

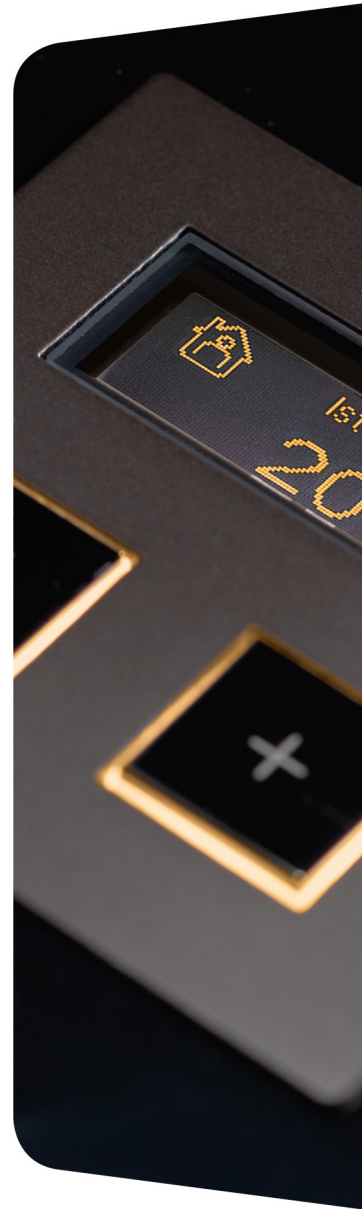
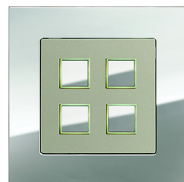
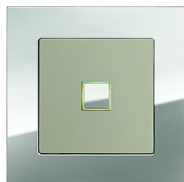
Application description

EDIZIOdue elegance

KNX push-button RGB 1- to 4-gang

480x-B...

10.KNX4801B-E.1309/130924



EDIZIO as well as the corresponding logo are registered trademarks of Feller S.A.

All rights reserved, including translation into other languages. It is not permitted to copy, duplicate or distribute the document or parts thereof in any form or to transmit it by means of electronic systems without the written approval of the publisher.
We reserve the right to make technical changes.

© Feller S.A. 2013

1	General	1
1.1	Technical data	1
1.2	Typographical conventions	2
2	The application "Push-button 1- to 4-gang V1.0"	3
2.1	Overview	3
2.2	Communication objects	3
2.2.1	Object table push-button	3
2.2.2	Object table scene module	6
2.2.3	Object table sequence module	8
2.3	Parameters	9
2.3.1	Parameter page "Configuration of push-buttons"	9
2.3.2	Parameter page "Push-button x"	10
2.3.3	Parameter page "LED brightness and flashing speed"	18
2.3.4	Parameter page "LED colours"	19
2.3.5	Parameter page "General disabling"	19
2.3.6	Parameter page "Disable push-buttons"	21
2.4	Parameters sequence module	21
2.4.1	Parameter page "Sequence module"	21
2.4.2	Parameter page "Switching point x"	22
2.5	Parameters scene module	23
2.5.1	Parameter page "Scene module"	23
2.5.2	Parameter page "Data type scene value 1...10/1..15"	24
2.5.3	Parameter page "Scene x [value 1...10/1...15]"	24
3	Functional description	25
3.1	Behaviour after ETS download or bus voltage return	25
3.2	Push-button	25
3.2.1	Operating concept	25
3.2.2	LEDs	26
3.2.3	LED colours	27
3.3	Sequence module	28
3.4	Scene module	29
3.5	RGB colour theory	30

CONTENT

1 General

This document explains the individual parameters of the EDIZIOdue elegance KNX push-buttons RGB 1- to 4-gang, and serves as configuration aid.



EDIZIOdue elegance KNX push-button 1- to 4-gang RGB
Application: Push-button 1-to 4-gang V1.0

The EDIZIOdue colore KNX push-button RGB is an input unit and is used as sensor for activating and deactivating different loads, for dimming lights, operating blinds and for saving and recalling scenes and/or starting sequences in KNX systems.

The function insert can be equipped with 1, 2 or 4 buttons. Both single-button and two-button operation is possible (→ [chapter 3.2.1](#)).

If only single-button operation is possible, a maximum of four independent functions can be realised.

The KNX push-buttons RGB feature RGB LEDs, which can display 6 different basic colours as well as 4 freely definable user colours. For the user colours, the values red, green and blue can be set in the ETS or sent via the KNX bus with a 3 byte object.

1.1 Technical data

Ambient conditions:

- | | |
|----------------------------------|---|
| - Type of protection (IEC 60529) | IP20, dry installation |
| - Ambient temperature | operation: -5 °C up to +45 °C
storage: -25 °C up to +70 °C |

KNX supply

- | | |
|--------------|-----------------------------|
| - Voltage | 21–30 V DC SELV |
| - Connection | KNX bus connecting terminal |

Power consumption

- | | |
|---|-------------|
| - Basic power requirement | max. 250 mW |
| - additionally per lighting push-button | max. 75 mW |

Service life at least 10⁵ switching operations

Installation depth 22 mm

Attention:

- > KNX devices with the additional designation **RGB** can only be programmed using the corresponding application with the additional designation RGB.
- > Older applications (without the additional designation RGB) cannot be loaded to the present hardware with the additional designation **RGB**. Feller shall not assume any liability or consequential costs for projecting errors.



Note:

For further information on the installation, please refer to the installation instructions.

2 The application "Push-button 1- to 4-gang V1.0"

2.1 Overview

Number of communication objects:	89
max. number of group addresses:	254
max. number of allocations:	255

For planning as well as for commissioning and the diagnostics of a KNX system, a programming software is required: KNX Tool Software ETS version 3 or later. It is used to select and/or create the application programme and its parameters as well as loading them into the device.

The product database required for the KNX push-button RGB is available at www.feller.ch The KNX label guarantees that the products of different manufacturers are able to communicate with each other and that the commands are interpreted in the same way by devices of different manufacturers (command compatibility).

2.2 Communication objects

Communication flags:

Flag	Name	Meaning
R	Read	Object status can be viewed (ETS / display etc.)
W	Write	Object can receive
T	Transmit	Object can send
U	Update	Objekt can accept answer to own read requests

2.2.1 Object table push-button



The following objects are visible depending on the parameterisation.

No.	Object name	Function	Type	DPT	Flags			
					R	W	T	U
0, 3 6, 9	Push-button couple x	ON/OFF, switching	1 bit	1.001		x	x	
	Push-button x							
1 bit object for sending and receiving switching telegrams (ON, OFF).								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Switching</i>								
					R	W	T	U
0, 3 6, 9	Push-button couple x	ON/OFF, dimming	1 bit	1.001		x	x	
	Push-button x							
1 bit object for sending and receiving switching telegrams (ON, OFF).								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Dimming</i>								
					R	W	T	U
0, 3 6, 9	Push-button couple x	UP/DOWN, blind	1 bit	1.008		x	x	
	Push-button x							
1 bit object for sending and receiving telegrams with which blinds can be moved up- or downwards.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Blind</i>								

Communication objects

No.	Object name	Function	Type	DPT	Flags			
					R	W	T	U
0, 3 6, 9	Push-button couple x Push-button x	Recall, scene Recall/save, scene	8 bit	18.001			x	
8 bit object for recalling or saving one of a maximum of 64 scenes in the actuator.								
The object is visible if the following parameter setting is selected: "Scene module" – Scene function = <i>Decentralised scene saving (in actuator)</i> "Push-button x" – Push-button function = <i>Scene</i>								
					R	W	T	U
0, 3 6, 9	Push-button couple x Push-button x	Recall scene x	1 bit	1.010		x	x	
1 bit object for starting a local scene.								
The object is visible if the following parameter setting is selected: "Scene module" – Scene function = <i>Local scene saving (in push-button)</i> "Push-button x" – Push-button function = <i>Scene</i> Further information on the scene function → chapter 3.4								
					R	W	T	U
0, 3 6, 9	Push-button couple x Push-button x	Send, value	8 bit	5.001		x	x	
8 bit object for sending and receiving values 0–255.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Value</i>								
					R	W	T	U
0, 3 6, 9	Push-button couple x Push-button x	Forced position	2 bit	2.001		x	x	
2 Bit for sending and receiving forced telegrams. Polarity → chapter 2.3.2								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Forced position</i>								
					R	W	T	U
1, 4, 7, 10	Push-button couple x Push-button x	Brighter/darker, dimming	4 bit	3.007			x	
4 bit object for sending relative dimming telegrams.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Dimming</i>								
					R	W	T	U
1, 4, 7, 10	Push-button couple x Push-button x	Step/stop, blind	1 bit	1.009		x	x	
1 bit object for sending and receiving telegrams with which blinds can be stopped or slats can be readjusted.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Blind</i>								
					R	W	T	U
1, 4, 7, 10	Push-button x (longer press)	ON/OFF, switching	1 bit	1.001		x	x	
1 bit object for sending and receiving switching telegrams (ON, OFF).								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Switching / Value / Forced position</i> "Push-button x" – Longer press push-button x = <i>Active</i> "Push-button x" – Longer press function = <i>Switching</i>								

