INTERRA

Developer of Uniqueness

Multi Presence Detector

Product Manual

PM2407040175DEN

Contents

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded versions by updates. It is your responsibility to ensure that your application meets your specifications.

INTERRA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR NONSTATUTORY, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE.

Interra disclaims all liability arising from this information and its use. Use of Interra devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Interra from any damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise under any Interra intellectual rights.

Trademarks

The Interra name, logo and the Interra ITR415 - 0XXX Multi Presence Detector devices are registered trademarks of Interra Technology in Turkey and other countries.

All other trademarks mentioned herein are property of the Interra Technology.

Printed on recycled paper.

TS EN ISO 9001:2008

TO OUR CUSTOMERS

One of our most important aims is to provide you with the best documentation possible to use successfully your Interra products. Focusing on this, we will keep on improving our documentation to better suit your needs. Our publications will be updated as new volumes as soon as changes are introduced.

If you have any questions or comments regarding this publication, do not hesitate to contact us:

E-mail: info@interratechnology.com

Tel: +90 (216) 326 26 40 Fax: +90 (216) 324 25 03

Most Current Product Manual

To obtain the most up-to-date version of this product manual, please visit our Web site at:

http://www.interratechnology.com

You can determine the version of an Interra document by examining its literature number found on the bottom right corner of any page.

The first two letters of the literature are the type of document. The numbers that follow are the creation date of the document and the last letter is the version (e.g., PM181017001A is the version A of a product manual created on the date 17/10/18.

1. Content of The Document

This document contains Interra ITR415-0XXX coded Multi Presence Detector devices' electronic and all essential feature information for programming the products. In each subtitle is explained the characteristics of the device. Modifications of the product and special change requests are only allowed in coordination with product management.

This manual provides detailed technical information concerning ITR415-0XX1 Mid-Range Multi Presence Detector, ITR415-0XX2 Mid-Range Plus Multi Presence Detector, ITR415-0XX3 Wide-Range Multi Presence Detector, ITR415-0XX4 High Bay KNX Sensor and ITR415-0XX5 High Bay Plus KNX Sensor. All the models have the same software functionality so, the features described in this document apply to all versions.

This user manual is intended for use by KNX installers and describes the functions and parameters of the Interra Multi Presence Detector family devices and how it is possible to change the settings and configurations using the ETS software tool. This document also describes the installation, programming, commissioning and use of the devices with detailed information.

2. Product Description

ITR415-0XXX series Multi Presence Detector devices are the newest products of Interra Technology. The Interra KNX Presence detectors are designed for using at mainly in interior areas of buildings.

The range of Interra presence sensors is suitable for ceiling mounting and with flush mount or surface mount optionality.

Interra Multi Presence Detectors support presence detection, brightness detection, movement detection, temperature detection and external temperature detection.

All versions have a rear connector with 3 digital inputs that can be connected to buttons and used for switch sensor, switch/dimming sensor, shutter sensor, value/forced operation, scene control, RGB colour control and HVAC mode selection control.

Temperature measuring through an integrated sensor, analog input, KNX temperature sensor with the possibility of sending the value on change and periodically to the bus for monitoring the room temperature.

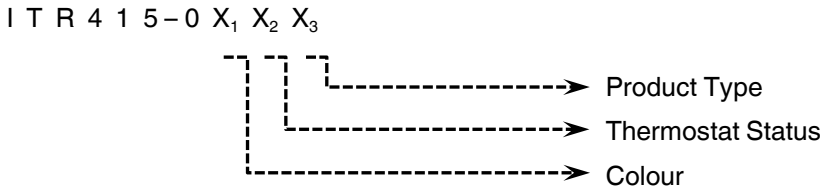
Interra Multi Presence Detectors have 5 logic function blocks and can be set the logical relation AND/OR/XOR. Each block can control 5 output objects.

2.1. Technical Information

The following table shows the technical information of the Multi Presence Detectors.

Product Name	Multi Presence Detector
Product Code	ITR415-0XXX
Power Supply	21..30 V DC
Current Consumption	5 mA
Brightness Detection	1-1200 lux
Inputs	3 x Inputs 2 x Digital and 1 x Digital/Analog
Number of Channels	4 x Presence 1 x Brightness 5 x Logic
Maximum Air Humidity	%90 RH
Pollution Degree	2
Type of Protection	Flush: IP 20 Surface: IP 44
Operation Temperature	-5°C…45°C
Storage Temperature	-10°C…60°C
Dimensions (Φ x H)	70 × 41.8 mm
Colour	Black and White
Certicitation	KNX Certified
Configuration	Configuration with ETS

2.2. Models And Variations



X_1	Colour	X_2	Thermostat Status
0	White	**0**	No Thermostat
1	Black	**1**	Thermostat
X_3	**Product Type**		
1	Mid-Range		
2	Mid-Range Plus		
3	Wide Range Plus		
4	High Bay		
5	High Bay Plus		
6	Microwave		

Table 1: Variations Table

2.3. Detection Ranges

This section provides information about the detection ranges of Interra presence sensor family devices. The motion detection ranges of each different model vary depending on the installation height. Therefore, detection distances should be carefully examined, and device installations should be made by these distances.

Mid-Range Multi Presence Detector:

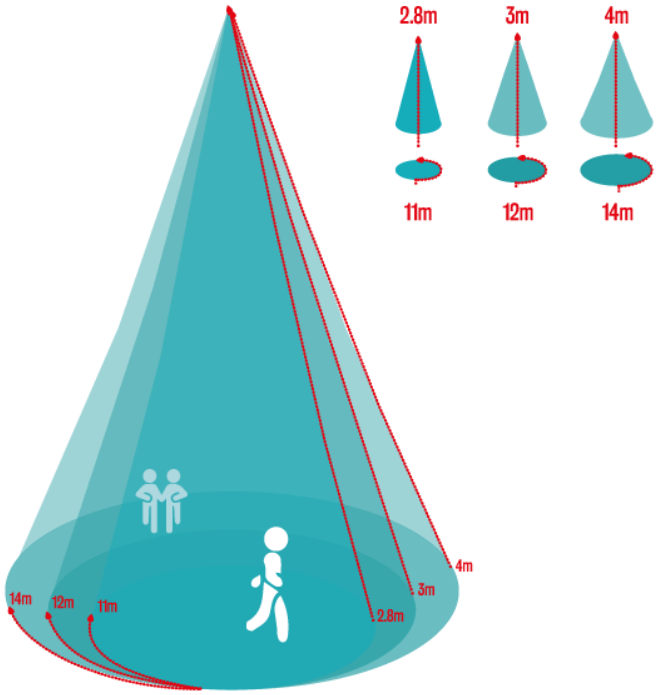


Fig. 1: Mid-Range Multi Presence Detector Detection Range

Mounting Height	Seated Activity	Walking Towards	Walking Across
2.8 m	5.5 m	6 m	11 m
3 m	6 m	7 m	12 m
4 m	7 m	8 m	14 m

Table 2: Mid-Range Detection Range Table

Mid Range Plus Multi Presence Detector:

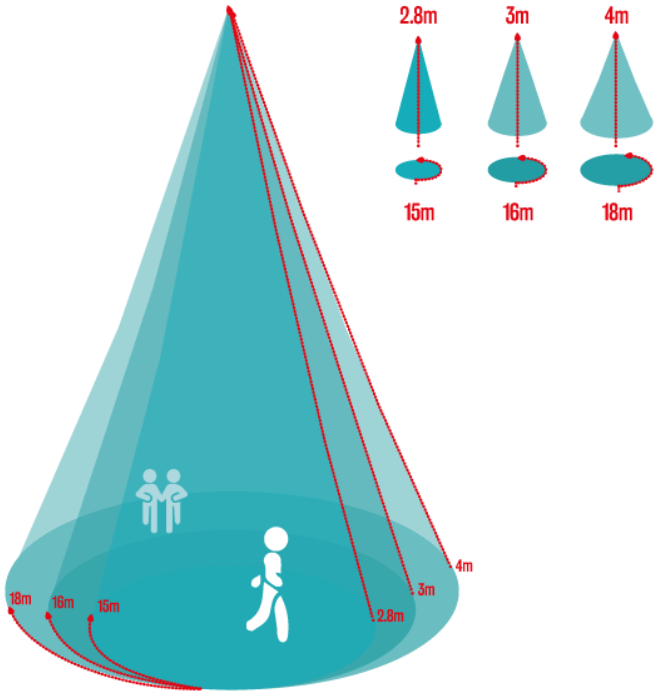


Fig. 2: Mid Range Plus Multi Presence Detector Detection Range

Mounting Height	Seated Activity	Walking Towards	Walking Across
2.8 m	6 m	9 m	15 m
3 m	6.5 m	11 m	16 m
4 m	7 m	12 m	18 m

Table 3: Mid-Range Plus Detection Range Table

Wide Range Multi Presence Detector:

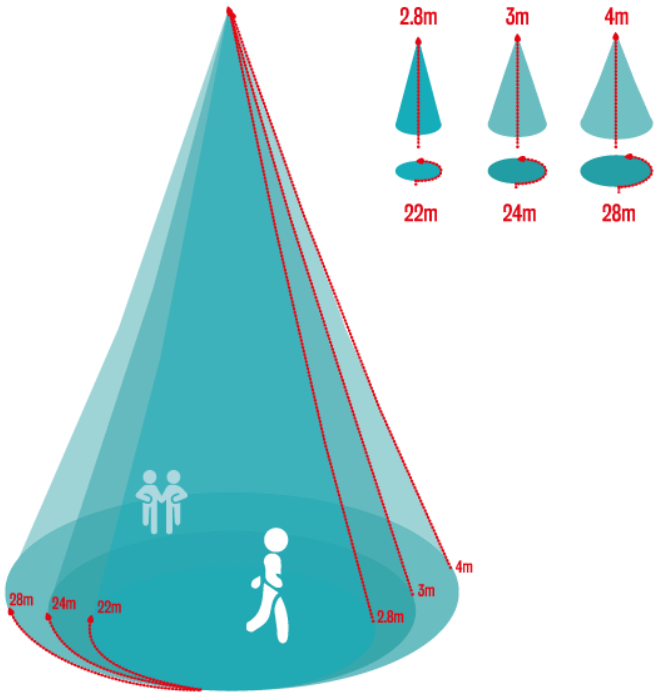


Fig. 3: Wide Range Multi Presence Detector Detection Range

Mounting Height	Seated Activity	Walking Towards	Walking Across
2.8 m	6 m	12 m	22 m
3 m	7 m	14 m	24 m
4 m	7.5 m	16 m	28 m

