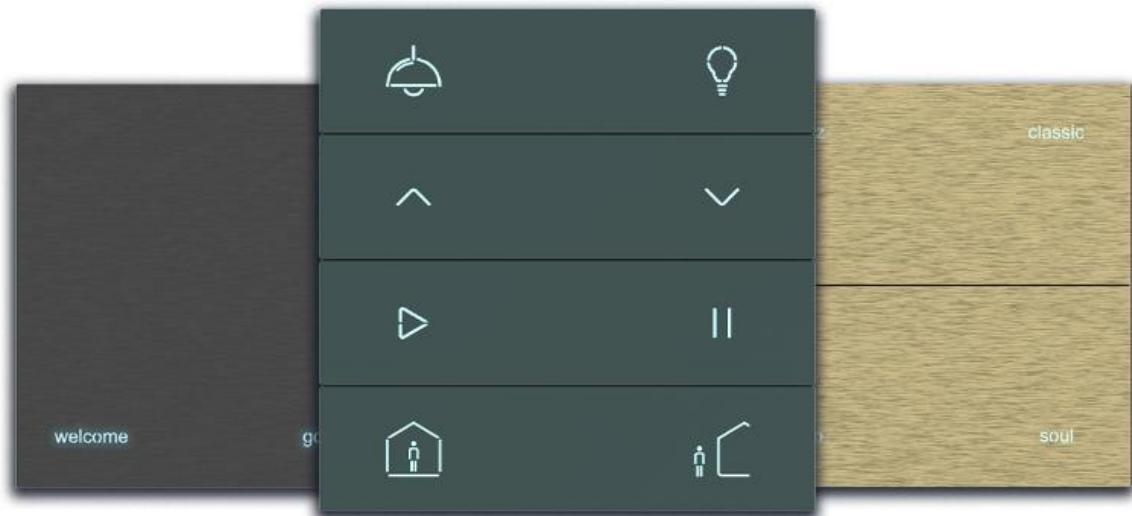


**core**

# User Manual

## Eclipse Push Button Switch v2.0



**Document Version:** 1.0

**Last Revision:** 15.08.2025

**Product Code:** CR-ECS-86-KNX-ThCO2

## Table Of Contents

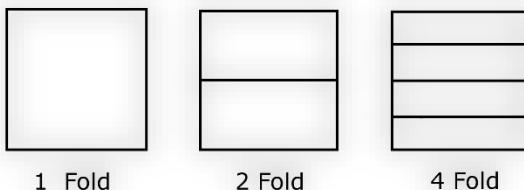
1.	Presentation.....	4
1.1.	Main Features.....	5
1.2.	Dimensions .....	6
2.	Technical Specification.....	7
2.1.	Installation.....	7
2.1.1	Installation Site .....	7
2.1.2.	Mounting, Electrical Connection .....	7
3.	ETS Parameters .....	10
3.1.	General .....	10
3.1.1	Proximity Sensor.....	10
3.1.2.	Humidity Sensor .....	13
3.1.3.	IAQ (Indoor Air Quality) Sensor .....	15
3.1.4.	Logic Function.....	17
3.1.5.	Leds Brightness.....	19
3.2.	Switch Configuration.....	22
3.2.1.	Rocker Oriented [Switch].....	23
3.2.2.	Rocker Oriented [Dimming] .....	26
3.2.3.	Rocker Oriented [Shutter] .....	29
3.2.4.	Rocker Oriented [Value].....	32
3.2.5.	Button Oriented [Switch] .....	35
3.2.6.	Button Oriented [Dimming] .....	38
3.2.7.	Button Oriented [Shutter].....	42
3.2.8.	Button Oriented [Scene] .....	45
3.2.9.	Button Oriented [Value] .....	47
3.3.	Temperature Sensor .....	50
3.4.	Thermostat.....	53
3.4.1.	Thermostat Settings.....	53
3.4.2.	Setpoint Temperature .....	55

3.4.3. Heating .....	57
3.4.4. Cooling .....	62
3.4.5. Heating & Cooling .....	66
<b>4. Commissioning.....</b>	<b>70</b>

## 1. Presentation

Core Eclipse Switch is designed to control the ambient in a stylish way, thanks to premium materials and finishes. It is also equipped with multiple sensors to automate the ambient; thus becomes a switch, a sensor and a design material.

### Fold Options



### Material and Colour Options

#### Brushed Finish

Pure form of stainless steel, brass and aluminium are brushed with perfect craftsmanship to provide satin effect in each touch to the device.

#### Silky-Matte Finish

Aluminium is painted with unique colours and coated with special techniques to provide silky feeling in each touch to the product.



### Ordering Tips

Eclipse Push-Button Switch Components



= Eclipse PB Switch  
Electronic Part

+ Eclipse PB Switch  
1 Fold

+ Custom Icon  
Engraving

Use online planner to create a custom-made Eclipse Switch and order.

<https://portal.core.com.tr/>

## 1.1. Main Features

### More than just a Switch...

Eclipse Switch can activate many functions like other smart switches does such as switching and dimming. Besides these, it can do much more with its built-in logic functions and multiple sensors.

### BUILT-IN THERMOSTAT

Eclipse Switch can control HVAC systems via built-in thermostat logic, temperature sensor and humidity sensor.

### AIR QUALITY MEASUREMENT

Eclipse Switch measures indoor air quality inside the room with its built-in sensor. Logic functions can be triggered according to the air-quality level via KNX.

### CUSTOMIZABLE ICONS

Icon or text can be engraved to each buttons in folds. Thanks to dedicated online planner, customer can select icons from extensive icon library or write text accordingly.

### RGB BACKLIGHTS

RGB Backlights of the icons indicate the status of connected load. Colour of the backlight can be configured via ETS programming tool.

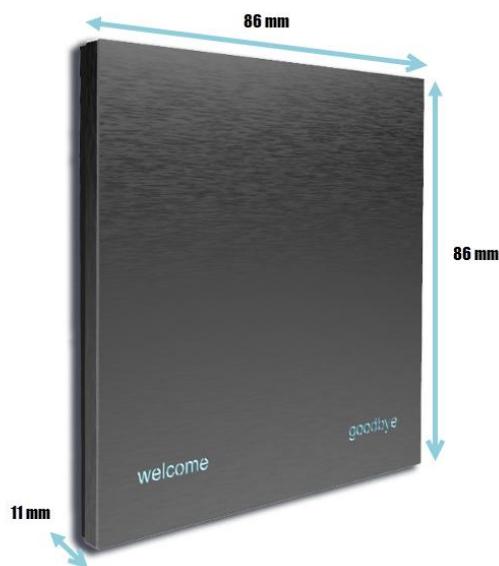
### ADAPTIVE BACKLIGHTS

Light sensor measures ambient light and icon backlights are dimmed according to the light level in the room.

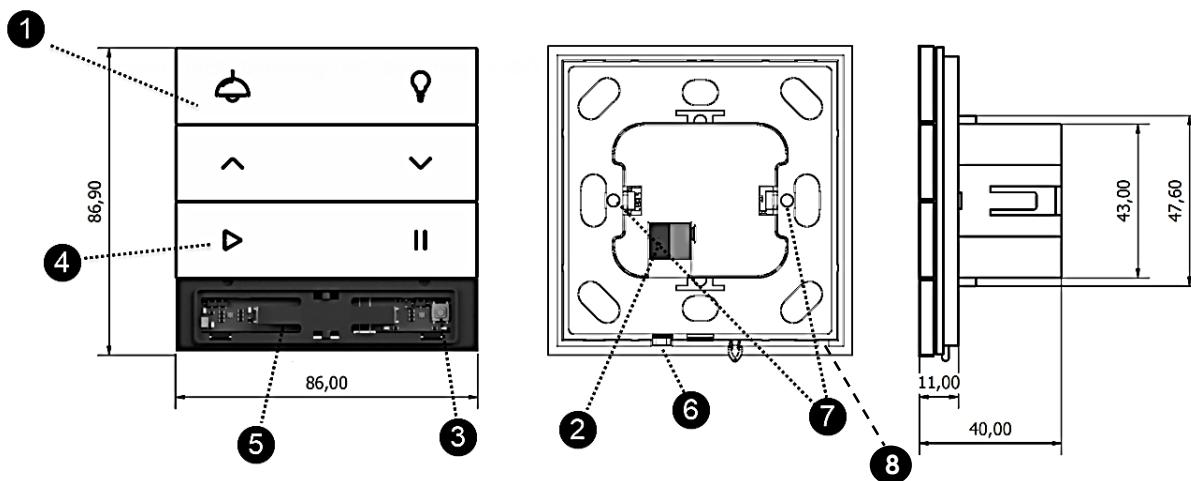
### PROXIMITY SENSOR

Proximity sensor senses the approach and backlights of icons are dimmed up to welcome customer in the dark ambient.

## 1.2. Dimensions



Dimensional drawing (all dimensions are in mm)



1. Folds (Sold Separately)
2. KNX connector
3. KNX Programming button
4. Customisable icon

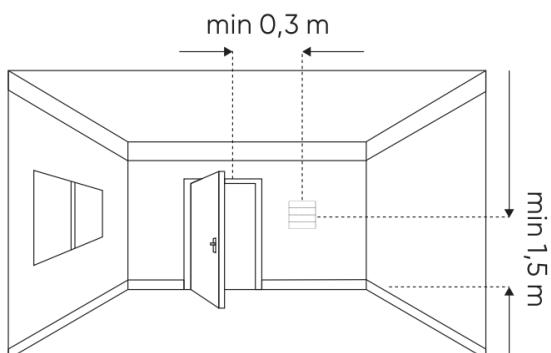
5. Position of temperature sensor
6. Position of light sensor
7. Screw holes
8. Position of IAQ sensor

## 2. Technical Specification

Sensors:	Temperature – Accuracy Rate: +/- 0,2°C Humidity – Accuracy Rate: +/- 2% Indoor Air Quality (IAQ) Proximity & Light
Dimensions:	86mm X 86mm X 11mm
Fold Material:	Aluminium, Brass and Nickel depending on the finish selection
Power:	29 VDC
Consumption:	< 12 mA from KNX Bus-line
Installation:	German IEC/EN 60670 In wall Box 77

### 2.1. Installation

#### 2.1.1 Installation Site

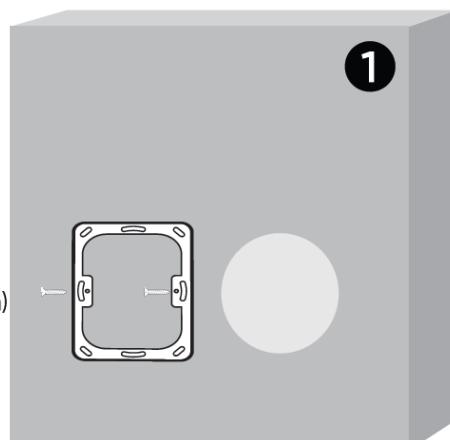


➤ The device should be positioned approximately 150 cm above the ground and 30 cm away from the door.

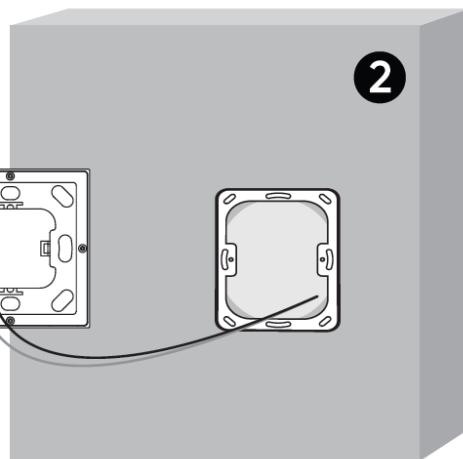
➤ The device should not be installed close to the heat source. The wall opposite the heat source will be appropriate for the installation.

➤ Contact with fluids to the device is to be avoided.

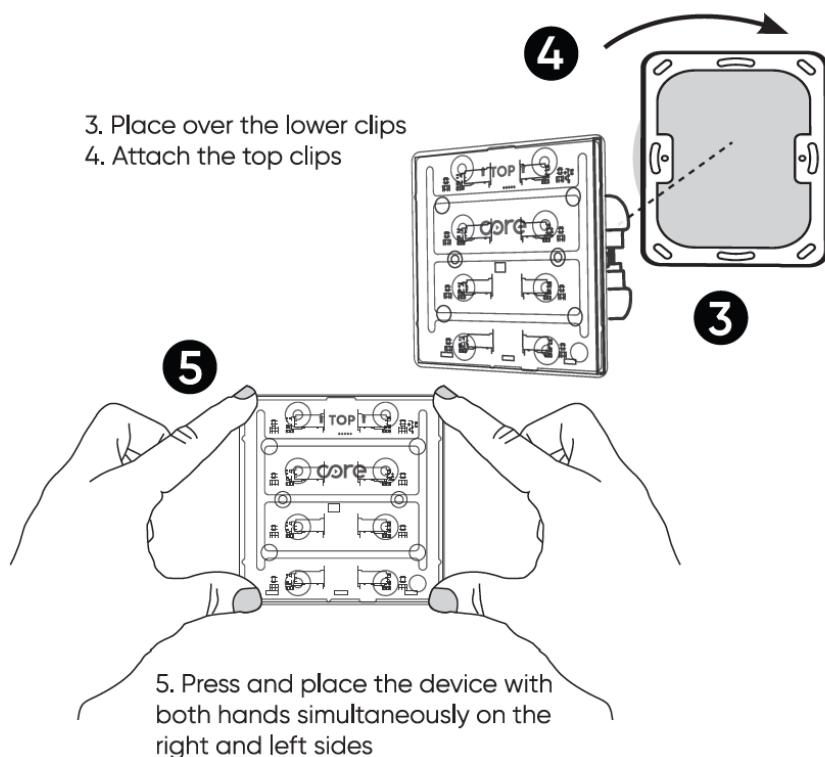
#### 2.1.2. Mounting, Electrical Connection

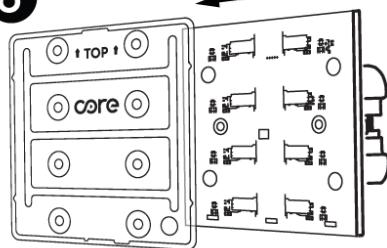


2. Connect KNX cable to the device. Check that polarity is correct.

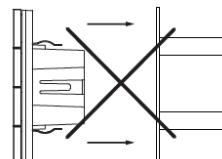
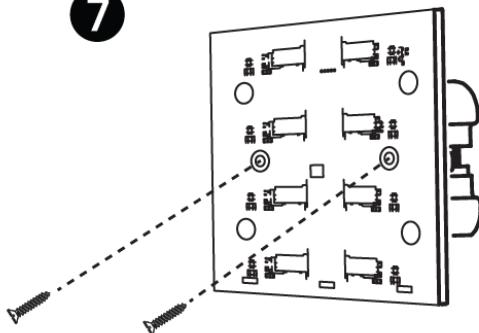
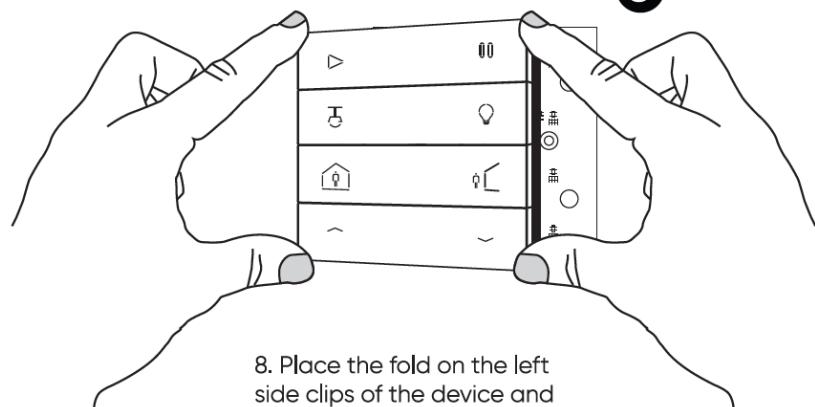


3. Place over the lower clips  
4. Attach the top clips



**6**

6. Remove the electronic part cover

Do not throw away  
the screwsPushing the device  
straight into the clips  
might damage**7**7. Mount the screws  
on the body**8**8. Place the fold on the left  
side clips of the device and  
push on the right side

Folds sold separately

For installation video: <https://www.youtube.com/watch?v=fQ1GhjV30yY&t=3s>

### 3. ETS Parameters

Core Eclipse Push Button Switch series are fully compatible KNX devices that must be configured and set up using the standard KNX configuration tool ETS. The ETS database for this device can be downloaded from ETS online catalogue.



