PRODUCT MANUAL

ACTINBOX

CLASSIC-HYBRID

ACTinBOX Classic-Hybrid KNX

ZN1I0-AB46A

Edition 2

Version Classic-Hybrid 1.3

Index

1. INTRODUCTION

1.1 PRODUCT

The **ACTinBOX Classic-Hybrid** is a KNX actuator that combines in a same device:

- 4 x 10A multifunction (INDIVIDUAL or SHUTTER) binary **OUTPUTS**.
- **4 Voltage free contact INPUTS** to connect sensors and/or push buttons.
- **1 INPUT** customizable as **binary input or temperature probe**. There is a thermostat associated to this temperature probe.
- **1 INPUT** customizable as **binary input or motion sensor**.
- Advanced **LOGICAL FUNCTIONS** included.

These sections work independently, and can interact the others as if there were different autonomous devices connected to the **KNX Bus**.

1.2 COMMUNICATION OBJECTS

The **Communication Objects** of **ACTinBOXClassic-Hybrid** are divided into four different sections:

- Individual Outputs
- Inputs
- Shutter channels
- Logical functions

Nomenclature:

To easily find the appropriate Communication Objects during the Group Addressing process, these are named depending on the Section they belong to, as:
["Group Type"] "Function to be executed"

Following abbreviations are associated to the different sections:

- Individual Outputs:

  [O1] → Output 1
  [O2] → Output 2

[O3] → Output 3
[O4] → Output 4

- Inputs:

  [I1] → Input 1
  ...
  [I6] → Input 6

- Shutter Channels

  [CA] → Channel A
  [CB] → Channel B

- Logical Functions

  [LF] → Logical Function

Examples (See Figure 1)

- **[O3] Status:** Output 3 Status object
- **[CA] Stop:** Channel A Shutter Stop Control
- **[I3] Block:** Object to block input 3 control
- **[LF] FUNCTION 5 RESULT (1 bit):** Object to store the 1 bit Logical Function 5 result.

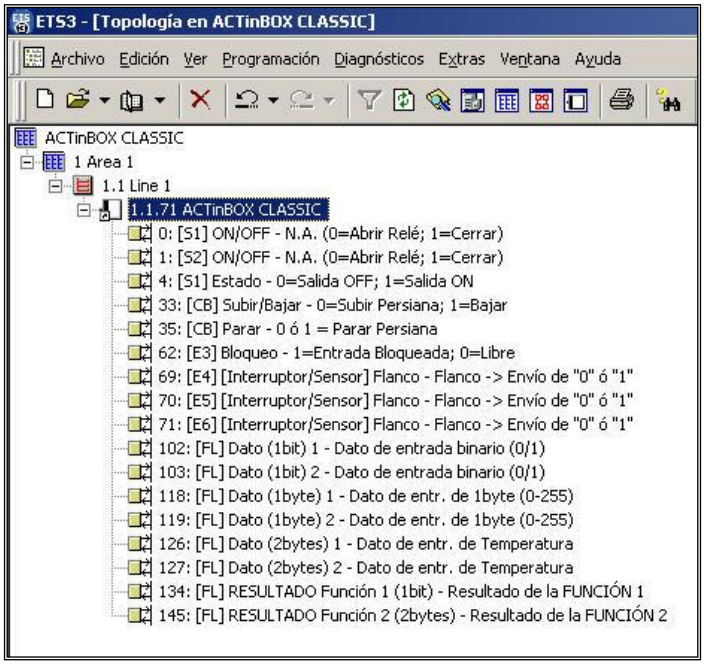

Figure 1. Communication Objects

2. OUTPUTS

The **ACTinBOX Classic-Hybrid** has **four 10A binary outputs** (see product Datasheet). These 4 outputs are divided into **2 groups (called Channels) with 2 outputs each**.

- **Channel A:** "Output 1" y "Output 2"
- **Channel B:** "Output 3" y "Output 4"

Both channels can be independently parameterized as:

- **Individual Outputs:** Every output is independent from the others, so all of them are independently managed. To control electrical loads…
- **Shutter Channels:** Outputs in the same channel are jointly managed. To control shutter drives, sun blinds or similar…

  In the shutter channel case:

  - **Output 1 (3 in Channel B):** In charge of raising the shutter.
  - **Output 2 (4 in Channel B):** In charge of lowering the shutter.

Example: *Consider a facility where we need to manage one shutter drive and a light spot.*

In this case, the ***ACTinBOX Classic-Hybrid*** *could be parameterized as follows:*

Channel A = Shutter Channel
Channel B = Individual Outputs
Output 3 = normally open
Output 4 = Disabled
Salida 4 = Deshabilitada

2.1 SHUTTER CHANNELS

The **ACTinBOX Classic-Hybrid** allows installing any type of shutter drive control (or similar) on its Channels. To operate them, two basic control objects (**"Move"** & **"Stop/Step"**) together with a set of additional functions (with their own Communications objects) are available.

The Basic shutter control with the above mentioned Communications objects is made as follows:

- **Raise Shutter:** Send value "**0**" to the object "**Move**".

- **Lower Shutter:** Send value "**1**" to the object "**Move**".

Take into account I: When the object "Move" gets a "0" or a "1", the shutter will start moving, and won't stop unless it reaches its lowest or highest position (depending on the order received), or that receives some other order cancelling the previous *one*

Take into account II: After programming the device with the ETS, the ACTinBOX considers the shutter is completely raised

- **To stop a shutter in motion:** A "**0**" or a "**1**" must be sent to the object "**Stop**".

2.1.1 TYPE

- **ROLLER SHUTTER (No lamellas) / SUNBLINDS**: These are the typical revolving shutters, with a simple (Raise/Lower) movement.

- **SHUTTER WITH LAMELLAS:** Special shutters with a secondary movement managed by the same drive.

  The **ACTinBOX Classic-Hybrid,** in this case, allows controlling both movements, lamella rotation (getting more or less incident light from the outside), and the Raise/Lower movement.

  By selecting this Shutter control, the **"Stop"** object is replaced with the **"Stop/Step"** one.

  This way, if the device receives a **"0"** or a **"1"** via this object when the shutter is in motion, it shall stop; when the shutter is stopped, receiving a **"0"** via this communication object will make lamellas to pull up; on the opposite, receiving a "**1"** will make lamellas to pull down.