Table 4: Wide Range Detection Range Table

High Bay KNX Sensor:

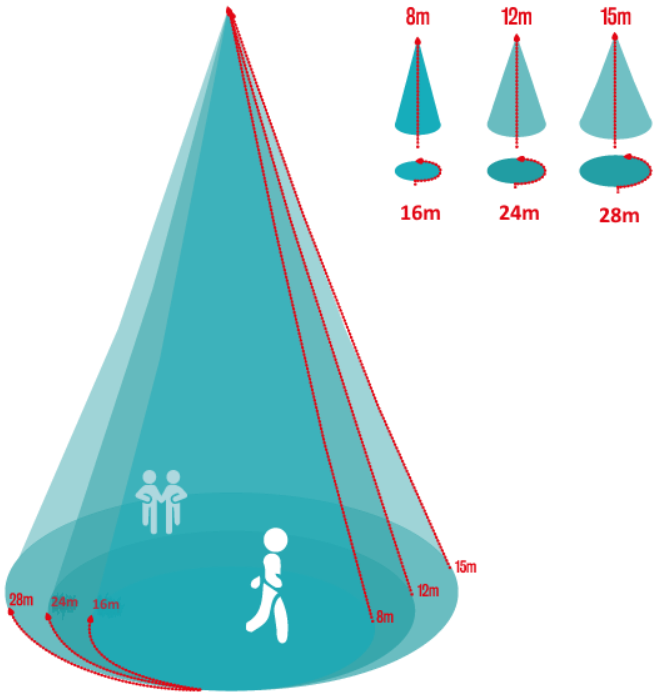


Fig. 4: High Bay KNX Sensor Detection Range

Mounting Height	Walking Across
4 m	8 m
8 m	16 m
12 m	24 m
15 m	28 m

Table 5: High Bay Detection Range Table

High Bay Plus KNX Sensor:

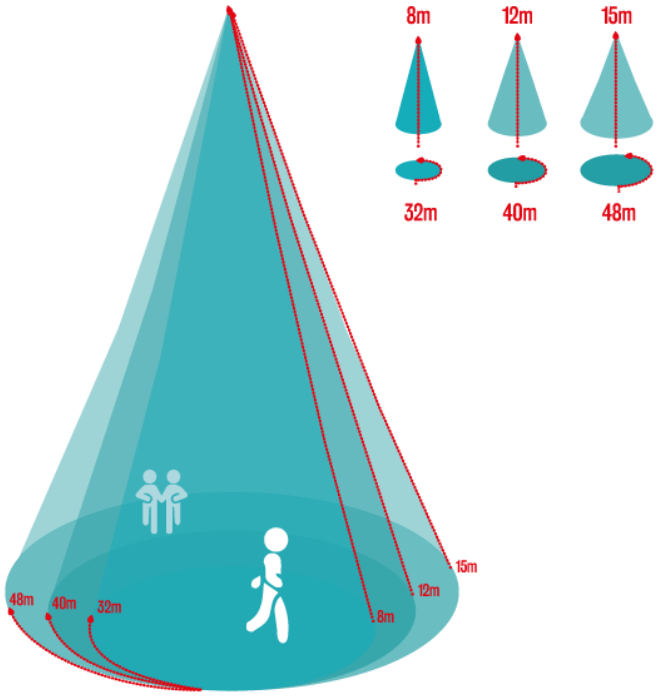


Fig. 5: High Bay Plus KNX Sensor Detection Range

Mounting Height	Walking Across
4 m	16 m
8 m	32 m
12 m	40 m
15 m	48 m

Table 6: High Bay Plus Detection Range Table

Microwave KNX Sensor:

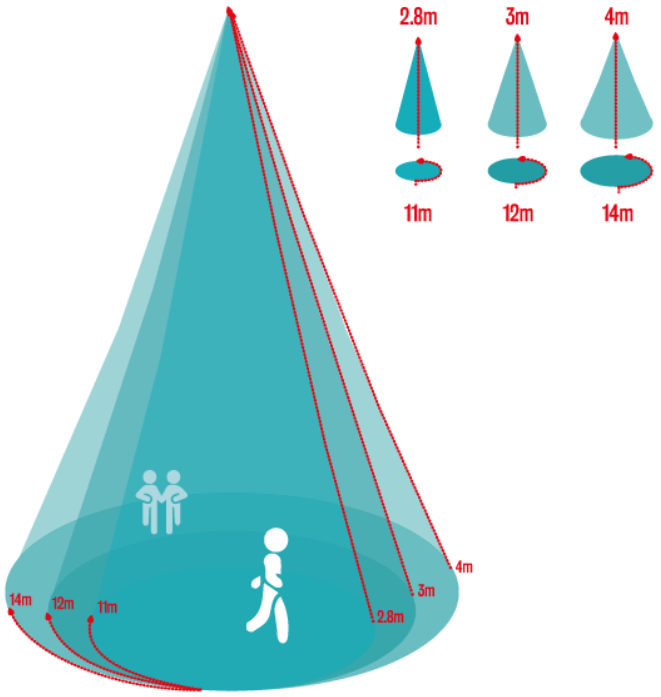


Fig. 6: Microwave KNX Sensor Detection Range

Mounting Height	Walking Across
2.8 m	11 m
3 m	12 m

Table 7: Microwave Detection Range Table

2.4. Connection Features

The figure below shows the Multi Presence Detector connectors. All of the ITR415-0XXX models have the same connection layout.

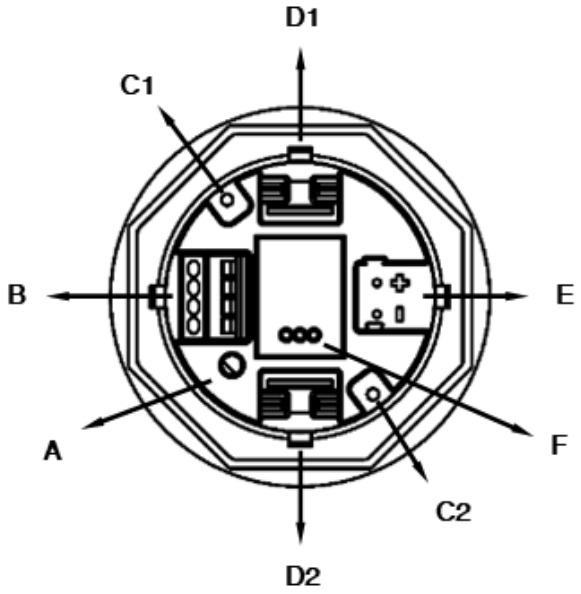


Fig. 7: Connection Features of Multi Presence Detector

Letter	Feature
A	KNX Programming Button
B	1 x Analog, 2 x Digital Input
C1	Metal Tab for Junction Box
C2	Metal Tab for Junction Box
D1	Metal Spring
D2	Metal Spring
E	KNX Connector
F	Programming connector

Table 8: Connection Features Table

2.5. Additional Temperature Probe

Fig. 8: Additional Temperature Probe

The table below lists the Temperature Probe compatible with our product. The Temperature Probe **is not supplied with the product** and customers must order this probe separately.

Product Code	**HM001718**
Description	Thermistor NTC 10K 5%
Cable Cross-section	24 AWG
Cable Length	1.5 m
Pin Count	2

Table 9: Additional Temperature Probe Technical Information Table

2.6. Dimensions

All values given in the device dimensions are millimetres. Interra Multi Presence Detectors can be used as flush mount or surface mount.

Flush Mounted:

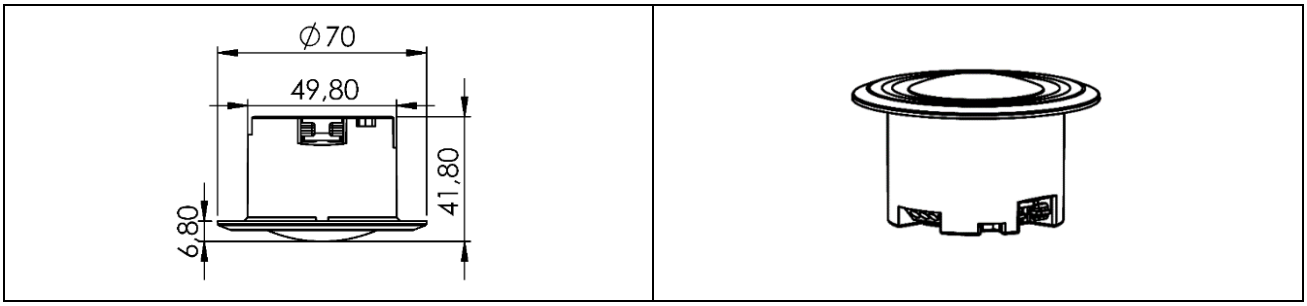


Fig. 9: Flush Mounted Dimensions Figure

Surface Mounted:

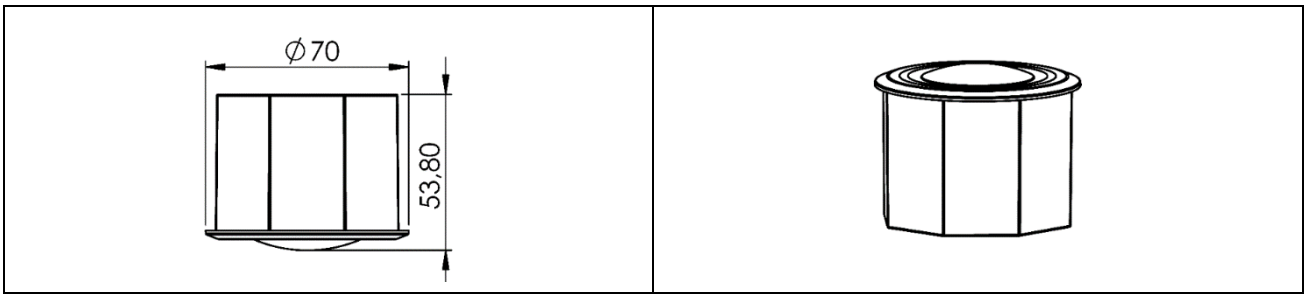


Fig. 10: Surface Mounted Dimensions Figure

2.7. Functionality

All the Interra Multi Presence Detectors offer various options to ensure that the brightness in the room is maintained at a more pleasant level. To ensure this, constant light switch and constant light control functions are mostly used There is a difference between the functions of the constant light switch and the constant light controller. Both functions ensure that the brightness does not drop below a certain level when persons are in the room. The use of a presence detector is especially practical for work stations in an office since even small movements are detected.