No.	Object name	Function	Type	DPT	Flags			
					R	W	T	U
1, 4, 7, 10	Push-button x (longer press)	Value, dimming	8 bit	5.001			x	
8 bit object for sending dimming values.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Switching / Value / Forced position</i> "Push-button x" – Longer press push-button x = <i>Active</i> "Push-button x" – Longer press function = <i>Dimming value in %</i>								
					R	W	T	U
1, 4, 7, 10	Push-button x (longer press)	UP/DOWN, blind	1 bit	1.008			x	
1 bit object for sending telegrams with which blinds can be moved up- or downwards.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Switching / Value / Forced position</i> "Push-button x" – Longer press push-button x = <i>Active</i> "Push-button x" – Longer press function = <i>Blind UP / DOWN</i>								
					R	W	T	U
1, 4, 7, 10	Push-button x (longer press)	Send, value	8 bit	5.001			x	
8 bit object for sending values 0–255.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Switching / Value / Forced position</i> "Push-button x" – Longer press push-button x = <i>Active</i> "Push-button x" – Longer press function = <i>Value</i>								
					R	W	T	U
1, 4, 7, 10	Push-button x (longer press)	Recall, scene	8 bit	18.001			x	
8 bit object for recalling or saving one of a maximum of 64 scenes in the actuator.								
The object is visible if the following parameter setting is selected: "Push-button x" – Push-button function = <i>Switching / Value / Forced position</i> "Push-button x" – Longer press push-button x = <i>Active</i> "Push-button x" – Longer press function = <i>Scene</i>								
					R	W	T	U
4, 10	Push-button couple x, double-click	UP/DOWN, move shading	1 bit	1.008			x	
1 bit object for sending telegrams with which the shading can be moved up- or downwards by means of blind actuators.								
The object is visible if the following parameter setting is selected: "Configuration of push-buttons" – Operating concept push-button x = <i>Two-button operation</i> "Push-button x" – Push-button function = <i>Blind</i> "Push-button x" – Advanced functions blind = <i>Move shading (double-click: long/short)</i>								
					R	W	T	U
2, 5, 8, 11	Push-button couple x, Signal LED	Show on LED	1 bit	1.001		x		x
	Push-button x, signal LED							
1 bit object used for activating the push-button LED. Polarity: 1 = LED illuminated ; 0 = LED extinguished								
The object is visible if the following parameter setting is selected: "Push-button x" – LED function = <i>Status signal LED object (external signal)</i>								

Communication objects

No.	Object name	Function	Type	DPT	Flags			
					R	W	T	U
2, 5, 8, 11	Push-button couple x, Signal LED	Show on RGB LED	3 byte	232.600		x		x
	Push-button x, signal LED							
3 byte object for receiving RGB telegrams which may affect the colour of the push-button LED.								
The object is visible if the following parameter setting is selected: "Push-button x" – LED-function = <i>RGB signal LED object (external signal)</i>								
					R	W	T	U
2, 5, 8, 11	Push-button couple x, Signal LED	Override/show on LED	1 bit	1.001		x		x
	Push-button x, signal LED							
1 bit object used for overriding the push-button LED function. Polarity can be parameterised.								
The object is visible if the following parameter setting is selected: "Push-button x" – LED function = <i>Orientation light (always switched on) / Push-button status (internal signal) / Press: ON / Release: OFF (feedback)</i> "Push-button x" – LED function overridable with object signal LED = Yes								
					R	W	T	U
24	All involved push-buttons	Disable push-buttons	1 bit	1.001		x		
1 bit object for enabling or disabling the push-button functions. Polarity can be parameterised.								
The object is visible if the following parameter setting is selected: "General disabling" – Disable push-buttons function unequal <i>Not active</i>								
					R	W	T	U
25	Night reduction LEDs	Decrease brightness	1 bit	1.001		x		
1 bit object for activating or deactivating the night reduction (modified brightness of all active LEDs). Polarity can be parameterised.								
The object is visible if the following parameter setting is selected: "LED brightness and flashing speed" – Night reduction LEDs function unequal <i>Not active</i>								

2.2.2 Object table scene module



Notes:

- The objects are only visible during the parameter setting
"Scene module" – **Scene function** = *Local scene saving (in push-button)*
- The number of visible objects varies between 10 (objects 31–40) and 15 (objects 31–45).
This depends on the parameter setting "Scene module" – **Number of scene values per scene**.

No.	Object name	Function	Type	DPT	Flags			
					R	W	T	U
31–45	Scene value x	ON/OFF, UP/DOWN	1 bit	1.001		x	x	x
1 bit object for sending and receiving switching telegrams (ON, OFF) or telegrams with which blinds can be moved up- or downwards. The object is visible if the following parameter setting is selected: "Data type scene value" – Data type scene value x = 1 bit (switching ON/OFF, blind UP/DOWN)								
					R	W	T	U
31–45	Scene value x	Send, value	8 bit	5.001		x	x	x
8 bit object for sending and receiving dimming values or telegrams with which blinds can be moved into the respective position. The object is visible if the following parameter setting is selected: "Data type scene value" – Data type scene value x = 8 bit (dimming value, blind value)								

2.2.3 Object table sequence module



Notes:

- The objects are only visible with the parameter setting "Sequence module" – **Sequence module = Active**
- The number of visible objects varies between 0 and 10 (first object: 79). This depends on the parameter setting "Sequence module" – **Switching point x = Active**.

No.	Object name	Function	Type	DPT	Flags			
					R	W	T	U
87	Sequence module	Recall sequence	1 bit	1.010		x	x	
	1 bit object for starting or stopping the sequence module. Polarity: 1 = start; 0 = stop.							
					R	W	T	U
88	Sequence module	Status	1 bit	1.010			x	
	1 bit object for reading out the status of the sequence module. Polarity: 1 = sequence has been started and will be processed; 0 = sequence completed							
					R	W	T	U
79–86	Sequence switching point x	ON/OFF, switching	1 bit	1.001			x	
	1 bit object for sending switching telegrams (ON, OFF). The object is visible if the following parameter setting is selected: "Sequence module" – Switching point x = Active "Switching point x" – Switching point x function = Switching							
					R	W	T	U
79–86	Sequence switching point x	Value, dimming	8 bit	5.001			x	
	8 bit object for sending dimming values. The object is visible if the following parameter setting is selected: "Sequence module" – Switching point x = Active "Switching point x" – Switching point x function = Dimming value in %							
					R	W	T	U
79–86	Sequence switching point x	UP/DOWN, blind	1 bit	1.008			x	
	1 bit object for sending telegrams with which blinds can be moved up- or downwards. The object is visible if the following parameter setting is selected: "Sequence module" – Switching point x = Active "Switching point x" – Switching point x function = Blind UP/DOWN							
					R	W	T	U
79–86	Sequence switching point x	Send, value	8 bit	5.001			x	
	8 bit object for sending values 0–255. The object is visible if the following parameter setting is selected: "Sequence module" – Switching point x = Active "Switching point x" – Switching point x function = Value							
					R	W	T	U
79–86	Sequence switching point x	Recall, scene	8 bit	18.001			x	
	8 bit object for recalling one of a maximum of 64 scenes in the actuator. The object is visible if the following parameter setting is selected: "Sequence module" – Switching point x = Active "Switching point x" – Switching point x function = Scene							

2.3 Parameters

Parameter page "Quick reference guide"

At first, the desired parameters must be set in the ETS application. These parameter settings will be saved when downloading the ETS application data to the KNX push-button RGB. Before starting the ETS download, the KNX push-button must be programmed with a unique physical address by the ETS.

Since certain dependencies exist between the parameters in the ETS, care must be taken in order to ensure that the parameters are set according to the following procedure:

1. Parameter page "Scene module"
2. Parameter page "Configuration of push-buttons"
3. The configuration of the remaining parameters can be carried out in any order.



Caution! Make sure to follow the correct order of parameterisation. If this order is not observed, any settings already made during the configuration will be lost.



Note: Always carry out the parameterisation in a top-down order.

2.3.1 Parameter page "Configuration of push-buttons"

On the parameter page "Configuration of push-buttons", you can specify the layout of the push-button.

Parameter **Number of push-buttons** determines the number of push-buttons for operating the connecting consumers.

Number of push-buttons 1 / 2 / 4

Parameter **Operating concept push-button x** defines the operating concept of the push-buttons (→ [chapter 3.2.1](#)).

Operating concept push-button x **Two-button operation**

Single-button operation

Two-button operation

The two push-buttons complement each other; they control the same actuator. One push-button issues the inverted command of the other.
e.g. left: Light ON – right: Light OFF

The parameter page "Push-button couple x" is visible.

Single-button operation

The two push-buttons are independent of each other; each controls a different actuator,
e.g. left: Light ON/OFF – right: Blind UP/DOWN

The parameter pages "Push-button left x" and "Push-button right x" are visible.

2.3.2 Parameter page "Push-button x"



Note: A specific parameter page (parameter page "Push-button couple x" and/or parameter pages "Push-button left x" and "Push-button right x") is available for the parameterisation of the push-buttons. However, since all parameter pages are identically structured, they will be explained altogether at this point.

Parameter **Push-button function** specifies the basic function of a push-button.

Push-button function

Parameter page "Push-button couple x"

Switching

Dimming

Blind

Forced position

Parameter page "Push-button left x" / "Push-button right x"

Switching

Dimming

Blind

Scene

Value

Forced position

Sequence module

Switching

The push-button has the basic function of switching.

With the parameter **Switching function**, the behaviour for a brief press is specified.

The object <Push-button x – ON/OFF, switching> is visible.

An alternative function can be defined for a longer press (parameter **Longer press**).

Dimming

The push-button has the basic function of dimming.

With the parameter **Dimming function**, the behaviour is specified for when the push-button is pressed.