2.1.2 TIMES

It is necessary to set two different times (three when working with shutters with moveable lamellas) for a proper channel working.

- **Main Time (Shutter Height)**: This is the time the shutter needs to cover its height completely. Both times can be used in this field (Raising time or Lowering time); but if there was some difference between these two times, "Lowering time" will be considered as "master", and should be used to fill in this field. "Raising time" in this case will be set in the "Total Time up" field, enabled for this purpose. This variable won't need to be periodically calibrated since the exact shutter position remains on the ACTinBOX (even when Power Failure occurs).

- **Secondary Time (Lamellas movement):** (Only for Lamella Shutters) Time used by lamellas to cover a complete deployment (up or down).

- **Security Time (before changing the movement direction):** This time is applied by the actuator to protect the drive when the movement direction of the shutter is changed. If the device receives the order to **"Lower"** the shutter while this is being raised, the ACTinBOX Classic-Hybrid will stop it for a while (security time), to later **"Lower"** it. It is recommended to keep the default value in this field: 5 [x 0.1 sec].

- **Different Up/Down Times?** : Whenever the shutter raising and lowering times are different (i.e. heavy shutters), this field should be dropped down to set the raising shutter time in this field, as mentioned above the lowering one must be set in the **"Main Time field"**.

Example: *Shutter in Channel B takes 15 seconds to be lowered and 20 to be raised. In this case the ACTinBOX Classic-Hybrid parameterization should remain as follows:*

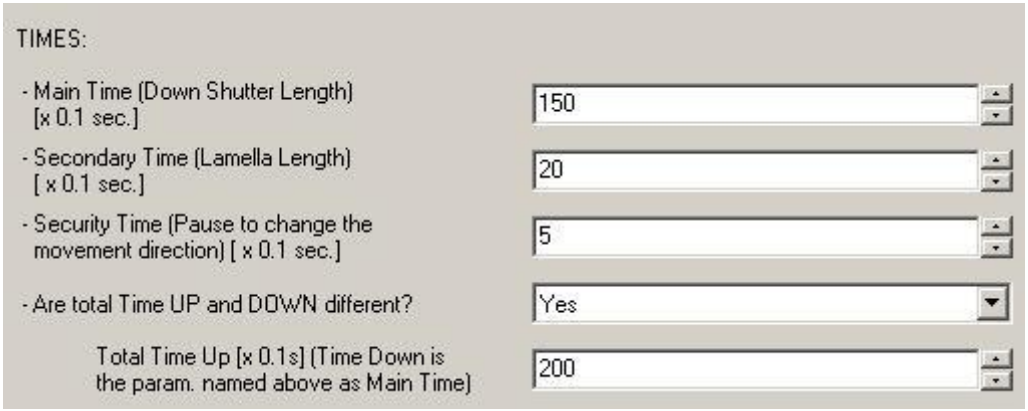

Figure 2. Time up and Time down

- **Additional time when shutter gets the limit:** This parameter guarantees the shutter always gets its lowest or highest level by setting an extra time for the drive to keep moving once the shutter took up its raising/lowering times, preventing small maladjustments.

2.1.3 FUNCIONES

Following parameters add functionality or special features to a Shutter Channel Control:

STATUS OBJECT

This function provides a communication object "Current Position", to indicate the user the exact position the shutter is at all times.

This is a 1 byte object measured in "%". The object value is 0% when shutter is completely raised and 100% when shutter is completely lowered. The other values correspond to the different shutter positions between up and down.

When the shutter is in motion, and eventually stops, the ACTinBOX Classic-Hybrid can send (via this Object) the exact position of the shutter to update the rest of devices in the Bus when needed.

Take into account: The shutter "**Current Position**" status object has been programmed so that every time the shutter is in motion, this is transmitted to the bus to update its position in real time (every second).

PRECISE CONTROL

This function makes possible to move the shutter to any position on its length, via the 1 byte "**Positioning**" communication object.

Every time the **ACTinBOX Classic-Hybrid** gets a new value through this object (e.g. 50%), the shutter is moved to the corresponding position (the middle in the example).

SCENES

This function makes possible to use a standard (1 byte) scene object to control shutters, giving the users the possibility to choose a precise position where to locate the shutter depending on the scene number received by the **ACTinBOX Classic-Hybrid** through the object "**Scenes**".

- **Total Scenes**: To choose the total number of scenes to be used; up to a maximum of five scenes can be set in this field.
- **Scene**: Set the scene number.
- **Response**: To set the precise position where to locate the shutter when the corresponding scene number is called from the Bus.

Example: *Consider a facility where 4 scenes will be used (**4, 6, 8 & 9**), but only three of them will be controlled from the ACTinBOX Classic-Hybrid to locate the shutter in a precise position:*

- ***Scene 4*** → *Up*
- ***Scene 6*** → *Down*
- ***Scene 8*** → *In the middle (50%)*

Parameterization in this case should remain as follows:

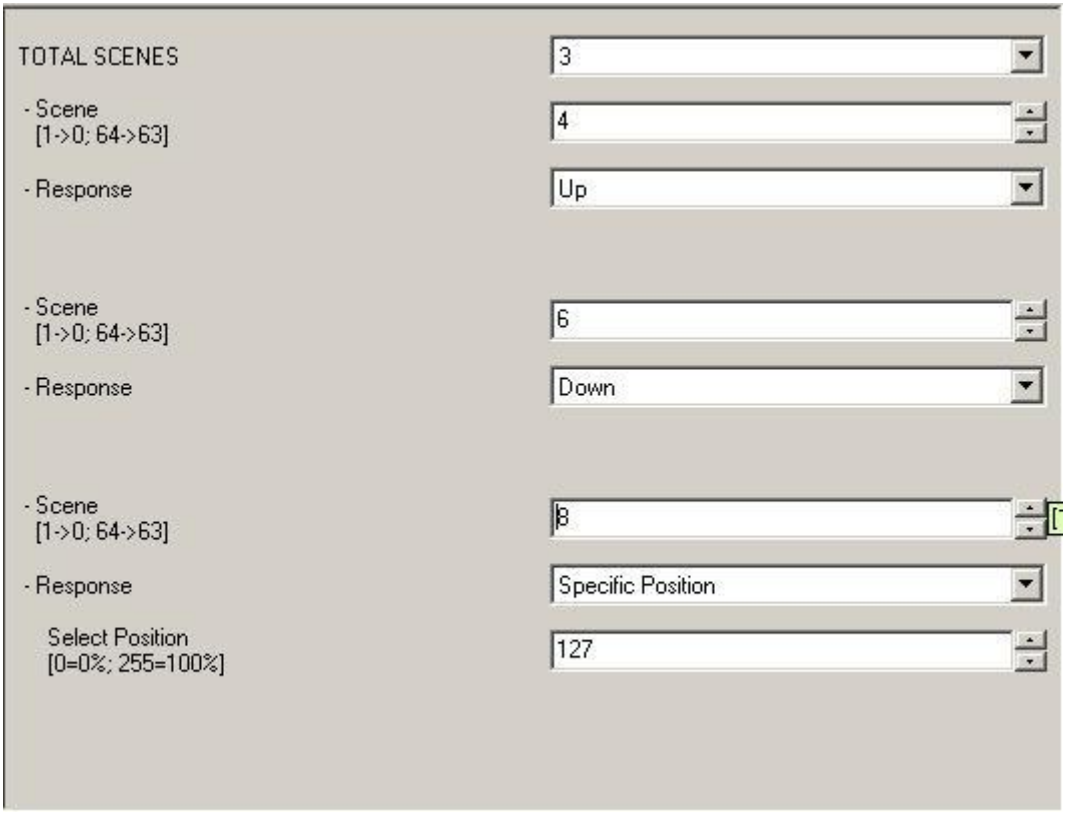

Figure 3. Scenes parameterization

BLOCK

This function makes possible to "**block**" an output by disabling its control.

> **Take into account:** Only the Alarm function is higher priority than the block one. This means that , if shutter in the channel is blocked and the Alarm goes On, the shutter shall be carried to the Alarm parameterized position. When Alarm goes off, the shutter will recover its blocked status.
>
> To unblock the shutter it's necessary to send a **"0"** to the object **"Block"**.

An output will be blocked by sending a "**1**" to the corresponding object in the channel. The Output is enabled again by sending a "**0**".

ALARM

This function is designed for cases in which the ACTinBOX Classic-Hybrid must response to an alarm situation. When an alarm occurs, this function locates the shutter in the parameterized position, and after this, shutter is blocked (it won´t be controlable until Alarm goes off).

- **Number of alarms:** Set whether to use one or two alarms. Both of them

> **Take into account:** "Alarm 1" is higher priority than "Alarm 2". This means that if channel is in "Alarm 2" status and "Alarm 1" occurs the shutter will change to "Alarm 1" status and it shall not come back to "Alarm 2" until "Alarm 1" goes off. Whereas if the Channel is in "Alarm 1" status, and "Alarm 2" occurs , "Alarm 1" prevails.

can be independently managed through their corresponding communication objects.

- **Trigger Value:** Set the value to activate the Alarm status. An Alarm status will be activated when the value set in this field is sent to the object Alarm (or Alarm 2).

  The opposite of the **"trigger"** value is the **"Passive"** value.

- **Cyclical Monitoring:** To be sure that the sensor works properly and that no alarm is active, the sensor to send the **"Trigger Value"** to the ACTinBOX Classic-Hybrid, shall continuously send the **"Passive Value"** to the Bus when there is no alarm active. It is in this case that this parameter should be enabled. if the ACTinBOX Classic-Hybrid doesn't get the **"passive value"** during the parameterized

  - **Cycle Time:** alarm will be automatically activated if the ACTinBOX Classic-Hybrid doesn't get the **"passive value"** during this time.

Take into account: It's recommended to set a time higher than double sensor cycle time, to avoid alarm frames missing.

- **Response:** Set the response of the actuator channel output when the alarm is activated.

- **Deactivation:** Two different methods to deactivate an active Alarm:

  - **Normal:** By sending the **"Passive value"** (opposite to the Trigger one) to the object **"Alarm"**.

  - **Frozen:** Consists in applying the normal method to later send a **"1"** to the object "**Unfreeze alarm**". This method makes the channel output remains blocked even when the alarm status is finished; in this case it will be necessary then that the output is manually enabled from another point in the installation.

- **End:** This parameter sets the output response when the alarm status is finished.

REVERSED MOVING

This function makes possible to control shutters the other way around from usual; this means **"1"** to raise the shutter and **"0"** to lower it. This control is

made through the object “Reversed Moving” and is compatible with the usual control.

This is really useful when a “Central Off” is required in the installation, i.e. to turn the lights off & lower the shutters. In this case, a “0” should be sent to the light "ON/OFF” objects, and to the shutter “Reversed moving” objects.

DIRECT POSITIONING

Function to move a shutter to a prefixed specific position via 1 bit objects.

When value “**1**” is received through one of these objects (“**Direct positioning**” or “**Direct positioning 2**”), the shutter will be moved to the parameterized position.

- **Total direct positioning:** Set the number of direct positioning to be used.
- **Position:** Choose the exact position to move the shutter to (Remember that 0=0% and 255=100%).
- **New positions saving:** Set whether to allow or not new positions saving:
  - **Save Position 1**
  - **Save Position 2**

  A "**1**" must be sent to these objects in order to save new positions.

START UP CONFIGURATION

This function is meant to define the behaviour of the shutter channel output after a Bus Power Failure, or after programming the device with the ETS.

- **Initial position:** This field is to define the exact position the shutter should be located after a Bus Power Failure. After programming the device with the ETS, **“current position”** option means the shutter will remain the position it was before de programming.

> **Take into account:** The initial position is always sent through the object "Current Position".

- **Update:** By enabling this field, the output initial status can be sent to the Bus to update the rest of devices in the installation if needed.
- **Start up sending delay:** As some devices in the installation may take longer to restart after a Power Failure, this field allows setting a delay in seconds for the initial configuration to be sent to the Bus, in order to ensure

the rest of devices in the installations are ready to receive the corresponding telegrams.

2.2 INDIVIDUAL OUTPUTS

When this option is selected, both outputs in the channel are completely independent from each other.