For tutorial videos: <https://www.youtube.com/playlist?list=PLtwbriT0bxi9VQ78KBSBaC1WMLtOtB9AN>

#### 3.1. General

...- Eclipse Push-Button Switch > General

General	General
Switch Configuration	Send Alive Beacon <input checked="" type="radio"/> Disable <input type="radio"/> Enable
Rocker 1	Delay After Bus Voltage Recovery (s) 1
Temperature Sensor	Humidity Sensor <input checked="" type="radio"/> Disable <input type="radio"/> Enable
Thermostat	IAQ Sensor <input checked="" type="radio"/> Disable <input type="radio"/> Enable
	Proximity Sensor <input checked="" type="radio"/> Disable <input type="radio"/> Enable
	Logic Function <input checked="" type="radio"/> Disable <input type="radio"/> Enable
	<b>Leds Brightness</b>
	Brightness Sensor <input checked="" type="radio"/> Disable <input type="radio"/> Enable
	Brightness Level (%) 100

#### SEND ALIVE BEACON

Send Alive Beacon	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Send Alive Beacon Interval (s)	300
Send Alive Beacon Value	<input type="radio"/> 0 <input checked="" type="radio"/> 1

Parameter used to observe that the device and the application are running. It is disabled by default. When activated, Object Number 1 “General – Alive Beacon” will send selected value with defined time interval. [0-65535 s]

#### DELAY AFTER BUS VOLTAGE RECOVERY

The parameter defines the behaviour of the switch after bus power return. The delay time determines the period between bus voltage recovery and the point after which telegrams can be sent. [0-65535 s]

##### 3.1.1 Proximity Sensor

Through the proximity sensor it is possible to keep the Eclipse Switch in a stand-by state, setting a level of brightness of the LEDs and reactivate the backlighting of the buttons only when the user approaches the switch.

-.- Eclipse Push-Button Switch > General

General	General
Switch Configuration	Send Alive Beacon <input checked="" type="radio"/> Disable <input type="radio"/> Enable
Rocker 1	Delay After Bus Voltage Recovery (s) 1
Temperature Sensor	<input checked="" type="radio"/> Disable <input type="radio"/> Enable
Proximity	IAQ Sensor <input checked="" type="radio"/> Disable <input type="radio"/> Enable
Thermostat	Proximity Sensor <input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Logic Function <input checked="" type="radio"/> Disable <input type="radio"/> Enable
	Leds Brightness
	Brightness Sensor <input checked="" type="radio"/> Disable <input type="radio"/> Enable
	Brightness Level (%) 100

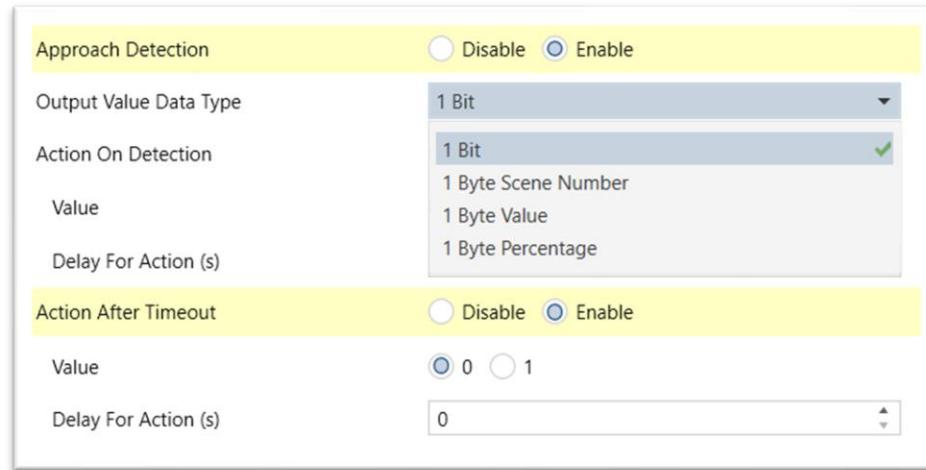
When “Proximity Timeout” is over, [1-120 s] brightness of LEDs will be dimmed to “Standby Dimming Value” until next proximity approach is detected. Stand by dimming value [%0, %10, %20, %30, %40, %50]

-.- Eclipse Push-Button Switch > Proximity

General	Proximity
Switch Configuration	Proximity Timeout (s) 30
Rocker 1	Standby Dimming Factor 50%
Temperature Sensor	Approach Detection <input type="radio"/> Disable <input checked="" type="radio"/> Enable
Proximity	Output Value Data Type 1 Bit
Thermostat	Action On Detection <input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Value <input type="radio"/> 0 <input checked="" type="radio"/> 1
	Delay For Action (s) 0
	Action After Timeout <input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Value <input checked="" type="radio"/> 0 <input type="radio"/> 1
	Delay For Action (s) 0

#### Approach Detection: [Disable, Enable]

Object “Detection Output” can be activated by enabling “Approach Detection” parameter.



Output data type can be selected 1 Bit or 1 Byte.

#### Action On Detection:

When enabled, "Detection Output" object will transmit selected value to KNX bus when the user approaches the device. A delay can be set to transmit the value with "Delay For Action" parameter.

#### Action After Timeout:

When enabled, "Detection Output" object will transmit selected value to KNX bus after proximity timeout expired. A delay can be set to transmit the value with "Delay For Action" parameter.

156 Proximity      Detection Output (Switching)      1 bit      switch      C - - T -

Brightness calculation when Proximity sensor is activated.

Maximum brightness value= MBV [selected in "General" parameter tab "Leds Brightness" section.]

Standby dimming percentage= SDP [possible values: 0, 10, 20, 30, 40, 50]

Proximity sensor value= PSV [possible values: 0, 1] 0=not detected, 1=default value

Actual brightness formula:

$$\text{Actual Brightness} = \text{MBV} \times \frac{\text{SDP} + (100 - \text{SDP}) \times \text{PSV}}{100}$$

Example 1: MBV=100, SDP=20

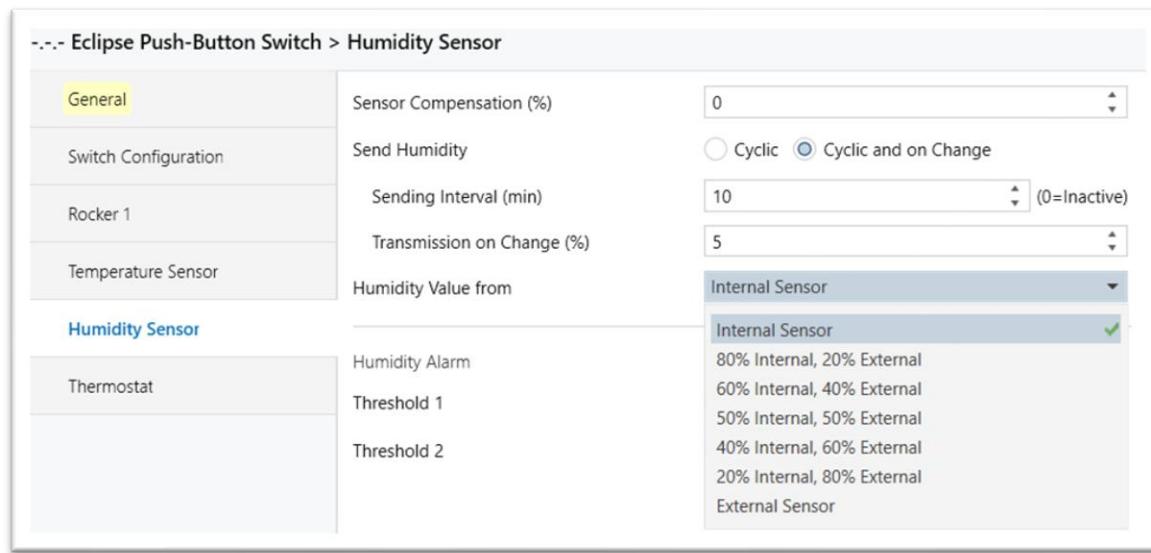
$$\text{Actual Brightness} = 100 \times \frac{20 + (100 - 20) \times 0}{100} = 20 \%$$

Example 2: MBV=60, SDP=20

$$\text{Actual Brightness} = 60 \times \frac{20 + (100 - 20) \times 0}{100} = 12 \%$$

### 3.1.2. Humidity Sensor

Humidity sensor tab contains following parameters.



#### Sensor Calibration:

Measured humidity value can be shifted up or down by using sensor calibration value. [-32768...+32768]

Example: Assume that “10” is written to the sensor calibration box. Calculation:  $10 \times 0.1 = 1\%$ , measured humidity percentage will be increased + 1%. If “-10” is written to the sensor calibration box. Calculation:  $-10 \times 0.1 = -1\%$ , measured humidity percentage will be decreased - 1%.

#### Send Humidity:

Object “Humidity – Internal Value” can be sent cyclically or by change of measured humidity.

Sending interval time [0-65535 s]

Transmission on change [0-255]

#### Humidity value from:

Humidity value can be received by an external humidity sensor directly or partially according to selected percentage.

Humidity Alarm:

Humidity Alarm

Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value (%)	60
Hysteresis ± (%)	3
Output Value Data Type	1 Bit
Action On Below Threshold 1	<input type="radio"/> 1 Bit <input type="radio"/> 1 Byte Scene Number <input type="radio"/> 1 Byte Value <input type="radio"/> 1 Byte Percentage
Value	<input checked="" type="radio"/> 0 <input type="radio"/> 1
Delay For Action (s)	0
Action On Above Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value	<input checked="" type="radio"/> 0 <input type="radio"/> 1
Delay For Action (s)	0
Threshold 2	<input checked="" type="radio"/> Disable <input type="radio"/> Enable

2 Thresholds can be defined.

When a threshold is enabled, "Humidity - Threshold x Output" object will appear. Value and hysteresis can be defined. Output data type can be selected as 1 Bit or 1 Byte.

Humidity Alarm

Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value (%)	60
Hysteresis ± (%)	3
Output Value Data Type	1 Bit
Action On Below Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value	<input type="radio"/> 0 <input checked="" type="radio"/> 1
Delay For Action (s)	0
Action On Above Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value	<input checked="" type="radio"/> 0 <input type="radio"/> 1
Delay For Action (s)	0
Threshold 2	<input checked="" type="radio"/> Disable <input type="radio"/> Enable

Action On Below Threshold:

When enabled, "Threshold x Output" object will transmit selected value to KNX bus when measured humidity value is less than entered "value(%) - hysteresis(%)". A delay can be set to transmit the value with "Delay For Action" parameter.

Example: Threshold value is %60 and Hysteresis is %3. When measured humidity value is less than %57, "Humidity - Threshold x Output" object will transmit selected value to KNX bus.

#### Action On Above Threshold:

When enabled, "Threshold x Output" object will transmit selected value to KNX bus when measured humidity value is greater than entered "value(%) + hysteresis(%)". A delay can be set to transmit the value with "Delay For Action" parameter.

Example: Threshold value is %60 and Hysteresis is %3. When measured humidity value is greater than %63, "Humidity - Threshold x Output" object will transmit selected value to KNX bus.

152	Humidity	Threshold 1 Output (Switching)	1 bit	switch	C - - T -
153	Humidity	Threshold 2 Output (%)	1 byte	percentage (0..100%)	C - - T -

### 3.1.3. IAQ (Indoor Air Quality) Sensor

The screenshot shows the configuration interface for the Eclipse Push-Button Switch. The left sidebar lists various sensor types: Switch Configuration, Rocker 1, Temperature Sensor, IAQ Sensor (highlighted in yellow), Proximity Sensor, Thermostat, Leds Brightness, and Brightness Sensor. The right panel shows settings for each type. For the IAQ Sensor, the 'Enable' radio button is selected. Other settings include 'Send Alive Beacon' (disabled), 'Delay After Bus Voltage Recovery' (set to 1), and 'Brightness Level' (set to 100).

When enabled, "Indoor Air Quality – IAQ Level" object will appear.

154	Indoor Air Quality	IAQ Level (0-Off, 1-Very Good, 2-Good, 3-Medium, 4-Poor, 5-Bad)	1 byte	C R - T -
-----	--------------------	---	--------	-----------

IAQ Levels:

Level 1 – Very Good (Telegram "1")

Level 2 – Good (Telegram "2")

Level 3 – Medium (Telegram "3")

Level 4 – Poor (Telegram "4")

Level 5 – Bad (Telegram "5")



The object transmits "0" telegram only when IAQ sensor stops working

IAQ sensor tab contains following parameters.

#### IAQ Alarm Object:

-.- Eclipse Push-Button Switch > IAQ Sensor

General	IAQ Alarm Object	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Switch Configuration	Alarm Level	Bad
Rocker 1	Output Value Data Type	1 Bit
Temperature Sensor	Action On Alarm State	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
<b>IAQ Sensor</b>	Value	<input type="radio"/> 0 <input checked="" type="radio"/> 1
Thermostat	Delay For Action (s)	0
	Action On Normal State	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Value	<input checked="" type="radio"/> 0 <input type="radio"/> 1
	Delay For Action (s)	0

When enabled, "Indoor Air Quality – IAQ Alarm Output" object will appear.

#### Alarm Level:

IAQ Alarm Object	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Alarm Level	Bad
Output Value Data Type	Bad
Action On Alarm State	Poor Medium

Alarm level can be defined as Bad, Poor or Medium.

Alarm Level	Bad
Output Value Data Type	1 Bit
Action On Alarm State	1 Bit
Value	1 Byte Scene Number 1 Byte Value 1 Byte Percentage
Delay For Action (s)	
Action On Normal State	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value	<input checked="" type="radio"/> 0 <input type="radio"/> 1
Delay For Action (s)	0

Output data type can be selected as 1 Bit or 1 Byte.

#### Action On Alarm State:

When enabled, “IAQ Alarm Output” object will transmit selected value to KNX bus when measured IAQ Level reaches selected alarm level. A delay can be set to transmit the value with “Delay For Action” parameter.

