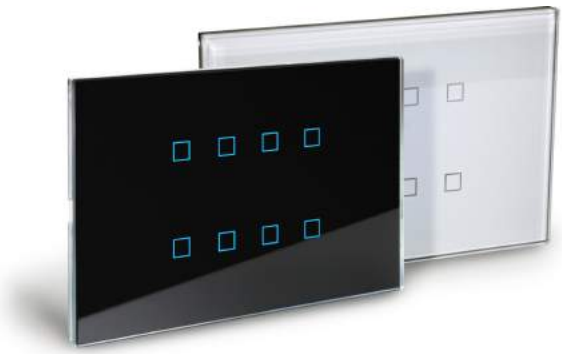


Technical Manual

KRISTAL Line

KNX Capacitive Keypads

blumotix

KNX®

CONTENTS

1 General Information

Kristal is a complete line of capacitive keypads designed to control KNX devices.
The keypads are available in the format 80mm x 80mm for installation in round connector blocks (BX-qx) or in the format 120mm x 80mm for installation in Italian connector blocks Model 503 (BX-Rx).
Both models are available in the White (W) and Black (B) colour options.

Here below are the products' identification codes:

BX-Q02B	2-control model 80x80, Black
BX-Q02W	2-control model 80x80, White
BX-Q04B	4-control model 80x80, Black
BX-Q04W	4-control model 80x80, White
BX-Q06B	6-control model 80x80, Black
BX-Q06W	6-control model 80x80, White

BX-R03B	3-control model 120x80, Black
BX-R03W	3-control model 120x80, White
BX-R06B	6-control model 120x80, Black
BX-R06W	6-control model 120x80, White
BX-R08B	8-control model 120x80, Black
BX-R08W	8-control model 120x80, White

Each device provides control functions to power on users, open and close shutters, adjust sources of lighting and select scenes.
Some of these features leverage the possibility of recognising a prolonged contact to perform auxiliary actions.
This product can be used for both home and industrial applications.

The products described in this manual all use the ETS library named
BX-QxRx_AV01.vd5

2 Product Features

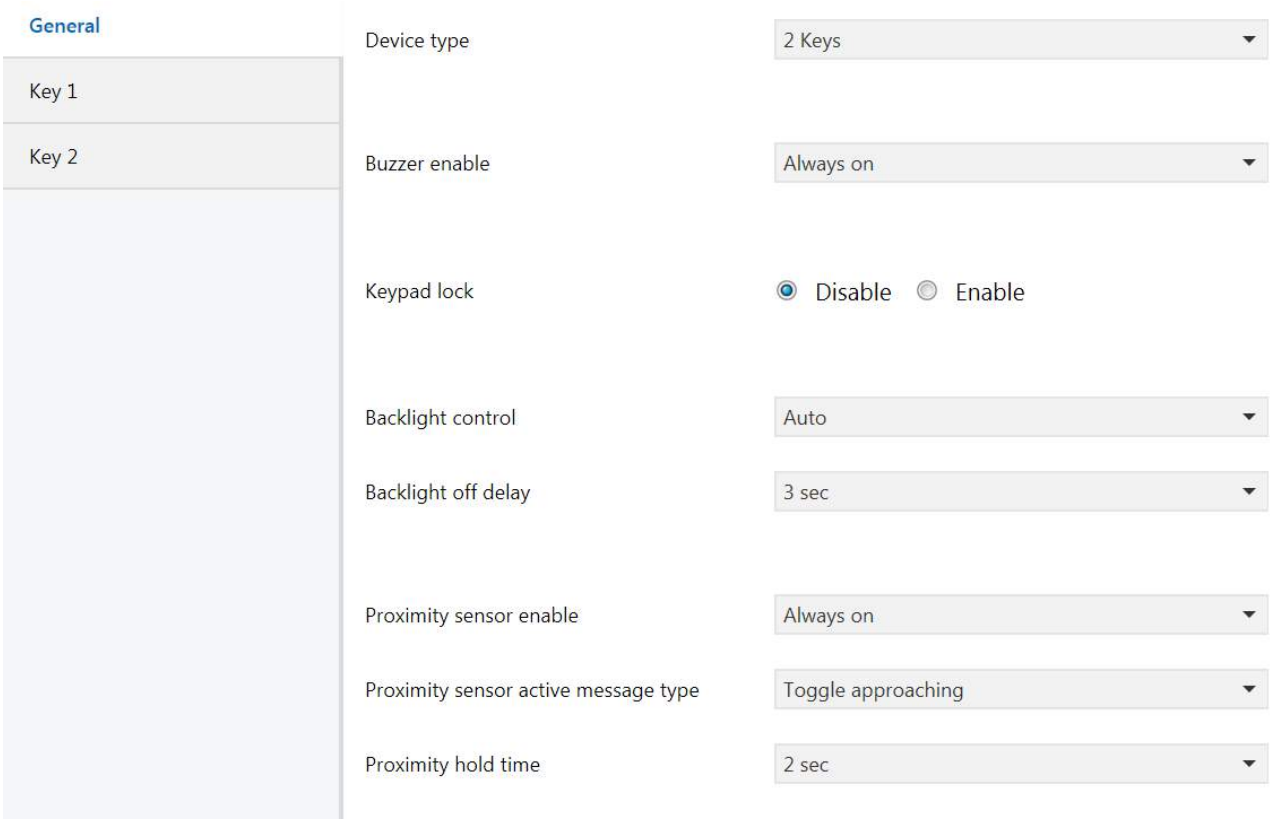
The buttons interface is powered via the Konnex bus and does not need any additional power supplies.
The keypads have BLUE backlighting behind each control and WHITE backlighting all along their transparent edges to illuminate the wall.
Both are controllable with suitable ETS communication objects.
The device is equipped with a buzzer, also programmable via ETS, in order to offer the best acoustic feedback.
The interface is also equipped with a proximity sensor to turn on as the user's hand moves closer.
The IP20 casing is designed for installation inside connector blocks intended to accept conventional push-buttons.
Fixing is done with a spring-type latch to a frame which must be previously fixed to the connector block via the appropriate self-tapping screws.

