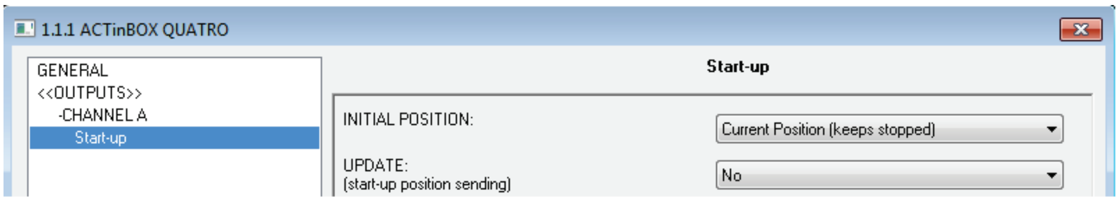
- **New positions saving:** allows ("Yes") or not ("No") saving new direct positions. After enabling this option, one or two new communication objects appear (depending on the number of direct positioning controls): "[CX] Save Position" and "[CX] Save Position 2". To overwrite the parameterized direct position it is necessary to send a "1" to these objects when the shutter is at the desired position.

<u>**Note**</u>: *saving a new position is not possible while the shutter is moving.*

- **Start-up configuration**: defines the shutter state that the device will assume/order when it recovers from a bus power failure, or after an ETS download. A <u>default</u> or a <u>custom</u> configuration may be selected.

  If "default" is chosen, after a partial or complete download from ETS, ACTinBOX QUATRO will assume the shutter is completely up (0%), independently of its actual status. For blinds with slats, ACTinBOX QUATRO assumes that the slats are completely open (0%), no matter what their actual status is. On the other hand, after a bus power failure, the device will assume the shutter (and the slats) maintains the same state as before the power failure.

  When "custom" is chosen, ETS will show the following window:



Start-up

INITIAL POSITION: Current Position (keeps stopped)

UPDATE: (start-up position sending) No

Figure 18 Custom Start-up configuration

Where it is possible to configure the following parameters:

- **Initial position**: this field defines the desired initial position for the shutter after a bus power failure or after programming the device. The following statuses can be chosen: <u>Current position (keeps stopped)</u>, <u>up</u>, <u>down</u>, or <u>specific position</u> (which moves the shutter to the position set for "Select shutter position [%]" and the slats to the position set for "Select slats position [%]", both shown after this option is chosen).

- **Update**: by enabling this field ("Yes"), the position status object will be sent to the bus as a feedback for the rest of the devices in the installation. In addition a delay can be applied to this sending. If the value 0 is set, the status is sent immediately.

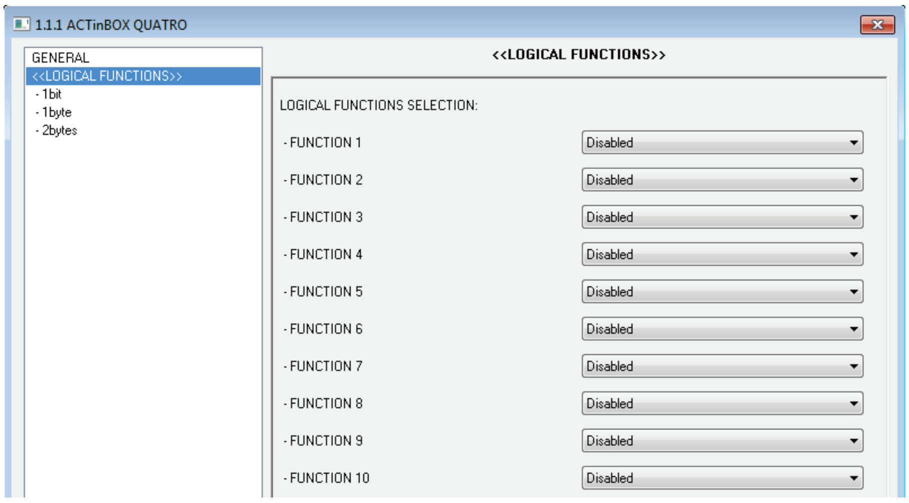
Note: *the initial status is always sent through the object "[CX] Current Shutter position" (and "[CX] Current Slats position", for slats).*

Note: *in the event of a bus power failure while the shutter is moving, ACTinBOX QUATRO opens the output relay prior to being unpowered, as a security measure. The interrupted movement is not resumed after the bus recovery.*

3.3 LOGICAL FUNCTIONS

This option in ACTinBOX QUATRO makes it possible to perform mathematical or binary logic operations to incoming values received from the KNX bus, and to send the result through other communication objects specifically enabled in the actuator for this purpose.