The object <Push-button x – ON/OFF, dimming> is visible.

The object <Push-button x – Brighter/darker, dimming> is visible.

Blind

The push-button has the basic function blind.

With the parameter **Blind function**, the behaviour is specified for when the push-button is pressed.

The object <Push-button x – UP/DOWN, blind> is visible.

The object <Push-button x – Step/stop, blind> is visible.

Scene

The push-button has the basic function scene.

With the parameter **Scene function**, the scene to be called up is specified.

The object <Push-button x – ..., scene> is visible.

Value

The push-button has the basic function value.

The parameter **Value function** is set to the value *Send 8 bit value*.

The parameter **8 bit value** defines the value (0..255) that is sent to the bus upon a brief press.

The object <Push-button x – Send, value> is visible.

An alternative function can be defined for a longer press (parameter **Longer press**).

Forced position

The push-button has the basic function forced position.

With the parameter **Forced position function**, the behaviour for a brief press is specified.

The object <Push-button x – Forced position> is visible.

An alternative function can be defined for a longer press (parameter **Longer press**).

Sequence module

The push-button has the basic function sequence module.

With the parameter **Sequence module**, the behaviour is specified for when the push-button is pressed.

Parameter	Switching function specifies which command is triggered if a push-button with the basic function of switching is briefly pressed.
	<p>Switching function</p> <p>Operating concept push-button x = Two-button operation Press: ON <i>Press: OFF</i></p> <p>Operating concept push-button x = Single-button operation Press: INV <i>Press: ON</i> <i>Press: OFF</i> <i>Press: ON / Release: OFF</i> <i>Press: OFF / Release: ON</i></p> <p><i>Press: INV</i></p> <p>If the push-button is pressed, the state of the object <Push-button x – ON/OFF, switching> is inverted and transferred.</p> <p><i>Press: ON</i></p> <p>If the push-button is pressed, an ON telegram is sent to the object <Push-button x – ON/OFF, switching>.</p> <p><i>Press: OFF</i></p> <p>If the push-button is pressed, an OFF telegram is sent to the object <Push-button x – ON/OFF, switching>.</p> <p><i>Press: ON / Release: OFF</i></p> <p>If the push-button is pressed, an ON telegram is sent to the object <Push-button x – ON/OFF, switching>. An OFF telegram is sent once it is released again.</p> <p><i>Press: OFF / Release: ON</i></p> <p>If the push-button is pressed, an OFF telegram is sent to the object <Push-button x – ON/OFF, switching>. An ON telegram is sent once it is released again.</p>
Parameter	Dimming function specifies which commands are triggered by the push-button with the basic function of dimming.
	<p>Dimming function</p> <p>Operating concept push-button x = Two-button operation ON/brighter (short/long) <i>OFF/darker (short/long)</i> <i>INV/brighter (short/long)</i> <i>INV/darker (short/long)</i></p> <p>Operating concept push-button x = Single-button operation Single-button op. (short/long: INV/dimming INV) <i>ON/brighter (short/long)</i> <i>OFF/darker (short/long)</i></p> <p><i>Single-button op.</i> <i>(short/long: INV/dimming INV)</i></p> <p>If the push-button is pressed shortly, the state of the object <Push-button x – ON/OFF, dimming> is inverted and transferred. If the push-button is pressed and held, the dimming brightness is increased or decreased (brighter/darker) (object <Push-button x – Brighter/darker, dimming>). The corresponding dimming direction is determined by inverting the direction of the object value.</p> <p><i>ON/brighter (short/long)</i></p> <p>If the push-button is pressed shortly, an ON telegram is sent to the object <Push-button x – ON/OFF, dimming>. If the push-button is pressed and held, the dimming brightness is increased (brighter) (object <Push-button x – Brighter/darker, dimming>). If the push-button is released again, the dimming process is stopped.</p> <p><i>OFF/darker (short/long)</i></p> <p>If the push-button is pressed shortly, an OFF telegram is sent to the object <Push-button x – ON/OFF, dimming>. If the push-button is pressed and held, the dimming brightness is decreased (darker) (object <Push-button x – Brighter/darker, dimming>). If the push-button is released again, the dimming process is stopped.</p> <p><i>INV/brighter (short/long)</i></p> <p>If the push-button is pressed shortly, the state of the object <Push-button x – ON/OFF, dimming> is inverted and transferred. If the push-button is pressed and held, the dimming brightness is increased (brighter) (object <Push-button x – Brighter/darker, dimming>). If the push-button is released again, the dimming process is stopped.</p>

Parameters

INV/darker (short/long) If the push-button is pressed shortly, the state of the object <Push-button x – ON/OFF, dimming> is inverted and transferred. If the push-button is pressed and held, the dimming brightness is decreased (darker) (object <Push-button x – Brighter/darker, dimming>). If the push-button is released again, the dimming process is stopped.

Parameter **Blind function** specifies which commands are triggered by the push-button with the basic function blind.

Blind function

Operating concept push-button x = Two-button operation

UP (short: step/stop, long: move)

DOWN (short: step/stop, long: move)

UP (short: move, long: move/stop)

DOWN (short: move, long: move/stop)

Operating concept push-button x = Single-button operation

UP & DOWN (short: move, long: move/stop)

UP (short: move, long: move/stop)

DOWN (short: move, long: move/stop)

UP (short: step/stop, long: move)

DOWN (short: step/stop, long: move)

UP & DOWN (short: move, long: move/stop)

If the push-button is pressed shortly, the blind will move (object <Push-button x – UP/DOWN, blind>). If the push-button is briefly pressed again, the blind is stopped. If the push-button is pressed and held, the blind will move (object <Push-button x – UP/DOWN, blind>). If the push-button is released, the blind will be stopped (object <Push-button x – Step/stop, blind>). The corresponding move direction is determined by inverting the direction of the last actuation of the push-button.

UP (short: move, long: move/stop)

If the push-button is pressed shortly, the blind will move upwards (object <Push-button x – UP/DOWN, blind>). If the push-button is briefly pressed again, the blind is stopped. If the push-button is pressed and held, the blind will move upwards (object <Push-button x – UP/DOWN, blind>). If the push-button is released, the blind will be stopped (object <Push-button x – Step/stop, blind>).

DOWN (short: move, long: move/stop)

If the push-button is pressed shortly, the blind will move downwards (object <Push-button x – UP/DOWN, blind>). If the push-button is briefly pressed again, the blind is stopped. If the push-button is pressed and held, the blind will move downwards (object <Push-button x – UP/DOWN, blind>). If the push-button is released, the blind will be stopped (object <Push-button x – Step/stop, blind>).

UP (short: step/stop, long: move)

If the push-button is pressed shortly, the angle of the slats is adjusted upwards (object <Push-button x – Step/stop, Blind>) or the blind is stopped if it was moving before. If the push-button is pressed and held, the blind will move upwards (object <Push-button x – UP/DOWN, Blind>)

DOWN (short: step/stop, long: move)

If the push-button is pressed shortly, the angle of the slats is adjusted downwards (object <Push-button x – Step/stop, Blind>) or the blind is stopped if it was moving before. If the corresponding push-button is pressed and held, the Blind will move downwards (object <Push-button x – UP/DOWN, Blind>).

Parameter **Advanced functions blind** is only visible if **Operating concept push-button x = Two-button operation** is set. It can only be changed if the **Blind function left** is configured to *UP (short: step/stop, long: move)* or *DOWN (short: step/stop, long: move)*.

Advanced functions blind**Not active**

Move shading (double-click: long/short)

Not active

The advanced function is not activated.

Move shading

(double-click: long/short)

If the push-button is pressed and held (>0.5 sec) at first and then pressed again within one second, an UP/DOWN telegram is sent via the object

<Push-button x, double-click – UP/DOWN, move shading>.

Parameter **Scene function** specifies which commands are triggered by the push-button with the basic function scene. (explanations regarding scenes → [chapter 3.4](#))

Scene function

Scene function = Decentralised scene saving (in actuator)

Recall scene

Recall / save scene

Scene function = Local scene saving (in push-button)

Recall / save scene 1

Recall / save scene 2

...

Recall / save scene 8

Recall scene

A short press of the push-button results in a simple scene recall (object <Push-button x – Recall, scene>). Pressing the push-button longer does not have any function.

Note: Not all of the actuators do support the maximum number of 64 scenes.

Recall / save scene

A short press of the push-button results in a simple scene recall (object <Push-button x – Recall/save, scene>). If the push-button is pressed and held, a storage telegram is sent to the bus and the involved actuators save the current value. After 3 seconds, the LED starts flashing and after another 3 seconds the LED will light up permanently, indicating that the scene has been saved.

Note: Not all of the actuators do support the maximum number of 64 scenes.

Recall / save scene x

If the push-button is pressed shortly, the scene values saved on the push-button will be sent to all assigned actuators.

When using the saving function on the parameter page "Scene module" (→ [chapter 2.5.1](#)), the scenes can also be saved.

Parameter **Scene number** determines the respective scene number in the actuator.

Scene number

1..64

Parameters

Parameter **Forced position function** specifies the behaviour for a brief press.