3 ETS Library

The ETS Library features a number of parameters used to characterise the operation of each control of our keypad.
The parameters are conveniently divided into pages for the configuration of each control, with an added main page for the selection of which device type should be installed.

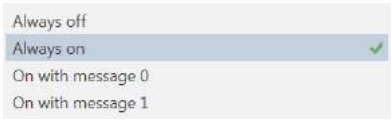
3.1 General Parameters

Below is the General parameters page reached by activating the ETS control **Edit Parameters.**



This ETS library is applicable to a number of products which differ in shape, colour and above all, number of controls.
Therefore, it is first of all necessary to correctly select the number of available keys on the device that you are installing, so that the Library can display all the parameters and communication objects necessary for correct programming accordingly.

Parameters are listed here below that may affect the general operation of the device.

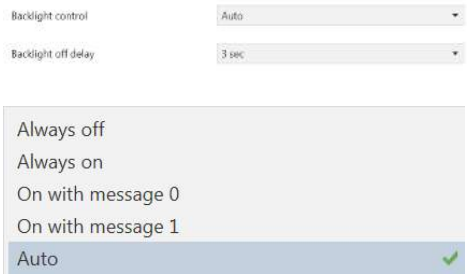


The first relates to the enabling of the **Buzzer** to ensure an acoustic feedback response when a control is activated.
The buzzer can be turned off by enabling the Always off function.
The buzzer can be turned permanently on by enabling the Always on function.
Alternatively, the Buzzer operation can be enabled or disabled through a communication object (On with message 1 and On with message 0).

31	Buzzer	Enable	1 bit	C	R	W	-	-	switch	Low

The second relates to the enabling of the **Keypad Lock** to lock the keypad operation, for example while the glass top is cleaned.
To enable the Keypad Lock means to display a Lock set communication object to be able to receive a lock control.
In our case, ETS also displays an object for Lock status reading.

68	Keypad	Lock set	1 bit	C	R	W	-	-	switch	Low
69	Keypad	Lock status	1 bit	C	R	-	T	-	switch	Low

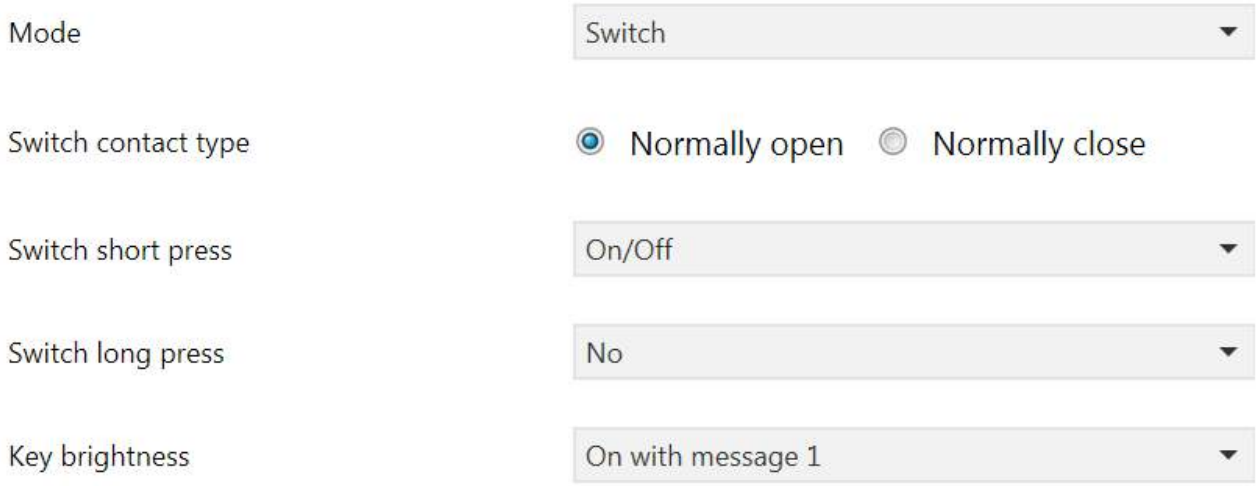


The third relates to the programming of the **Backlight**.
The Backlight can be left always off (OFF) or always on (ON), or it may be controlled by communication objects (On with message 1 and On with message 0).
Alternatively, operation can be automatic as a function of proximity sensing, so as to be activated when sensing a hand.
In this case, a time delay can be set after which the Backlight will go off.