Up to 10 different and independent logical functions can be enabled, being each of them capable of carrying out **up to 4 operations**. To use any of them, it is necessary to enable it from the following ETS screen, which becomes available after selecting "Yes" under Logical Functions in the General parameter screen.


1.1.1 ACTinBOX QUATRO

GENERAL
<<LOGICAL FUNCTIONS>>
- 1bit
- 1byte
- 2bytes

<<LOGICAL FUNCTIONS>>

LOGICAL FUNCTIONS SELECTION:

- FUNCTION 1	Disabled
- FUNCTION 2	Disabled
- FUNCTION 3	Disabled
- FUNCTION 4	Disabled
- FUNCTION 5	Disabled
- FUNCTION 6	Disabled
- FUNCTION 7	Disabled
- FUNCTION 8	Disabled
- FUNCTION 9	Disabled
- FUNCTION 10	Disabled

Figure 19 Logical functions

To get further information about how to use and parameterize the logical functions, please refer to the specific document "**Logical Functions X10**", available at http://www.zennio.com.

ANNEX I. SLATS PRECISE CONTROL

Zennio actuators allow controlling the movement of shutters, blinds or similar window/door automated systems, which may be of one of the following types:

- Shutter (No slats).
- Blind (with slats/lamellas).

Depending on the shutter type, the ACTinBOX QUATRO application program will show different options.

This particular section is referred to the parameterization of blinds with slats.

To begin with, it is important to keep in mind the criteria followed by the actuator for shutter positioning:

- The shutter is in the “up” position (**0%**, in percentage) when it is completely **open**.
- The shutter is in the “down” position (**100%**, in percentage) when it is completely **closed**.

While the slat positioning criteria is as follows:

- The slats are in the “up” position or “open” (**0%**, in percentage) when their position is such that they can only move downwards.
- The slats are in the “down” position or “close” (**100%**, in percentage) when their position is such that they can only move upwards.

Figure 20 shows a scheme of the positions the slats may adopt.

It is necessary to take into account that shutter actuators control shutter drives without any feedback about their exact position, and that the movement of the slats relays entirely on the movement of shutter drive itself. This means that a **movement in the slats will always provoke a certain change in the position of the shutter**.

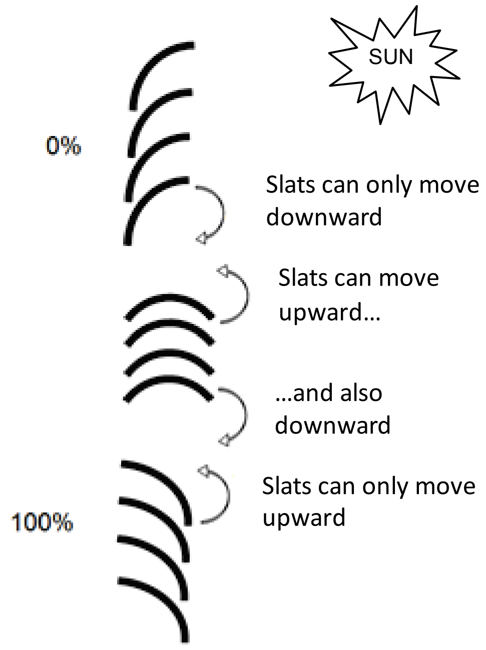
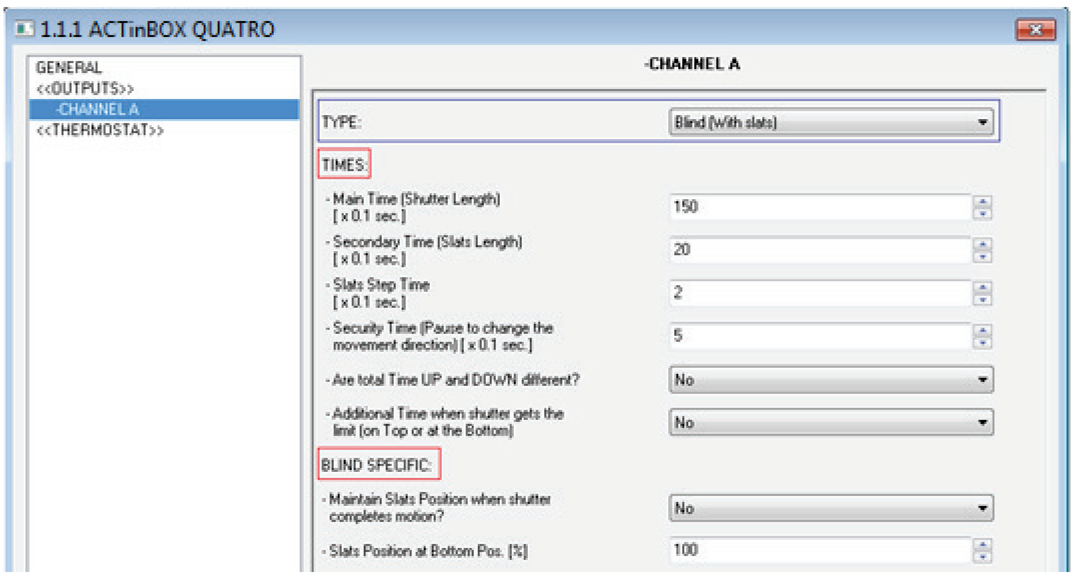