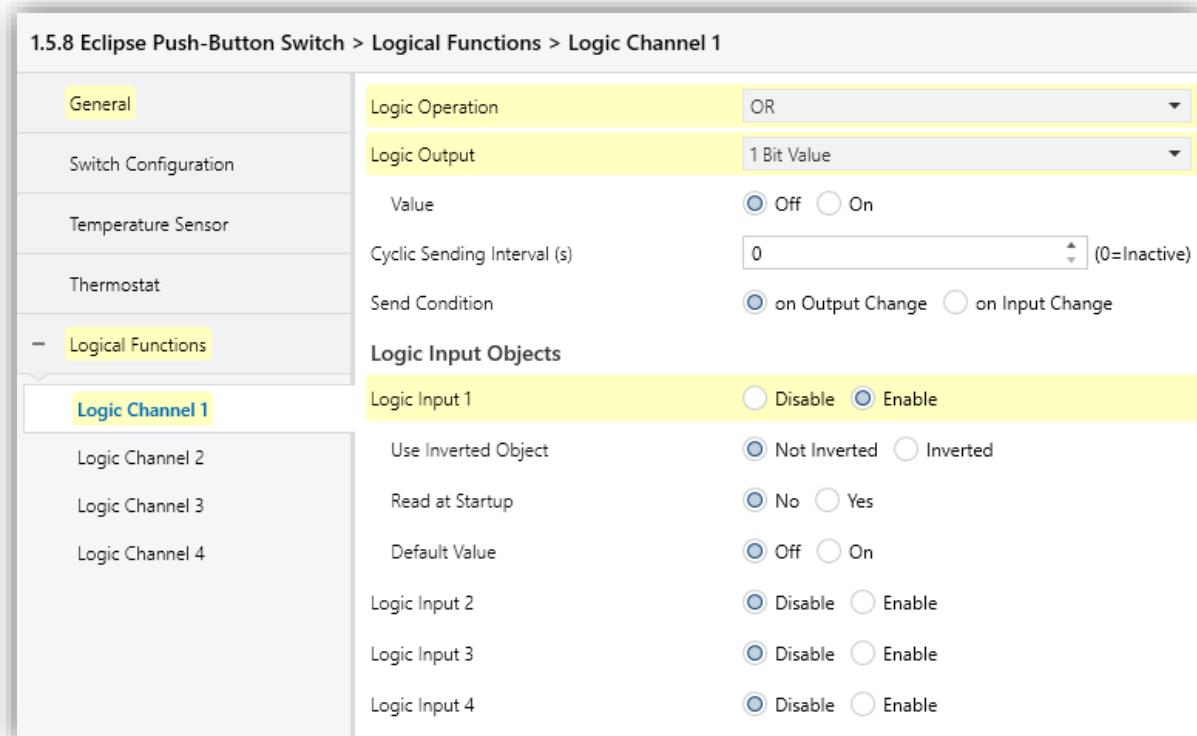
#### Action On Normal State:

When enabled, “IAQ Alarm Output” object will transmit selected value to KNX bus when measured IAQ Level is less than selected alarm level. A delay can be set to transmit the value with “Delay For Action” parameter.

155 Indoor Air Quality      IAQ Alarm Output (Switching)      1 bit      switch      C - - T -

### 3.1.4. Logic Function

Logic function tab contains “Logic channels (4)” which are available with three different logic gates. [OR, AND, XOR]



#### Logic operation:

[OR, AND, XOR]

#### Logic Output:

1 bit

1 byte unsigned value

1 byte signed value

1 byte percentage

2 byte unsigned value

2 byte signed value

2 byte floating value

**Send condition:**

[On output change] Output object sends on output change.

[On input change] Output object sends on input change.

**Logic Input Objects: [Logic input 1-4]**

Each logic channel has a maximum of 4 logic input objects.

**Logic input 1 [Use inverted Object]:** Object value will be inverted before the processing in the logic gate.

**Logic input 1 [Read at Start-up]:** Object will be read out after switch is power return.

**Logic input 1 [Default value]:** Defines default value of the input object.

Example:

“Input 1” and “Input 2” objects are defined for “AND” logic operation. If both objects are “True” at the same time then output object will send “1 byte percentage” %75 value for every 20 seconds periodically.

1.5.8 Eclipse Push-Button Switch > Logical Functions > Logic Channel 1

General	Logic Operation	AND
Switch Configuration	Logic Output	1 Byte Percentage
Temperature Sensor	Value (%)	75
Thermostat	Cyclic Sending Interval (s)	20 (0=Inactive)
Send Condition		
<input checked="" type="radio"/> on Output Change <input type="radio"/> on Input Change		
Logical Functions		
Logic Input Objects		
Logic Channel 1	Logic Input 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Logic Channel 2	Use Inverted Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Logic Channel 3	Read at Startup	<input checked="" type="radio"/> No <input type="radio"/> Yes
Logic Channel 4	Default Value	<input checked="" type="radio"/> Off <input type="radio"/> On
	Logic Input 2	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Use Inverted Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
	Read at Startup	<input checked="" type="radio"/> No <input type="radio"/> Yes
	Default Value	<input checked="" type="radio"/> Off <input type="radio"/> On
	Logic Input 3	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Logic Input 4	<input type="radio"/> Disable <input checked="" type="radio"/> Enable

	Address ^	Name	Description	Central	Pass	Data Type	Length	No.	Last Value
☒	0/7/3	Input 1		No	No	boolean	1 bit	1	\$01   True
☒	0/7/4	Input 2		No	No	boolean	1 bit	1	\$01   True
☒	0/7/5	Output (1 Byte)		No	No	percentage (0..100%)	1 byte	1	\$BF   75%

### 3.1.5. Leds Brightness

#### Brightness Sensor [Disable]

Brightness level can be defined between %0 - %100 for LEDs when brightness sensor is disabled.

Leds Brightness

Brightness Sensor	<input checked="" type="radio"/> Disable <input type="radio"/> Enable
Brightness Level (%)	100

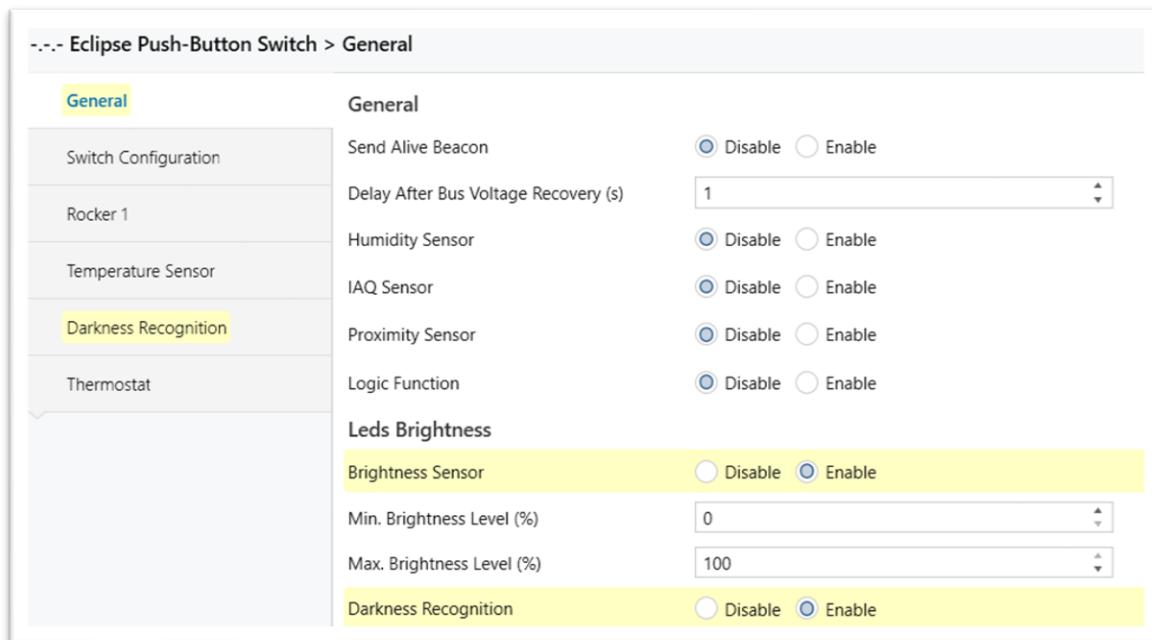
### Brightness Sensor [Enable]

Eclipse Switch adjusts brightness of LEDs according to ambient light level which is measured by built-in light sensor.

“Brightness sensor” can be enabled under “General” parameter tab. Minimum and maximum brightness levels can be defined here. LEDs brightness will be adjusted between these two percentage levels.

Min Brightness Value can be defined between %0 - %50.

Max Brightness Value can be defined between %50 - %100.



Brightness calculation when Brightness sensor is enabled.

Maximum Brightness Value= **Max** possible values: [50-100]

Brightness value for darkness (min value) = **Min** possible values: [0-50] default=0

Ambient sensor value= **AS** [0-100 lux] 0=10 lux, 100=X lux

Actual brightness formula:

$$\text{Actual Brightness} = \text{Min} + \frac{(\text{Max} - \text{Min}) \times \text{AS}}{100}$$

Example 1: Max=80, Min=30, AS=30

$$\text{Actual Brightness} = 30 + \frac{(80 - 30) \times 10}{100} = 45 \%$$

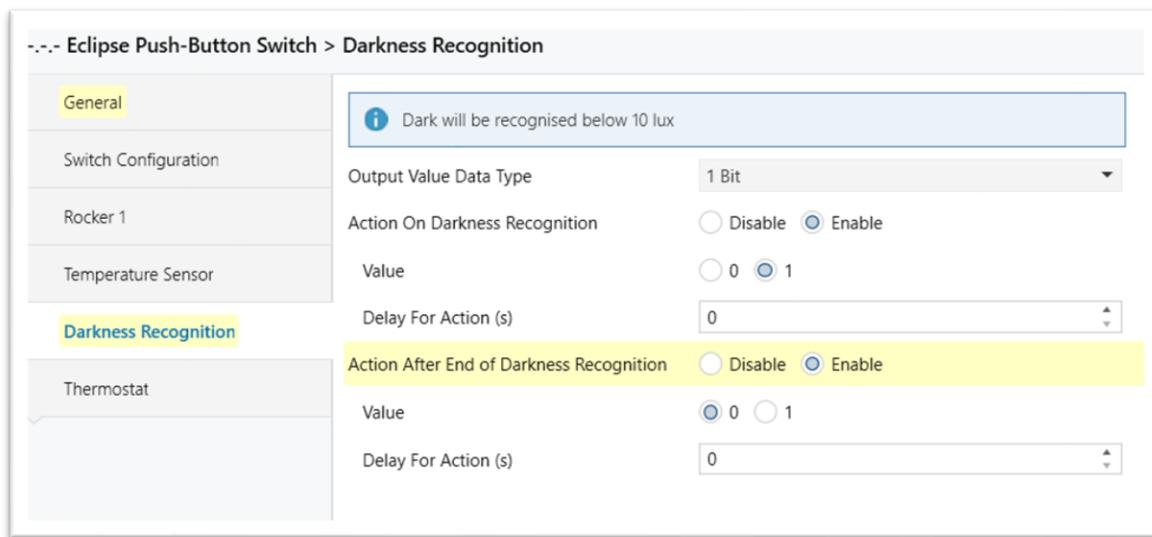
Example 2: Max=80, Min=30, AS=70

$$\text{Actual Brightness} = 30 + \frac{(80 - 30) \times 70}{100} = 65\%$$

MBV [50-100]	Y [0-50]	AS [0-100]	Actual Brightness
80	30	0	30%
80	30	10	35%
80	30	20	40%
80	30	30	45%
80	30	40	50%
80	30	50	55%
80	30	60	60%
80	30	70	65%
80	30	80	70%
80	30	90	75%
80	30	100	80%

Table 1 – Change of Actual Brightness according to ambient light level

## Darkness Recognition



Object "Darkness Output" can be activated by enabling "Darkness Recognition" parameter. If measured lux value is less than 10 lux, darkness will be recognised and transmitted via this object to KNX bus.

Output data type can be selected 1 Bit or 1 Byte.

### Action On Darkness Recognition:

When enabled, "Darkness Output" object will transmit selected value to KNX bus when measured lux value is less than 10 lux. A delay can be set to transmit the value with "Delay For Action" parameter.

### Action After End of Darkness Recognition:

When enabled, "Darkness Output" object will transmit selected value to KNX bus when measured lux value is greater than 10 lux. A delay can be set to transmit the value with "Delay For Action" parameter.

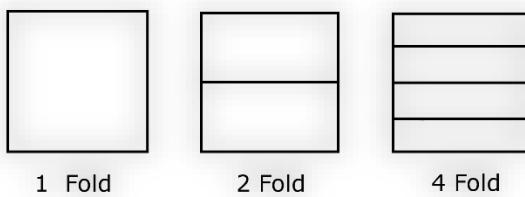
157 Darkness Recognition Darkness Output (Switching) 1 bit switch C - - T -

Note: If "Brightness Sensor" is enabled at the same time with "Proximity Sensor", brightness value of LEDs will change accordingly to the formula below.

$$\text{Actual Brightness} = (\text{Min} + \frac{(\text{Max} - \text{Min}) \times \text{AS}}{100}) \times \frac{\text{SDP} + (100 - \text{SDP}) \times \text{PSV}}{100}$$

### 3.2. Switch Configuration

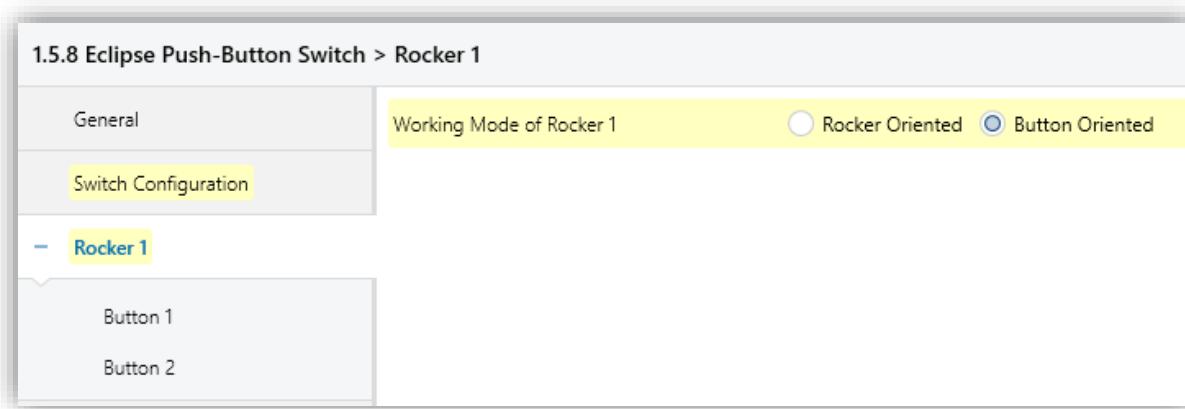
Select rocker/fold count for Eclipse switch. [1, 2, 4]



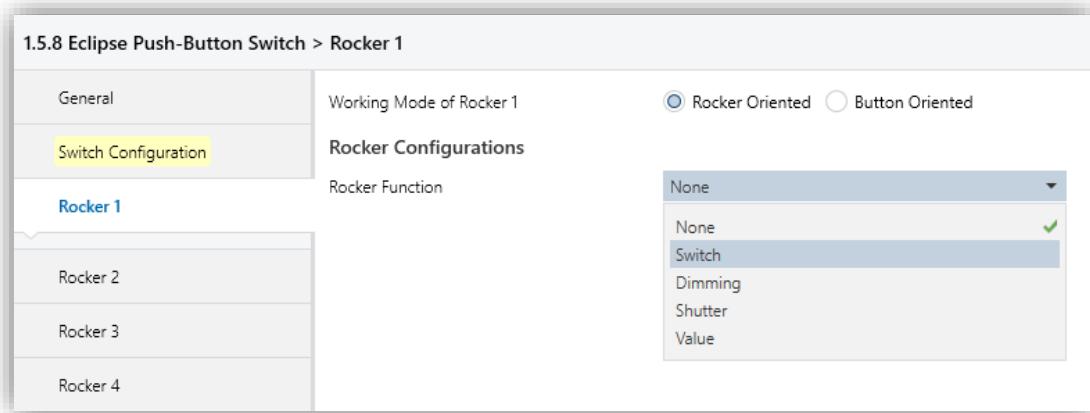
1.5.8 Eclipse Push-Button Switch > Switch Configuration