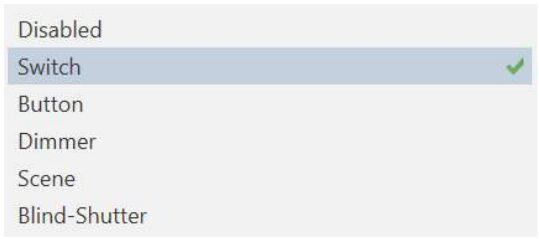
30	Backlight	Set	1 bit	C	R	W	-	-	switch	Low

3.2 Key1, Key 2, Key3, ...Key12

Below is the image typically opening the key configuration dedicated page.

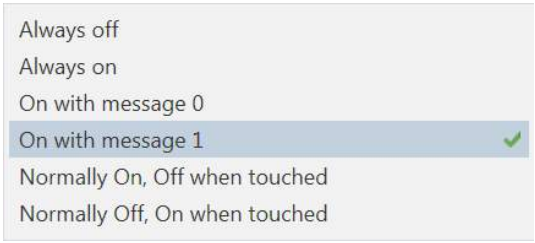


The first parameter, **Mode**, is the most important one and determines the mode of operation of the corresponding key.
There are 6 different modes of operation that will be described below - including the possibility of disabling the control.

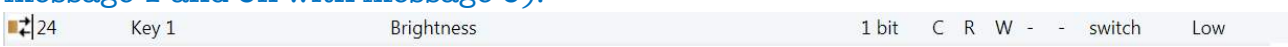


The remaining parameters are determined dynamically according to the type of function chosen and will change from time to time.

The last parameter is, however, shared by all the modes of operation and can be described in general terms in this introduction.

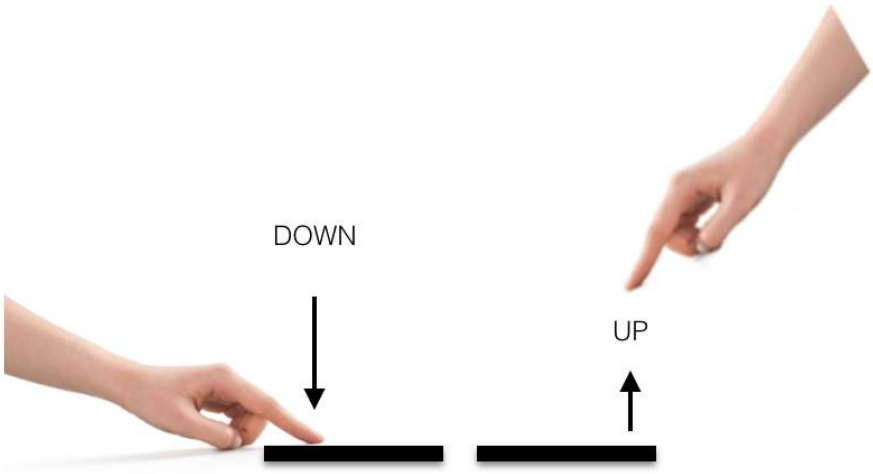


It is the **Key brightness**, a feature that allows the control light feedback to be controlled.
By selecting the Always off option, the feedback can stay off all the time, while by selecting Always on it can stay always switched on.
Alternatively, you can decide that it should change state when touched, going on if it was off (Normally Off, On when touched) or off if it was previously on (Normally On, Off when touched).
Finally, it can be programmed to be controlled via a communication object (On with message 1 and On with message 0).



3.2.1 Switch Mode

The **Switch** mode enables to send a KNX telegram to coincide with one of the possible events linked with the contact between finger and keypad.
By defining a **Contact type** of the normally open type, the telegram will be transmitted upon keypad/finger contact start.
By defining a Contact type of the normally closed type, the telegram will be