Figure 20 Slat positions

The ETS configuration screen for Blinds with slats is shown next, as well as the details about the available options:



1.1.1 ACTinBOX QUATRO

GENERAL
<<OUTPUTS>>
-CHANNEL A
<<THERMOSTAT>>

-CHANNEL A

TYPE:	Blind (With slats)
TIMES:	
- Main Time (Shutter Length) [x 0.1 sec.]	150
- Secondary Time (Slats Length) [x 0.1 sec.]	20
- Slats Step Time [x 0.1 sec.]	2
- Security Time (Pause to change the movement direction) [x 0.1 sec.]	5
- Are total Time UP and DOWN different?	No
- Additional Time when shutter gets the limit (on Top or at the Bottom)	No
BLIND SPECIFIC:	
- Maintain Slats Position when shutter completes motion?	No
- Slats Position at Bottom Pos. [%]	100

Figure 21 Blinds with slats. Configuration screen (ETS)

As shown in Figure 21, for this type of shutters multiple time parameters need to be parameterized. On the one hand, time parameters referred to the movement of the blind: "**Main Time (shutter length)**", "**Security time**", "**Are total time up and down different?**" and "**Additional time when shutter gets the limit**". All of them were already explained in Section 3.2.2 of this manual.

On the other hand, time parameters referred to the movement of the slats:

- **Secondary time (slats length)**: sets the time, in tenths of a second, the drive takes to perform a complete slat movement cycle from 0% (completely "up") to 100% (completely "down"), or vice versa. This time must be *manually* measured.

- **Slats Step Time**: sets the time, in tenths of a second, the drive will keep moving the slats every time it receives the order to perform an up/down step movement ("[CX] Stop/Step"=0 or 1, respectively), assuming that the blind is stopped. These step orders allow gradually rotating the slats and modifying their position (%), which may be very useful to prevent glare, for example after a change in the position of the sun.

  <u>Note</u>: *in case of joint-controlling slat step movements of multiple shutter channels together through the same group address, the time configured for this parameter is recommended to be* ***slightly greater than N*** *tenths of a second (where N is the amount of the channels that have been enabled and configured as "blind with slats") in order to ensure that rapidly sent consecutive orders are properly processed.*

<u>Note</u>: *times referred to slat movements must be shorter than those configured for blind movements (usual configuration).*

Besides defining these times, it will be necessary to configure the following options, which are specific for blinds with slats:

- **Maintain slats position when shutter completes motion?**: this option allows choosing whether the slats should recover their position after the blind reaches the desired position, or not.