Forced position function	Forced switch on (11) <i>Forced switch off (10)</i> <i>Cancel forced position (00)</i>		
<i>Forced switch on (11)</i>	1	1	If the push-button is only pressed shortly, a forced telegram with bit 0 = 1 and bit 1 = 1 is sent to the object <Push-button x – Forced position>.
<i>Forced switch off (10)</i>	1	0	If the push-button is only pressed shortly, a forced telegram with bit 0 = 0 and bit 1 = 1 is sent to the object <Push-button x – Forced position>.
<i>Cancel forced position (00)</i>	0	0	If the push-button is pressed shortly, the forced position is disabled and sent to bit 0 = 0 and bit 1 = 0. The control system will then be released for normal operation again.
		bit 0	forced state, polarity: 1 = ON/DOWN ; 0 = OFF/UP
		bit 1	forced position, polarity: 1 = active ; 0 = inactive

Parameter **Sequence module function** specifies the behaviour for a press.

Sequence module function	Start <i>Start (short) / Stop (longer press)</i>
<i>Start</i>	If the push-button is pressed, the sequence defined on the parameter page "Sequence module" (→ chapter 2.4.1) will start. Since the sequence cannot be stopped by a press of a push-button, care must be taken in order to ensure that the parameter Restart sequence module after last switching point? is set to <i>No</i> .
<i>Start (short) / Stop (longer press)</i>	If the push-button is pressed shortly, the sequence defined on the parameter page "Sequence module" (→ chapter 2.4.1) will start. Pressing the push-button longer will stop the sequence.

Parameter **LED function** specifies when the LED of the corresponding push-buttons should light up.

LED function	Not active (always switched off) <i>Orientation light (always switched on)</i> <i>Push-button status (internal signal)</i> <i>Status signal LED object (external signal)</i> <i>RGB signal LED object (external signal)</i> <i>Press: ON / Release: OFF (feedback)</i>
<i>Not active (always switched off)</i>	The LED is always switched off.
<i>Orientation light (always switched on)</i>	The LED is always switched on. The parameters LED colour and LED function overridable with object signal LED are visible.
<i>Push-button status (internal signal)</i>	The LED is internally linked to the first push-button object (group address) of <Push-button x – ON/OFF, ...>. The parameters LED display mode , LED colour and the LED function overridable with object signal LED are visible. Note: This function is only useful in combination with the push-button function switching or dimming.
<i>Status signal LED object (external signal)</i>	If an ON telegram is sent to the object <Push-button x, signal LED – Show on LED>, the LED will light up. If an OFF telegram is sent to the object <Push-button x, signal LED – Show on LED>, the LED will go out. The parameters LED display mode and LED colour are visible. The parameter LED function overridable with object signal LED is permanently set to <i>No</i> .

<i>RGB signal LED object (external signal)</i>	<p>If an RGB telegram is sent to the object <Push-button x, signal LED – Show on RGB LED>, the LED will light up in the corresponding colour. The RGB telegram must contain the proportions of the colour values for red, green and blue. In order to turn off the LED, the value 0 must be sent to all 3 colour values.</p> <p>The parameter LED display mode is visible. The parameter LED function overridable with object signal LED is permanently set to <i>No</i>.</p>
<i>Press: ON / Release: OFF (feedback)</i>	<p>If the push-button is pressed, the LED will light up, if it is released, the LED will go out.</p> <p>The parameters LED colour and LED function overridable with object signal LED are visible.</p>

Parameter **LED display mode** specifies the lighting mode of the LED.

LED display mode	<p>LED function = <i>Push-button status (internal signal)</i> LED function = <i>Status signal LED object (external signal)</i> Status normal <i>Status inverted</i> <i>Status normal flashing</i> <i>Status inverted flashing</i> <i>Status normal soft-flashing</i> <i>Status inverted soft-flashing</i></p> <p>LED function = <i>RGB signal LED object (external signal)</i> Status normal <i>Status normal flashing</i> <i>Status normal soft-flashing</i></p>
<i>Status normal</i>	The LED lights up if an ON telegram is present at the corresponding object.
<i>Status inverted</i>	The LED lights up if an OFF telegram is present at the corresponding object.
<i>Status normal flashing</i>	The LED lights up if an ON telegram is present at the corresponding object.
<i>Status inverted flashing</i>	The LED starts flashing if an OFF telegram is present at the corresponding object.
<i>Status normal soft-flashing</i>	The LED starts soft-flashing (→ chapter 3.2.2) if an ON telegram is present at the corresponding object.
<i>Status inverted soft-flashing</i>	The LED starts soft-flashing (→ chapter 3.2.2) if an OFF telegram is present at the corresponding object.

Parameter **LED colour** specifies in which colour the LED lights up.

LED colour	<i>Red / Green / Blue / White / Yellow / Vviolet</i> <i>User colour 1–4</i>
<i>Red / Green / Blue / White / Yellow / Violet</i>	The LED lights up in the selected colour.
<i>User colour 1–4</i>	The LED lights up in the colour mixed (see also chapter 3.2.3) on the parameter page "LED colours" (→ chapter 2.3.4).

Parameter **LED function overridable with object signal LED** specifies if the LED can be overridden.

LED function overridable with object signal LED	No Yes
<i>No</i>	The LED cannot be overridden.
<i>Yes</i>	<p>The LED is overridden as soon as the corresponding telegram is present at the object <Push-button x, signal LED – Override/show on LED>.</p> <p>The parameters Signal LED display mode and Signal LED colour are visible.</p>

Parameters

Parameter **Signal LED display mode** specifies how the LED is to be overridden.

Signal LED display mode	<i>Status normal</i> <i>Status inverted</i> Status normal flashing <i>Status inverted flashing</i> <i>Status normal soft-flashing</i> <i>Status inverted soft-flashing</i>
<i>Status normal</i>	The LED lights up and overrides the normal LED function if an ON telegram is present at the object <Push-button x, signal LED – Override/show on LED>.
<i>Status inverted</i>	The LED lights up and overrides the normal LED function if an OFF telegram is present at the object <Push-button x, signal LED – Override/show on LED>.
<i>Status normal flashing</i>	The LED starts flashing and overrides the normal LED function if an ON telegram is present at the object <Push-button x, signal LED – Override/show on LED>.
<i>Status inverted flashing</i>	The LED starts flashing and overrides the normal LED function if an OFF telegram is present at the object <Push-button x, signal LED – Override/show on LED>.
<i>Status normal soft-flashing</i>	The LED starts soft-flashing (→ chapter 3.2.2) and overrides the normal LED function if an ON telegram is present at the object <Push-button x, signal LED – Override/show on LED>.
<i>Status inverted soft-flashing</i>	The LED starts soft-flashing (→ chapter 3.2.2) and overrides the normal LED function if an OFF telegram is present at the object <Push-button x, signal LED – Override/show on LED>.

Parameter **Signal LED colour** specifies with which colour the LED is overridden.

Signal LED colour	<i>Red / Green / Blue / White / Yellow / Violet</i> <i>User colour 1–4</i>
<i>Red / Green / Blue / White / Yellow / Violet</i>	The LED lights up in the selected colour.
<i>User colour 1–4</i>	The LED lights up in the colour mixed (see also chapter 3.2.3) on the parameter page "LED colours" (→ chapter 2.3.4).

Parameter **Longer press left push-button** and/or **Longer press right push-button** specifies if an additional function is carried out by pressing the push-button longer.

Longer press left push-button

Longer press right push-button *Not active*

Active

Not active

Pressing the push-button longer does not have any effect.

Active

By pressing the push-button longer, an additional command is sent to an additional object.

The object <Push-button x (longer press)> is visible.

The parameters **Time for longer press** and **Longer press function** are visible.

Parameter **Time for longer press** specifies for how long the push-button is to be pressed in order to trigger the command defined with the parameter **Longer press function**.

Time for longer press *0.5 sec. / 1 sec. / 2 sec. .. 10 sec.*

Parameter **Longer press function** specifies the additional command to be triggered by pressing the push-button longer.

Longer press function

Switching

Dimming value in %

Blind UP / DOWN

Value

Scene

Switching

By pressing the push-button longer, the telegram specified with the parameter **Switching function** is sent to the object <Push-button x (longer press) – ON/OFF, switching>.

Dimming value in %

By pressing the push-button longer, the telegram specified with the parameter **Dimming value function** is sent to the object <Push-button x (longer press) – Value, dimming>.

Blind UP / DOWN

By pressing the push-button longer, the telegram specified with the parameter **Blind function** is sent to the object <Push-button x (longer press) – UP/DOWN, Blind>.

Value

By pressing the push-button longer, the value specified with the parameter **8 bit value** (0..255) is sent to the object <Push-button x (longer press) – Send, value>.

Scene

By pressing the push-button longer, the scene saved in the actuator with the parameter **Scene number** (1..64) is recalled.
(object <Push-button x (longer press) – Recall, scene>)

Parameters

2.3.3 Parameter page "LED brightness and flashing speed"

Parameter **LED brightness during normal operation** specifies how bright the LED will light up during normal operation. The luminosity is indicated as a percentage of the maximum possible luminosity of the LED.

LED brightness during normal operation *0..100 (100)*

Parameter **Night reduction LEDs function** specifies if the LEDs are to shine with a decreased brightness during night-time.

Night reduction LEDs function **Not active**
ON = night reduction active / OFF = inactive
ON = inactive / OFF = night reduction active

Not active

The LEDs always shine with the specified brightness value for normal operation.

Note: *Not active* is not to be interpreted as «.. are not lit». Only the object 25 <Night reduction LEDs – Decrease brightness> is not displayed thus making the night reduction function unavailable.

ON = night reduction active /
OFF = inactive

If an ON telegram is sent to object 25 <Night reduction LEDs – Decrease brightness>, the LEDs only shine with the degree of brightness specified for night reduction.