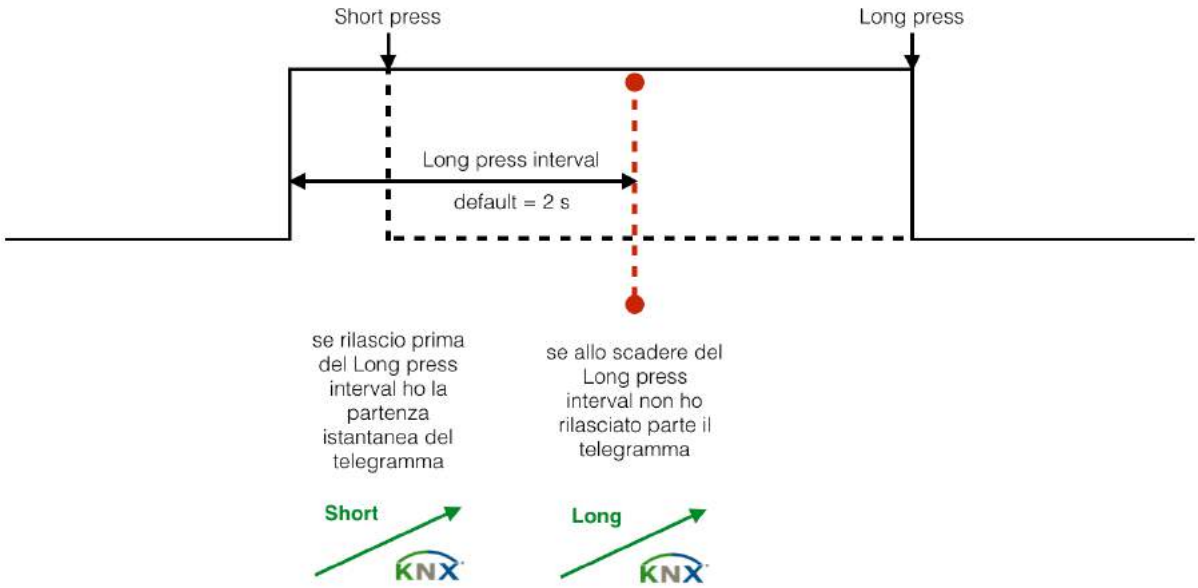
transmitted upon finger/keypad contact end, or when the user's finger moves away.

The definition of the Contact type property determines the event upon which the telegram will be transmitted on the KNX bus.

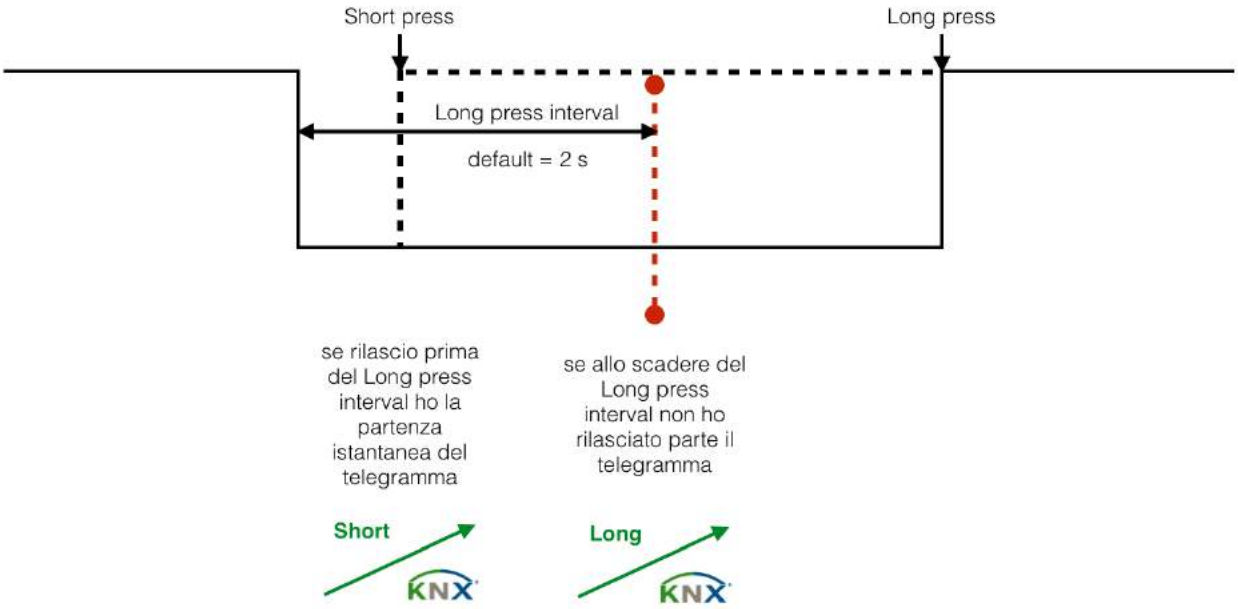
Every time that the keypad is touched, a telegram is sent the value of which can be specified via the **Short press function** parameter.
This value can be always the same or always change for each event.
By selecting ON you will be able to always send the ON value (1) at each event.
By selecting OFF you will be able to always send the OFF value (0) at each event.
By selecting ON/OFF you will be able to change the value sent upon each event, by alternating each time the values ON and OFF (Toggle Mode).

Finally, it is possible to activate the **Long press function** mode.
This allows two events to be managed with each input:
the short press event to activate a certain KNX control, or the long press event to activate another KNX control.
This feature allows the number of controls manageable with a button interface to be doubled.

Enabling this feature activates the configuration of two new parameters.
The former, identified as Long press function, enables the definition of the value to be transmitted - similarly to the procedure already described for the short function.
The latter allows the press time required to trigger a long press event to be indicated.



In the event that Contact type is normally closed, the operation logic that we have just described is completely reversed.