<u>Example</u>:

Suppose the parameter "Maintain slats positioning when shutter completes motion?" has been enabled. The initial position of the slats is 50% and the initial position of the blind is 0% (up). An order to lower the blind is received, thus making the blind (and therefore the slats) start moving downwards, until the position of the blind reaches 100%. At this point, the blind has completed its movement, and ACTinBOX QUATRO will revert the position of the slats by additionally moving the drive until they reach the *position they had before (50%, in this example), thus causing a minor deviation in the position of the blind.*

If parameter "Maintain slats positioning when shutter completes motion?" had not been enabled, when the position of the blind reaches 100% (down), the slats will maintain their resulting position after the movement of the blind.

- **Slats position at Bottom pos. [%]**: allows establishing the slats position (in percentage) when the blind is completely closed (i.e., its position is equal to 100%).

  This means that, when the blind stops moving downwards and reaches 100%, the slats will correct their position to the one established by parameter.

Apart from these configuration options, it is necessary to define the "Slats specific position" for those enabled functions where particular positions are required to be parameterized. These functions are:

- **Scenes.** "Response: Specific position". The positions of the blind and the slats (in percentage) can be configured independently.
- **Alarms.** "Response: Specific position". The same as above.
- **Direct positioning.** Independent configuration of positions 1 or 2 (depending on the parameterized number), in percentage, of the blind and the slats.
- **Start-up configuration.** "Initial position: Specific position". The position percentages of the blind and slats can be configured independently.

To obtain further information about the configuration and options of the functions of the shutter channels, please refer to the section 3.2.2 of this manual.

ANNEX II. COMMUNICATION OBJECTS

Column RESET shows the values of the communication objects after a bus failure. However, this does not necessarily imply that such values are sent to the bus.

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES			NAME	FUNCTION
					RANGE	1ST TIME	RESET		
0	1 Byte	I	C - - W -	[18.1] DPT_SceneControl	0-63 (run) 128-191 (learn)	-	-	Scenes (Individual Outputs)	0-63(Run 1-64); 128-191(Learn)
1	1 Byte	I	C - - W -	[18.1] DPT_SceneControl	0-63 (run) 128-191 (learn)	-	-	Scenes (Shutter Channels)	0-63(Run 1-64); 128-191(Learn)
2	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 1	Binary Data Entry (0/1)
3	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 2	Binary Data Entry (0/1)
4	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 3	Binary Data Entry (0/1)
5	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 4	Binary Data Entry (0/1)
6	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 5	Binary Data Entry (0/1)
7	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 6	Binary Data Entry (0/1)
8	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 7	Binary Data Entry (0/1)
9	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 8	Binary Data Entry (0/1)
10	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 9	Binary Data Entry (0/1)
11	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 10	Binary Data Entry (0/1)
12	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 11	Binary Data Entry (0/1)
13	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 12	Binary Data Entry (0/1)
14	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 13	Binary Data Entry (0/1)
15	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 14	Binary Data Entry (0/1)
16	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 15	Binary Data Entry (0/1)
17	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 16	Binary Data Entry (0/1)
18	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 17	Binary Data Entry (0/1)
19	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 18	Binary Data Entry (0/1)
20	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 19	Binary Data Entry (0/1)
21	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 20	Binary Data Entry (0/1)
22	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 21	Binary Data Entry (0/1)