General	Switch Configuration
<b>Switch Configuration</b>	Switch Configuration
Rocker 1	4 Rockers / 8 Buttons
Rocker 2	1 Rocker / 2 Buttons
Rocker 3	2 Rockers / 4 Buttons
Rocker 4	4 Rockers / 8 Buttons
Temperature Sensor	
Thermostat	

Two working modes are available for each rocker. [Rocker oriented, Button Oriented]

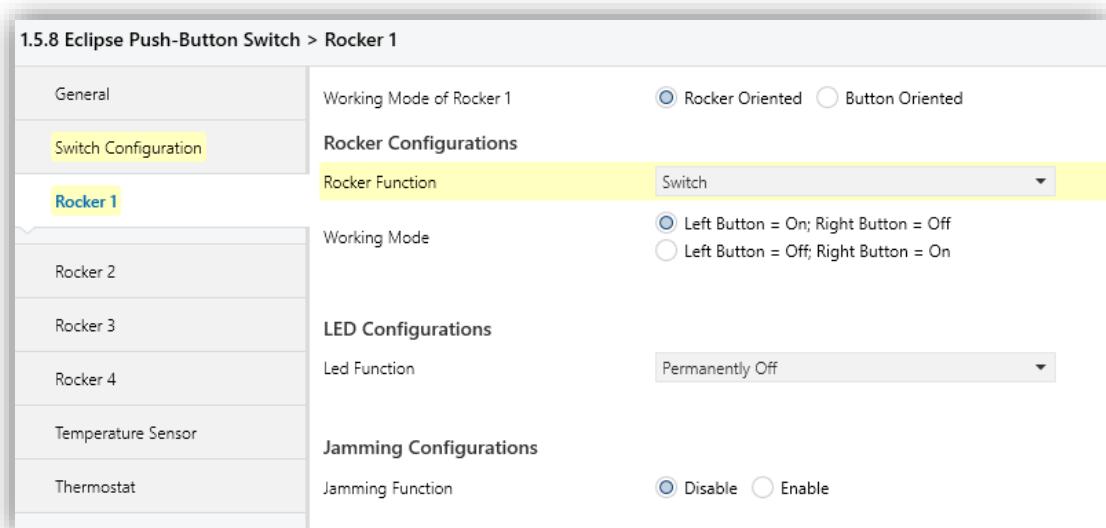


**Rocker oriented functions:** [Switch, Dimming, Shutter, Value]



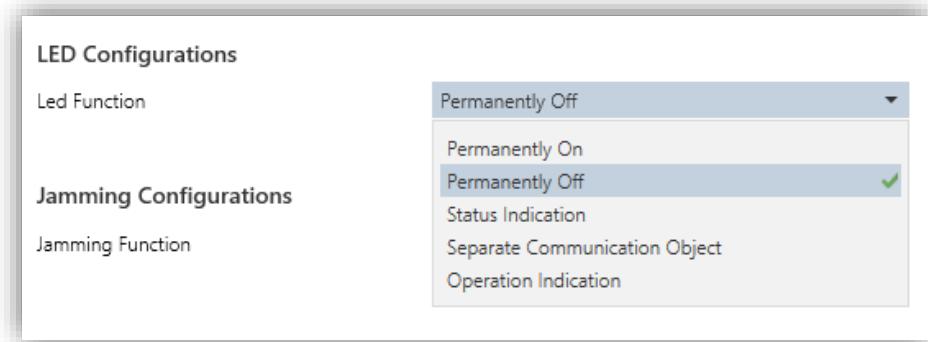
### 3.2.1. Rocker Oriented [Switch]

Working mode is selectable for left and right buttons. [Left Button = ON; Right Button=OFF]



**Led configurations:**

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.



**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]



**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Status indication]:** LED color will change according to status information.

On command [white] – OFF Command [Off]

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Status Indication
Use Inverted Status Indication	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▲"/> (0=Inactive) <input type="button" value="▼"/>
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

### LED Configurations

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▲"/> (0=Inactive) <input type="button" value="▼"/>
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for “OFF command” until it is pressed. The pressed button will stay on color selected for “ON command” until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at “Blink Duration(s)” parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 <input type="button" value="▲"/> (0=Inactive) <input type="button" value="▼"/>
On Command	
Color	White
Off Command	
Color	Off

## Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function

Disable  Enable

Use Inverted Jamming Function

Not Inverted  Inverted

### 3.2.2. Rocker Oriented [Dimming]

Working mode is selectable for left and right buttons. [Left Button = ON/Brighter; Right Button=OFF/Darker]

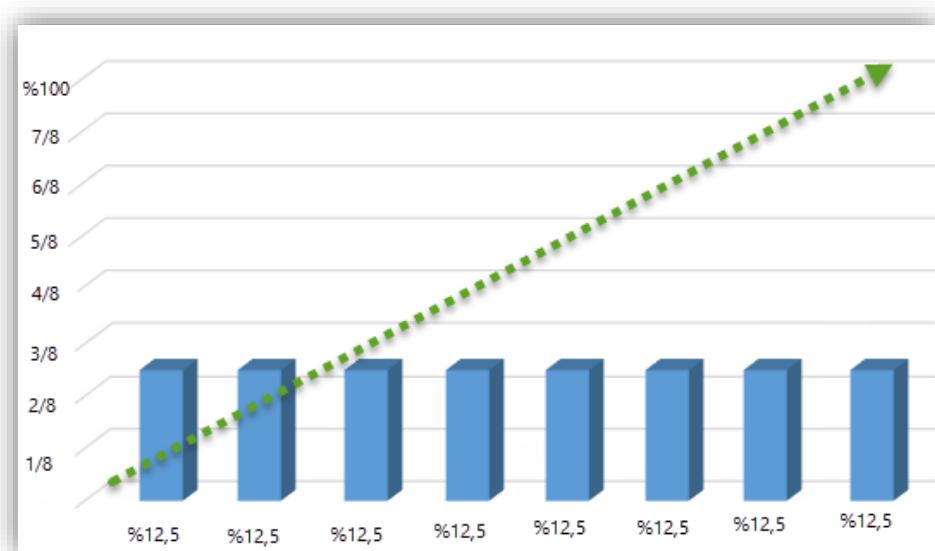
**Long Press Duration(x100 ms):** Long press duration can be changed. [0...10...65535]

As default; 100 ms x 10=1000 ms (1 second)

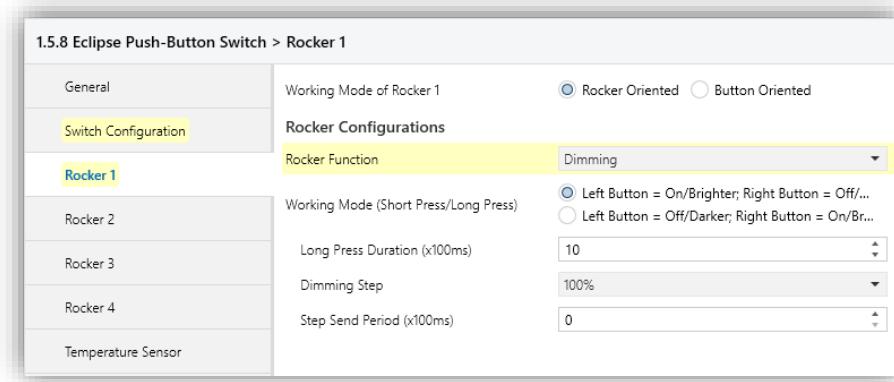
Switch will start to send dimming up/down commands after each press longer than “1 second”.

**Dimming Step:** Dimming step percentage can be changed to specify the maximum dimming step width of a dimming telegram. With a dimming message, you can dim by a maximum of X %.

(%100 option represents “Start-stop” dimming function. Other percentage values correspond to “step dimming function”. )

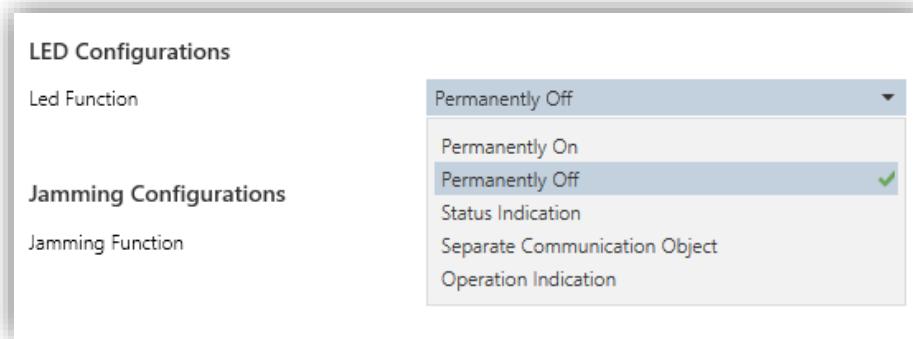


**Step Send Period:** Defines time interval between two dimming step commands.

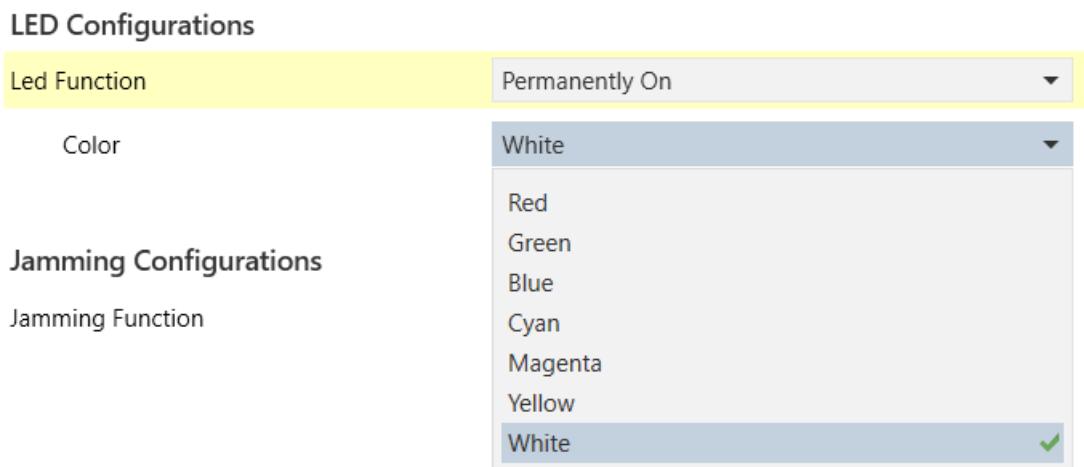


### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.



**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]



**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Status indication]:** LED color will change according to status information.

On command [white] – OFF Command [Off]

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

#### LED Configurations

Led Function	Status Indication
Use Inverted Status Indication	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White <input type="button" value="▼"/>
Off Command	
Color	Off <input type="button" value="▼"/>

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

#### LED Configurations

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White <input type="button" value="▼"/>
Off Command	
Color	Off <input type="button" value="▼"/>

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for "OFF command" until it is pressed. The pressed button will stay on color selected for "ON command" until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

### Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

### 3.2.3. Rocker Oriented [Shutter]

Working mode is selectable for left and right buttons. [Left Button = UP; Right Button=DOWN]

**Long Press Duration(x100 ms):** Long press duration can be changed. [0...10...65535]

As default; 100 ms x 10=1000 ms (1 second)

Switch will start to send move up/down commands after each press longer than “1 second”. Switch will send Step/Stop command on each short press,

1.5.8 Eclipse Push-Button Switch > Rocker 1

General	Working Mode of Rocker 1	<input checked="" type="radio"/> Rocker Oriented <input type="radio"/> Button Oriented
Switch Configuration	<b>Rocker Configurations</b>	
Rocker 1	Rocker Function	Shutter <input type="button" value="▼"/>
Rocker 2	Working Mode	<input checked="" type="radio"/> Left Button = Up; Right Button = Down <input type="radio"/> Left Button = Down; Right Button = Up
Rocker 3	Long Press Duration (x100ms)	10 <input type="button" value="▲"/> <input type="button" value="▼"/>
Rocker 4	<b>LED Configurations</b>	
Temperature Sensor	Led Function	Permanently Off <input type="button" value="▼"/>

### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.

**LED Configurations**

Led Function	Permanently Off <input type="button" value="▼"/>
Permanently On Permanently Off <input checked="" type="checkbox"/> Status Indication Separate Communication Object Operation Indication	

**Jamming Configurations**

Jamming Function	White <input type="button" value="▼"/>
Red Green Blue Cyan Magenta Yellow White <input checked="" type="checkbox"/>	

**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]

**LED Configurations**

Led Function	Permanently On <input type="button" value="▼"/>
Color	White <input type="button" value="▼"/>

**Jamming Configurations**

Jamming Function	White <input type="button" value="▼"/>
Red Green Blue Cyan Magenta Yellow White <input checked="" type="checkbox"/>	

**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Status indication]:** LED color will change according to status information.

On command [white] – OFF Command [Off]

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at “Blink Duration(s)” parameter. (Blinking interval is fixed.)

#### LED Configurations

Led Function	Status Indication
<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted	
Use Inverted Status Indication	
Blink Duration (s)	<input type="text" value="0"/> (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

#### LED Configurations

Led Function	Separate Communication Object
<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted	
Use Inverted Communication Object	
Blink Duration (s)	<input type="text" value="0"/> (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for “OFF command” until it is pressed. The pressed button will stay on color selected for “ON command” until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at “Blink Duration(s)” parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

### Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable	<input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted	<input type="radio"/> Inverted

### 3.2.4. Rocker Oriented [VALUE]

Working mode is selectable for left and right buttons. [Left Button = VALUE 1; Right Button= VALUE 2]

- Selectable data types:
- 1 bit value
  - 1 byte Unsigned Value
  - 1 Byte Signed Value
  - 1 Byte Percentage
  - 2 Byte Unsigned Value
  - 2 Byte Signed Value
  - 2 Bytes Floating Value

1.5.8 Eclipse Push-Button Switch > Rocker 1

General	Working Mode of Rocker 1	<input checked="" type="radio"/> Rocker Oriented <input type="radio"/> Button Oriented
Switch Configuration	<b>Rocker Configurations</b>	
Rocker 1	Rocker Function	Value
Rocker 2	Working Mode	<input checked="" type="radio"/> Left Button = Value 1; Right Button = Value 2 <input type="radio"/> Left Button = Value 2; Right Button = Value 1
Rocker 3	Data Type	None
Rocker 4	<b>LED Configurations</b>	
Temperature Sensor	Led Function	None 1 Bit Value 1 Byte Unsigned Value 1 Byte Signed Value 1 Byte Percentage 2 Bytes Unsigned Value 2 Bytes Signed Value 2 Bytes Floating Value
Thermostat	<b>Jamming Configurations</b>	
	Jamming Function	

### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.

#### LED Configurations

Led Function

Permanently Off

#### Jamming Configurations

Jamming Function

- Permanently Off
- Permanently On
- Permanently Off
- Separate Communication Object
- Operation Indication

**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]

**LED Configurations**

Led Function	Permanently On
Color	White
	Red
	Green
	Blue
	Cyan
	Magenta
	Yellow
	White

**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

**LED Configurations**

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for "OFF command" until it is pressed. The pressed button will stay on color selected for "ON command" until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

### Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

### 3.2.5. Button Oriented [Switch]

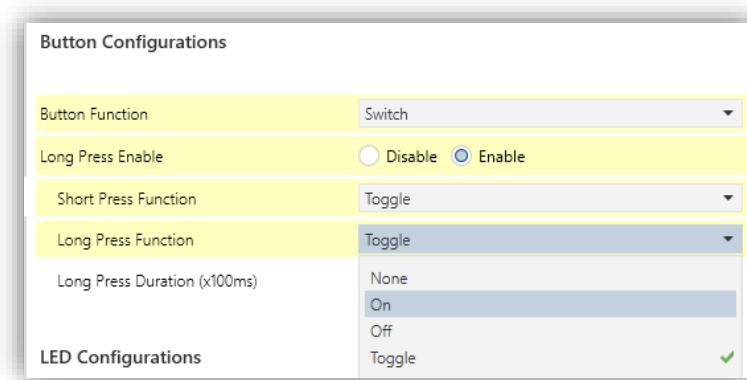
Action on press: [On, Off, Toggle] selected data will be sent to KNX bus for each press of button.