If an OFF telegram is sent to object 25 <Night reduction LEDs – Decrease brightness>, the LEDs will return to the illumination value for normal operation.

The object 25 <Night reduction LEDs – Decrease brightness> is visible. The parameter **LED brightness during night reduction** is visible.

ON = inactive /
OFF = night reduction active

If an OFF telegram is sent to object 25 <Night reduction LEDs – Decrease brightness>, the LEDs and only shine with the degree of brightness specified for night reduction.

If an ON telegram is sent to object 25 <Night reduction LEDs – Decrease brightness>, the LEDs and will return to the illumination value for normal operation.

The object 25 <Night reduction LEDs – Decrease brightness> is visible. The parameter **LED brightness during night reduction** is visible.

Parameter **LED brightness during night reduction** specifies how bright the LEDs will be lit during night operation, which is activated via the object 25 <Night reduction LEDs – Decrease brightness>. The luminosity is indicated as a percentage of the maximum possible luminosity of the LED.

LED brightness during night reduction *0..100 (50)*

Parameter **Flashing speed LEDs** specifies at what cadence the LEDs will be flashing.

Flashing speed LEDs *very fast (0.5 sec.)*
fast (1 sec.)
slow (2 sec.)
very slow (4 sec.)

2.3.4 Parameter page "LED colours"

On the parameter page "LED colours", two user-specific colours (LED user colour 1–4) can be «mixed» in an additive manner in order to match them to the environment.

Parameter **Red**, **Green** and **Blue** determine the numeric portion of the colours red, green and blue in the user colour. Further information on additive colour mixing → [chapter 3.5](#).

Red

Green

Blue 0..255

In the ETS the colours *Red*, *Green*, *Blue*, *White*, *Yellow* and *Violet* are predefined (→ [chapter 3.2.3](#)).

Parameter **Use colour correction** allows you to compensate colour differences of LEDs between two different push-buttons.

Use colour correction

No

Yes

No

The colour correction is not used.

Yes

The colour correction is used for all LEDs.

The parameters **Red**, **Green** and **Blue** are visible in percent (-100..30) for the correction.



Note: With these parameters, no colours may be set. They should only be used for any correction of colour deviations which may be required in the case of a deviating aging of the LEDs or with minor colour differences of LEDs of different batches.

2.3.5 Parameter page "General disabling"

With the object 24 <All involved push-buttons – Disable push-buttons>, all or individual push-buttons (parameter page "Disable push-buttons" → [chapter 2.3.6](#)) can be disabled. If a push-button is disabled, it is no longer able to send a signal until the push-button is enabled again. A disabled push-button can be signalled by means of LEDs.

Parameter **Disable push-buttons function** determines the polarity of the disabling object 24 <All involved buttons – Disable buttons>.

Disable push-buttons function

Not active

ON = disable / OFF = operation

ON = operation / OFF = disable

Not active

The push-buttons cannot be disabled.

ON = disable / OFF = operation

If an ON telegram is sent to the object 24, the push-buttons will be disabled depending on the configuration on the parameter page "Disable push-buttons".

If an OFF telegram is sent to object 24, these push-buttons will be enabled again.

The object 24 <All involved push-buttons – Disable push-buttons> is visible.

ON = operation / OFF = disable

If an OFF telegram is sent to the object 24, the push-buttons will be enabled depending on the configuration on the parameter page "Disable push-buttons".

If an ON telegram is sent to object 24, these push-buttons will be enabled again.

The object 24 <All involved push-buttons – Disable push-buttons> is visible.

Parameters

Parameter **Behaviour for disabling event** specifies if and which telegrams are sent before the push-buttons are disabled.

Behaviour for disabling event	<i>Maintain state and disable</i> <i>ON/DOWN, then disable</i> <i>OFF/OFF, then disable</i>
<i>Maintain state and disable</i>	Only the push-button is disabled. The state of the actuator is not changed.
<i>ON/DOWN, then disable</i>	If the push-button is disabled, an ON telegram is sent to the corresponding group address (1 / DPT 1.001) and the push-button is disabled.
<i>OFF/OFF, then disable</i>	If the push-button is disabled, an OFF telegram is sent to the corresponding group address (0 / DPT 1.001) and the push-button is disabled.



Note: During disabling, the telegram is always sent via the group address of the push-button object with the lowest object number. The telegram is only sent via 1 bit objects. If the object has another data type, no telegram will be sent.

Parameter **LED display mode, if disabled** specifies if and how the LEDs will react if the push-button has been disabled via the object 24 <All involved push-buttons – Disable push-buttons>.

LED display mode, if disabled	<i>Not active (is not overridden)</i> <i>ON (switched on if disabled)</i> <i>OFF (switched off if disabled)</i> <i>Sequence (3x flashing / 3x pause if disabled)</i> <i>Flashing (flashing if disabled)</i> <i>Soft-flashing (soft-flashing if disabled)</i>
<i>Not active (is not overridden)</i>	If the push-button is disabled, this does not have an effect on the state of the LED. If available and parameterised, the LED will show its «normal» function.
<i>ON (switched on if disabled)</i>	If the push-button is disabled, the LED will be lit permanently.
<i>OFF (switched off if disabled)</i>	If the push-button is disabled, the LED will be switched off.
<i>Sequence (3x flashing / 3x pause if disabled)</i>	If the push-button is disabled, the LED starts flashing in a specific disable flashing sequence.
<i>Flashing (flashing if disabled)</i>	If the push-button is disabled, the LED will flash continuously.
<i>Soft-flashing (soft-flashing if disabled)</i>	If the push-button is disabled, the LED will soft-flash continuously (→ chapter 3.2.2).

The flashing speed is determined by the general parameter **Flashing speed LEDs** on the parameter page "LED brightness and flashing speed" (→ [chapter 2.3.3](#)).

Parameter **LED colour** specifies in which colour the LED lights up.

LED colour	<i>Red / Green / Blue / White / Yellow / Violet</i> <i>User colour 1–4</i>
<i>Red / Green / Blue / White / Yellow / Violet</i>	The LED lights up in the selected colour.
<i>User colour 1–4</i>	The LED lights up in the colour mixed (see also chapter 3.2.3) on the parameter page "LED colours" (→ chapter 2.3.4).

2.3.6 Parameter page "Disable push-buttons"

On the parameter page "Disable push-buttons", individual push-buttons can be excluded from the disabling function on the parameter page "General disabling" (→ [chapter 2.3.5](#)).



Note: The following parameter is available for each of the individual push-buttons. To simplify matters, the parameterisation is described using only one parameter as example.

Parameter **Push-button x** determines whether or not the push-button can be disabled via object 24 <All involved push-buttons – Disable push-buttons>.

Push-button x	Yes
	<i>No</i>

2.4 Parameters sequence module

2.4.1 Parameter page "Sequence module"

Parameter **Sequence module** enables the definition of a sequence (→ [chapter 3.3](#)) with up to 8 switching points which can be parameterised.

Sequence module	Not active
	<i>Active</i>
<i>Not active</i>	No sequence has been defined. All follow-up parameters are hidden.
<i>Active</i>	The sequence can be defined with up to 8 parameterisable switching points. The object 87 <Sequence module – Recall sequence> is visible. Object 88 <Sequence module – Status> is visible.

Parameter **Switching point x** specifies if the switching point is passed through in the sequence.

Switching point x	Not active
	<i>Active</i>
<i>Not active</i>	The switching point is not active.
<i>Active</i>	The switching point is active and will be «passed through». The command to be executed is specified with the parameter Switching point x function on the parameter page "Switching point x" (→ chapter 2.4.2). The object <Switching point x – ON/OFF, switching> is visible.

Parameter **Restart sequence module after last switching point?** specifies whether the sequence is restarted from the beginning after the last switching point has been processed.

Restart sequence module after last switching point?	No
	<i>Yes</i>
<i>No</i>	The sequence can be stopped by pressing the push-button longer when an OFF telegram is received at the object 87 <Sequence module – Recall sequence>. If it is not stopped manually, this will happen after the last switching point.
<i>Yes</i>	The sequence restarts from the beginning. It is only stopped by the press of a push-button (if the push-button has been parameterised accordingly) or if an OFF telegram is received at the object 87 <Sequence module – Recall sequence>.

2.4.2 Parameter page "Switching point x"

Parameter **Time interval to starting point** and **Time interval to previous active switching point** indicate the time interval to the starting point or to the previous switching point in seconds.

Time interval to starting point

Time interval to previous active switching point 0..3600 (0)

Parameter **Switching point x function** specifies the function to be executed at the corresponding switching point.

Switching point x function

Switching

Dimming value in %

Blind UP / DOWN

Value

Scene

Switching

The telegram specified with the parameter **Switching function** is sent to the object <Switching point x – ON/OFF, switching>.

Dimming value in %

The value specified with the parameter **Dimming value function** is sent to the object <Switching point x – Value, dimming>.

Blind UP / DOWN

The telegram specified with the parameter **Blind function** is sent to the object <Switching point x – UP/DOWN, Blind>.

Value

The value specified with the parameter **8 bit value** (0..255) is sent to the object <Switching point x – Send, value>.

Scene

The scene saved in the actuator with the parameter **Scene number** (1..64) is recalled (object <Switching point x – Recall, scene>)

2.5 Parameters scene module

2.5.1 Parameter page "Scene module"

On the parameter page "Scene module", the number of group addresses and the functioning of the scene saving is specified when using local scene saving.

Parameter **Scene function** specifies the type of scenes (→ [chapter 3.4](#)).