2.2.1 TYPE

It is necessary to indicate whether the outputs are "**Normally open**" or "**Normally closed**":

- Normally Open: **ON**→Close Relay **OFF**→Open Relay
- Normally Closed: **ON**→ Open Relay **OFF**→ Close Relay

2.2.2 FUNCTIONS

Following parameters add functionality or special features to Shutter Channels Control.

STATUS OBJECT

The "**status**" object always shows the current status output, and is meant to feedback any other device in the installation when needed.

- Output **ON** -> status output = "**1**".
- Output **OFF** -> status output = "**0**".

This object will be sent to the Bus every time the output status changes.

TIMERS

This section is meant to control the outputs by mean of a timer.

Two different timers can be selected: **Simple Timer** and **Flashing.** They work independently from each other as from the "On/Off" control, since these are all managed from three different Communications objects.

By sending an ON order to the object “Timer”, a scheduled ON begins. If before the time for the ON comes to its end a new OFF order is sent to the object “ON/OFF”, the output will turn off finishing the previous set timer.

Their parameterization is like follows:

- **Simple timer:** This is applied in the output when an ON or OFF order is received through the object **“Timer”**.
  - **On Delay:** Time to pass since the ON order is sent (through the object **“Timer”)** and the (ON) response in the output takes place.
  - **Off Delay:** Time to pass since the OFF order is sent (through the object **“Timer”)** and the (OFF) response in the output takes place.
  - **On Duration:** Time the output remains ON before recovering the OFF status. A “**0**” set in this field means the Output will remain always ON, no timing is applied in the output.

Take into account: Simple timer funcioning is detailed next:

- By sending a “1” to the object “Timer”, an ON order will be sent to the corresponding output after de “On Delay” and after the “On Duration” time, an OFF order will be sent.
- By sending a “0” to the object ”Timer”, an OFF order will be sent to the corresponding output with the “Off Delay” .

  - **Multiply:** Consists in multiplying a timer as many times as a “**1**”or a “**0**” is received through the object “**Timer**”.

Take into account: The way the multiply parameter works is detailed next:

- No Multiply: When value “1” is received by object “Timer” while a simple ON timer is still running, the ACTinBOX Classic-Hybrid resets its counter to 0 to restart counting again. The same happens with “0” and timer off.
- Multiply: When value “1” is received by object “Timer”while a simple ON timer is still running, the ACTinBOX Classic-Hybrid will count double time. If another “1” is received before the ON timer ends, the ACTinBOX Classic-Hybrid will count triple time and so on…. The same happens with “0” and timer off.

- **Flashing:** This function is meant for the output to run the sequence ON-OFF-ON-OFF…. when needed
  - **On Duration:** Set the On duration time in the sequence.

- **Off Duration:** Set the off duration time in the sequence.
- **Repetitions:** Set the number of repetitions in the sequence. For an unlimited value, set **"0"** in this field.
- **Status after last repetition**: To set the output status after the last repetition in the sequence took place.

Take into account: Flashing starts by sending a "**1**" to the object "**Flashing**", and stops by sending a **"0"**.

SCENES

This function makes possible to use a standard (1 byte) scene object to control the outputs.

- **Number of scenes:** To choose the total number of scenes to be used; up to a maximum of five scenes can be set in this field.
- **Scene:** Set the Scene number.
- **Response:** To set the exact output response (On/Off) when the corresponding scene number is called from the Bus.

Example: *Consider a facility where 4 scenes will be used (**4, 6, 8 & 9**), but only three of the outputs willl be controlled from the ACTinBOX Classic-Hybrid*

*- **Scene 4** → ON*
*- **Scene 6** → ON*
*- **Scene 8** → OFF*

Parameterization of the device in this case should remain as follows

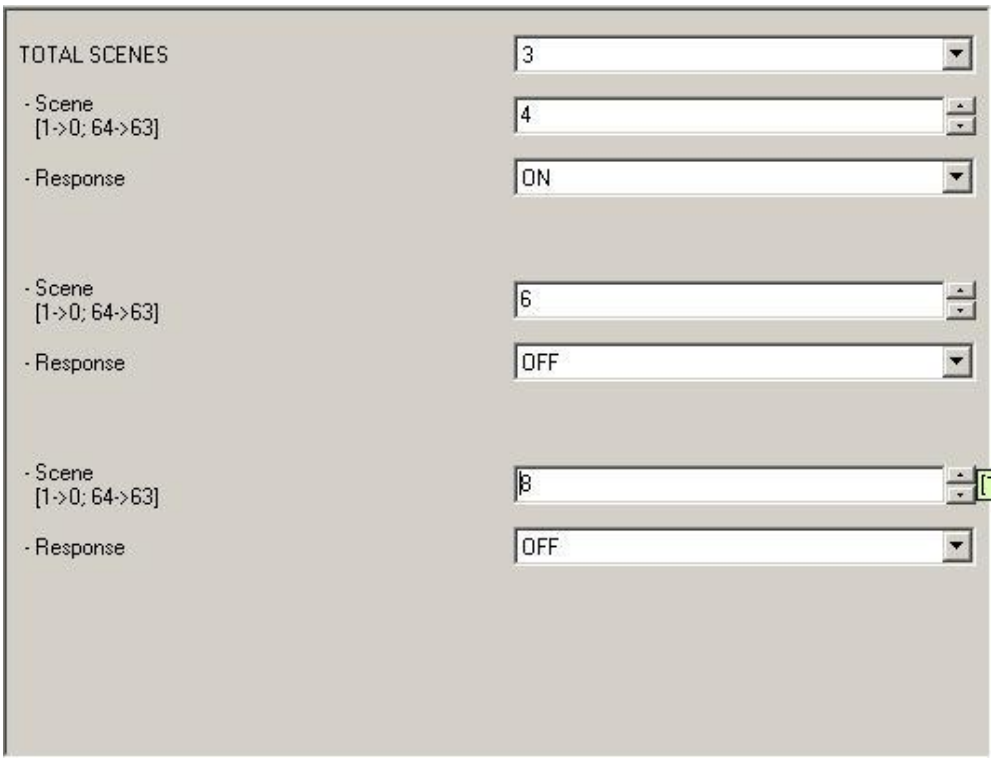

Figure 4. Scenes Parameterization

BLOCK

This function makes possible to “block” an output by disabling its control.