Action on release: [On, Off, Toggle] selected data will be sent to KNX bus for each release of button.

Button Function	Switch
Long Press Enable	<input checked="" type="radio"/> Disable <input type="radio"/> Enable
Action on Press	None
Action on Release	None On Off Toggle
LED Configurations	

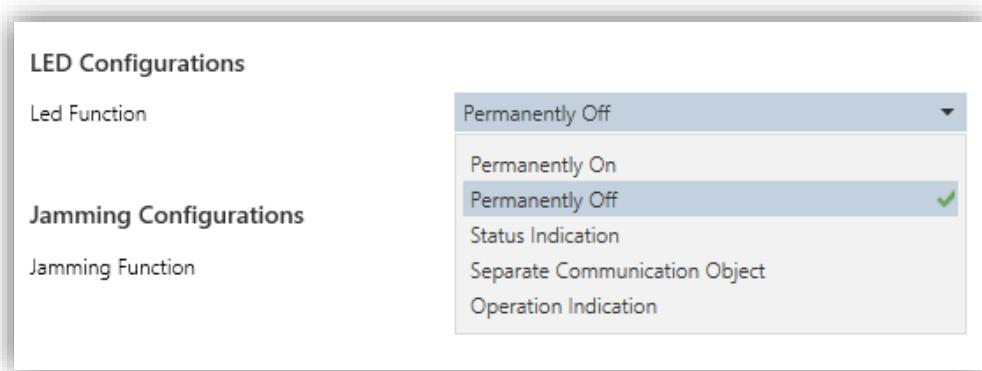
Long press function can be activated. Disabled as default.

ON, Off, Toggle commands can be sent separately using short and long press functions.



### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.



**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]



**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Status indication]:** LED color will change according to status information.

On command [white] – OFF Command [Off]

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

#### LED Configurations

Led Function	Status Indication
Use Inverted Status Indication	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White <input type="button" value="▼"/>
Off Command	
Color	Off <input type="button" value="▼"/>

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

#### LED Configurations

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White <input type="button" value="▼"/>
Off Command	
Color	Off <input type="button" value="▼"/>

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for "OFF command" until it is pressed. The pressed button will stay on color selected for "ON command" until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

### Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

### 3.2.6. Button Oriented [Dimming]

“Action on press” can be selected for ON, OFF, TOGGLE commands. Using Long press function, Dim direction is changeable for UP, DOWN, UP/DOWN commands.

1.5.8 Eclipse Push-Button Switch > Rocker 1 > Button 1

General	Button Configurations
Switch Configuration	Button Function Dimming
Rocker 1	Action on Press Toggle
Button 1	Dim Direction on Long Press Dim Up
Button 2	Long Press Duration (x100ms) 10
Rocker 2	Dimming Step 12,5%
	Step Send Period (x100ms) 8

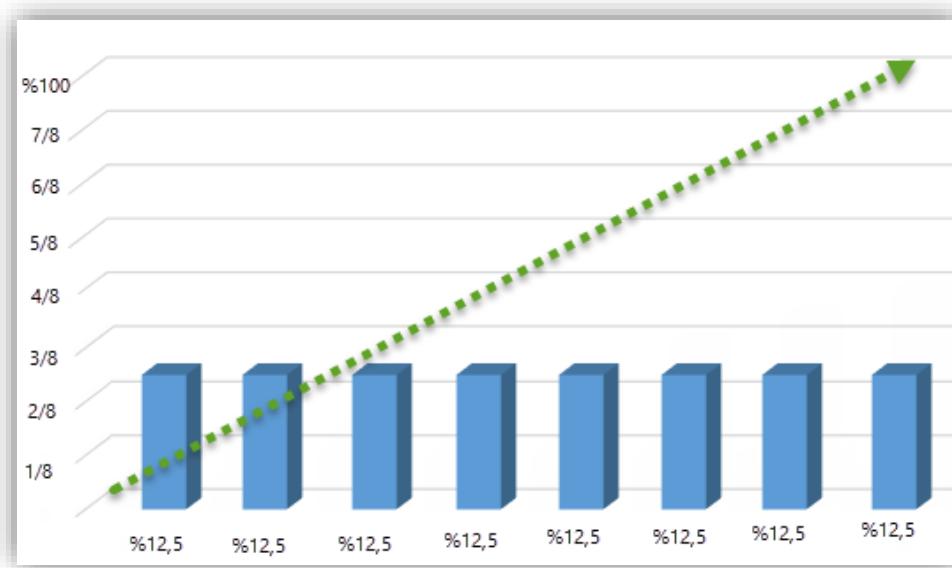
**Long Press Duration(x100 ms):** Long press duration can be changed. [0...10...65535]

As default; 100 ms x 10=1000 ms (1 second)

Switch will start to send dimming up/down commands after each press longer than "1 second".

**Dimming Step:** Dimming step percentage can be changed to specify the maximum dimming step width of a dimming telegram. With a dimming message, you can dim by a maximum of X %.

(%100 option represents "Start-stop" dimming function. Other percentage values correspond to "step dimming function". )



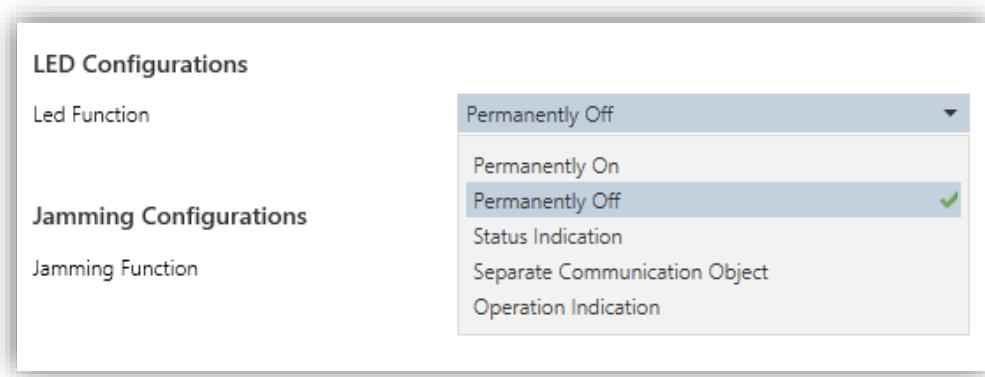
**Step Send Period:** Defines time interval between two dimming step commands. This interval is another parameter to change dimming speed of the lighting source.

1.5.8 Eclipse Push-Button Switch > Rocker 1

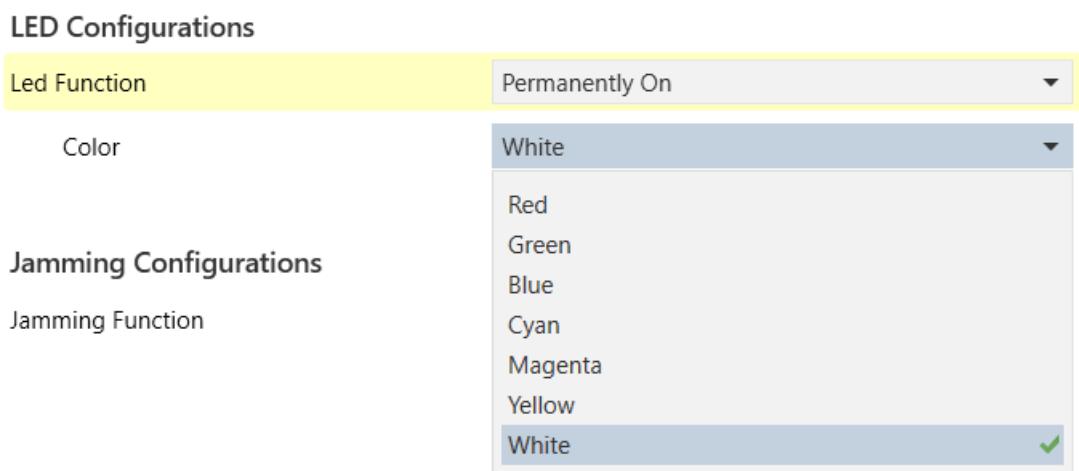
General	Working Mode of Rocker 1	<input checked="" type="radio"/> Rocker Oriented <input type="radio"/> Button Oriented
Switch Configuration	Rocker Configurations	
Rocker 1	Rocker Function	Dimming
Rocker 2	Working Mode (Short Press/Long Press)	<input checked="" type="radio"/> Left Button = On/Brighter; Right Button = Off/Darker <input type="radio"/> Left Button = Off/Darker; Right Button = On/Brighter
Rocker 3	Long Press Duration (x100ms)	10
Rocker 4	Dimming Step	100%
Temperature Sensor	Step Send Period (x100ms)	0

#### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.



**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]



**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Status indication]:** LED color will change according to status information.

On command [white] – OFF Command [Off]

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Status Indication
Use Inverted Status Indication	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▲"/> <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

### LED Configurations

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▲"/> <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for “OFF command” until it is pressed. The pressed button will stay on color selected for “ON command” until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at “Blink Duration(s)” parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 <input type="button" value="▲"/> <input type="button" value="▼"/> (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

## Jamming Configurations

Jamming function is used to block the respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

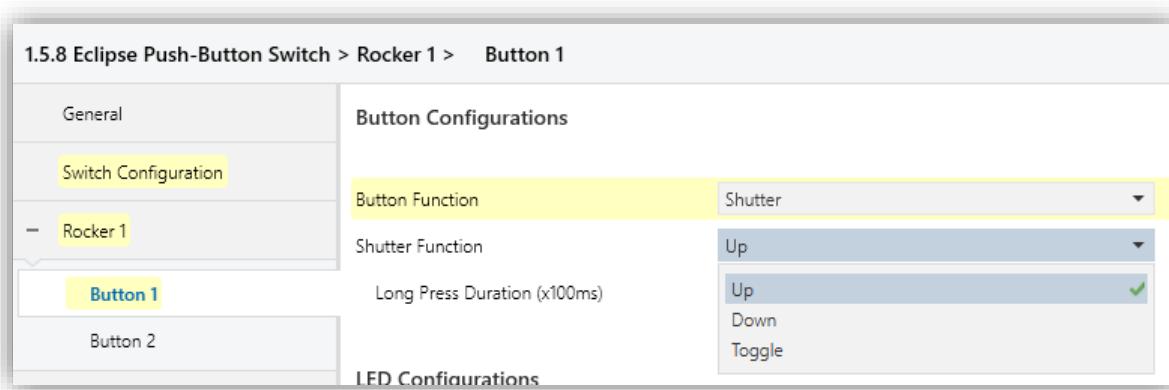
### 3.2.7. Button Oriented [Shutter]

Shutter command can be selected for each long press [UP, DOWN, TOGGLE].

Switch will start to send [UP, DOWN, TOGGLE] commands after each press longer than “1 second”. Switch will send Step/Stop command on each short press,

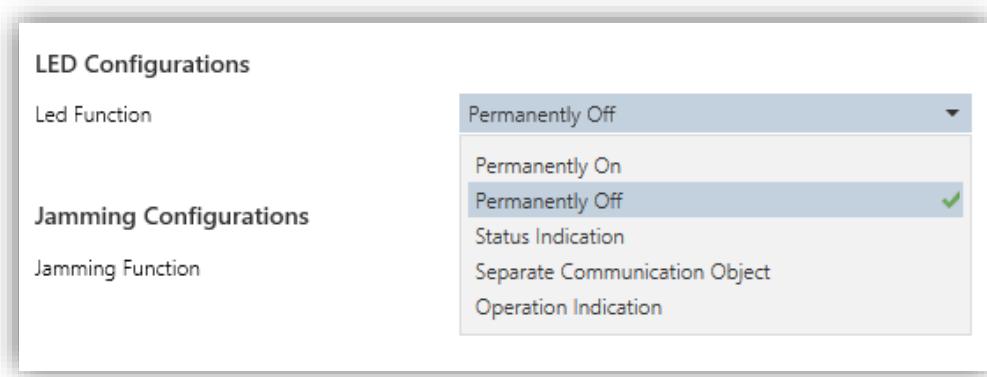
**Long Press Duration(x100 ms):** Long press duration can be changed. [0...10...65535]

As default; 100 ms x 10=1000 ms (1 second)



### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.



**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]



**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Status indication]:** LED color will change according to status information.

On command [white] – OFF Command [Off]

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Status Indication
Use Inverted Status Indication	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▲"/> (0=Inactive) <input type="button" value="▼"/>
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

### LED Configurations

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 <input type="button" value="▲"/> (0=Inactive) <input type="button" value="▼"/>
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for “OFF command” until it is pressed. The pressed button will stay on color selected for “ON command” until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at “Blink Duration(s)” parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 <input type="button" value="▲"/> (0=Inactive) <input type="button" value="▼"/>
On Command	
Color	White
Off Command	
Color	Off

## Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

## 3.2.8. Button Oriented [Scene]

Scene number determines which scene (1...64) is to be recalled and stored. 64 different scenes can be managed by using single group address on different buttons.

Each short press will call the selected scene. Storing of the current scene can be triggered by long press action. Please check example group monitor record.

1.5.8 Eclipse Push-Button Switch > Rocker 1 > Button 1

General  Switch Configuration  - Rocker 1  <b>Button 1</b>  Button 2	Button Configurations  Button Function: Scene Scene Number: 1 Mode: Send Scene and Save at Long Press Long Press Duration (x100ms): 10
--	---

Example: Short press -> Button 1 is calling scene number 1.

Long press ->Button 1 is sending the “store scene command” for scene number 1.

#	Time	Destination A	Destination B	Info	Type	DPT
1	10/03/2023 11:35:32.962	0/7/7	Scene	\$00   Activate #1	GroupValue_Write	18.001 scene control
2	10/03/2023 11:35:34.704	0/7/7	Scene	\$80   Learn #1	GroupValue_Write	18.001 scene control

**Long Press Duration(x100 ms):** Long press duration can be changed. [0...10...65535]

As default; 100 ms x 10=1000 ms (1 second)

**Led configurations:**

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.

**LED Configurations**

Led Function

Permanently Off

**Jamming Configurations**

Jamming Function

Permanently On

Permanently Off

Separate Communication Object

Operation Indication



**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]

**LED Configurations**

Led Function

Permanently On

Color

White

**Jamming Configurations**

Jamming Function

Red

Green

Blue

Cyan

Magenta

Yellow

White



**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

**LED Configurations**

Led Function

Separate Communication Object

Use Inverted Communication Object

 Not Inverted  Inverted

Blink Duration (s)

0 (0=Inactive)

On Command

Color

White

Off Command

Color

Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for “OFF command” until it is pressed. The pressed button will stay on color selected for “ON command” until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at “Blink Duration(s)” parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

### Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

### 3.2.9. Button Oriented [Value]

“Action on press” determines the data type for the short press. When button is pressed, this type of data will be sent KNX bus via respective communication object.

Long press function can be enable to send another data type by pressing longer to the same button.

1.5.8 Eclipse Push-Button Switch > Rocker 1 > Button 1

General	Button Configurations	
Switch Configuration	Button Function	Value
Rocker 1	Action on Press	None
<b>Button 1</b>	Long Press Enable	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Button 2	Long Press Function	None
Rocker 2	Long Press Duration (x100ms)	None 1 Bit Value 1 Byte Unsigned Value 1 Byte Signed Value 1 Byte Percentage 2 Bytes Unsigned Value 2 Bytes Signed Value 2 Bytes Floating Value
Rocker 3	LED Configurations	
Rocker 4	Led Function	
Temperature Sensor	Jamming Configurations	
Thermostat	Jamming Function	<input checked="" type="radio"/> Disable <input type="radio"/> Enable

**Long Press Duration(x100 ms):** Long press duration can be changed. [0...10...65535]

As default; 100 ms x 10=1000 ms (1 second)

#### Led configurations:

Available functions: Permanently OFF, Permanently ON, Status Indication, Separate Communication Object and Operation Indication.