The Switching mode exposes an extremely simple 1 bit communication object to transmit controls to the other KNX devices.

0	Input 1	Switching	1 bit	C	R	W	T	-	switch	Low

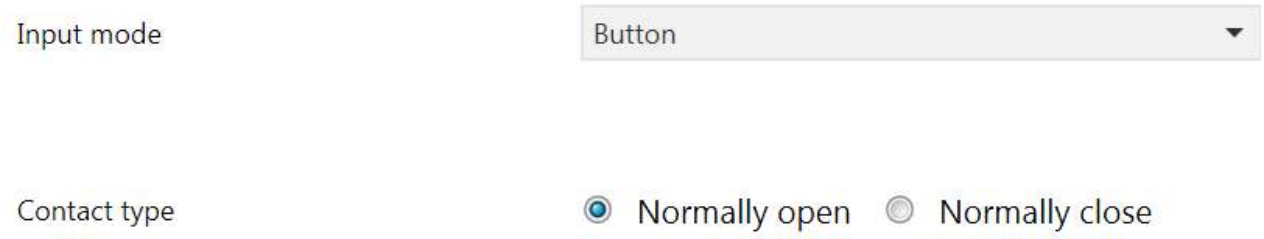
This object is created to notify a State change that occurs in the field to coincide with the already mentioned contacts.
Therefore, no other object will be found to notify the change of state!
This single communication object is also highly useful to synchronise the input state with events notified on the KNX bus.

Warning!

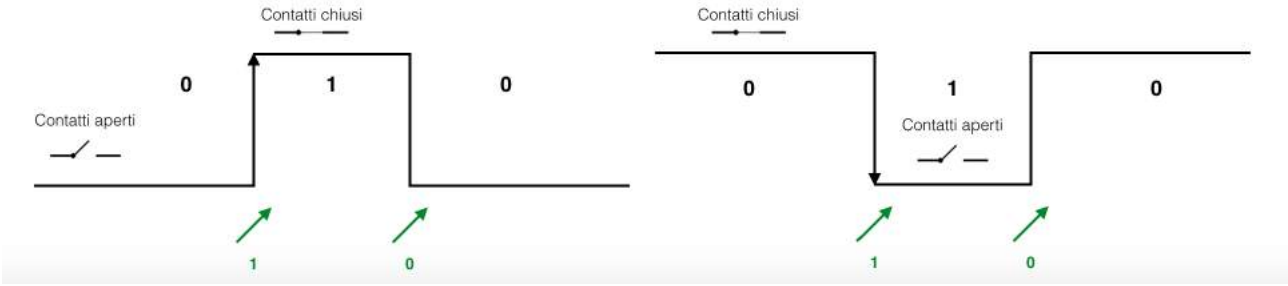
The performance of the button interface changes when the Long press function is enabled.
If the long press control is not expected, the KNX telegram is sent immediately, as soon the pressure applied on the button is felt (rising edge).
Conversely, if the Long press function is activated, the interface must be left waiting to know if the control will be long or short, by introducing a short **latency** time.
This latency can induce a feeling of failed control reactivity.
The user must be informed that this is a normal consequence of the need to wait for the release of the button (trailing edge) - in order to assess whether the control to run will be of the long or short type.

3.2.2 Button Mode

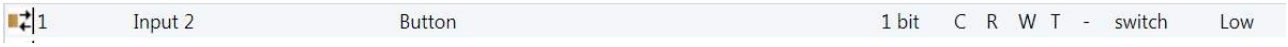
This mode is suitable for active control - to control the closing of the relay while the button is held down.
The device constantly reads the state of the input and sends a telegram whenever it detects a change.



This means that a telegram is sent whenever there is a change in the status of our inputs.
In the event that a button is used, a telegram is sent at the time of pressure application and then a telegram is also sent at the time of release.
Like in the previous case, Contact type is the property that determines with which event the notification of telegrams on the KNX bus is synchronised.
If Contact type is defined as normally open (standard) the datum is sent on the rising edge, therefore it is worth 1 when the contact closes and 0 when the contact opens.