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES			NAME	FUNCTION
					RANGE	1ST TIME	RESET		
23	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 22	Binary Data Entry (0/1)
24	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 23	Binary Data Entry (0/1)
25	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 24	Binary Data Entry (0/1)
26	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 25	Binary Data Entry (0/1)
27	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 26	Binary Data Entry (0/1)
28	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 27	Binary Data Entry (0/1)
29	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 28	Binary Data Entry (0/1)
30	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 29	Binary Data Entry (0/1)
31	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 30	Binary Data Entry (0/1)
32	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 31	Binary Data Entry (0/1)
33	1 Bit	I	C - - W -	[1.002] DPT_Bool	0/1	-	Last	[LF] (1 bit) Data Entry 32	Binary Data Entry (0/1)
34	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 1	1 byte Data Entry (0-255)
35	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 2	1 byte Data Entry (0-255)
36	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 3	1 byte Data Entry (0-255)
37	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 4	1 byte Data Entry (0-255)
38	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 5	1 byte Data Entry (0-255)
39	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 6	1 byte Data Entry (0-255)
40	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 7	1 byte Data Entry (0-255)
41	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 8	1 byte Data Entry (0-255)
42	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 9	1 byte Data Entry (0-255)
43	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 10	1 byte Data Entry (0-255)
44	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 11	1 byte Data Entry (0-255)
45	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 12	1 byte Data Entry (0-255)
46	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 13	1 byte Data Entry (0-255)
47	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 14	1 byte Data Entry (0-255)
48	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 15	1 byte Data Entry (0-255)
49	1 Byte	I	C - - W -	[5.010] DPT_Value_1_Uncount	0-255	-	Last	[LF] (1 byte) Data Entry 16	1 byte Data Entry (0-255)
50	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 1	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
51	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 2	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES			NAME	FUNCTION
					RANGE	1ST TIME	RESET		
52	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 3	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
53	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 4	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
54	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 5	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
55	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 6	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
56	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 7	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
57	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 8	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
58	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 9	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
59	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 10	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
60	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 11	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
61	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 12	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
62	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 13	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
63	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 14	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
64	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 15	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
65	2 Bytes	I	C - - W -	[7.001] DPT_Value_2_Uncount	0 - 65535	-	Last	[LF] (2 bytes) Data Entry 16	2 bytes Data Entry
				[9.001] DPT_Value_Temp	0.00 - 120.00	-	Last		
66	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 1 RESULT (1 bit)	FUNCTION 1 Result
67	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 2 RESULT (1 bit)	FUNCTION 2 Result
68	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 3 RESULT (1 bit)	FUNCTION 3 Result

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES RANGE	1ST TIME	RESET	NAME	FUNCTION
69	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 4 RESULT (1 bit)	FUNCTION 4 Result
70	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 5 RESULT (1 bit)	FUNCTION 5 Result
71	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 6 RESULT (1 bit)	FUNCTION 6 Result
72	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 7 RESULT (1 bit)	FUNCTION 7 Result
73	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 8 RESULT (1 bit)	FUNCTION 8 Result
74	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 9 RESULT (1 bit)	FUNCTION 9 Result
75	1 Bit	O	C T R - -	[1.002] DPT_Bool	0/1	0	Last	[LF] Function 10 RESULT (1 bit)	FUNCTION 10 Result
76	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 1 RESULT (1 byte)	FUNCTION 1 Result
77	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 2 RESULT (1 byte)	FUNCTION 2 Result
78	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 3 RESULT (1 byte)	FUNCTION 3 Result
79	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 4 RESULT (1 byte)	FUNCTION 4 Result
80	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 5 RESULT (1 byte)	FUNCTION 5 Result
81	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 6 RESULT (1 byte)	FUNCTION 6 Result
82	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 7 RESULT (1 byte)	FUNCTION 7 Result
83	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 8 RESULT (1 byte)	FUNCTION 8 Result
84	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 9 RESULT (1 byte)	FUNCTION 9 Result
85	1 Byte	O	C T R - -	[5.010] DPT_Value_1_Uncount	0-255	0	Last	[LF] Function 10 RESULT (1 byte)	FUNCTION 10 Result
86	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 1 RESULT (2 bytes)	FUNCTION 1 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
87	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 2 RESULT (2 bytes)	FUNCTION 2 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
88	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 3 RESULT (2 bytes)	FUNCTION 3 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
89	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 4 RESULT (2 bytes)	FUNCTION 4 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
90	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 5 RESULT (2 bytes)	FUNCTION 5 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
91	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 6 RESULT (2 bytes)	FUNCTION 6 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
92	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 7 RESULT (2 bytes)	FUNCTION 7 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES			NAME	FUNCTION
					RANGE	1ST TIME	RESET		
93	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 8 RESULT (2 bytes)	FUNCTION 8 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
94	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 9 RESULT (2 bytes)	FUNCTION 9 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
95	2 Bytes	O	C T R - -	[7.001] DPT_Value_2_Uncount	0 - 65535	0	Last	[LF] Function 10 RESULT (2 bytes)	FUNCTION 10 Result
				[9.001] DPT_Value_Temp	0.00 - 120.00	0	Last		
96	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CA] Alarm	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CA] Alarm	1=Alarm; 0=No Alarm
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O1] On/Off	N.O. (0=Open Relay; 1=Close)
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O1] On/Off	N.C. (0=Close Relay; 1=Open)
97	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CA] Alarm 2	1=Alarm; 0=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CA] Alarm 2	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O2] On/Off	N.O. (0=Open Relay; 1=Close)
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O2] On/Off	N.C. (0=Close Relay; 1=Open)
98	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CB] Alarm	1=Alarm; 0=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CB] Alarm	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O3] On/Off	N.O. (0=Open Relay; 1=Close)
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O3] On/Off	N.C. (0=Close Relay; 1=Open)
99	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CB] Alarm 2	1=Alarm; 0=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[CB] Alarm 2	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O4] On/Off	N.O. (0=Open Relay; 1=Close)
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O4] On/Off	N.C. (0=Close Relay; 1=Open)
100	1 Bit	O	C T R - -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O1] Status	0=Output Off; 1=Output ON
101	1 Bit	O	C T R - -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O2] Status	0=Output Off; 1=Output ON
102	1 Bit	O	C T R - -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O3] Status	0=Output Off; 1=Output ON
103	1 Bit	O	C T R - -	[1.001] DPT_Switch	0/1	Parameteriz.	Parameteriz.	[O4] Status	0=Output Off; 1=Output ON
104	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CA] Save Position	1=Save Position; 0=No Action
	1 Bit	I	C - - W -	[1.003] DPT_Enable	0/1	0	Last	[O1] Lock	1=Lock; 0=Unlock
105	1 Bit	I	C - - W -	[1.003] DPT_Enable	0/1	0	Last	[O2] Lock	1=Lock; 0=Unlock
	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CA] Save Position 2	1=Save Position; 0=No Action
106	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CB] Save Position	1=Save Position; 0=No Action