2.7.1. Constant Light Switch

The constant light switch switches on lamps in the room as soon as movement of a person is detected and the desired brightness value is not attained by entering daylight alone. The programmed setpoint minus hysteresis is maintained as long as people are in the detection range. The application detects when entering daylight is sufficient. The lamps are then switched off again to save energy.

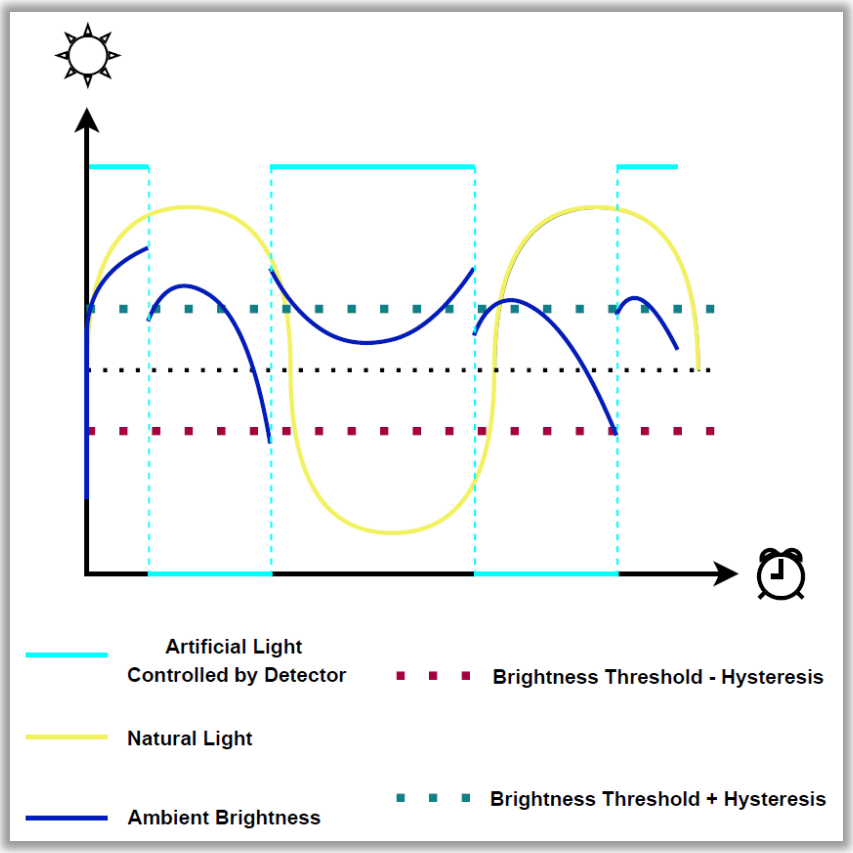


Fig. 11: Constant Light Switch Operation

2.7.2. Constant Light Control

In contrast to the constant light switch, there is the possibility of switching in several stages. Both the constant light controller and the constant light switch ensure that the level of brightness in the room does not drop below the desired level. However, the brightness controller is additionally able to send telegrams for dimming lamps to the KNX bus. This enables a constant level to be attained due to the dimming of lights brighter and darker, always in dependence on the natural light in the room. And the accuracy of the control increases with the operating time.

The Multi Presence Detectors can operate either in "Automatic" or "Automatic switch-off" mode in the Constant Light Control application. If an automatic switch-off is selected, for example, the light must be switched on manually via a control element. The light remains on as long as movement is detected and daylight is not sufficient. If no movement is detected, the light-on time delay expires. Only then is an OFF telegram sent to the bus via the output. In automatic mode, the movement sensor also takes over the switch-on function as soon as someone enters the room.

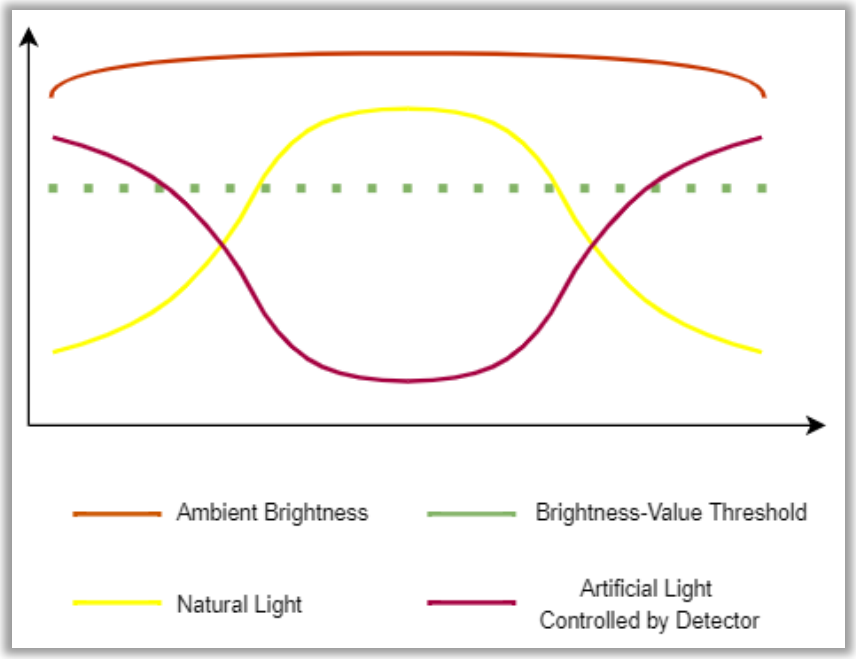


Fig. 12: Constant Light Control Operation

3. ETS Parameters & Descriptions

In this chapter, the ETS parameters of ITR415-XXXX Multi Presence Detector devices are described using the parameter pages and options. The parameter pages features are dynamic structures which mean further parameters and parameter pages are enabled depending on the configuration.

The words sensor and detector are used interchangeably in this document to mean 'device'. Therefore, both have the same meaning.

In the ETS parameter configuration pages, each of the parameters has got a default parameter value. These default values are written in bold.

- E.g. : Enable in operation ●**no** yes

3.1. General Page

When the ITR415-XXXX Multi Presence Detector ETS configuration file is attached to the project from the ETS software, a configuration setting must be made primarily before loading. When entering the “GENERAL” in the parameter page, the configuration screen will be appeared shown below. General settings for the devices are made in this window.

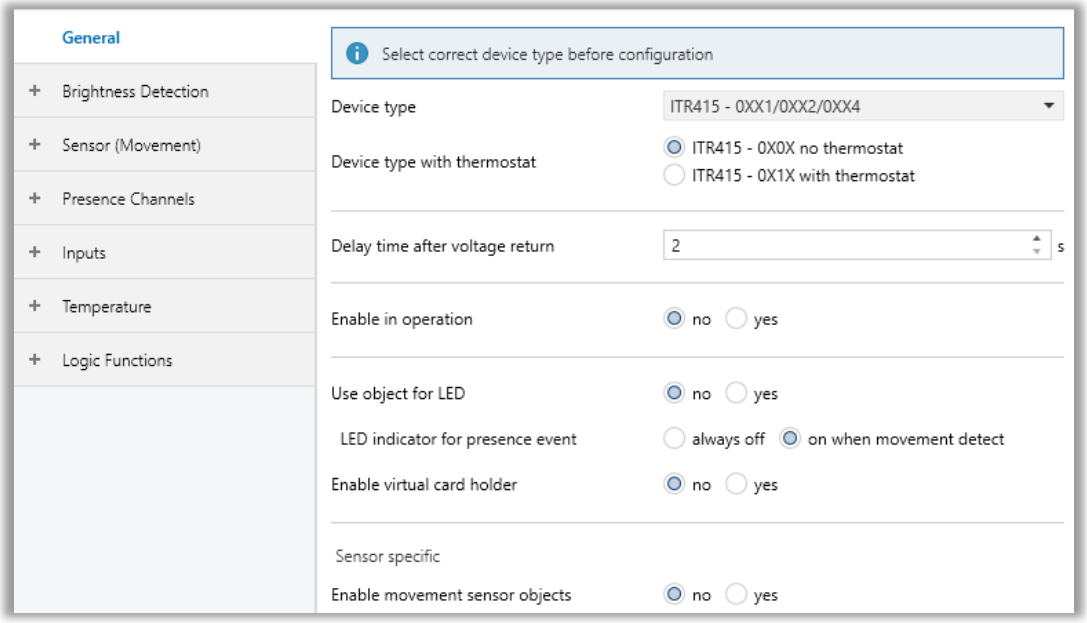


Fig. 13: General Configuration Page Page

3.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Device type	This parameter determines the device type.	**ITR415–0XX1/0XX2/0XX4** ITR415-0XX3/0XX5 ITR415-0XX6
Device type with thermostat	This parameter determines whether the product has a thermostat or not. If the thermostat type is selected, the "Room Controller" menu will be accessible.	**ITR415-0X0X no thermostat** ITR415-0X1X with thermostat
Delay time after voltage return	This parameter is used to determine the delay time after voltage return in seconds. When in a delayed state, the Multi Presence Detector does not send any KNX telegrams. Incoming telegrams are received and updated in the background. The updated values are only executed when the wait state ends and then sent according to the parametrization.	**2**...60
Enable In Operation	This parameter is used to determine the existence of the Multi Presence Detector on the KNX bus line. The cyclic telegram can be monitored by an external KNX device. If a telegram is not received, the device may be defective or the KNX cable to the transmitting device may be interrupted. **Yes:** The group object is enabled. **No:** The group object is not enabled.	**No** Yes
-> In operation send[1]	This parameter is used to determine the send value of the “General - In operation” group object on the KNX bus line.	Value ‘0’ **Value ‘1’**
-> In operation send interval (min)[1]	This parameter is used to set the cyclically sending time interval value of the “General - In operation” group object.	1...5...255
Use object for LED	This parameter is used to activate or deactivate the use of LED indicator KNX object during presence detection.	No Yes
-> LED indicator for the presence event[2]	This parameter is used to give status information via the blue LED when motion is detected by the Multi Presence Detector. **always off:** The blue LED will be off continuously. **on when movement detected:** The blue LED turn on when there is a motion detected by the sensor.	Always off On when movement detect
Enable Virtual Card Holder	This parameter is used to enable the virtual card hold of the Multi Presence Detector. If this parameter is activated, a separate parameter page will be opened for configuration.	No Yes