Output will be blocked by sending a “1” to the corresponding object in the channel.

Take into account: Only the Alarm function is higher priority than the block one. This means that , if output in the channel is blocked and the Alarm goes On, the output shall be carried to the Alarm parameterized position. When Alarm goes off, the output is blocked again.

To unblock the output it's necessary to send a **“0”** to the object **“Block”**.

ALARM

This function is designed for cases in which the ACTinBOX Classic-Hybrid must response to an alarm situation. When an alarm occurs, this function sets the output in the previously parameterized position, and after this, output is blocked (it won’t be controllable until Alarm goes off).

- **Trigger Value:** Set the value to activate the Alarm status. An Alarm status will be activated when the value set in this field is sent to the object Alarm (or Alarm 2). The opposite of the **"trigger"** value is the **"Passive"** value.

- **Cyclical Monitoring:** To be sure that the sensor works properly and the alarm is not activated, the sensor to send the **"Trigger Value"** to the ACTinBOX Hybrid, shall continuously send the **"Passive Value"** to the Bus when there is no alarm active. It is in this case that this parameter should be enabled; this way, if the ACTinBOX Classic-Hybrid doesn´t get the **"passive value"** during the parameterized.

- **Cycle Time:** alarm will be automatically activated

Take into account: It's recommended to set a time higher than double sensor cycle time, to avoid alarm frames missing.

- **Response** Set the response of the actuator channel output when the alarm is activated. When the **"Flashing"** response is set in this field, new parameters will appear in the ETS parameterization environment:

  - **ON Duration:** Set the On duration time in the sequence.

  - **OFF Duration:** Set the Off duration time in the sequence.

  - **Repetitions:** Set the number of repetitions in the sequence. For an unlimited value, set **"0"** in this field.

  - **Status after last repetition:** To set the output status after the last repetition in the sequence took place.

- **Deactivation:** Two different methods to activate an alarm:

  - **Normal:** By sending the **"Passive value"** (opposite to the Trigger one) to the object **"Alarm".**

  - **Frozen:** Consists in applying the normal method to later send a **"1"** to the object "**Unfreeze alarm**". This method makes the channel output remains blocked even when the alarm status is finished; in this case it will be necessary then that the output is manually enabled from another point in the installation.

- **End:** This parameter sets the output response when the alarm status is finished.

START UP CONFIGURATION

This function is meant to define the behaviour (ON/OFF) of the channel output after a Bus Power Failure, or after programming the device with the ETS.

- **Initial Position:** This field is to define the exact initial position for the channel output after a Bus Power Failure. After programming the device with the ETS, option **"current position"** means output OFF (relay open) for normally open outputs, and output OFF (relay closed) for normally closed outputs.

- **Update:** By enabling this field, the output initial status signal can be sent to the Bus to feedback the rest of devices in the installation when needed.

- **Start Up Sending Delay:** As some devices in the installation may take longer to restart after a Power Failure, this field allows setting a delay in seconds for the initial configuration to be sent to the Bus, in order to ensure the rest of devices in the installation are ready to receive the corresponding telegrams.

3. INPUTS

The **ACTinBOX Classic-Hybrid** has **6 Inputs:**

- **Voltage Free Contact Inputs:** Input 1, 2, 3, 4, 5 and 6. They can be connected to a “Push Button” or a “Switch/Sensor”.
- **Analogical input for Temperature Probe:** input 5 can be connected to a Zennio Temperature Probe and it has an optional thermostat function.
- **Analogical/Digital input for Motion Sensor:** input 6 can be connected to a Zennio Motion Sensor.

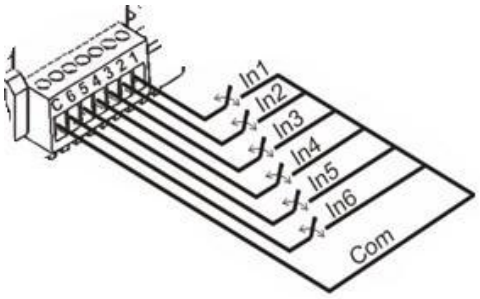

Figure 5. Terminal block with binary inputs

Following functions can be selected through a **Push Button input type**:

- **0 / 1 Sending:** To send “1 bit” values to the Bus.
- **Shutter Control:** This function results on sending a **“1 bit”** object to the Bus in order to control shutters.
- **Dimmer Control:** This function results on sending a **“4 bits”** Dimming Control Object to the Bus.
- **Scene Sending:** This function results on sending a **“1 byte”** Scene Control Object to the Bus; a scene on the Bus, can be then managed from the input through this Object.

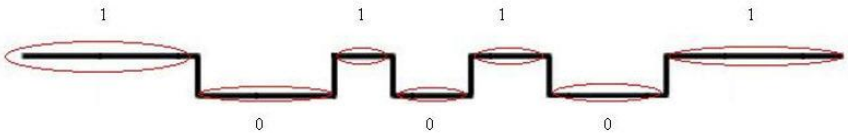

Figure 6. Push button Status

It is possible for the ACTinBOX Classic-Hybrid to carry out a function with a short press and another different function with a long press. The ACTinBOX Classic-Hybrid then can control up to 12 different functions through its inputs.

Example: *Input 3 can control a light with a short press and run a scene with a long press.*

For a **sensor input type** the following function is available:

- **0 / 1 Sending:** For every rising/falling edge detected on an input, user can select whether to send a **"0"**, a **"1"**, or an alternative switching of **“0”** and **“1”**

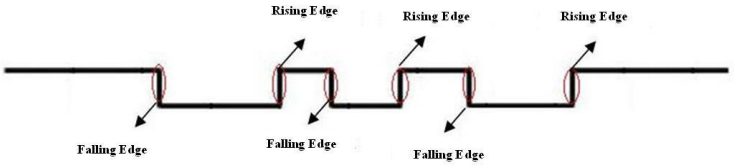

Figure 7. Sensor Status

For a **Temperature Probe** the following functions are available:

- **Temperature Sensor**
- **Temperature Sensor and Thermostat.**

For a **Motion Sensor** the following functions are available:

- **0/1 Sending:** for every status of the motion sensor (detection and no detection), the user can select whether to send **"0"** or **"1”.**
- **Scene Sending:** for every status of the motion sensor (detection and no detection), the user can select a scene number to be sent.
- **Luminosity Sending**: allows the sending of the detected luminosity level in level from 0 to 100.
- **Short circuit**: indicates that a short circuit has happened between input 6 and C, which means that the motion sensor won’t work properly.
- **Open circuit:** indicates that an open circuit has happened between input 6 and C, which means that the motion sensor won’t work properly.