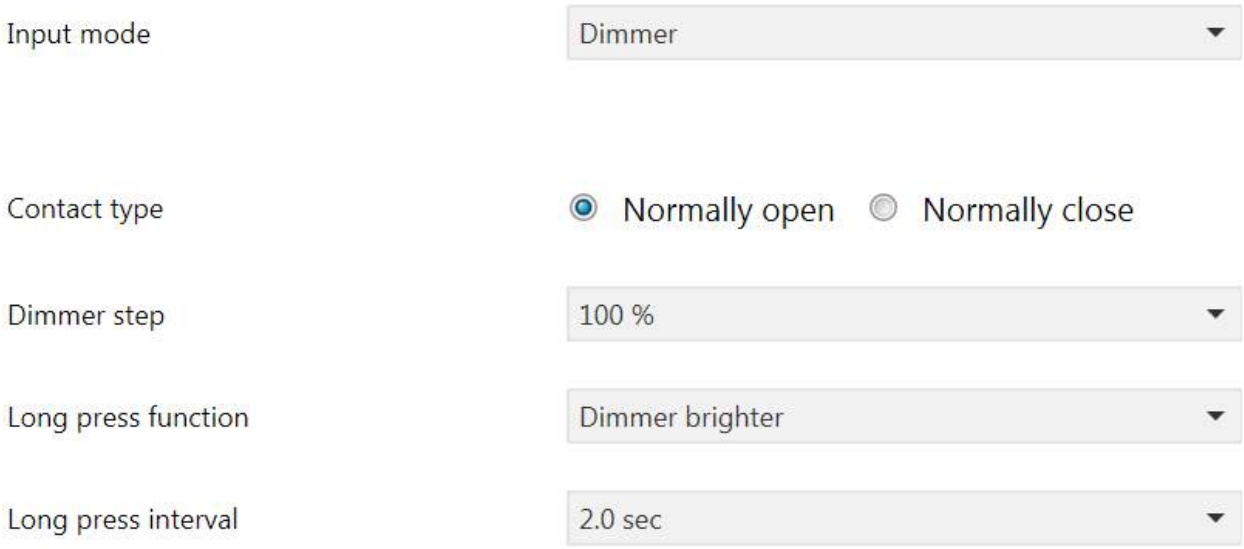
If Contact type is defined as normally closed, the telegram 1 is sent at the end of the Closed contacts event and the logic is reversed.
Therefore, you will read 1 when the contacts are open and you will read 0 when the contacts are closed.



The Button mode, too, exposes a single plain communication object that operates in the manner already described for the Switching function.

3.2.3 Dimmer Mode

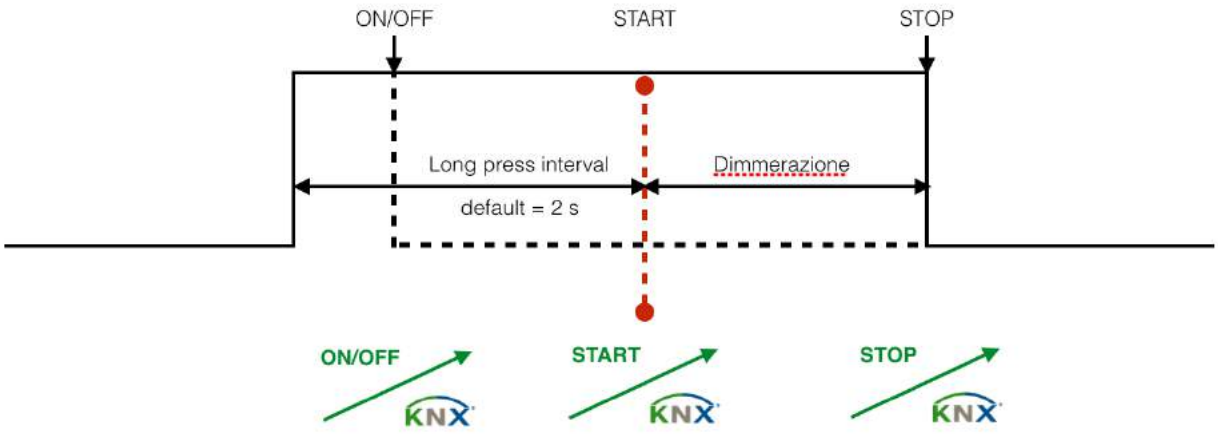
This is the function suitable for adjusting the brightness of a dimmable light.



The Dimmer function is one of the modes that automatically enables the use of a Long press function to have a second control available.
The Short control is used to switch the light on and off and enables a 1bit communication object named Switching; it will function exactly like the previously described switching function.
The Long control is used to adjust the light intensity and enables a 4 bit communication object named Dimming.

2	Dimmer 3	Switching	1 bit	C	R	W	T	-	switch	Low
10	Dimmer 3	Dimming	4 bit	C	R	-	T	-	dimming c...	Low

This means that, in order to turn on and turn off the light, a short press only will be required, while in order to adjust the brightness, the control should be held down until the required brightness is obtained.



After the time interval required to activate the Long control, a telegram will be sent with the dimming START control.