Sensor Specific		
Enable Movement Sensor Objects	This parameter is used to enable or disable each integrated presence detectors to send the detected movements to the KNX bus.	**No** Yes
Sensor Specific – ITR415-0XX6		
⚠ [0…25]% = low, [26…50]% = medium, [51…75]% = high, [76…100]% = very high		
-> Boost factor[3]	This parameter becomes available when the "Device Type" parameter is set to "ITR415 - 0XX6". This parameter is used to adjust the sensor detection range and sensitivity. "very high" indicates the highest sensitivity value.	Low Medium High **Very high**
-> Boost factor changed by bus[3]	If this parameter is selected as "Yes", the "Microwave sensor boost" object is activated. This parameter allows the Boost factor to be changed at run time. When the device is first turned on, it sends the last recorded value to the KNX line.	**No** Yes
-> Microwave sensor sensitivity[3]	This parameter becomes available when the "Device Type" parameter is set to "ITR415 - 0XX6". This parameter is used to adjust the sensor sensitivity. "very high" indicates the highest sensitivity value. **Note:** This value has little effect on the sensitivity setting.	Low Medium **High** Very high
-> Microwave sensor sensitivity changed by bus[3]	If this parameter is selected "Yes", the "Microwave sensor sensitivity" object is activated. This parameter allows the Sensitivity value to be changed at run time. When the device is first turned on, it sends the last recorded value to the KNX line.	**No** Yes

[*1] This parameter is visible when the function “Enable in operation” is set to “Yes”.

[*2] This parameter is visible when the function “Use object for LED” is set to “No”.

[*3] This parameter is visible when the function “Device type” is set to “ITR415-0XX6”.

3.2. Brightness Detection

On this parameter page, brightness functionality related configurations are made. Brightness calibration type, brightness measurement algorithm and brightness variation features can be configured on this page.

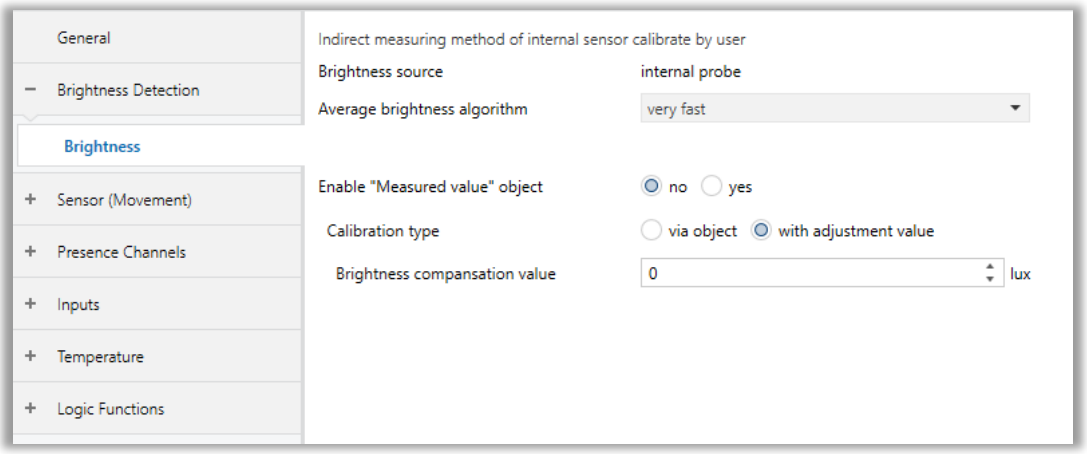


Fig. 14: Brightness Configuration Page

3.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Average brightness algorithm	This parameter is used to define the response speed of the controlled output after a measured ambient brightness variation. It defines the calculation speed of the average illuminance value, the faster the algorithm and the faster it reacts to a change in lux level. The "very fast" selection can lead to very frequent switching on and off of the light, the "very slow" selection can introduce delays in switching the light on or off.	**very fast** fast normal slow very slow
Enable "Measured Value" Object	This parameter is used to enable or disable the measured value object. The measured brightness value can be sent to the KNX bus via this object.	**No** Yes
-> Cyclic sending of brightness[1]	This parameter is used to determine the periodic transmission time of the brightness value to the KNX bus line. The values entered are in hours, minutes and seconds.	00:00:05...**00:00:30…** 18:12:15
-> Sending on variation[1]	This parameter is used to determine whether it will be sent to the bus line depending on the change of the ambient brightness value. **do not send:** The brightness value is never sent to the KNX bus line. **xx lux:** If there is a change from the last measured lux value equal to the lux value entered in the parameter, the new measured lux value is sent to the KNX bus line. **Note:** xx means the listed values in the parameter list such as 10 lux, 20 lux, 5 lux etc.	**Do not send** 5 lux, 10 lux, 15 lux, 20 lux, 25 lux, 30 lux, 35 lux, 40 lux, 45 lux, 50 lux, 55 lux, 60 lux, 65 lux, 70 lux, 75 lux
Calibration type	This parameter is used to determine the brightness sensor calibration method. Brightness calibration must be done after the sensor installation, for the sensor to measure correctly and therefore work ideally with other sensor features. **Via object:** Calibration is made through the KNX communication object. The brightness value measured with lux meter from a point in the sensing range of the sensor (usually just below the location where the sensor is installed) should be written into the KNX calibration object as a new lux value. **With adjustment value:** Calibration is made through the brightness compensation value parameter.	via object **with adjustment value**

-> Brightness compensation value[2]	This parameter is used to determine the brightness compensation value. The entered value is subtracted from the value measured by the sensor or added. **Ex:** The ambient brightness value measured by the sensor is 175 lux. Moreover, with a lux meter, the measured ambient brightness value is 200 lux. In this case, the sensor measures the ambient brightness value 25 lux less. For the sensor to measure accurately, the value 25 should be entered in the brightness compensation value parameter.	-200…0…200

[*1] This parameter is visible when the function “Enable “Measured Value” Object” is set to “Yes”.

[*2] This parameter is visible when the function “Brightness compensation value” is set to “With adjustment value”.

3.3. Sensor (Movement)

This section describes the parameters for how to use the sensor channel. Since the sensor channel has many features for basic uses in general, it has a simpler parametric structure compared to presence channels. Without the need to configure the presence channels, many basic needs can be set up by simply configuring the sensor channel.

If desired, presence channels can be used for environments with more complex user demands.

3.3.1. Sensor Operation

On this parameter subpage, the basic parameters required for sensor operation are explained in detail. Parameters such as master/slave, motion detection source, brightness level source, light-on time etc. are configured in this section.

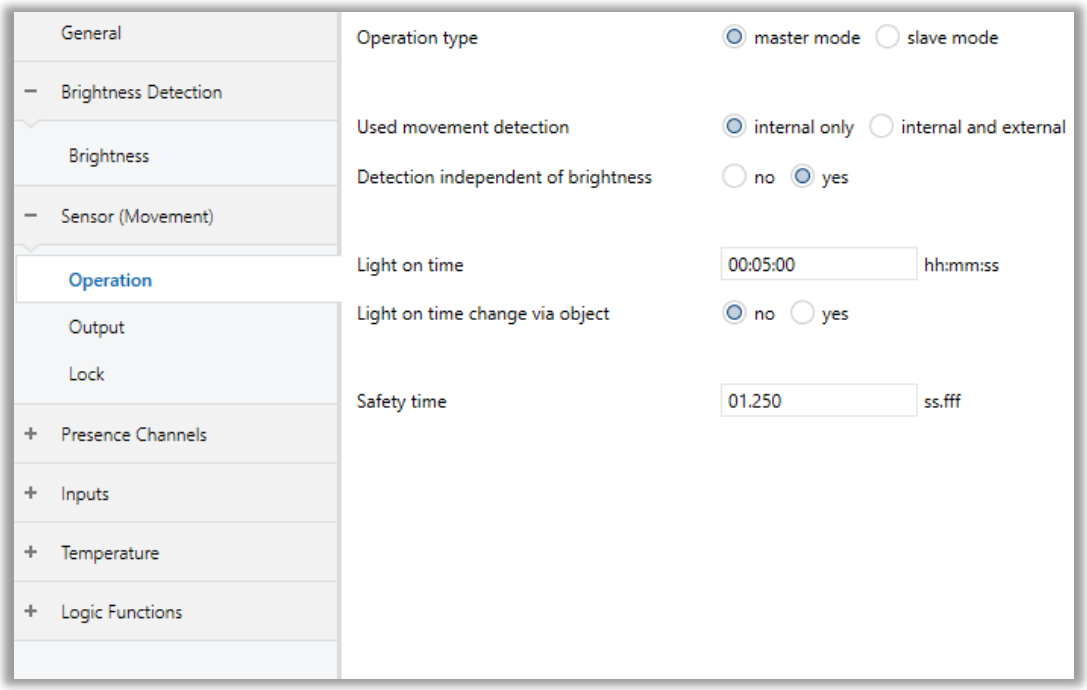


Fig. 15: Sensor - Operation Configuration Page

3.3.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation type	This parameter is used to determine the operating mode of the sensor. Operation mode is set as master or slave mode. In slave mode, the device can only send periodic ON telegrams depending on motion detection. In slave mode, sensor features can be used in a limited way. In master mode, in addition to motion detection, ON or OFF telegrams can be sent depending on the brightness if desired. Also, unlike slave mode, all features can be used in master mode.	**master mode** slave mode
-> Used movement detection[1]	This parameter is used to determine the type of source that will be used for motion detection. There are 2 types of motion detection sources: **Internal only:** Motion detection is made only with the built-in detectors inside the sensor. **Internal and external:** Motion detection is done both by the internal detectors inside the sensor and by an external source (slave sensor, a button used for presence).	**Internal only** Internal and external
-> Detection independent of brightness[1]	This parameter is used to determine whether the motion detection performed by the sensor should be independent of the ambient brightness.	no **yes**
->> Used brightness[2]	This parameter is used to determine the source to be used for measuring the ambient brightness value. There are 2 types of brightness detection sources. **Internal:** It is the internal source of brightness. Ambient brightness is measured with the built-in brightness sensor on the sensor. **External:** It is an external source of brightness. Ambient brightness is determined by sending data from a device that can measure brightness to the relevant KNX object of the sensor.	**internal** external
->> Brightness-value threshold[2]	This parameter is used to specify the lux value at which the sensor is to start to respond. If the detector detects a movement and the measured brightness lies above the threshold, no telegram is sent during movement.	1…**400**…1200
->> Threshold change via object[2]	This parameter is used to change the brightness threshold value via a KNX object.	**no** yes