#### LED Configurations

Led Function	Permanently Off
	Permanently On
	Permanently Off
	Separate Communication Object
	Operation Indication

**Led Function [Permanently ON]:** LED is always ON for selected color. [Red, Green, Blue, Cyan, Magenta, Yellow, White]

### LED Configurations

Led Function	Permanently On
Color	White
	Red
	Green
	Blue
	Cyan
	Magenta
	Yellow
	White

**Led Function [Permanently OFF]:** LED is always OFF.

**Led Function [Separate Communication Object]:** LED color will change according to value received by LED status object.

### LED Configurations

Led Function	Separate Communication Object
Use Inverted Communication Object	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

**Led Function [Operation indication]:** Status LED of rocker button will stay on color selected for "OFF command" until it is pressed. The pressed button will stay on color selected for "ON command" until it is released.

**Blink duration:** Status LED of the pressed rocker button will blink for the time period selected at "Blink Duration(s)" parameter. (Blinking interval is fixed.)

### LED Configurations

Led Function	Operation Indication
Blink Duration (s)	0 (0=Inactive)
On Command	
Color	White
Off Command	
Color	Off

### Jamming Configurations

Jamming function is used to block to respective button or rocker via Object Number 9 – “Rocker X - Jamming” by writing “true or false” data from the bus. Button will not work until it is enabled via jamming object.

### Jamming Configurations

Jamming Function	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Use Inverted Jamming Function	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted

### 3.3. Temperature Sensor

Temperature unit can be selected as Celsius or Fahrenheit.

#### Sensor Calibration:

Measured temperature value can be shifted up or down by using sensor calibration value. [-32768...+32768]

Example: Assume that “10” is written to the sensor calibration box. Calculation:  $10 \times 0.1 = 1$  Celsius, measured temperature will be increased “+ 1 °C”. If “-10” is written to the sensor calibration box. Calculation:  $-10 \times 0.1 = -1$  Celsius, measured temperature will be decreased “-1 °C”.

#### Send Temperature:

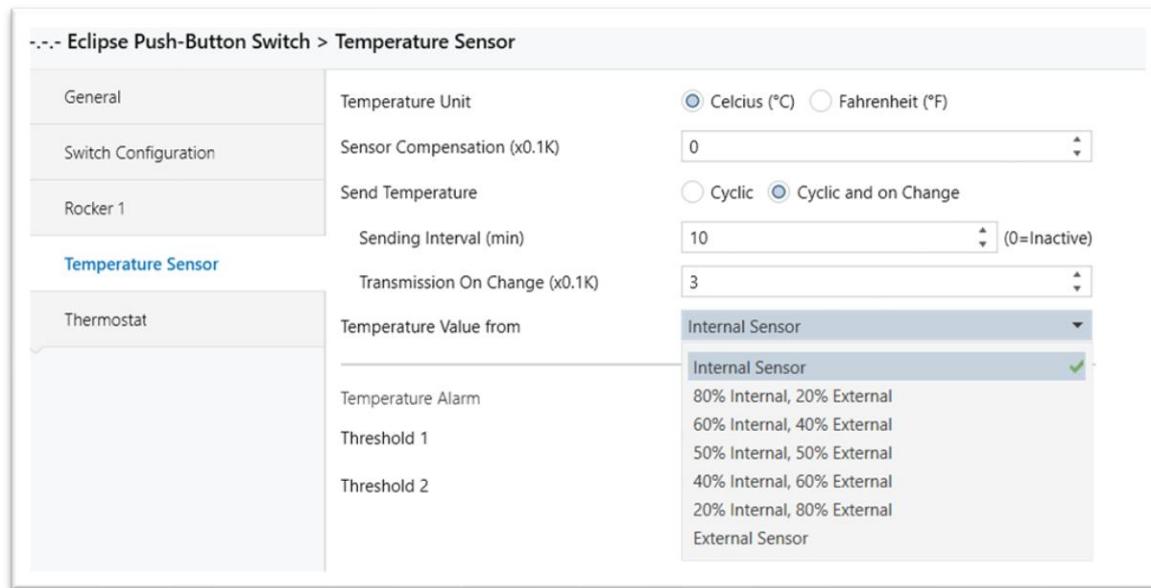
Object Number 147 “Temperature – Actual Temperature” can be sent cyclically or by change of measured temperature.

Sending interval time [0...**60**...65535 s]

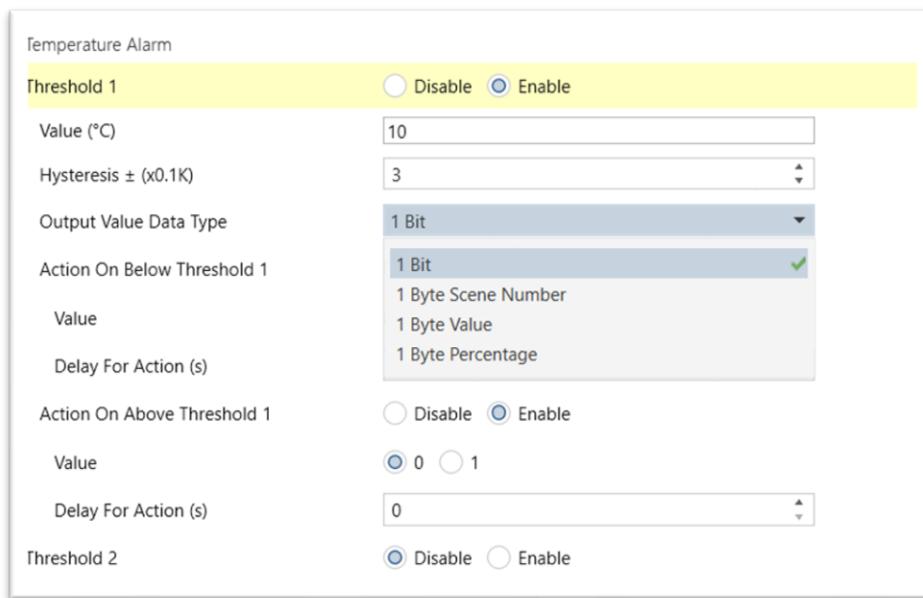
Transmission on change [0...**5**...255]

#### Temperature Value from:

Temperature value can be received from an external temperature sensor directly or partially according to selected percentage. Object Number 146 “Temperature – External Value”.



### Temperature Alarm:



2 Thresholds can be defined.

When a threshold is enabled, "Temperature - Threshold x Output" object will appear. Value and hysteresis can be defined. Output data type can be selected as 1 Bit or 1 Byte.

Temperature Alarm

<b>Threshold 1</b>	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value (°C)	10
Hysteresis ± (x0.1K)	3
Output Value Data Type	1 Bit
Action On Below Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value	<input type="radio"/> 0 <input checked="" type="radio"/> 1
Delay For Action (s)	0
Action On Above Threshold 1	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Value	<input checked="" type="radio"/> 0 <input type="radio"/> 1
Delay For Action (s)	0
Threshold 2	<input checked="" type="radio"/> Disable <input type="radio"/> Enable

Action On Below Threshold:

When enabled, "Threshold x Output" object will transmit selected value to KNX bus when measured temperature value is less than entered "value(°C) – hysteresis (0.1K)". A delay can be set to transmit the value with "Delay For Action" parameter.

Example: Threshold value is 10°C and Hysteresis is 0,3°C. When measured value is less than 9,7°C, "Temperature - Threshold x Output" object will transmit selected value to KNX bus.

Action On Above Threshold:

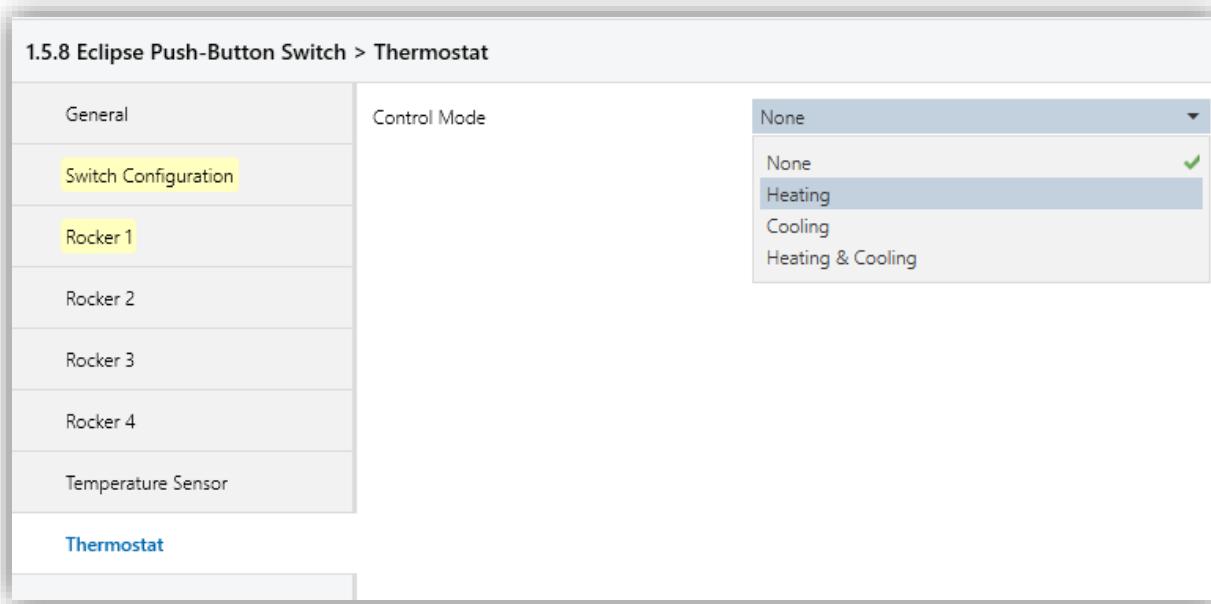
When enabled, "Threshold x Output" object will transmit selected value to KNX bus when measured temperature value is greater than entered "(°C) + hysteresis (0.1K)". A delay can be set to transmit the value with "Delay For Action" parameter.

Example: Threshold value is 10°C and Hysteresis is 0,3°C. When measured temperature value is greater than 10,3°C, "Temperature - Threshold x Output" object will transmit selected value to KNX bus.

148	Temperature	Threshold 1 Output (Switching)	1 bit	switch	C - - T -
149	Temperature	Threshold 2 Output (Scene No)	1 byte	scene number	C - - T -

### 3.4. Thermostat

Control mode of thermostat can be selected for Heating, Cooling, and Heating & Cooling together.



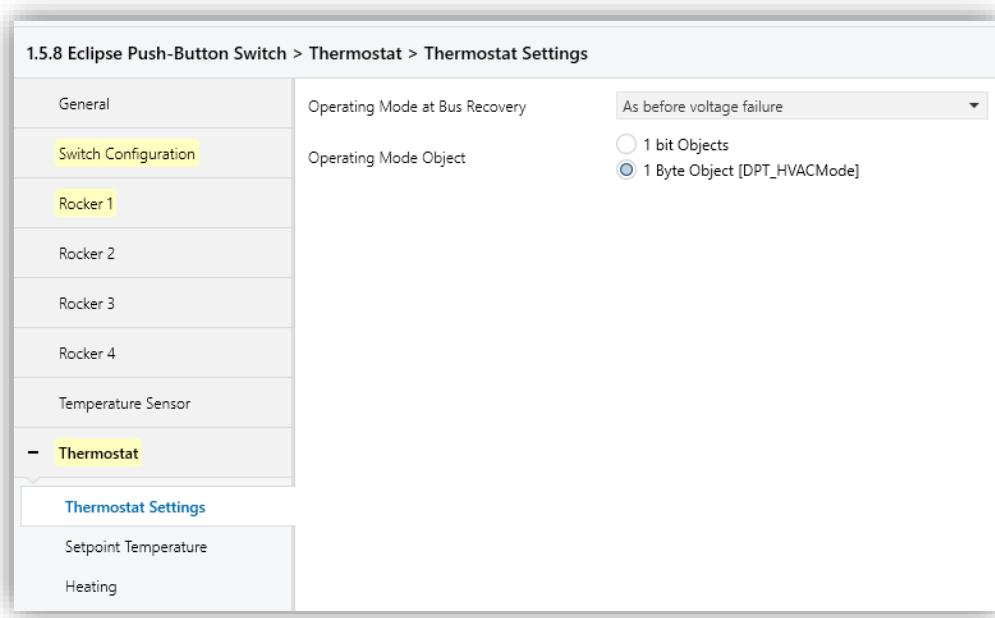
#### 3.4.1. Thermostat Settings

##### Operating Mode at Bus Recovery:

The parameter defines the behavior of the thermostat after bus power return. Operating mode can be changed to following options after a power return:

- As before voltage failure
- Comfort
- Standby
- Night
- Protection

Each operating mode has a different temperature setpoint. Please check [#3.4.2. Setpoint Temperature](#)



### Operating Mode Object:

Parameter determines the data type of operating mode objects. Data type of operating mode objects can be used as "1 bit" with separate objects for each operating mode or "1 byte" for all modes. As default,

### **1 Byte Object [DPT\_HVACMode]:**

Object 168 "Thermostat – Operating Mode" can be used to change between different modes.

\$01 – Comfort	[20.102 DPT_HVAC]
\$02 – Standby	[20.102 DPT_HVAC]
\$03 – Economy	[20.102 DPT_HVAC]
\$04 – Protection	[20.102 DPT_HVAC]

168	Thermostat	Operating Mode	1 byte	HVAC mode
169	Thermostat	Operating Mode (prev/next)	1 bit	boolean
170	Thermostat	Operating Mode Status	1 byte	HVAC mode

Object 169 "Thermostat – Operating Mode (prev/next)" can be used to change operating modes one by one via writing "True/False".

True= Next

False= Previous

Object 170 "Thermostat – Operating Mode Status" will give status of operating mode after change.

**1 bit Object [DPT\_Enable]:**

	171	Thermostat	Comfort Mode	1 bit	enable	C R W T U
	172	Thermostat	Standby Mode	1 bit	enable	C R W T U
	173	Thermostat	Night Mode	1 bit	enable	C R W T U
	174	Thermostat	Protection Mode	1 bit	enable	C R W T U

**3.4.2. Setpoint Temperature****Min. Setpoint Value: [4...16...40]**

Defines the minimum temperature setpoint value for the thermostat function. Any temperature value lower than Min. Setpoint Value cannot be written or selected on following objects.

	179	Thermostat	Setpoint	2 bytes	temperature (°C)	C R W T U
	180	Thermostat	Setpoint (-/+)	1 bit	boolean	C R W T U

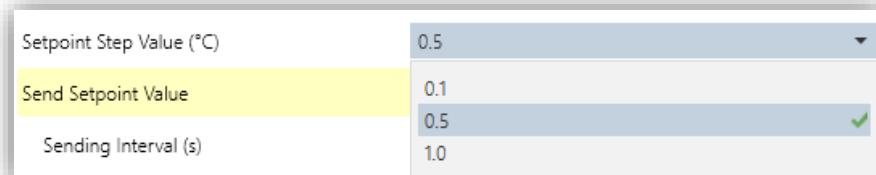
**Max. Setpoint Value: [4...32...40]**

Defines the maximum temperature setpoint value for the thermostat function. Any temperature value higher than Max. Setpoint Value cannot be written or selected on following objects.