Scene function	<i>Decentralised scene saving (in actuator)</i> <i>Local scene saving (in push-button)</i>
<i>Decentralised scene saving (in actuator)</i>	The scene values are remotely saved in the actuators (8-bit scene).
<i>Local scene saving (in push-button)</i>	The scene values are locally saved in the KNX push-button (conventional scene).

Parameter **Number of scene values per scene** specifies the maximum number of scene values per scene. The value applies to all scenes.

Number of scene values per scene	<i>max. 10 values/objects per scene</i> <i>max. 15 values/objects per scene</i>
<i>max. 10 values/objects per scene</i>	Per scene, a maximum of 10 different scene values can be recalled and saved.
<i>max. 15 values/objects per scene</i>	Per scene, a maximum of 15 different scene values can be recalled and saved.

Parameter **Scene mode for the user during the operation** specifies if and how scenes can be saved by the user.

Scene mode for the user during the operation	<i>Only recall scene</i> <i>Recall scene and save all</i> <i>Recall scene and save selectively</i>
<i>Only recall scene</i>	The scene can be recalled by the push-button but it cannot be saved. The saving of scenes is only carried out via ETS.
<i>Recall scene and save all</i>	The scene can be recalled and saved by the push-button. If the push-button is pressed and held, the current state of all group addresses assigned to the scene is queried and saved. After approx. 3 seconds, the LED starts flashing quickly, after another 4 seconds it will light up permanently, indicating that the scene has been saved. If the push-button is pressed for a very long time (approx. 12 seconds), the scene will be deleted.
<i>Recall scene and save selectively</i>	The scene can be recalled and saved by the push-button. Only changed values will be taken into account in the new scene. Group addresses, that were not changed during the scene saving procedure, will not be saved. If the push-button is pressed and held, the LED will start flashing after approx. 3 seconds. If the push-button is released now, the desired loads can be set within a time of 4 minutes. If the push-button is pressed and held again, the LED will light up permanently after approx. 3 seconds indicating that the scene has been saved. If the push-button is pressed shortly during the programming, the programming mode will be exited without saving. If the push-button is pressed for a very long time (approx. 12 seconds), the scene will be deleted.



Note: The actuator value will not be saved in the scene if **Presetting scene value x = Disabled** (parameter page "Scene x [value 1...10/1...15]" → [chapter 2.5.3](#)).

Parameters scene module

Parameter **Transmission delay between scene telegrams** specifies the duration of the pauses between the individual telegrams of a scene when the scene is recalled.

Transmission delay between scene telegrams *25 ms / 50 ms / 75 ms / 100 ms*



Note: The more quickly the telegrams follow in sequence, the higher the bus load.

Parameter **Recall scene via object** is permanently set to 1 = *recall scene*. Via the corresponding objects in the ETS, scenes can be recalled using additional switches by sending an ON telegram to the corresponding object number.

2.5.2 Parameter page "Data type scene value 1...10/1..15"



Note: For the parameterisation of the data types of the scene values per scene, a designated parameter page is available. However, since both parameter pages are progressively structured, they will both be explained together at this point.

Parameter **Data type scene value x** specifies the data type (DPT) of the individual save points.

Data type scene value x *1 bit (switching ON/OFF, blind UP/DOWN)*
8 bit (dimming value, blind value)

1 bit (switching ON/OFF, blind UP/DOWN)

When the scene is triggered, a 1 bit telegram is sent to the corresponding group address. This way, the state of the object <Scene value x – ON/OFF, UP/DOWN> is switched according to the state saved in the scene.

8 bit (dimming value, blind value)

When the scene is triggered, a 8 bit telegram is sent to the corresponding group address. This way, the value of the object <Scene value x – Send, value> is switched according to the value saved in the scene.

2.5.3 Parameter page "Scene x [value 1...10/1...15]"



Note: For the parameterisation of the presetting of the scene values per scene, a designated parameter page is available. However, since all parameter pages are identically structured, they will be explained altogether at this point.

Parameter **Presetting scene value x** specifies the scene value to be sent. During operation, new values can be saved via the push-button.

Presetting scene value x **Data type scene value x = 1 bit (switching ON/OFF, blind UP/DOWN)**
Disabled

Switching ON, blind DOWN

Switching OFF, blind UP

Data type scene value x = 8 bit (dimming value, blind value)

Disabled

0 % / 5 % / 10 % / 15 % .. 100 %

Disabled

The scene value x is not involved in the scene x. Therefore, the corresponding group address remains unchanged upon recall of scene.

Switching ON, blind DOWN

When the scene is triggered, a 1 bit telegram with the value (1) is sent to the corresponding group address (DPT 1.001/1.008).

This causes the light to be switched on or the blind to be closed.

Switching OFF, blind UP

When the scene is triggered, a 1 bit telegram with the value (0) is sent to the corresponding group address (DPT 1.001/1.008).

This causes the light to be switched off or the blind to move up.

0 % / 5 % / 10 % /

15 % .. 100 %

When the scene is triggered, a 8 bit telegram with the preset value is sent to the corresponding group address (DPT 5.001). This causes the light to be adjusted to the desired brightness or the blind to move to the corresponding position.

3 Functional description

3.1 Behaviour after ETS download or bus voltage return

After the application has been downloaded to the device by means of the ETS, the device will restart. After a few seconds, the device will be ready for operation.

If all of the LEDs on the push-button are flashing red, this means that the download could not be carried out properly or that the ETS application is not compatible with the hardware.

Procedure:

1. Shortly disconnect the device from the KNX bus voltage
2. Check the application compatibility
3. Check the physical address
4. Download the application again

Attention:

- > KNX devices with the additional designation **RGB** can only be programmed using the corresponding application with the additional designation RGB.
- > Older applications (without the additional designation RGB) cannot be loaded to the present hardware with the additional designation **RGB**. Feller shall not assume any liability or consequential costs for projecting errors.

After an interruption of the bus voltage, the device will start automatically after the voltage has returned. The settings made during parameterisation will remain unchanged.



Note: Depending on the settings on the parameter pages "General disabling" and "Disable push-buttons", it may occur that telegrams are sent to the bus after the restart.

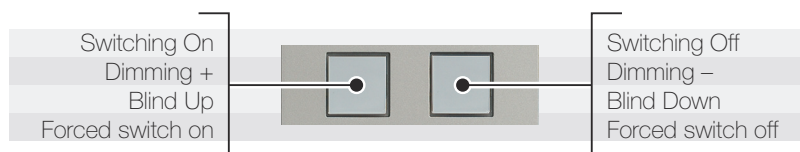
3.2 Push-button

3.2.1 Operating concept

Thanks to a flexible operating concept, the EDIZIOdue elegance KNX push-button RGB can be used in different ways.

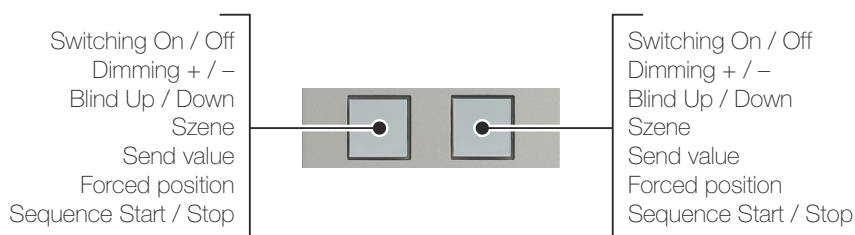
Two-button operation

The same consumer is always actuated, but the function depends on whether the left push-button or the right push-button is pressed.



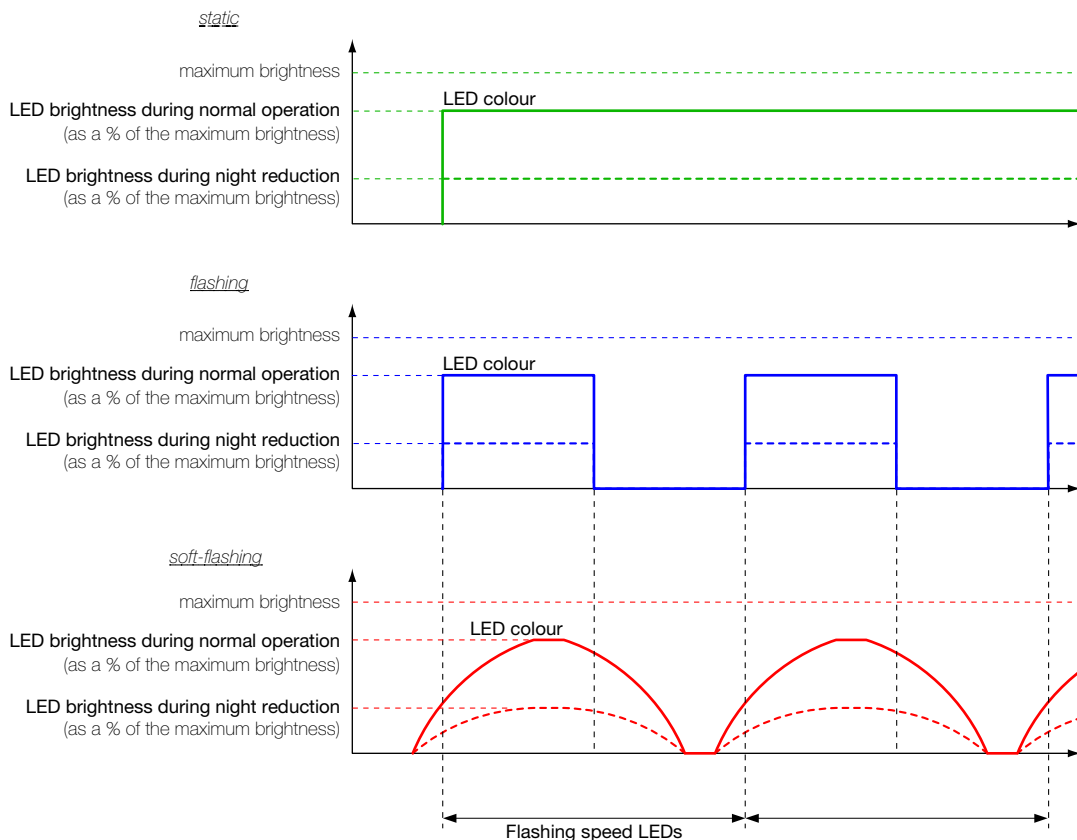
Single-button operation

The two push-buttons are independent of each other, each controls a different consumer.