	The value is sent to this object in lux. Object type is a 2-byte float.	
->> Hysteresis[2]	This parameter is used to determine the hysteresis value. The hysteresis is a percentage value (+/-) that is related to the setpoint in lux. The hysteresis is a tolerance for maintaining the setpoint. The pre-setpoint is sufficient for most applications. Switching threshold = brightness threshold ± hysteresis The hysteresis prevents excessive switching when the current ambient brightness is close to the brightness threshold.	15…**60**…500
-> Light on time[1]	This parameter is the time between the last movement detected and the sending of the telegram "value for switch-off" by the sensor. The presence detector will switch off the lamps in the room if no movement is detected. A light-on time delay can be set to prevent the lamp from being switched off immediately. This time starts as soon as at the end of motion. If movement is detected again within this period, the light-on time delay is reset.	00:00:10… **00:05:00**…18:12:15
-> Light on time change via object[1]	This parameter is used to change the light-on time duration via a KNX object. The time entered is in seconds. **Note:** The values which can be sent are between 10 and 65535 seconds. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.	**no** yes
-> Cyclical repeating time[3]	This parameter becomes visible when the "Operation Type" parameter "Slave Mode" is selected. It sets the period in which a "Presence output" object will be sent to the sensor operating in "Master Mode" only when there is movement. Telegram is not sent to the line when there is no activity. **Note:** This time value must be entered lower than the MASTER mode "Light on time" value in order for MASTER and SLAVE operation to work properly.	00:00:10… **00:00:30**…18:12:15
Safety time	This parameter is used to determine the time it takes the detector to perform the detection process again after the light on time has expired. The safety time is started after the detector has been switched off due to the expiry of light on time. The main purpose of safety time is to prevent immediate reactivation.	00.100…**01.250**… 59.999

[*1] This parameter is visible when the function “Operation type” is set to “Master”.

[*2] This parameter is visible when the function “Detection independent of brightness” is set to “No”.

[*3] This parameter is visible when the function “Operation type” is set to “Slave”.

3.3.2. Sensor Output

On this parameter subpage, the sensor channel output related parameters are described. The relevant parameters that need to be configured to control the ambient brightness through the switching are explained. The switching operation for the movement channel is performed over the output object of the presence detector.

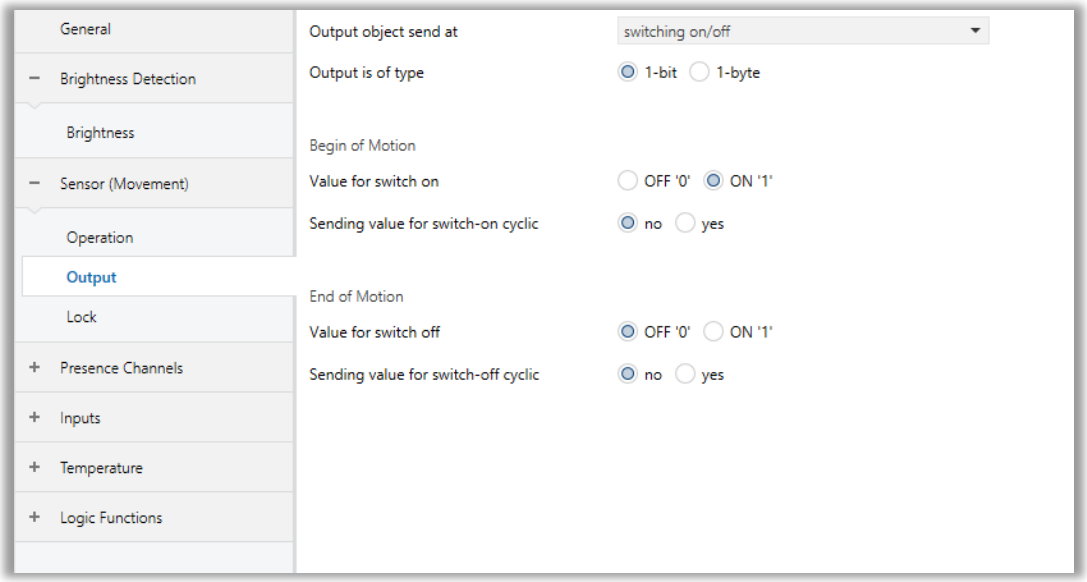


Fig. 16: Sensor - Output Configuration Page

3.3.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Output object send at	This parameter is used to specify in which cases a "Presence output" object will be sent to the field.	**Switching on/off** Switching on Switching off
Output is of type	This parameter is used to determine the output type. Output type can be set as 1-bit or 1-byte.	1-bit 1-byte
Begin of Motion		
Value for switch-on	This parameter is used to determine the switch on the value that will be sent to the KNX bus line when the presence detection is over. 2 types of data are sent according to the output type. **1-bit:** OFF or ON value is sent. **1-byte:** The value between 0% and 100% is sent.	**1 bit:** OFF '0' **ON '1'** 1 byte: **100% (255)**, 99% (252), 98% (250)…. 3% (8), 2% (5), 1% (3), 0% (OFF)
Sending value for switch-on cyclic	This parameter is used to determine whether the switch on value will be sent to the KNX bus periodically. If the parameter value is selected as "yes", the switch on value will be sent periodically.	no yes
End of Motion		
Used three-stage off (*)	This parameter is used by the sensor to switch off with 3 different level transitions after the end of motion. The parameters related to this feature are explained in detail in the table below.	no yes
Value for switch off	This parameter is used to determine the switch off the value that will be sent to the KNX bus line when the presence detection is over. 2 types of data are sent according to the output type. **1-bit:** OFF or ON value is sent. **1-byte:** The value between 0% and 100% is sent.	**1 bit:** OFF '0' **ON '1'** 1 byte: **100% (255)**, 99% (252),….3% (8), 2% (5), 1% (3), 0% (OFF)
Sending value for switch-off cyclic	This parameter is used to determine whether the switch off value will be sent to the KNX bus periodically. If the parameter value is selected as "yes", the switch-off value will be sent periodically.	no yes
-> Cyclical repeating time	This parameter is used to send the switch on and switch off values periodically. The time information to be entered is in hours, minutes and seconds.	00:00:10…00:00:30… 18:12:15

[*1] This parameter is visible when the function "Sending value for switch-on cyclic" or "Sending value for switch-off cyclic" is set to "Master".

(*) Three Stage Off

State: It indicates 3 different transition states as A, B and C. When the presence information is no longer detected by the sensor, after the light-on time has elapsed, the sensor will first switch to state A. When the stay at A time has elapsed, it switches to state B. When the stay B time has elapsed, it switches to state C. After the time entered in state C has expired, the sensor sends the switch off value.

Stay Time: This parameter is used to set the transition times of states A, B and C. Time information is configured by entering hours, minutes and seconds.

Value: This parameter is used to adjust the ambient light level in states A, B and C. It is configured by selecting one of the percentage values in the parameter list.

State	Stay Time	Value
Stage A	00:00:05…00:00:30…18:12:15	100% (255), 99% (252), 98% (250)…3% (8), 2% (5), 1% (3), 0% (OFF)
Stage B	00:00:05…00:00:30…18:12:15	100% (255), 99% (252), 98% (250)…3% (8), 2% (5), 1% (3), 0% (OFF)
Stage C	00:00:05…00:00:30…18:12:15	100% (255), 99% (252), 98% (250)…3% (8), 2% (5), 1% (3), 0% (OFF)

Table 10: Three Stage Of Table

3.3.3. Sensor Lock

On the sensor lock parameter page, there are features for the sensor to be locked in various situations. In various cases, users can lock the sensor depending on their wishes and keep the brightness of the lighting devices connected to the sensor output at a certain level.

Special Note

Lock functionality is available separately for the Sensor channel and all Presence channels. For this reason, the lock functionality mentioned in this section is only related to the "Sensor" parameter page.

Special Note

The parametric structure of the lock function is the same as the lock functionality in Presence channels. For this reason, lock parameters are not explained again in the relevant sections.