	179	Thermostat	Setpoint	2 bytes	temperature (°C)	C R W T U
	180	Thermostat	Setpoint (-/+)	1 bit	boolean	C R W T U

**Setpoint Step Value: [0.1...0.5...1]**

Increase/Decrease value of current setpoint by writing “True/False” to Object 180 “Thermostat – Setpoint (-/+).”

**Send Setpoint Value (°C): [Cyclic...Cyclic on change]**

Current setpoint can be sent cyclically or by change of measured temperature via Object Number 181 “Thermostat – Setpoint Indication”.

Sending interval (s) [0...60...65535 s]      0=Inactive

1.5.8 Eclipse Push-Button Switch > Thermostat > Setpoint Temperature

General	Min. Setpoint Value	16
Switch Configuration	Max. Setpoint Value	32
+ Rocker 1	Setpoint Step Value (°C)	0.5
Rocker 2	Send Setpoint Value	<input checked="" type="radio"/> Cyclic <input type="radio"/> Cyclic and on Change
Rocker 3	Sending Interval (s)	60 (0=Inactive)
Rocker 4	<b>Heating Mode Setpoints</b>	
Temperature Sensor	Comfort	22 °C
- Thermostat	Standby	20 °C
Thermostat Settings	Night	18 °C
Setpoint Temperature	Protection	7 °C
Heating		

### Heating Setpoints:

Thermostat has "4" operating modes; "Comfort Mode, Standby Mode, Night Mode and Protection Mode". Each operating mode has their own predefined setpoint temperature.

Changover of operating modes can be achieved through "Operating Mode" communication objects which are explained in [#3.4.1. Thermostat Settings](#).

Note: Eclipse Switch starts with "Comfort Mode" as default after download.

Rocker 3	<b>Heating Mode Setpoints</b>	
Rocker 4	Comfort	22 °C
Temperature Sensor	Standby	20 °C
- Thermostat	Night	18 °C
Thermostat Settings	Protection	7 °C
Setpoint Temperature		
Heating		

Cooling Setpoints:

Rocker 3	Cooling Mode Setpoints		
Rocker 4	Comfort	22	°C
Temperature Sensor	Standby	24	°C
- Thermostat	Night	26	°C
	Protection	35	°C
Thermostat Settings			
<b>Setpoint Temperature</b>			
Cooling			

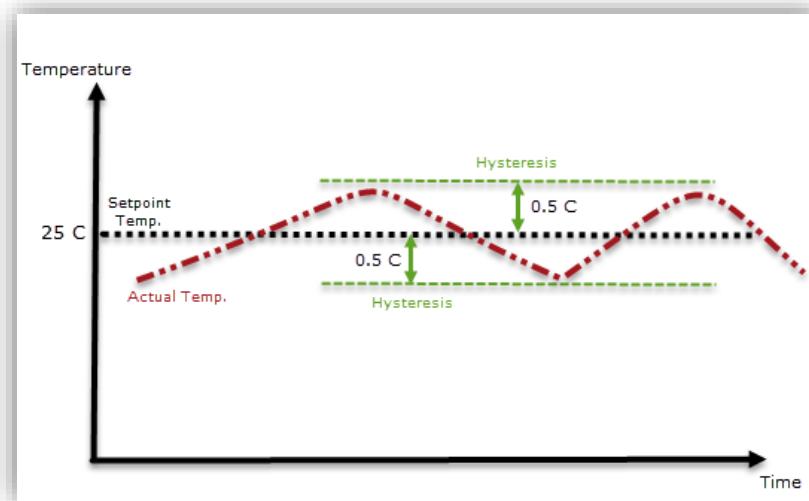
**3.4.3. Heating**Control Type: [2-Point Control (On/Off), Switching PI Control (PWM), Continuous PI Control]

Control Type	2-Point Control (On/Off)
Invert Control Value	<input checked="" type="checkbox"/> 2-Point Control (On/Off)
Cyclic Sending Interval (min)	Switching PI Control (PWM) Continuous PI Control

**3.4.3.1. Heating - Control Type: [2-Point Control ON/OFF]**

1.5.8 Eclipse Push-Button Switch > Thermostat > Heating		
General	Control Type	2-Point Control (On/Off)
Switch Configuration	Invert Control Value	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
+ Rocker 1	Cyclic Sending Interval (min)	1 (0=Inactive)
Rocker 2	Hysteresis ± (x0.1°C)	5
Rocker 3	Additional Stage	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Rocker 4	Disabled from Bus	<input checked="" type="radio"/> No <input type="radio"/> Yes
Temperature Sensor	Offset from Setpoint (x0.1°C)	20
- Thermostat	Hysteresis ± (x0.1°C)	5
Thermostat Settings	Cyclic Sending Interval (min)	0 (0=Inactive)
<b>Heating</b>		

Operates as a simple switch around the setpoint temperature using hysteresis values. "Hysteresis" prevents the output value from oscillation and give larger margin to turning heat or cool on and off. If system is more an active system, hysteresis values should be given larger and more inactive values.



**Invert Control Value:** Output value can be used inverse.

**Cyclic Sending Interval (min):** [0...1...255]

Determines cyclic sending period of Object Number 157 "Thermostat - Heating Control Value".

**Hysteresis +/- (x 0.1 °C):** [0...5...255]

Determines Hysteresis value to control Heating Control Value Output more accurate. "Hysteresis" prevents the output value from oscillation and give larger margin to turning heat or cool ON and OFF. If system is more an active system, hysteresis values should be given larger and more inactive values.

### 3.4.3.2. Heating - Control Type: [Switching PI Control PWM]

1.5.9 Eclipse Push-Button Switch > Thermostat > Heating

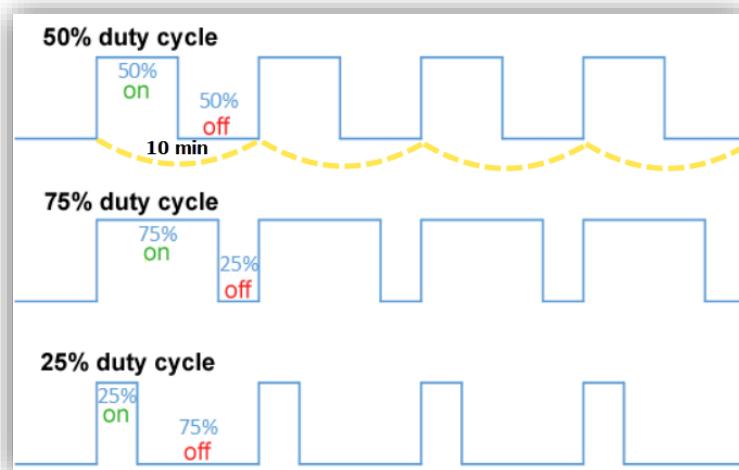
General	Control Type	Switching PI Control (PWM)
Switch Configuration	Invert Control Value	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
+ Rocker 1	PWM Period Time (min)	10
Rocker 2	Type of Heating	Floor Heating (5K/240min)
Rocker 3	Additional Stage	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Rocker 4	Disabled from Bus	<input type="radio"/> No <input checked="" type="radio"/> Yes
Temperature Sensor	Offset from Setpoint (x0.1°C)	20
- Thermostat	Hysteresis ± (x0.1°C)	5
Thermostat Settings	Cyclic Sending Interval (min)	0 (0=Inactive)
Setpoint Temperature		
<b>Heating</b>		

PI algorithm is used to calculate control signal. After calculation, control signal is converted into a pulse-interval signal. This means PWM cycle is divided into “1 bit ON/OFF” output commands based on control value. PWM period and type of heating should be selected according to the used room and type of heating.

**Invert Control Value:** Output value can be used inverse.

**PWM Period Time (min):** [1...10...255]

Defines PWM period time. If control value is calculated %50. Then control value will be ON for 5 minutes and OFF for second 5 minutes. Please check following graphic.



**Type of Heating:** Multiple heating types with preset parameters are available to the user.

- Floor Heating (5K/240)
- Hot Water Heating (5K/150)
- Electrical Heating (4K/100)
- Fan coil (4K/90)
- User Defined

If the required heating type is not available, individual parameters can be specified in the “User Defined” configuration.

#### **Proportional Range (x0.1 °C): [10...50...100]**

Defines the proportional range of control. Parameter changes the control speed of the controller.

#### **Integration Time (min): [0...240...255]**

Defines the reset time of controller. Integration Time has the effect of moving the room temperature slowly toward, and ultimately reaching the setpoint value. Depending on the type of system used, parameter needs to have different values. In general, the more inactive the overall system, the greater time is needed.

### **3.4.3.3. Heating - Control Type: [Continuous PI Control PWM]**

1.5.9 Eclipse Push-Button Switch > Thermostat > Heating		
General	Control Type	Continuous PI Control
Switch Configuration	Invert Control Value	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
+ Rocker 1	Type of Heating	Floor Heating (5K/240min)
Rocker 2	Send Value on Change (%)	0
Rocker 3	Cyclic Sending Interval (min)	1 (0=Inactive)
Rocker 4	Additional Stage	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Temperature Sensor	Disabled from Bus	<input type="radio"/> No <input checked="" type="radio"/> Yes
- Thermostat	Offset from Setpoint (x0.1°C)	20
Thermostat Settings	Hysteresis ± (x0.1°C)	5
Setpoint Temperature	Cyclic Sending Interval (min)	0 (0=Inactive)
Heating		

PI algorithm is used to calculate control signal and adjusts its output value between 0% and 100% to match the difference between the actual temperature and the setpoint temperature and enables an accurate regulation of the room temperature to the setpoint value. PI values should be selected compatible with the room and the type of heating system that needs to be controlled. Default PI values are defined for most common heating types. User defined values can be used for different rooms and different heating types for

better performance. Using default values as a reference point and adjusting these values according to system might increase controller performance.

**Invert Control Value:** Output value can be used inverse.

**Type of Heating:** Multiple heating types with preset parameters are available to the user.

- Floor Heating (5K/240)
- Hot Water Heating (5K/150)
- Electrical Heating (4K/100)
- Fan coil (4K/90)
- User Defined

If the required heating type is not available, individual parameters can be specified in the “User Defined” configuration.

**Send Value on Change (%): [0...100]**      0=inactive

Heating control value will be sent on change of percentage via Object Number 157 “Temperature – Heating Control Value”.

**Cyclic Sending Interval (min): [0...1...255]**

Determines cyclic sending period of Object Number 157 “Thermostat - Heating Control Value”.

#### 3.4.3.4. Heating - Additional Stage

**Additional Stage:** [Disable...Enable]

Additional Heating Control object can be enabled if an extra Heating/Cooling Control Value is needed on top of main Heat/Cool Control Value.

Object Number 164 “Thermostat – Additional Heating Control Value” is created when parameter is enabled.

**Disable from Bus:** Object Number 164 “Thermostat – Additional Heating Control Value” can be disable any time while writing True/False to Object Number 162 “Thermostat – Additional Heating Controller Disable”.

**Offset from Setpoint (x 0.1 °C): [1...20...255]**

Defines a separate setpoint value based on main Setpoint temperature for Object Number 164 “Thermostat – Additional Heating Control Value”. Thus, Additional Heating Source will be activated/deactivated depending on new temperature setpoint.

Example: Assume that a room has two type of different heating sources. (Main heating source, additional heating source)

Setpoint temperature is 24 degree for the “Heating Control Value” (main heating source.)

If “Offset from Setpoint” parameter is default value;  $20 \times 0.1 \text{ } ^\circ\text{C} = 2 \text{ } ^\circ\text{C}$ .

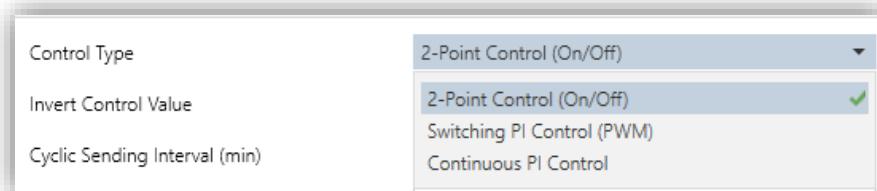
In this scenario, the setpoint for heating is set to  $24 \text{ } ^\circ\text{C}$ . When the temperature falls below  $22 \text{ } ^\circ\text{C}$ , additional heating should be switched on so that the room heats up again rapidly.

**Hysteresis +/- (x 0.1 °C): [0...5...255]**

Determines Hysteresis value to control Additional Heating Control Value Output more accurate. "Hysteresis" prevents the output value from oscillation and give larger margin to turning heat or cool ON and OFF. If system is more an active system, hysteresis values should be given larger and more inactive values.

**Cyclic Sending Interval (min): [0...255]**

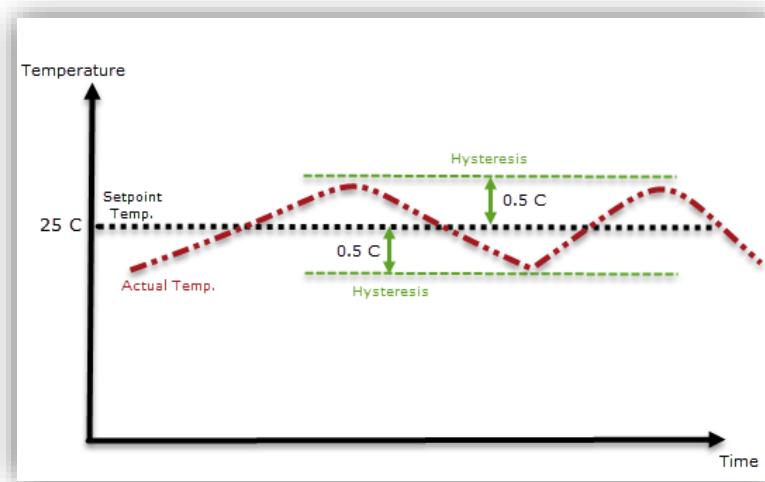
Determines cyclic sending period of Object Number 164 "Thermostat - Additional Heating Control Value".

**3.4.4. Cooling****Control Type: [2-Point Control (On/Off), Switching PI Control (PWM), Continuous PI Control]****3.4.4.1. Cooling - Control Type: [2-Point Control ON/OFF]**

1.5.8 Eclipse Push-Button Switch > Thermostat > Cooling

General	Control Type	2-Point Control (On/Off)
Switch Configuration	Invert Control Value	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
+ Rocker 1	Cyclic Sending Interval (min)	1 (0=Inactive)
Rocker 2	Hysteresis ± (x0.1°C)	5
Rocker 3	Additional Stage	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Rocker 4	Disabled from Bus	<input checked="" type="radio"/> No <input type="radio"/> Yes
Temperature Sensor	Offset from Setpoint (x0.1°C)	20
- Thermostat	Hysteresis ± (x0.1°C)	5
Thermostat Settings	Cyclic Sending Interval (min)	0 (0=Inactive)
Setpoint Temperature		
Cooling		

Operates as a simple switch around the setpoint temperature using hysteresis values. "Hysteresis" prevents the output value from oscillation and give larger margin to turning heat or cool on and off. If system is more an active system, hysteresis values should be given larger and more inactive values.



**Invert Control Value:** Output value can be used inverse.

**Cyclic Sending Interval (min):** [0...1...255]

Determines cyclic sending period of Object Number 157 "Thermostat - Heating Control Value".

**Hysteresis +/- (x 0.1 °C):** [0...5...255]

Determines Hysteresis value to control Cooling Control Value Output more accurate. "Hysteresis" prevents the output value from oscillation and gives larger margin to turning heat or cool ON and OFF. If system is more an active system, hysteresis values should be given larger and more inactive values.