3.1 PUSH BUTTON

A Push Button connected to an input consists of an electrical mechanism which keeps its contacts open under normal conditions. While keeping it pressed, contacts remain closed to later recover its normal position when released.

Depending on the Threshold Time, two different actions can be distinguished:

- **Shot Press**
- **Long Press**

Take into account: Normally closed Push buttons cannot be connected to the ACTinBOX inputs.

3.1.1 . “0/1” FUNCTION

This function results on sending 1 bit to the Bus:

- **Response:** This parameter sets whether the value sent is **“0”**, **“1”** or an alternative switching between **"0"** y **"1"**.
- **Cyclical Response Sending:** Users can ask the ACTinBOX Classic-Hybrid to periodically send **"0"** or **"1"**, or even both (when “always” is selected).
  - **Cycle Time:** Set the time between two consecutive telegrams when using the Cyclical Sending.

Take into account: The Internal Links in ACTinBOX Classic have been removed, since the new version 5.0 of the Operating System includes full dúplex.

3.1.2 SHUTTER FUNCTION

This function results on sending 1 bit to the Bus in order to control shutters

- **Response**: The corresponding control object can be used to:
  - **Raise:** Raise the shutter. “0” is sent to the Bus.
  - **Lower:** Lower the shutter. “1” is sent to the Bus.
  - **Raise/Lower (toggle switch):** Alternative switching between the Raise/Lower orders (to manage the shutter with an only input).
  - **Stop/Step Up:** Stops the shutter (“0” is sent to the Bus). When talking about shutters with lamellas, this mode also allows users to control them by pulling lamellas a step up.
  - **Stop/Step Down:** Stops the shutter (“1” is sent to the Bus). When talking about shutters with lamellas, this mode also allows the user to control them by pulling lamellas a step down.
  - **Stop/Step (toggle switch):** Stops the shutter. When talking about shutters with lamellas, this mode also allows users to control them; this parameter alternatively switches the lamellas up/down steps.

Take into account I: When up/down (toggle) option is selected for a short press, the user will not be able to stop the shutter with another short press on the same input.

3.1.3 DIMMER FUNCTION

This function results on sending a **(4 bits) Dimming Control Object** to the Bus.

- **Response:** Depending on the chosen option, the Control Object may be:
  - **Light ON:** Turn the light ON. “1” is sent to the Bus.
  - **Light OFF:** Turn the light OFF. “0” is sent to the Bus
  - **Light ON/OFF (toggle):** Alternative switching between the ON/OFF orders (to manage the lighting level with an only input).
  - **Brighter:** Depending on the “Dimming Step” set, every press on the input will make the brightness level go up. The first short press on the input will increase the brightness level ; a second press stops the “Increase”
  - **Darker:** Depending on the “Dimming Step” set, every press on the input will make the brightness level go down The first short press on the input will decrease the brightness level ; a second press stops the “Decrease”
  - **Brighter/Darker (toggle):** Alternative switching between the Brighter/Darker orders.
  - **Dimming Step:** Depending on the value selected, different brightness levels are offered. Once the “Dimmer Control” option is selected, it is necessary to set this parameter to fix the increasing/decreasing percentage level in every dimming step.

Dimming Step	Necessary button presses for a complete regulation (0 – 100%)
6. 100%	1
5. 50%	2
4. 25%	4
3. 12.5%	8
2. 6.25%	16
1. 3.1%	32
0. 1.5%	64

3.1.4 SCENE FUNCTION

This function results on sending a (1 byte) Scene Control Object to the Bus; a scene on the Bus can be managed with an input through this object.

- **Response:** Choose whether the scene will be “Run” or “Saved”.
- **Scene:** This parameter identifies the scene to Run/Save with the corresponding input.

Take into account: The Internal Links in ACTinBOX Classic have been removed, since the new version 5.0 of the Operating System includes full dúplex.

3.1.5 ADDITIONAL CONFIGURATION

- **Threshold Time:** This parameter defines the time limit where a short press turns into a long press. If a press on the display ends before the long press time, then it is a short press. This value must be set with precision to tenths of a second (i.e. to get “0.5” seconds, set “5”)
- **Response Delay:** This parameter sets the time to wait for the object to be sent to the Bus since the action on the input took place. This value must be set with precision to tenths of a second (i.e. to get “1” second, set “10”).

  Set value “0” in this field in order to get an immediate sending (no delay).
- **Block:** By selecting “Yes” on this field, a new Communications object “Block” appears on the ETS. This object can be used to disable an input:
  - On receiving a **“1”** through this object, the ACTinBOX Classic-Hybrid will be ignoring any press on the input.
  - On receiving a **“0”** through this object, the input recovers its enabled status, so that it is ready again to receive orders. Actions taken on the input while being disabled will not be taken into account.

3.2 SWITCH/SENSOR

A **Switch/Sensor** connected to an input, consists of an electrical mechanism which may have its **contacts open or closed** under normal conditions. These mechanisms don’t recover their normal position automatically as with the push button.

A transition of a digital signal from low/high/low is called "**Edge**":

- **Falling Edge:** Closed contact to Open Contact.
- **Rising Edge:** Open contact to Closed Contact.