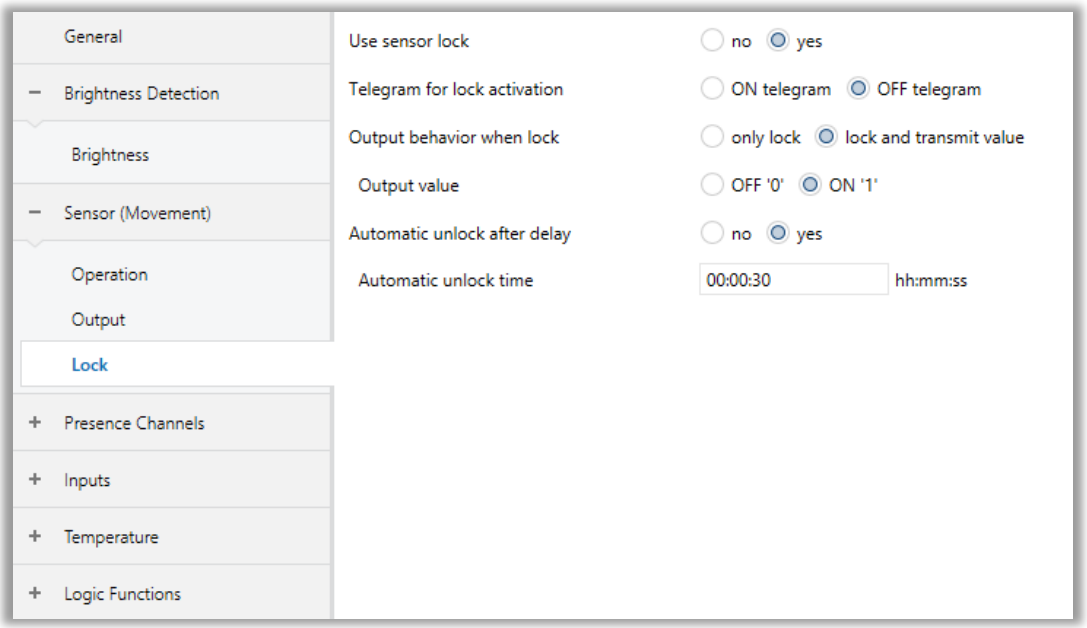


Fig. 17: Sensor - Lock Configuration Page

3.3.3.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Use sensor lock	This parameter is used to activate or deactivate the sensor lock feature. When this parameter is selected as "yes", parameters related to the locking feature will be visible on the page.	**no** yes
Telegram for lock activation[1]	This parameter specifies the telegram value that should be used to lock the sensor. For example, if this parameter is selected as "ON telegram", the sensor will be locked when an ON telegram is sent from the KNX bus line to the relevant Lock object. In this state, when the OFF telegram is sent, the sensor lock will be removed. The opposite of this configuration is also valid.	**ON telegram** OFF telegram
Output behaviour when lock[1]	This parameter is used to determine how the output behaviour will be when the sensor is locked.	**only lock** lock and transmit value
-> Output value[1,2]	This parameter is used to determine the value to be sent to the KNX bus through the output object when the sensor is locked.	**100% (255)**, 99% (252), 98% (250)…. 3% (8), 2% (5), 1% (3), 0% (OFF)
Automatic unlock after delay[1]	This parameter is used to activate or deactivate the feature of automatically unlocking the sensor after a certain period.	**no** yes
-> Automatic unlock time[1,3]	This parameter is used to determine the time required to unlock the sensor from the moment the sensor is locked. The time information to be entered is in hours, minutes and seconds.	00:00:10…**00:00:30**…18:12:15

[*1] This parameter is visible when the function “Use sensor lock” is set to “yes”.

[*2] This parameter is visible when the function “Output behaviour when lock” is set to “lock and transmit value”.

[*3] This parameter is visible when the function “Automatic unlock after delay” is set to “yes”.

3.4. Presence Channel

Multi Presence Detectors have 4 different presence channels for controlling the relevant environment based on the presence and/or brightness. These presence channels can be configured independently of each other.

There are 2 main functions in the presence channels. These:

- Constant Light Switch
- Constant Light Controller

The constant light switch is the same as the 'sensor' channel. Therefore, the explanations for the parameters are not made in this section. For detailed information about the parameters, see the "sensor" page. Detailed parameter descriptions for the use of constant light control are explained in this section, in 5 different subsections.

3.4.1. Constant Light Controller – Operation

On this parameter subpage, the basic parameters required for constant light control are explained in detail. Since the presence channels are the same, only the parameters related to the presence 1 channel are explained. When other channels are used, this section can be referenced for the relevant presence channel.

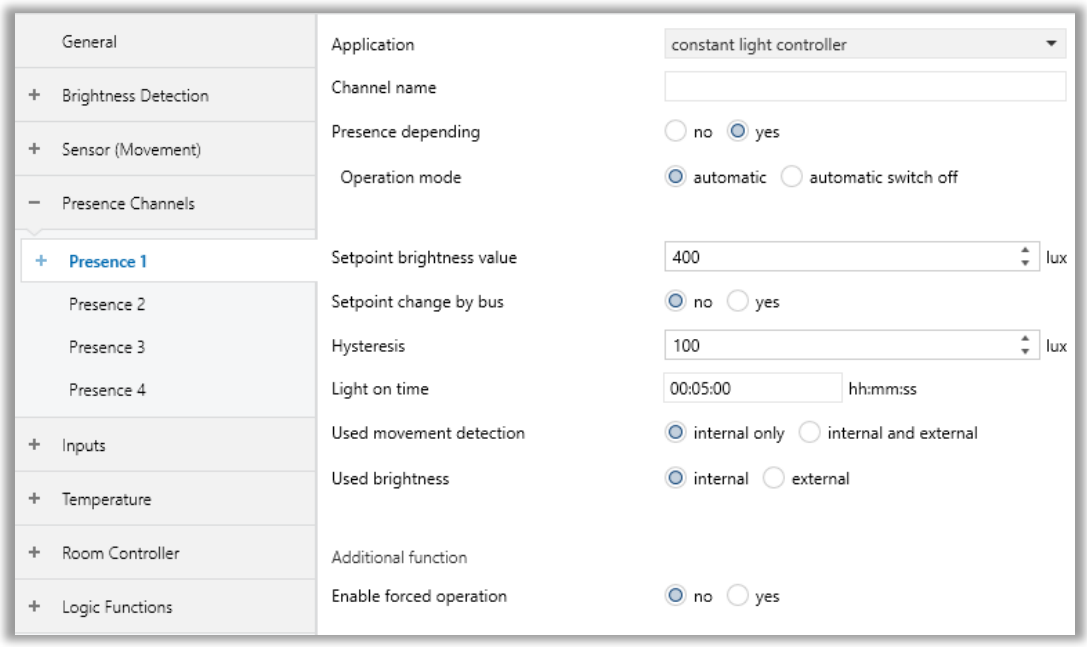


Fig. 18: Constant Light Controller - Operation Configuration Page

3.4.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Application	This parameter is used to determine the presence of channel functionality. **Inactive:** The presence channel is disabled. **Constant light switch:** The presence channel application is selected as the constant light switch. **Constant light control:** The presence channel application is selected as constant light control.	**Inactive** constant light switch constant light control
Channel name	This parameter is used to type a channel name. The name can be consisting of 40 characters.	**40 bytes allowed**
Presence depending	This parameter is used to determine whether constant light control will be performed based on presence.	no **yes**
- > Operation mode	This parameter is used to determine the operating mode of constant light control based on presence. **Automatic:** Automatic switch-on and automatic switch-off. In "Automatic" mode, in case it is too dark, the presence detector switches on automatically when detecting a movement. The switch off is affected when the set light-on time delay is expired. **Automatic Switch Off:** Manual switch-on and automatic switch off. In "Automatic switch-off" mode, the presence detector must be switched on manually using the manual control objects (‘Absolute dimming’, ‘Relative dimming’ etc.). The switch-off is affected automatically under consideration of the light-on time delay.	**automatic** automatic switch off
Setpoint brightness value	This parameter is used to set the desired brightness of the room. Below this threshold - hysteresis the presence detector switched at movement. The presence detector switches off again when the measured brightness is above the brightness-value threshold + hysteresis.	1…**400**…1200
Setpoint change by bus	This parameter is used to enable or disable the brightness-value threshold object.	**no** yes

	This object can be used to change the switching threshold of the presence sensor. The value is sent to this object in lux.	
Hysteresis	This parameter is used to determine the hysteresis value. By hysteresis value, the switching threshold of the Multi Presence Detector is configured. Switching threshold = brightness threshold ± hysteresis The hysteresis prevents excessive switching when the current ambient brightness is close to the brightness threshold.	15…**100**…500
Used brightness	This parameter is used to specify the source to be used for ambient brightness measurement. **Internal:** Built-in brightness measurement sensor is used for brightness measurement **External:** The brightness values are received from an external source via the KNX bus line.	**internal** external
Used movement detection	This parameter is used to specify the source to be used for movement detection. **Internal only:** Built-in movement measurement sensor is used for movement detection. **Internal and external:** Both internal movement detection sensor and external movement detection source (such as another Multi Presence Detector configured as a slave) are used.	**Internal only** Internal and external
Light on time	This parameter is the period between the last movement detected and the sending of the telegram "value for switch-off" by the sensor. The presence detector will switch off or dim the lamps in the room if no movement is detected. A light-on time delay can be set to prevent the lamp from being switched off immediately. This time starts as soon as at the end of motion. If movement is detected again within this period, the light-on time delay is reset.	00:00:10…**00:05:00**…18:12:15
Enable forced operation	This parameter is used to enable or disable the forced operation feature of the Multi Presence Detector.	**no** yes

3.4.2. Constant Light Controller – Output

On this parameter subpage, the relevant parameters that need to be configured to control the ambient brightness through the dimming are explained. The dimming operation for the selected presence channel is performed over the output object of the presence detector. For this reason, separate parameter settings should be made for each presence channel to be used.