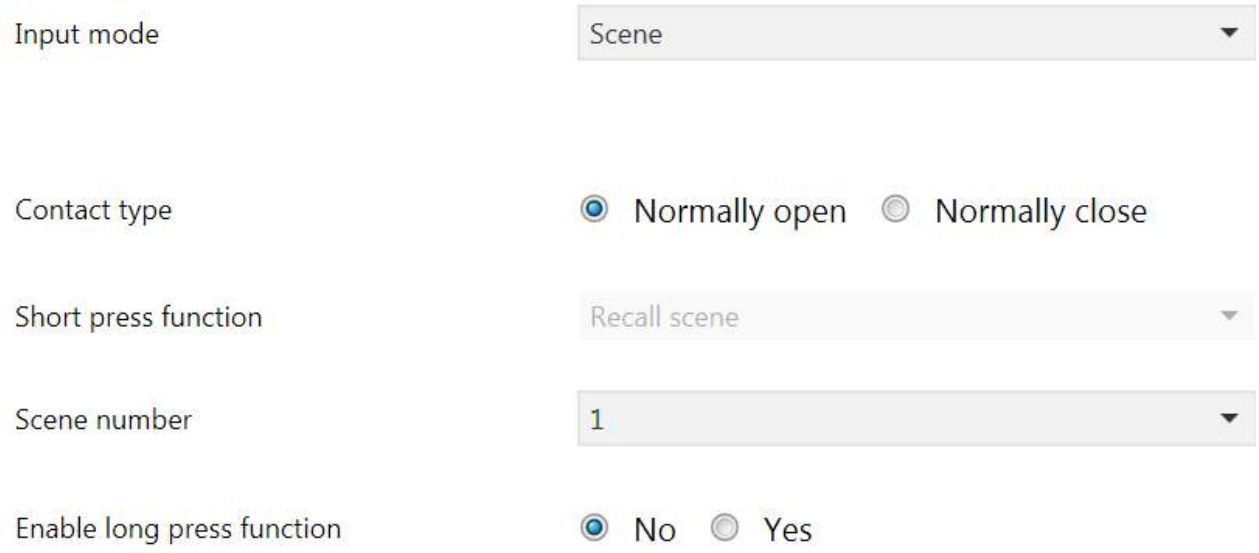
After obtaining the desired brightness we will be able to release the button by automatically causing the sending of the dimming STOP control.
Like in all previous cases, the **Contact type** property defined as Normally closed will determine the reversal of the operation logic.
The effectiveness of the adjustment depends above all on the speed with which the dimmer changes the brightness and is a parameter generally settable from the actuator.

The Long control can operate in 3 different modes.
It can act in the Brighter mode to increase the brightness, or in the Darker mode to decrease the brightness, or it can toggle from time to time between these two modes to be able to fully control the Dimmer with a single control.

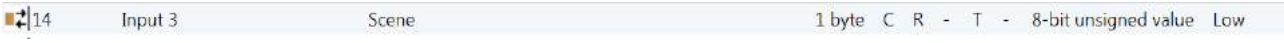
Finally, we are describing here below the **Dimmer Step** parameter.
This parameter determines by how much we can increase or decrease the brightness with a single Long press action.
100% means to be able to increase or decrease the brightness throughout the required width.
50% means to be able to increase or decrease the brightness up to half the required width. This means that in order to cover all of the available range, we will have to perform two actions.
The possible variation with a single action can decrease up to 1%.

3.2.4 Scene Mode

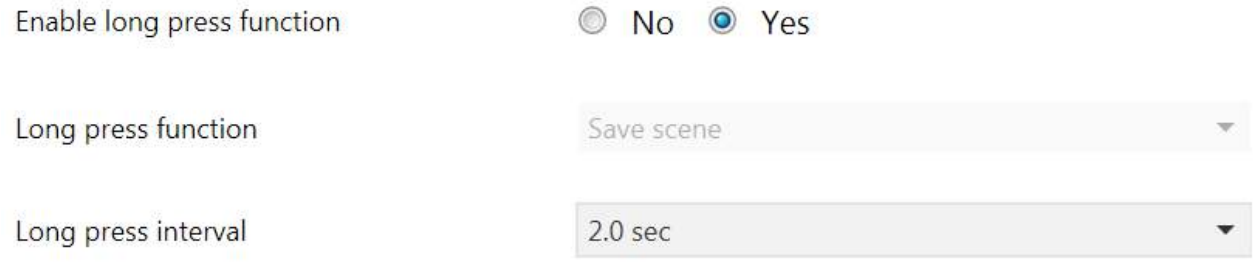
This is the function that allows control over the KNX Scenes.
Controlling a KNX scene means to be able to retrieve it when needed or be able to store a new configuration.



Each scene is defined by an identification number (ID) which can vary between 1 and 64.
To activate a scene consists in sending a byte to the KNX bus with value (ID - 1).



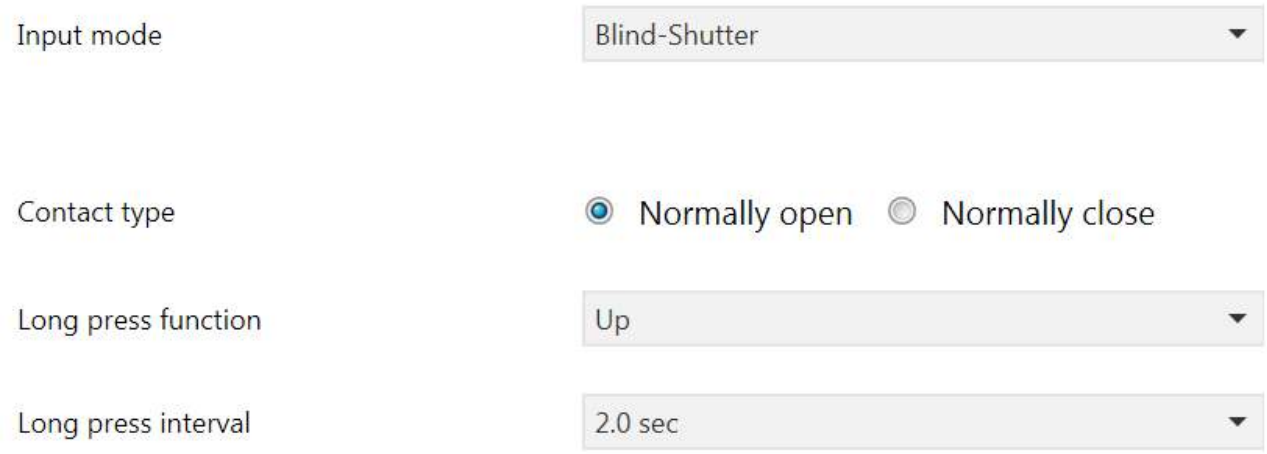
This control is assigned to the Short function.
Therefore every time that a short press is applied the configured scene will be recalled.