By selecting a Switch/Sensor input, “[Switch/Sensor] Edge” Communication object will be sent to the Bus every time a rising/falling Edge is detected

- **Falling Edge:** Set the value to be sent to the Bus in the transition of the digital signal from high to low
- **Rising Edge:** Set the value to be sent to the Bus in the transition of the digital signal from low to high
- **Sending of “0” delay:** Time for the ACTinBOX Classic-Hybrid to wait before sending value “0” through the “[Switch/Sensor] Edge” communication object when this value has been detected on an incoming edge.
- **Sending of “1” delay:** Time for the ACTinBOX Classic-Hybrid to wait before sending value “1” through the “[Switch/Sensor] Edge” communication object when this value has been detected on an incoming edge.
- **Periodical Sending of “0”:** Set a period of time to cyclically send value “0” to the Bus when object “[Switch/Sensor] Edge” detects this value on an incoming edge. If not cyclically sending is needed, please select value “0” in this field.
- **Periodical Sending of “1”:** Set a period of time to cyclically send value “1” to the Bus when object “[Switch/Sensor] Edge” detects this value on an incoming edge. If not cyclically sending is needed, please select value “0” in this field.
- **Block:** By selecting “Yes” on this field, a new Communications object “Block” appears on the ETS. This object can be used to disable an input:
  - On receiving a “**1**” through this object, the **ACTinBOX** Classic-Hybrid will be ignoring any Edge on the input.
  - On receiving a “**0**” through this object, the input recovers its enabled status, so that it is ready again to receive orders.

  Actions taken on the input while being disabled will not be taken into account.

Take into account: The Internal Links in ACTinBOX Classic have been removed, since the new version 5.0 of the Operating System includes full dúplex.

3.3 TEMPERATURE PROBE INPUT

A **temperature probe** allows users to make measurements of **real temperatures** in a room or place.

In addition to the temperature probe, the ACTinBOX Classic-Hybrid application program includes a thermostat option.

Thus, depending on the installation needs, the thermostat might be enabled or not (by parameter):

- Temperature sensor
- Temperature sensor and thermostat

3.3.1 TEMPERATURE SENSOR

TEMPERATURE SENSOR CALIBRATION

Allow a user to recalibrate the Temperature Sensor referred to the own temperature measured by the sensor itself when the calibration parameter remains unchanged (value “0”).

Example: If we have a really accurate thermometer at home, we can take its measure as a reference for the ACTinBOX Hybrid.

Measurement shown by the temperature probe	x ºC
Measurement shown by the accurate thermometer	y ºC
Positive difference between measures	x – y = z ºC
Positive Calibration parameter	z ºC
Negative difference between measures	x – y = - z ºC
Negative calibration parameter	- z ºC

TEMPERATURE SENDING PERIOD

This field is meant to set a period of time for the ACTINBOX Classic-Hybrid to send the registered temperature to the Bus through the **“Current Temperature”** Communication object.

TEMPERATURE SENDING WITH CHANGE

This field is meant to enable the measured temperature sending to the BUS when it changes more than the parameterized value in this section.

TEMPERATURE PROTECTION

A **Temperature protection** for Overheating, Overcooling or both can be selected.

If **Overheating** is selected, a 1-bit communication object with the same name is enabled. When the temperature measured by the temperature probe

raises over Overheating Temperature, value 1 is sent through this communication object.

If **Overcooling** is selected, a 1-bit communication object with the same name is enabled. When the temperature measured by the temperature probe decreases below the Overcooling Temperature, value 1 is sent through this communication object.

In both cases a **Hysteresis** value is selected, which avoids the continue switching of the communication objects value when measured temperature is next to the protection temperature.

3.3.2 TEMPERATURE SENSOR & THERMOSTAT

When selecting this option for input 5, we can parameterize the temperature probe options and Zennio thermostat settings.

The available options for **temperature probe** are explained in the previous section.

The options for **Zennio thermostat** and its parameterization can be found in the documentation *Clima I. Zennio Thermostat*, available in our website.

3.4 MOTION SENSOR

All the information about parameterization of input 6 as a motion sensor is in the manual of the motion sensor. This manual is available in our website.

4. LOGICAL FUNCTIONS

This section in the **ACTinBOX Classic-Hybrid** is meant to perform **binary logic operations with incoming data from the Bus**, to send the Result through other Communication Objects specifically enabled in the actuator for this operation. These Functions work with two different types of data:

- **Bus**, through special **Communication Objects** enabled for these functions.
- **Internal variables**, to store the intermediate partial operation results.

The **parameters** related with **Logical Funcions** must be configured in this section:

- **Logical Function Selection:** up to five different logical functions can be enabled.

- **Total Data Entry Objects:** It is necessary to define the number of Data Entry Objects of each type necessary to be used in all functions.

  - **1 BIT (16 available objects)**: It is necessary to previously define the number of 1 bit objects to be used as data entry in the function operations.

  - **1 BYTE (8 available objects)**: It is necessary to previously define the number of 1 byte objects to be used as data entry in the function operations.

  - **2 BYTES (8 available objects)**: It is necessary to previously define the number of 2 byte objects to be used as data entry in the function operations.

Take into account I: Also available as internal variables to store partial results in the operations:

- *16 "1 bit" variables*
- *8 "1 byte" variables*
- 8 "2 bytes" variables.

Take into account II: It is necessary to previously define by parameter the number of data entry objects to be used in the functions before these appear on the ETS environment.

Take into account III: It is always recommended to define more data entries than needed, as a later redefinition involves the deletion of the possible Group addresses association previously made; with the consequent loss of time to re-associate them all

4.1 CALL

This section is meant to select the objects to **trigger** the function execution. Up to **8 different objects** may be selected.

Take into account: For the function to be executed, it will be necessary that at least one of the enabled objects in this section is updated. It is NOT necessary that the objects in charge of triggering the function execution are included in it.

4.2 OPERATIONS

This section is meant to define the operations to be performed in every enabled function. Up to 4 different operations can be enabled:

- **Operation:** Enable the corresponding operation
- **Type:** 4 different operation types:
  - **Logic**: 1 bit available logical operations are **ID, AND, OR, XOR, NOT, NAND, NOR y NXOR**. All of them work with 2 different values (except **ID** and **NOT**). These values can be chosen from the available **16 1 bit objects**, and the **16 1 bit internal variables**. In this case, the operation Result is also a 1 bit value that can be stored in any of the 16 available 1 bit internal variables.
  - **Arithmetic (1 byte/2bytes (unsigned integer)/2bytes (Floating point):** Depending on the chosen type, these operations will work with 1 byte or 2 bytes values. Users can choose among the following arithmetic operations: **ID, ADD, SUBSTRACT, MULTIPLY, DIVIDE, MAXIMUM y MINIMUM**. All of them work with two values (excepto **ID**); these can be chosen from the available objects, variables or a constant value chosen by parameter. The Result in the arithmetic opeation will be 1byte or 2 bytes, (depending on the operation). This Result can be stored in any of the 8 corresponding variables.