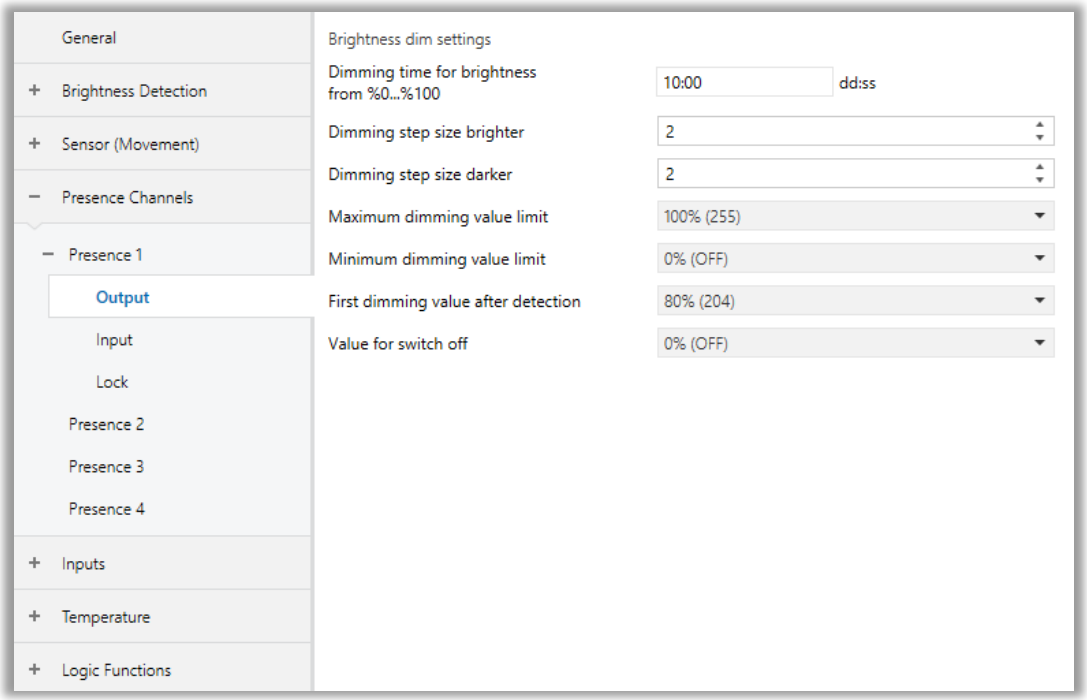


Fig. 19: Constant Light Controller - Output Configuration Page

3.4.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Dimming time for brightness from %0…%100	This parameter is used to determine how long it takes for the output object value to go from 0% to 100% during manual dimming. For example, let the lighting in the environment be completely off (that is, the output is at 0%) and the value entered in this parameter is 20 seconds. In this state, when the presence detector detects a movement, the brightness of the lighting unit controlled in the environment will be 100% after 20 seconds.	02:30..**10:00**..20:00
Dimming step size brighter	This parameter is used to specify the step size to be used for dimming upwards (dimming brighter).	1…**2**…15
Dimming step size darker	This parameter is used to specify the step size to be used for downward dimming (dimming darker).	1…**2**…15
Maximum dimming value limit	This parameter is used to determine the minimum dimming value that can be sent while the presence detector is dimming. A value higher than this value set in the parameter cannot be sent to the bus line.	**100% (255)**, 99% (252), 98% (250)…. 3% (8), 2% (5), 1% (3), 0% (OFF)
Minimum dimming value limit	This parameter is used to determine the minimum dimming value that can be sent while the presence detector is dimming. A value lower than this value set in the parameter cannot be sent to the bus line.	100% (255), 99% (252), 98% (250)…. 3% (8), 2% (5), 1% (3), **0% (OFF)**
First dimming value after detection	This parameter is used to set the initial value at which the presence detector will start dimming. For example, when a motion is detected by the presence detector in an environment where there is constant light control based on motion, the dimming process is started with this value.	100% (255), 99% (252), 98% (250)…. **80% (204)**….3% (8), 2% (5), 1% (3), 0% (OFF)
Value for switch off	This parameter is used to determine the value to be sent from the output object of the relevant present channel to the KNX bus line after the light-on time expires.	100% (255), 99% (252), 98% (250)…. 3% (8), 2% (5), 1% (3), **0% (OFF)**

3.4.3. Constant Light Controller – Input

On this parameter subpage, the parameters for manually controlling the detector when the Multi Presence Detector is set up to operate in constant light control mode is described. Manual control is naturally carried out via external sources (e.g., a device on the KNX bus with pushbutton control).

For this reason, to perform manual control objects of the detector for manual control must be connected to the relevant objects of external devices.

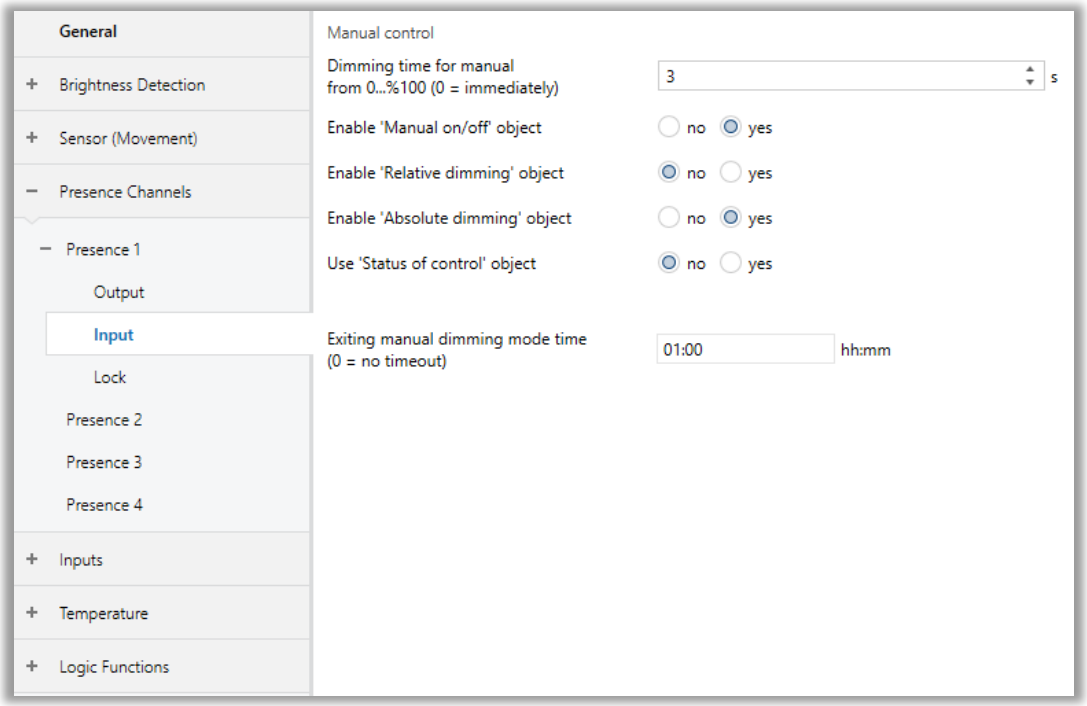


Fig. 20: Constant Light-Controller - Input Configuration Page

3.4.3.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Dimming time for manual from 0…%100 **(0 = immediately)**	This parameter is used to determine how long it takes for the output object value to go from 0% to 100% during manual dimming. For example, let the lighting in the environment be completely off (that is, the output is at 0%) and the value entered in this parameter is 20 seconds. In this state, when an ON telegram is sent to the 'manual on/off' object from the KNX bus line, the brightness of the lighting unit controlled in the environment will be 100% after 20 seconds. In another case, let this parameter be configured as 20 seconds. When the output is at 50%, the time required for it to reach 100% will be 10 seconds.	0…**3**…59
Enable 'Manual on/off' object	This parameter is used to activate or deactivate the 'manual on/off' object, which is used to manually control the presence detector while in constant light control mode. For detailed usage of this object, information checks section 4.	no **yes**
Enable 'Relative dimming' object	This parameter is used to activate or deactivate the 'Relative dimming' object, which is used to manually control the presence detector while in constant light control mode. For detailed usage of this object, information checks section 4.	**no** yes
Enable 'Absolute dimming' object	This parameter is used to activate or deactivate the 'Absolute dimming' object, which is used to manually control the presence detector while in constant light control mode. For detailed usage of this object, information checks section 4.	no **yes**
Use 'status of control' object	This parameter is used to activate or deactivate the status object, which is used to determine whether the KNX presence detector is operating in manual or automatic mode.	**no** yes
Exiting manual dimming mode time **(0 = no timeout)**	This parameter is used to switch back to automatic mode from manual control mode. Depending on the time entered in this parameter, the detector will exit the manual mode and switch to the automatic state. Only when the '0' value is entered into this parameter, the detector will remain in manual mode permanently.	00:00…**01:00**…12:00

3.4.4. Constant Light Controller – Lock

On this parameter subpage, the parameters for locking the corresponding presence channel are described. Since the locking parameters are the same as the parameters in the “sensor” channel, for a more detailed explanation the parameter properties can be viewed in the sensor channel lock section (3.3.3).

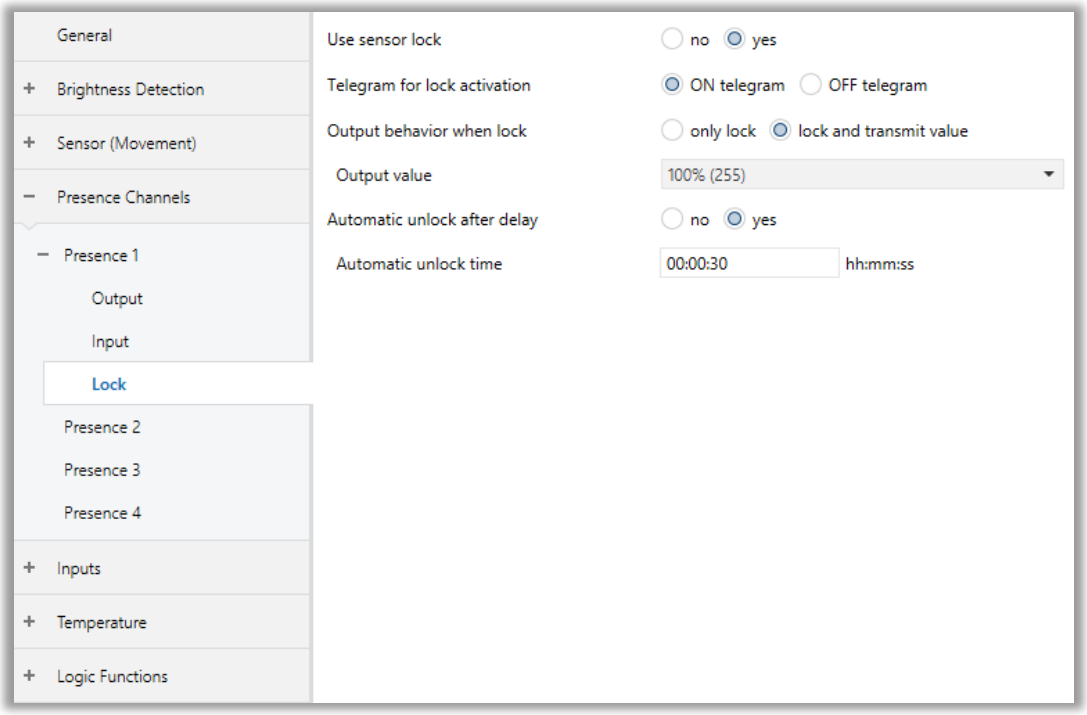


Fig. 21: Constant Light-Controller – Lock Configuration Page

3.4.4.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Use sensor lock	Same as sensor lock	Same as sensor lock
Telegram for lock activation	Same as sensor lock	Same as sensor lock
Output behaviour when lock	Same as sensor lock	Same as sensor lock
-> Output value	Same as sensor lock	Same as sensor lock
Automatic unlock after delay	Same as sensor lock	Same as sensor lock
-> Automatic unlock time	Same as sensor lock	Same as sensor lock

3.4.5. Constant Light Controller – Forced Operation

In this section, parameters for forcing the output of the relevant detector channel to remain at a constant value while the detector operates the constant light control are explained. Since the other presence channels are the same, only the parameters for the presence 1 channel are described. When other channels are used, this section can be referenced for the usage of relevant presence channels.