### 3.4.4.2. Cooling - Control Type: [Switching PI Control PWM]

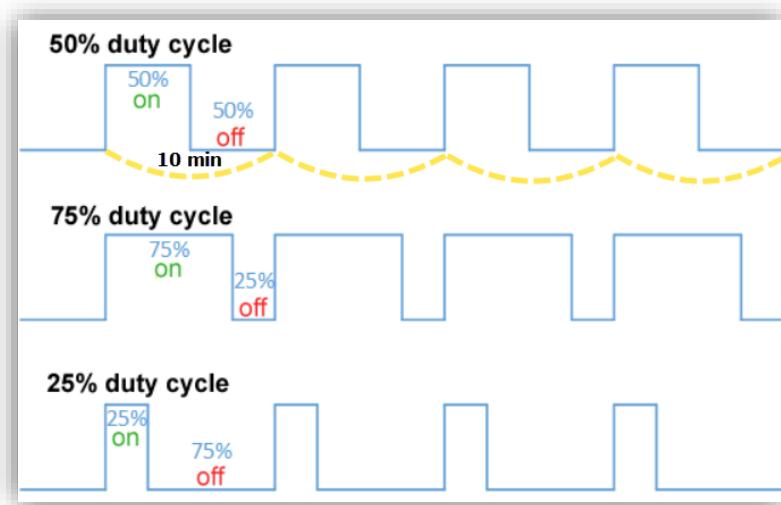
1.5.8 Eclipse Push-Button Switch > Thermostat > Cooling		
<ul style="list-style-type: none"> <li>General</li> <li>Switch Configuration</li> <li>+ Rocker 1</li> <li>Rocker 2</li> <li>Rocker 3</li> <li>Rocker 4</li> <li>Temperature Sensor</li> <li>- Thermostat</li> <li>Thermostat Settings</li> <li>Setpoint Temperature</li> </ul>	Control Type	Switching PI Control (PWM)
	Invert Control Value	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
	PWM Period Time (min)	10
	Type of Cooling	User Defined
	Proportional Range (x0.1°C)	50
	Integration Time (min)	240
	Additional Stage	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
	Disabled from Bus	<input type="radio"/> No <input checked="" type="radio"/> Yes
	Offset from Setpoint (x0.1°C)	20
	Hysteresis ± (x0.1°C)	5
Cyclic Sending Interval (min)	0 (0=Inactive)	

PI algorithm is used to calculate control signal. After calculation, control signal is converted into a pulse-interval signal. This means PWM cycle is divided into “1 bit ON/OFF” output commands based on control value. PWM period and type of cooling should be selected according to the used room and type of cooling source.

**Invert Control Value:** Output value can be used inverse.

**PWM Period Time (min):** [1...10...255]

Defines PWM period time. If control value is calculated %50. Then control value will be ON for 5 minutes and OFF for second 5 minutes. Please check following graphic.



**Type of Cooling:** Multiple cooling types with preset parameters are available to the user.

Cooling Ceiling (5K/240)

Fan coil (4K/90)

User Defined

If required cooling type is not available, individual parameters can be specified in the “User Defined” configuration.

**Proportional Range (x0.1 °C):** [10...50...100]

Defines the proportional range of control. Parameter changes the control speed of the controller.

**Integration Time (min):** [0...240...255]

Defines the reset time of controller. Integration Time has the effect of moving the room temperature slowly toward, and ultimately reaching the setpoint value. Depending on the type of system used, parameter needs to have different values. In general, the more inactive the overall system, the greater time is needed.

### 3.4.4.3. Cooling - Control Type: [Continuous PI Control PWM]

1.5.8 Eclipse Push-Button Switch > Thermostat > Cooling

General	Control Type	Continuous PI Control
Switch Configuration	Invert Control Value	<input checked="" type="radio"/> Not Inverted <input type="radio"/> Inverted
+ Rocker 1	Type of Cooling	User Defined
Rocker 2	Proportional Range (x0.1°C)	50
Rocker 3	Integration Time (min)	240
Rocker 4	Send Value on Change (%)	0 (0=Inactive)
Temperature Sensor	Cyclic Sending Interval (min)	1 (0=Inactive)
- Thermostat	Additional Stage	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Thermostat Settings	Disabled from Bus	<input type="radio"/> No <input checked="" type="radio"/> Yes
Setpoint Temperature	Offset from Setpoint (x0.1°C)	20
Cooling	Hysteresis ± (x0.1°C)	5
	Cyclic Sending Interval (min)	0 (0=Inactive)

PI algorithm is used to calculate control signal and adjusts its output value between 0% and 100% to match the difference between the actual temperature and the setpoint temperature and enables an accurate regulation of the room temperature to the setpoint value. PI values should be selected compatible with the room and the type of heating system that needs to be controlled. Default PI values are defined for most common cooling types. User defined values can be used for different rooms and different cooling types for better performance. Using default values as a reference point and adjusting these values according to system might increase controller performance.

**Invert Control Value:** Output value can be used inverse.

**Type of Heating:** Multiple heating types with preset parameters are available to the user.

Cooling Ceiling (5K/240)

Fan coil (4K/90)

User Defined

If the required cooling type is not available, individual parameters can be specified in the "User Defined" configuration.

**Send Value on Change (%): [0...100]** 0=inactive

Cooling control value will be sent on change of percentage via Object Number 158 "Temperature – Cooling Control Value".

**Cyclic Sending Interval (min): [0...1...255]**

Determines cyclic sending period of Object Number 158 "Thermostat – Cooling Control Value".

### 3.4.4.4. Cooling - Additional Stage

#### Additional Stage: [Disable...Enable]

Additional Cooling Control object can be enabled if an extra Cooling Control Value is needed on top of main Cooling Control Value.

Object Number 167 "Thermostat – Additional Cooling Control Value" is created when parameter is enabled.

**Disable from Bus:** Object Number 167 "Thermostat – Additional Cooling Control Value" can be disable any time while writing True/False to Object Number 165 "Thermostat – Additional Cooling Controller Disable".

#### Offset from Setpoint (x 0.1 °C): [1...20...255]

Defines a separate setpoint value based on main Setpoint temperature for Object Number 167 "Thermostat – Additional Cooling Control Value". Thus, Additional Cooling Source will be activated/deactivated depending on new temperature setpoint.

Example: Assume that a room has two type of different cooling sources. (Main cooling source, additional cooling source)

Setpoint temperature is 24 degree for the "Cooling Control Value" (main cooling source.)

If "Offset from Setpoint" parameter is default value;  $20 \times 0.1 \text{ } \text{C}^\circ = 2 \text{ } \text{C}^\circ$ .

In this scenario, the setpoint for cooling is set to 24 °C. When the temperature rises above 26 °C, additional cooling should be switched on so that the room cools off again rapidly.

#### Hysteresis +/- (x 0.1 °C): [0...5...255]

Determines Hysteresis value to control Additional Cooling Control Value Output more accurate. "Hysteresis" prevents the output value from oscillation and give larger margin to turning heat or cool ON and OFF. If system is more an active system, hysteresis values should be given larger and more inactive values.

#### Cyclic Sending Interval (min): [0...255]

Determines cyclic sending period of Object Number 167 "Thermostat - Additional Cooling Control Value".

---

### 3.4.5. Heating & Cooling

Control mode of thermostat can be selected for Heating, Cooling, and Heating & Cooling.

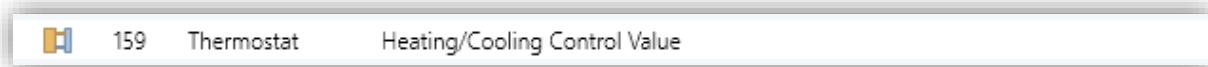
If Heating & Cooling control mode is selected parameter tabs of "Heating" and "Cooling" will be same. But parameter tabs of "Thermostat Settings" and "Setpoint Temperature" will have some additional parameters.

Please check below.

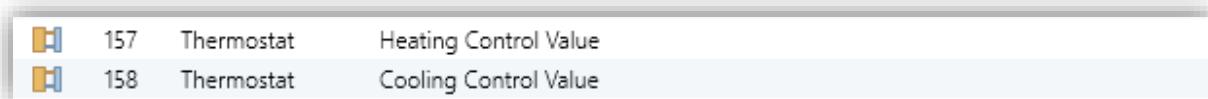
### 3.4.5.1. Thermostat Settings

**Heating & Cooling Control Value Output:** Output value for Heating and Cooling can be sent via same object or 2 separate objects.

If “via 1 Object” option is selected Object Number 159 “Thermostat – Heating/Cooling Control Value” will be activated.



If “via 2 Objects” option is selected Object Number 157 “Thermostat – Heating Control Value” and Object Number 158 “Thermostat – Cooling Control Value” will be activated.



1.5.8 Eclipse Push-Button Switch > Thermostat > Thermostat Settings

General	Heating & Cooling Control Value Output	<input type="radio"/> via 1 Object <input checked="" type="radio"/> via 2 Objects
Switch Configuration	Behaviour of Control Mode at Bus Recovery	As before voltage failure
+ Rocker 1	Switchover Control Mode	<input type="radio"/> Automatically <input checked="" type="radio"/> via Object
Rocker 2	Operating Mode at Bus Recovery	As before voltage failure
Rocker 3	Operating Mode Object	<input type="radio"/> 1 bit Objects <input checked="" type="radio"/> 1 Byte Object [DPT_HVACMode]
Rocker 4		
Temperature Sensor		
- Thermostat		
<b>Thermostat Settings</b>		
Setpoint Temperature		
Heating		
Cooling		

**Behavior of Control Mode at Bus Recovery:**

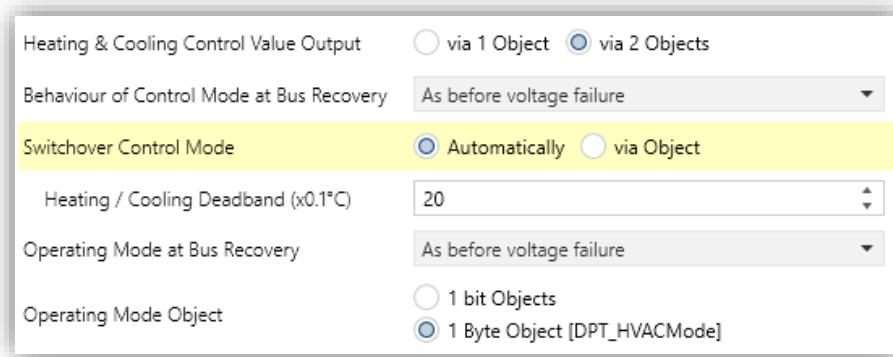
The parameter defines the behavior of the control mode after bus power return. Control mode can be changed to following options after a power return:

- As before voltage failure
- Heating
- Cooling

**Switchover of Control Mode:** [Automatically, via Object]

Parameter makes possible to switch between the heating and cooling mode of the device.

Automatically: The device switches automatically between heating and cooling and to the associated setpoint according to defined "Deadband". Object Number 156 "Thermostat – Heat/Cool Status" will transmit the status after switchover.

**Heating Cooling Deadband (x 0.1 °C):** [0...20...255]

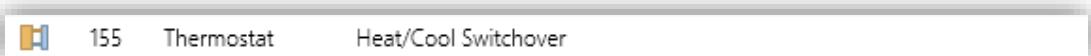
Deadband defines the range between setpoint temperature and measured temperature. If deadband is exceeded, switchover will be applied.

**Via Object:**

Switchover can be applied "via Object" manually using Object Number 155 "Thermostat – Heat/Cool Switchover".

\$01= Heating [1.100 DPT\_cooling/heating]

\$00= Cooling [1.100 DPT\_cooling/heating]



1.5.8 Eclipse Push-Button Switch > Thermostat > Setpoint Temperature

General	Min. Setpoint Value	16
Switch Configuration	Max. Setpoint Value	32
+ Rocker 1	Setpoint Step Value (°C)	0.5
Rocker 2	Send Setpoint Value	<input type="radio"/> Cyclic <input checked="" type="radio"/> Cyclic and on Change
Rocker 3	Sending Interval (s)	60 (0=Inactive)
Rocker 4	Heating Mode Setpoints	
Temperature Sensor	Comfort	22 °C
- Thermostat	Standby	20 °C
	Night	18 °C
	Protection	7 °C
Thermostat Settings	Cooling Mode Setpoints	
Setpoint Temperature	Comfort	22 °C
Heating	Standby	24 °C
Cooling	Night	26 °C
	Protection	35 °C

### 3.4.5.2. Heating/Cooling Object Description

#### Heating / Cooling Indication

Object Number 160 "Thermostat – Heating Indication" defines a state for recent heating command. It indicates that heating source is recently having an active command to heat. In same way, Object Number 161 "Thermostat – Cooling Indication" defines a state for recent cooling command. It indicates that cooling source is recently having an active command to cool.

	160	Thermostat	Heating Indication	1 bit	state
	161	Thermostat	Cooling Indication	1 bit	state

Example: Heating mode is active. Setpoint Temperature 22 °C, Actual Temperature 21 °C.

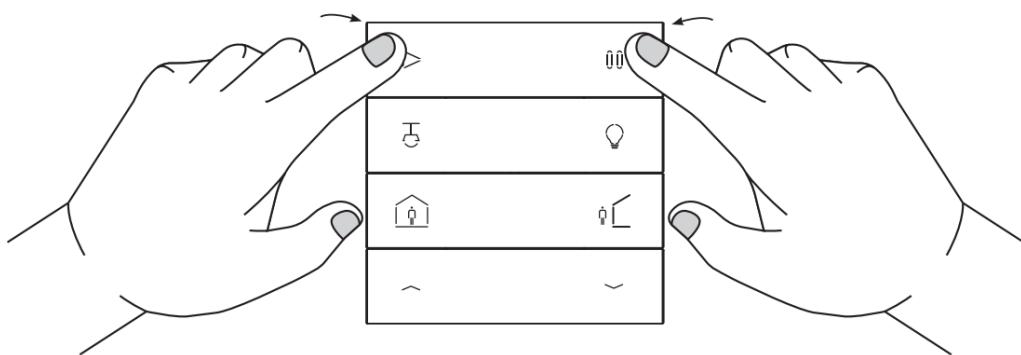
Heating control value is sending ON command to heating source and "heating indication" is instantly informing about heating command.

10:59:52.375	1.5.8	0/7/3	GroupValue_Write	Setpoint Indication	0C 4C   22 °C
10:59:52.398	1.5.8	0/7/6	GroupValue_Write	Actual Temperature	0C 6A   22.6 °C
11:00:26.114	1.5.8	0/7/4	GroupValue_Write	Heating Control Value	\$00   Off
11:00:52.635	1.5.8	0/7/3	GroupValue_Write	Setpoint Indication	0C 4C   22 °C
11:00:52.658	1.5.8	0/7/6	GroupValue_Write	Actual Temperature	0C 6A   22.6 °C
11:01:05.541	15.15.241	0/7/18	GroupValue_Write	External Value	0C 1A   21 °C
11:01:07.700	1.5.8	0/7/6	GroupValue_Write	Actual Temperature	0C 1A   21 °C
11:01:08.299	1.5.8	0/7/4	GroupValue_Write	Heating Control Value	\$01   On
11:01:08.320	1.5.8	0/7/23	GroupValue_Write	Heating Indication	\$01   Active

## 4. Commissioning

For commissioning the device, the following activities are required:

- Make electrical connections
- Turn on the bus power supply
- Switch the device operation to programming mode
  - Alternatively, instead of using programming button, it is possible to switch operation of the device to programming mode by pressing button 1 and button 2 simultaneously for 5 seconds



- Download into device the physical address and the configuration with ETS program
- At the end of the download operation of the device returns to normal mode
- Now the device is programmed and ready to use



Configuration and commissioning of the device require the use of ETS4 or later releases. These activities must be carried out according to the design of the building automation system done by a qualified planner.