3.2.2 LEDs

If desired, the KNX push-buttons RGB can be equipped with LEDs, in which case each LED can be individually configured. They can optionally be activated or deactivated (orientation light), serve as a status display or be used for feedback purposes (LED function). A control via separate communication objects is also possible. The LEDs are able to display a static, flashing or soft-flashing status (display mode). The soft-flashing status can be used as subtle source of information since the LEDs appear more vivid due to the increasing and decreasing dimming brightness.



An individual colour can be set for each LED in the ETS. Optionally, the function of the LED can be overridden via the bus thus enabling a change in the colour and the display mode of individual LEDs depending on priority. For the KNX push-button RGB, two user colours can be individually mixed on the parameter page "LED colours". This enables an optimal adjustment of the LEDs to both the colours of the EDIZIOdue elegance design parts as well as to the environment (see also [chapter 3.5](#)).

The brightness during normal operation and the flashing speed of all LEDs is globally defined on the parameter page "LED brightness and flashing speed". This ensures a unified visual appearance and a synchronised flashing of the LEDs ¹⁾. The brightness can optionally be adjusted during operation via a 1 bit communication object. This adjustment can be used to reduce the brightness during night-time, for example. If you wish to adjust the brightness via the object, the parameter **Night reduction LEDs function** needs to be set. In this case, the object 25 <Night reduction LEDs – Decrease brightness> will be visible in the ETS.

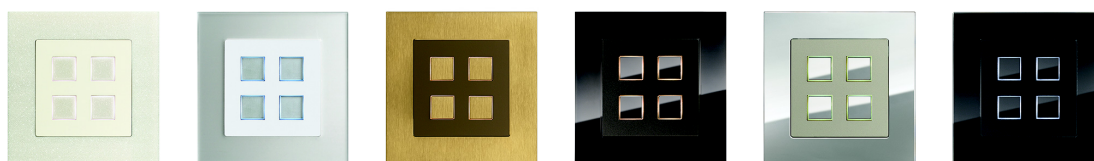
¹⁾ The increasing and decreasing dimming brightness of the soft-flashing LED starts at approx. 10% of the flashing speed prior to switch on/off of the flashing LED. When reaching the upper or lower peak, this state is maintained for approx. 10% of the flashing speed.

3.2.3 LED colours

In the ETS, the following colours are predefined:

Colour	RGB value		
	Red:	Green:	Blue:
<i>Red</i>	102	000	000
<i>Green</i>	000	098	008
<i>Blue</i>	000	000	255
<i>White</i>	105	128	110
<i>Yellow</i>	128	110	000
<i>Violet</i>	089	000	255

The following colour definitions are recommended for the EDIZIOdue elegance real materials:



..91/..2B ..92/..1G ..97/..0K ..99/..0J ..95/..0G ..90/..1E

	EDIZIOdue elegance Echtmaterial	RGB value		
		Red:	Green:	Blue:
..91/..2B	artic / marmor weiss	135	158	110
..92/..1G	pearl, effect / spiegel satin	090	128	140
..97/..0K	hazel / messing champagne	124	131	074
..99/..0J	mocca effect / schwarzgold poliert	144	131	074
..95/..0G	stone effect / chromstahl poliert	124	160	074
..90/..1E	graphite / glas schwarz	105	128	110

The following definitions generate the same colour effect:

Colour	RGB value		
	Red:	Green:	Blue:
Orange	204	038	000
Pink	191	000	077
Warm white	124	131	074
Cold white	090	128	140

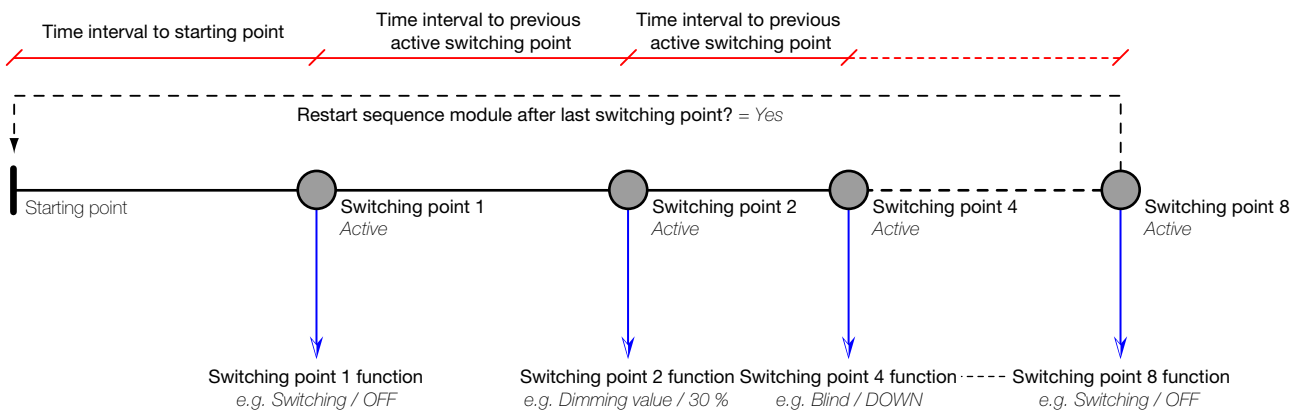
3.3 Sequence module

A possible application for the sequence module in residential buildings is, for example, a time-delayed «Central OFF». Compared to the pure scene solution, this provides the advantage of «organically shutting down» the house and helps to avoid current spikes. At first, the light in the hallway is reduced to 30%, then, the blinds are closed and the lights are turned off on all floors on a time-delayed basis, before the lights in the hallway are also turned off.

Since the time interval to the previous switching point can be up to an hour long, the restart can also be activated by means of a simple presence simulation.

In functional buildings, the sequence module can also be used for presentations, for example. At first, the beamer is turned on, then, the blinds are closed after 30 seconds, and after another 15 seconds, the lighting is dimmed down.

A sequence of up to 8 parametrisable switching points can be defined on the parameter page "Sequence module". There is an output object for each switching point. Each switching point is triggered with a time-delay following the starting point or previous switching point.



The sequence is started by pressing a push-button, provided that this push-button has been parameterised accordingly (→ [chapter 2.3.2](#)), or by writing ON into the object 87 <Sequence module – Recall sequence>.

The sequence is stopped by pressing a push-button for a longer time, provided that this push-button has been parameterised accordingly (→ [chapter 2.3.2](#)), or by writing OFF into the object 87 <Sequence module – Recall sequence>.

While the sequence is processed, the object 88 <Sequence module – Status> is set to ON. At the end, it is set back to OFF.

If the sequence is started again by a press of a push-button or by writing ON into the object 87 <Sequence module – Recall sequence> while it is being processed, the sequence will restart from the beginning (retrigger).

3.4 Scene module

With a scene, a group of actuators can be set to a desired state simultaneously by a press of a push-button. This way, the desired ambience can be achieved by pressing a push-button (e.g. meal, leaving the house, blinds down, lighting off, set heating to standby operation etc.). This scene functionality often provides advantages in functional buildings as well. A museum or a gallery could, for example, showcase the exhibition objects in the right light by a press of a push-button.

There are two concepts for the KNX push-button RGB with regard to triggering or saving scenes:

Decentralised scene saving in the actuator (8-bit scene)

The scene values are remotely saved in the scene storage of the actuator. At the press of a push-button, a preset scene number (1..64) is sent to the bus via a separate communication object. This way, the scene is called up in the actuator or – when using the saving function – also saved. The KNX push-button RGB and the actuators communicate with each other via an 8-bit telegram.

For the 8-bit scene, only one telegram is sent in order to control all corresponding actuators simultaneously.

For every push-button, it can be set whether a scene is only to be recalled or if it is to be recalled and saved using the parameter **Scene function** (→ [chapter 2.3.2](#)). When saving the scene, care must be taken in order to ensure that all involved devices are in the right state. A scene cannot be deleted by the user.

Local scene saving in the push-button (conventional scene)

The scene values are locally saved in the KNX push-button RGB. At the press of the push-button, the corresponding scene value is sent to all involved actuators via the bus. A snapshot of the default values and/or actuator states can be saved as scene value. The scenes are permanently stored and remain available even after a voltage interruption.

Up to 15 group addresses can be assigned to the scene function. A maximum of 8 different scenes is possible. The same actuators and/or group addresses participate in each scene.

For the conventional scene, up to 15 telegrams are serially sent to the bus (delay time between the sending of the individual telegrams can be set using the parameter **Transmission delay between scene telegrams**). This causes a «high» bus load and may result in visible delays when scenes are called up. (When using the 8-bit scene, this mechanism does not occur.)