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES			NAME	FUNCTION
					RANGE	1ST TIME	RESET		
	1 Bit	I	C - - W -	[1.003] DPT_Enable	0/1	0	Last	[O3] Lock	1=Lock; 0=Unlock
107	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CB] Save Position 2	1=Save Position; 0=No Action
	1 Bit	I	C - - W -	[1.003] DPT_Enable	0/1	0	Last	[O4] Lock	1=Lock; 0=Unlock
108	1 Bit	I	C - - W -	[1.010] DPT_Start	0/1	-	Last	[O1] Timer	0=to turn Off; 1=to turn ON
	1 Bit	I	C - - W -	[1.008] DPT_UpDown	0/1	-	Last	[CA] Move	0=Up Shutter; 1=Down Shutter
109	1 Bit	I	C - - W -	[1.010] DPT_Start	0/1	-	Last	[O2] Timer	0=to turn Off; 1=to turn ON
	1 Bit	I	C - - W -	[1.008] DPT_UpDown	0/1	-	Last	[CA] Reverse Movement	0=Down Shutter; 1=Up Shutter
110	1 Bit	I	C - - W -	[1.008] DPT_UpDown	0/1	-	Last	[CB] Move	0=Up Shutter; 1=Down Shutter
	1 Bit	I	C - - W -	[1.010] DPT_Start	0/1	-	Last	[O3] Timer	0=to turn Off; 1=to turn ON
111	1 Bit	I	C - - W -	[1.008] DPT_UpDown	0/1	-	Last	[CB] Reverse Movement	0=Down Shutter; 1=Up Shutter
	1 Bit	I	C - - W -	[1.010] DPT_Start	0/1	-	Last	[O4] Timer	0=to turn Off; 1=to turn ON
112	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CA] Direct Positioning	1=Go to Position; 0=No Action
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	-	Last	[O1] Flashing	1=Start Flashing; 0=End Flash.
113	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	-	Last	[O2] Flashing	1=Start Flashing; 0=End Flash.
	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CA] Direct Positioning 2	1=Go to Position; 0=No Action
114	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CB] Direct Positioning	1=Go to Position; 0=No Action
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	-	Last	[O3] Flashing	1=Start Flashing; 0=End Flash.
115	1 Bit	I	C - - W -	[1.17] DPT_Trigger	0/1	-	Last	[CB] Direct Positioning 2	1=Go to Position; 0=No Action
	1 Bit	I	C - - W -	[1.001] DPT_Switch	0/1	-	Last	[O4] Flashing	1=Start Flashing; 0=End Flash.
116	1 Bit	I	C - - W -	[1.7] DPT_Step	0/1	-	Last	[CA] Stop	0 or 1 = Stop Shutter
	1 Bit	I	C - - W -	[1.7] DPT_Step	0/1	-	Last	[CA] Stop/Step	0=Stop/StepUp; 1=Stop/StepDown
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O1] Alarm	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O1] Alarm	1=Alarm; 0=No Alarm
117	1 Bit	I	C - - W -	[1.003] DPT_Enable	0/1	0	Last	[CA] Lock	1=Lock; 0=Unlock
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O2] Alarm	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O2] Alarm	1=Alarm; 0=No Alarm
118	1 Bit	I	C - - W -	[1.7] DPT_Step	0/1	-	Last	[CB] Stop	0 or 1 = Stop Shutter
	1 Bit	I	C - - W -	[1.7] DPT_Step	0/1	-	Last	[CB] Stop/Step	0=Stop/StepUp; 1=Stop/StepDown
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O3] Alarm	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O3] Alarm	1=Alarm; 0=No Alarm
119	1 Bit	I	C - - W -	[1.003] DPT_Enable	0/1	0	Last	[CB] Lock	1=Lock; 0=Unlock