Take into account: Arithmetic operations (2 bytes unsigned integer) work with data in the range (0……65535). Constants set in the corresponding parametrizable field use format 0.1X (i.e. Value=22.5→Parameter=225)

Take into account I: Arithmetic operations (2 bytes Floating point) work with data in the range (0……120). Constants set in the corresponding parametrizable field use format 0.1X (i.e. Value=22.5→Parameter=225)

Take into account II: If the Result in the 2 bytes Arithmetic operations exceed the permitted range, this will be converted to the corresponding limit in the range. Dividing by "0" doesn´t send anything to the Bus

- **Comparison (1 byte/2bytes (unsigned integer)/2bytes (Floating point)**: These operations work with 1 byte or 2 bytes values, depending on the chosen type. Users in this case can choose among the following comparison operations: **HIGHER, HIGHER OR EQUAL, LOWER, LOWER OR EQUAL, EQUAL, UNEQUAL**. All of them work with two values to be chosen among the available objects, values or constant values chosen by parameter. The Result in the operation is a 1 bit type. This Result can be stored in any of the 16 available 1 bit variables.

- **Conversion (1 bit/1 byte/2bytes (unsigned integer)/2bytes (Floating point):** To convert the Communication Objects between formats.

Operation Result: To define the variable to store the operation result.

4.2.1 DESCRIPTION OF CONVERSION FUNCTIONS

Specific information on the conversion function in **ACTinBOX Classic-Hybrid** is detailed next:

"CONVERSION" (1 bit → 1byte)

1bit	1byte
0	00000000
1	00000001

- **“CONVERSION” (1bit → 2bytes unsigned integer)**

1bit	2bytes unsigned integer
0	00000000 00000000
1	00000000 00000001

- **“CONVERSION” (1 bit → 2 bytes floating point)**

1bit	2bytes floating point
0	0
1	0,1

- **“CONVERSION” (1 byte → 1 bit)**

1byte	1bit
0	0
1..255	1

- **“CONVERSION” (1 byte → 2 bytes unsigned integer)**

1byte	2bytes unsigned integer
$00	$00 00
$01	$00 01
…	…
$FF	$00 FF

- **“CONVERSION” (1 byte → 2 bytes floating point)**

1byte	2bytes floating point
0	0
1	0.1
255	25.5

Take into account: Conversion limit in this case is 25.5

- **“CONVERSION” (2 bytes unsigned integer → 1 bit)**

2bytes unsigned integer	1bit
0	0
1..65535	1

- **“CONVERSION” (2 bytes unsigned integer →1 byte)**

2bytes unsigned integer	1byte
$00 00	$00
$00 01	$01
…	…
$00 FF	$FF
> $00 FF	$FF

- **“CONVERSION” (2 bytes unsigned integer→ 2 bytes floating point)**

2bytes unsigned integer	2 bytes floating point
0	0
1	0.1
…	…
1200	120
>1200	120

- **“CONVERSION” (2 bytes floating point → 1 bit)**

2 bytes floating point	1bit
0	0
0,1…….120	1

- **“CONVERSION” (2 bytes floating point →1 byte)**

2 bytes floating point	1byte
0	0
0,1... 25,5	1..255
> 25,5	255

- **“CONVERSION” (2 bytes floating point → 2 bytes unsigned integer)**

2 bytes floating point	2bytes unsigned integer
0	0
0.1	1
…	
120	1200
>120	1200

4.3 RESULT

This section is meant to tell the **ACTinBOX Classic-Hybrid** where to store and what to do with the **Result** obtained in the previous sections.

- **Type:** Choose among 1 bit, 1 byte or 2 bytes (Unsigned integer) / (Floating point).
- **Value:** Set the variable where the Result will be stored.

> **Take into account:** Please notice that all the storing variables are shared with all the possible functions/operations in the ACTinBOX Hybrid, this means that a specific variable used to store the Result in an operation/function, should not be used to store a different result.

- **Sending:** Set the conditions to send the Result to the Bus.

  - **Result is different from last sent**: The Result will be sent every time the final Result in the operations changes.
  - **Whenever the function is executed**: The Result will be sent every time the Function is executed.

> **Take into account:** This parameter is related with section **CALL** (see section 4.1); actualy the Result will be sent every time the Function is executed, but the Function will only be executed when at least one of the enabled objects in the section CALL is updated.

  - **Periodical sending**: The result will be periodically sent depending on the time set in the **CYCLE TIME** field.

- **Restriction:** The sending of the **1 bit functions Result** can be restricted to ("**0**" or "**1**"). **1 byte and 2 bytes functions Result** sending can be also restricted depending on the following options:

  - Values equals reference one
  - Values not equal to reference one
  - Values lower than reference one
  - Values higher than reference one

  - **Reference Value**: For the **Result Type = 1 byte**, possible reference value range is **[0…..255]**. For the **Result Type = 2 bytes**, possible reference value range is **[0……65535]**.

- **Delay:** Time to pass before sending the Result to the Bus. If no delay is needed please set value “0” in this field.

> **Take into account:** The Internal Links in ACTinBOX Classic have been removed, since the new version 5.0 of the Operating System includes full dúplex.

5. COMMUNICATION OBJECTS

Communication Objects in the Logical Function section can be two types:

- **Data:** Data coming from the Bus, these are the data the operations work with.
- **Results:** These are the Functions Results. Depending on its size, these are divided in 3 types: 1 bit, 1 byte and 2 bytes.

5.1 NOMENCLATURE

- **Data Object Type**

[[LF] Data ("size") "X" where size can be 1 bit, 1 byte or 2 bytes; and "**X**" is the Data number (1…..16 for 1 bit data, 1…..8 for 1 byte & 2 bytes).

- **Result Object Type**

[LF] RESULT FUNCTION "X" ("size"). Where size can be 1 bit, 1 byte or 2 bytes (depending on the data function result), and "**X**" is the function number (1….5).

- **Internal Variables**

b1,……, b16 (1 bit type)

n1,……, n8 (1 byte type)

x1,……, x8 (2 bytes type)

BECOME A USER!
http://zennioenglish.zendesk.com
TECHNICAL SUPPORT

ZENNIO TECHNICAL DOCUMENTATION