The parameter **Scene mode for the user during the operation** can be used to set whether scenes can only be recalled or if they can be recalled and saved (all or selective) (→ [chapter 2.5.1](#)).

The link of the KNX push-button RGB with the actuators is established via the scene objects. They must be linked to the same ETS group address that is used to link the local push-button and display objects to the actuator.

In order to properly configure the KNX push-button RGB, please also note the following points:

- Enter the correct object type (1 bit for switching, 8 bit for dimming brightness or blind position) in the settings on the parameter page "Data type scene value 1...10/1...15" (→ [chapter 2.5.2](#)).
- In the settings on the parameter page "Scene x [value 1...10/1...15]" (→ [chapter 2.5.3](#)), define the parameters **Presetting scene value 1** to **Presetting scene value 10/15**.
Note: These parameters are only valid until a new scene is saved. If the device is programmed with the ETS again afterwards, all scenes are reset to the values saved in the ETS (presetting).
- The transfer (Ü) and/or read (L) flag must be set for the actuator for 1-byte scene groups. Both flags, however, may only be set for one actuator per scene group if several actuators are connected to a scene group.
- In the parameter settings **Scene mode for the user during the operation** = *Recall scene and save all* on the parameter page "Scene module" (→ [chapter 2.5.1](#)), the read flag (L) must be set for the 1-byte object of the actuator and the current brightness/position of the actuator must be legible.
- In the parameter settings **Scene mode for the user during the operation** = *Recall scene and save selectively* on the parameter page "Scene module" (→ [chapter 2.5.1](#)), the transfer (Ü) flag must be set for the 1-byte object of the actuator and the current brightness/position of the actuator must be legible.



Notes:

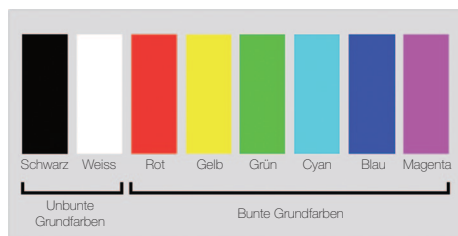
- > Depending on the programming via the ETS, a scene may also be called up by other push-buttons (so called extensions) by means of an ON telegram.
- > The «Program scene» function can be disabled via the ETS parameter settings so that a scene may only be recalled (parameter **Scene mode for the user during the operation** = *Only recall scene*). The scene can then not be programmed by the end user.
- > Not all of the actuators are scene-capable. Please note the relevant information provided in the product specifications of the manufacturers.

3.5 RGB colour theory

Source: Colour theory and colour design (www.ipsi.fraunhofer.de/~crueger/farbe/)

Man perceives light on a certain wavelength ranging from 380 nm (nanometer) to 750 nm as colours. There are three different types of colour-sensitive photoreceptors located in the retina of the human eye, also referred to as cones. They are sensitive for three different wavelength ranges of light, namely long-wave, medium-wave and short-wave light. The cones collect the rays of their wavelength that incidents in the human eye, and direct them to the brain, where the real colour perception evolves. We see long-wave light as red, medium-wave light as green and short-wave light as blue.

Primary colours

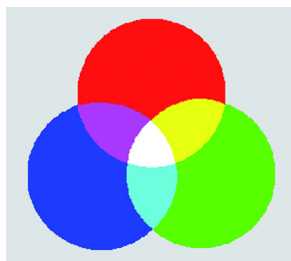


Combinations of 2 or 3 different wavelengths in equal proportions and full intensity result in overall 8 extreme colour perceptions, also referred to as primary colours.

The 8 primary colours are red, green, blue, cyan, magenta, yellow, white and black.

Black and white are the achromatic primary colours, the 6 others are chromatic primary colours.

The additive colour mixing (RGB)

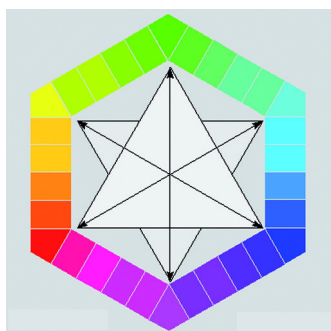


The RGB colour range is used for self-luminous (colour-displaying) systems that are subject to the principle of additive colour mixing, also referred to as light mixing. According to the three cone types of the human retina, it is based on the three primary colours red, green and blue. Brighter colour shades can be created by mixing. Yellow is created by mixing red and green, mixing green and blue results in cyan and blue mixed with red in magenta. If all three colours come together in full intensity and in equal proportions, they will create the colour white.

The LEDs of the KNX push-buttons RGB as well as colour television and the colour display of a computer are working based on this principle. In graphics software, it is known as the RGB model.

Colour hexagon

The colour hexagon consists of a triangle comprising the elementary colours red, green and blue and a triangle comprising the primary colours magenta, yellow and cyan.



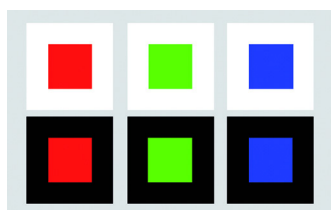
The colours are arranged in such a way that their mixed colour shades are located between the three elementary colours. Therefore, yellow is located between red and green, cyan between green and blue and magenta between blue and red. This way, two colours are facing each other that will complement each other and create the colour white when using the additive colour mixing. Such colour pairs are referred to as complementary colours.

The 6 primary colours are positioned in the corners of the hexagon, the mixed colour shades created from two neighbouring primary colours are located on the legs in between. The colour hexagon can be divided into two halves: one half contains cold colour shades while the other one contains warm ones. The warm colour shades range from green, yellow and red to magenta. The cold colour shades range from magenta, blue and cyan to green. Green and magenta are placed on the intersection points between

warm and cold and are considered neutral.

Simultaneous contrast

When informing yourself about colour design, you will soon discover that colours change their character depending on their environment. These are the effects of the simultaneous contrast.



Example:

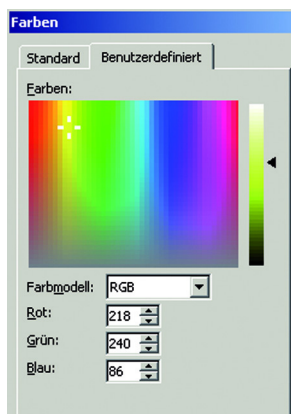
One and the same colour appears brighter in front of a dark background and darker in front of a bright background. A bright background will put a colour into the foreground and a dark background will decrease its effect. Achromatic environments let chromatic colours shine more brightly, which particularly applies to black.

This effect also occurs if the LED colour is combined with an EDIZIOdue cover set.

The effect of the simultaneous contrast is caused by the fact that the human eye is not made to reproduce colours as true to the physical values that they are based on as possible but is instead aiming at pointing out differences. This means that changing a colour shade in a colourful design – by adding a new colour shade or removing a colour shade – can fundamentally change the character of the design.

LED colours of the KNX push-buttons RGB

A user colour is defined by the numeric portion (0 ... 255) of the colours red, green and blue. The colour value 255 represents the full colour shade of a primary colour, while the colour value 0 indicates that no portions of this primary colour are included.



Auxiliary means such as colour mixers that are used in almost every computer programme can be used to define colours.

Numerous colour tables including colour patterns and their corresponding codes are available on the internet as well, e.g. www.ipsi.fraunhofer.de/~crueger/farbe/farb-must.html or www.farb-tabelle.de/de/farbtabelle.htm.

Please note that the colours mixed on your screen can only serve as general guide and **that the perception on site significantly depends on the combination of background – colour EDIZIOdue elegance design parts – lighting etc..**

Numerics	
8 bit value	10, 17
A	
Advanced functions blind	13
B	
Behaviour for disabling event	20
Blind function	12, 17
Blue	19
D	
Data type scene value	24
Dimming function	11
Dimming value function	17
Disable push-button	21
Disable push-buttons function	19
F	
Flashing speed LEDs	18
Forced position function	14
G	
Green	19
L	
LED brightness during night reduction	18
LED brightness during normal operation	18
LED colour	15, 20
LED display mode	15
LED display mode, if disabled	20
LED function	14
LED function overridable with object signal LED	15
Longer press function	17
Longer press left push-button	17
Longer press right push-button	17
N	
Night reduction LEDs function	18
Number of push-buttons	9
Number of scene values per scene	23
O	
Operating concept push-button	9
P	
Presetting scene value	24
Push-button function	10
R	
Recall scene via object	24
Red	19
Restart sequence module after last switching point?	21
S	
Scene function	13, 23
Scene mode for the user during the operation	23
Scene number	13, 17
Sequence module	21
Sequence module function	14
Signal LED colour	16
Signal LED display mode	16
Switching function	11, 17
Switching point	21
Switching point function	22

INDEX PARAMETERS

T

Time for longer press	17
Time interval to previous active switching point	22
Time interval to starting point	22
Transmission delay between scene telegrams	24

U

Use colour correction	19
---------------------------------	----

FELLER AG | Postfach | CH-8810 Horgen
Telefon +41 44 728 77 77 | Telefax +41 44 728 72 99

FELLER SA | Caudray 6 | CH-1020 Renens
Téléphone +41 21 653 24 45 | Téléfax +41 21 653 24 51

Service Line | Telefon +41 728 74 74 | info@feller.ch | www.feller.ch

10.KNX4801B-E.1309/130924


by Schneider Electric