To store a new scene consists in sending a byte to the KNX bus with value (ID + 64).
Sending the memory storage control is assigned to the Long function.
Therefore, every time that a long press is applied a new scene programming will be controlled.
This feature can be enabled or disabled via the **Enable long press function** property.

3.2.5 Blind - Shutter Mode

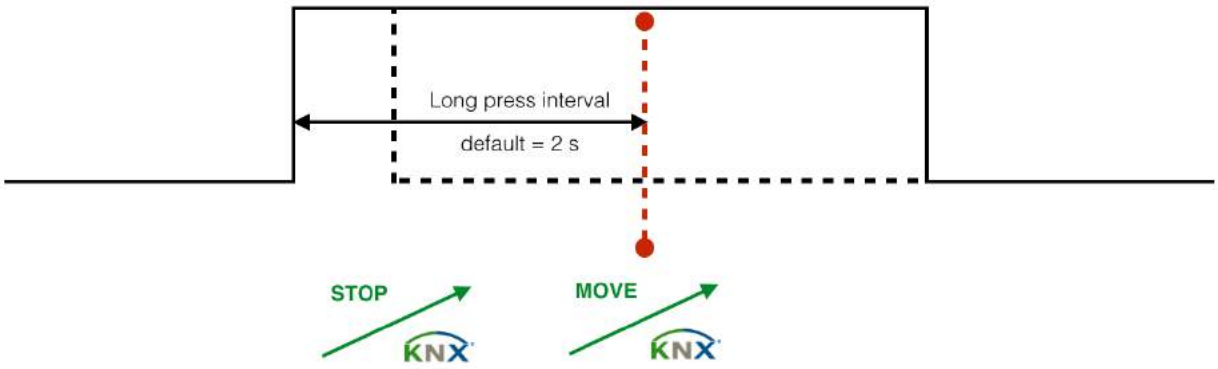
This is the function that allows control over the shutters and blinds actuators.



The main feature of a shutter/blind control is the fact that it is actuated via two different group addresses: the first to control the shutter movement, and the second to control the shutter stop and the venetian blind movement.
Therefore, the interface must necessarily display two distinct communication objects for movement and stopping.

3	Input 4	Blind-shutter stop	1 bit	C	R	W	T	-	switch	Low
7	Input 4	Blind-shutter move	1 bit	C	R	W	T	-	up/down	Low

The STOP control is assigned to the Short function.
The MOVE control is assigned to the Long function.
This means that every time a short press is applied, the roller shutter movement is stopped and every time a long press is applied, the roller shutter is set in motion.



This setting allows for the shutter to be set in motion and let run to the end of its stroke without any need to hold the button down.
Should you wish to stop it earlier, a short press can be applied to send a STOP control.

The movement control can be programmed both to raise the roller shutter (UP = 0) and to lower it (DOWN = 1).
When a given movement is assigned to an input, it should be borne in mind that two inputs are required to ensure full actuator control: one to raise and the other to lower the roller shutter.
An UP/DOWN mode is also available to save interface resources and be able to always toggle the control sent.
In this way, a single channel can be used for both functions.

The value assigned to STOP in order to stop the shutter movement is generally not influential.
Whatever value is assigned, it always results in the roller shutter stopping.

A different case is that of a shutter already at a stop.
In that case, sending a STOP control is interpreted as a control to actuate the slats - in order to adjust the brightness through the venetian blind.
In this case, the value assigned to STOP determines the direction of rotation of the slats, increasing or decreasing the venetian blind light-blocking ability.
Our interface always sends a slat control in an opposite direction of rotation to the last movement performed, to reflect the different position of the slats after the up-stroke or down-stroke of the venetian blind.
The down-stroke (DOWN = 1) of the venetian blind is always preceded by a release of the slats to a fully light-blocking position, therefore the subsequent adjustment of the slats must be in the opposite direction to increase the brightness (STOP = 0).
Conversely, the up-stroke (UP = 0) of the venetian blind is always preceded by a recall of the slats to a horizontal position, with minimal light-blocking effect, therefore the subsequent adjustment of the slats must be in the opposite direction to increase the shading effect (STOP = 1).