NUMBER	SIZE	I/O	FLAGS	DATA TYPE (DPT)	VALUES			NAME	FUNCTION
					RANGE	1ST TIME	RESET		
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O4] Alarm	0=Alarm; 1=No Alarm
	1 Bit	I	C - - W -	[1.5] DPT_Alarm	0/1	-	Last	[O4] Alarm	1=Alarm; 0=No Alarm
120	1 Byte	O	C T R - -	[5.001] DPT_Scaling	0-255	0	Last	[CA] Current Slats Position	0=0%=Open; 255=100%=Closed
121	1 Byte	O	C T R - -	[5.001] DPT_Scaling	0-255	0	Last	[CB] Current Slats Position	0=0%=Open; 255=100%=Closed
122	1 Byte	I	C - - W -	[5.001] DPT_Scaling	0-255	-	Last	[CA] Slats Positioning	0=0%=Open; 255=100%=Closed
123	1 Byte	I	C - - W -	[5.001] DPT_Scaling	0-255	-	Last	[CB] Slats Positioning	0=0%=Open; 255=100%=Closed
124	1 Bit	I	C - - W -	[1.16] DPT_Ack	0/1	0	Last	[O1] Unfreeze Alarm	Alarm=0 + Unf.=1 -> End Alarm
125	1 Bit	I	C - - W -	[1.16] DPT_Ack	0/1	0	Last	[O2] Unfreeze Alarm	Alarm=0 + Unf.=1 -> End Alarm
126	1 Bit	I	C - - W -	[1.16] DPT_Ack	0/1	0	Last	[O3] Unfreeze Alarm	Alarm=0 + Unf.=1 -> End Alarm
127	1 Bit	I	C - - W -	[1.16] DPT_Ack	0/1	0	Last	[O4] Unfreeze Alarm	Alarm=0 + Unf.=1 -> End Alarm
128	1 Byte	O	C T R - -	[5.001] DPT_Scaling	0-255	0	Last	[CA] Current Shutter Position	0=0%=Top; 255=100%=Bottom
129	1 Byte	O	C T R - -	[5.001] DPT_Scaling	0-255	0	Last	[CB] Current Shutter Position	0=0%=Top; 255=100%=Bottom
130	1 Byte	I	C - - W -	[5.001] DPT_Scaling	0-255	0	Last	[CA] Shutter Positioning	0=0%=Top; 255=100%=Bottom
131	1 Byte	I	C - - W -	[5.001] DPT_Scaling	0-255	0	Last	[CB] Shutter Positioning	0=0%=Top; 255=100%=Bottom
132	1 Bit	I	C - - W -	[1.16] DPT_Ack	0/1	0	Last	[CA] Unfreeze Alarm	Alarm=0 + Unf.=1 -> End Alarm
133	1 Bit	I	C - - W -	[1.16] DPT_Ack	0/1	0	Last	[CB] Unfreeze Alarm	Alarm=0 + Unf.=1 -> End Alarm
134	1 Bit		C T - - -	[1.001] DPT_Switch	0	0	0	Reset 0	Voltage Recovery->Sending of 0
135	1 Bit		C T - - -	[1.001] DPT_Switch	1	1	1	Reset 1	Voltage Recovery->Sending of 1

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