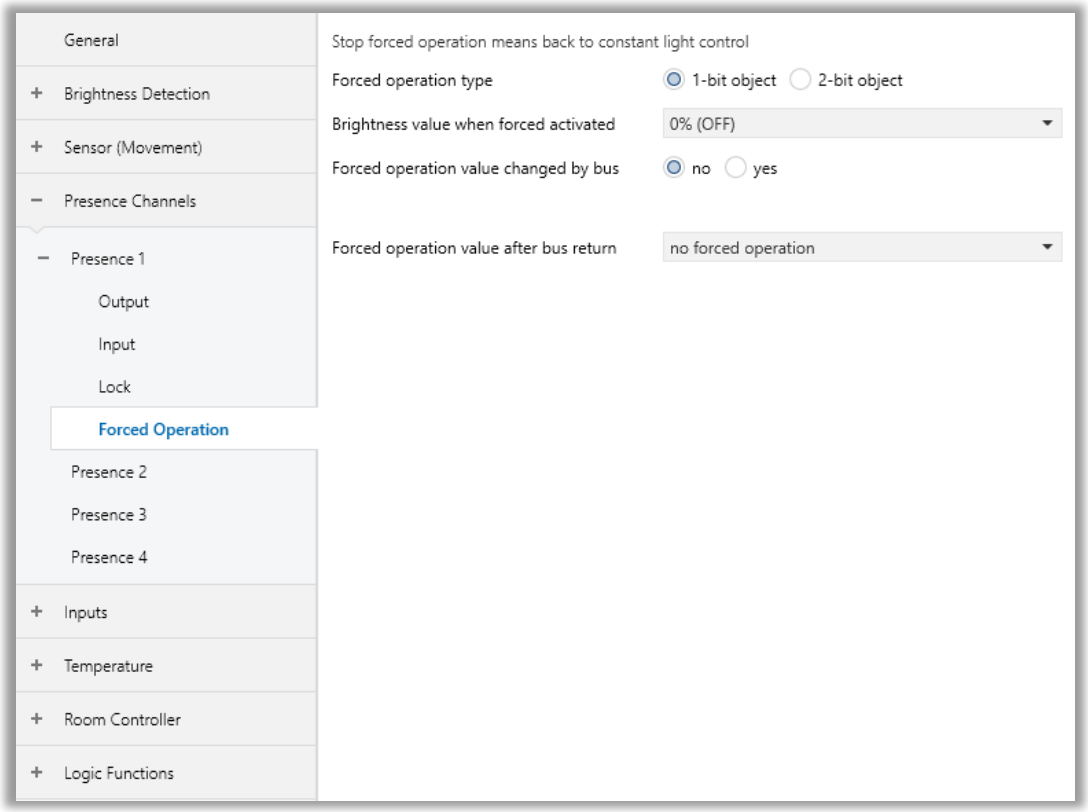


Fig. 22: Constant Light-Controller – Forced Operation Configuration Page

3.4.5.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Forced operation type	This parameter is used to determine the object type to trigger the detector from constant light control to forced operation state.	**1-bit object** 2-bit object
Brightness value when forced activated	This parameter is used to determine the brightness value of the output when the detector disables the constant light control and enters the forced operation state.	100% (255), 99% (252), 98% (250)…. 3% (8), 2% (5), 1% (3), **0% (OFF)**
Forced operation value changed by bus	This parameter is used to change the output brightness value over the KNX bus line when the detector goes into a forced operation state.	**no** yes
Forced operation value after bus return	This parameter is used to determine whether there will be forced operation after the KNX bus line voltage is restored. **No forced operation:** The detector will operate in the default state. **Forced on:** The detector will operate as a force on the state. **Forced off:** The detector will operate in constant light controller mode. **Position before failure:** The detector will operate at the last state(before bus voltage break) after bus voltage return.	**no forced operation** forced on forced off position before failure

3.4.6. Presence Channels Priority List

Presence channels' functions as explained in previous points. Also, all of the regarding functions have their different priority levels. "Constant Light Switch" is "Lock" prioritized first and "General Operation" prioritized second. Then, "Constant Light Controller" has "Lock" prioritized first, "Forced Operation" prioritized second, "Manuel Operation" prioritized third and "General Operation" prioritized fourth and last. Priority levels are also explained in the block diagram in Figure 19.

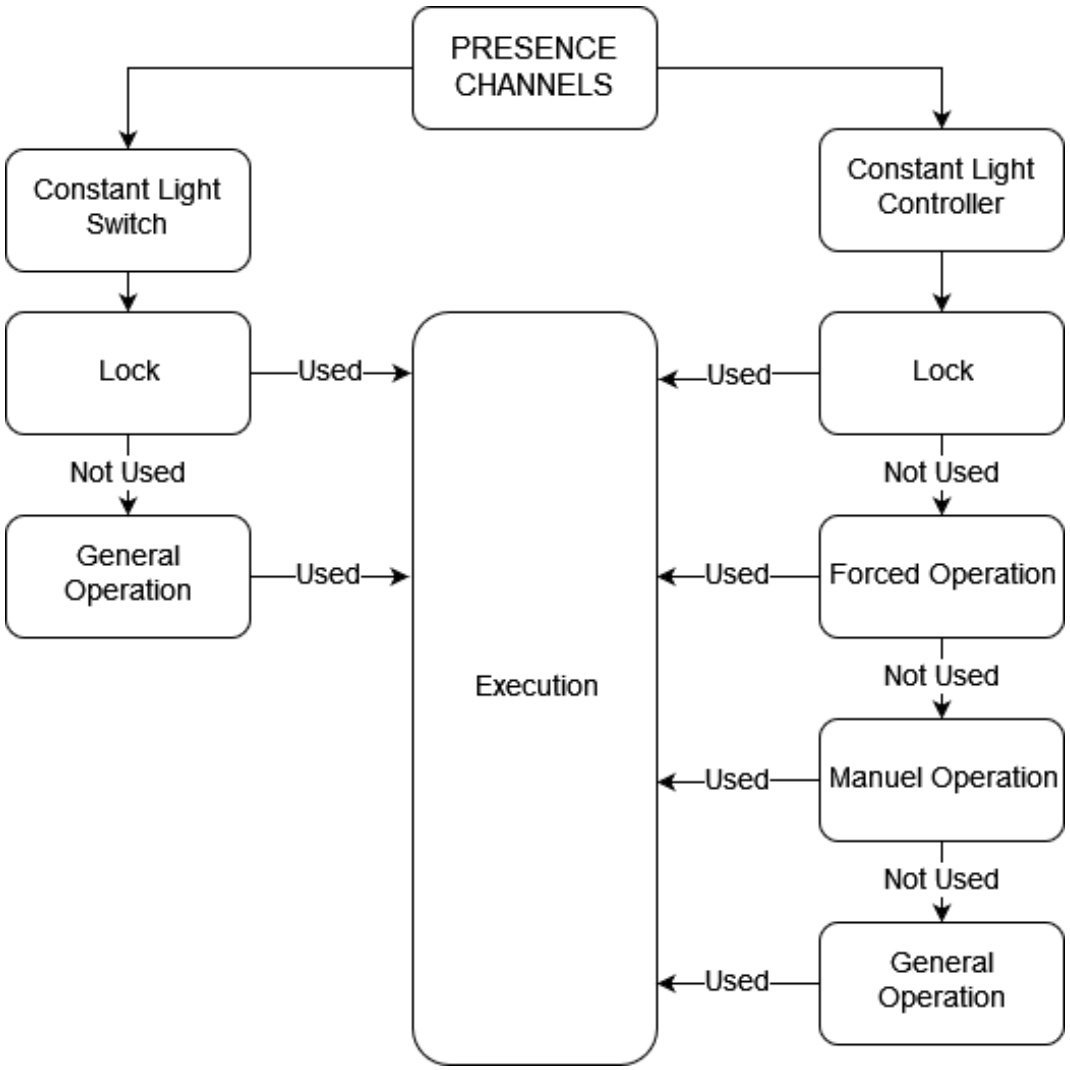


Fig. 23: Presence Channels' Priority Levels

3.5. Inputs

Interra Multi Presence Detector has 2 digital inputs and 1 analog input. By connecting buttons to digital inputs, you can choose the lighting, curtains/blinds, RGB LEDs, dim devices etc. you want to control. You can control the devices by making the necessary configurations via the KNX sensor. However, by connecting an NTC temperature sensor to the analog input, you can obtain temperature information from further distances from the Multi Presence Detector.

3.5.1. Input – Switch Sensor

In this section, it is explained how to control the related automation unit via the Multi Presence Detector by switching via buttons connected to digital inputs. Detailed information on the relevant parameter configurations is described in the table below.

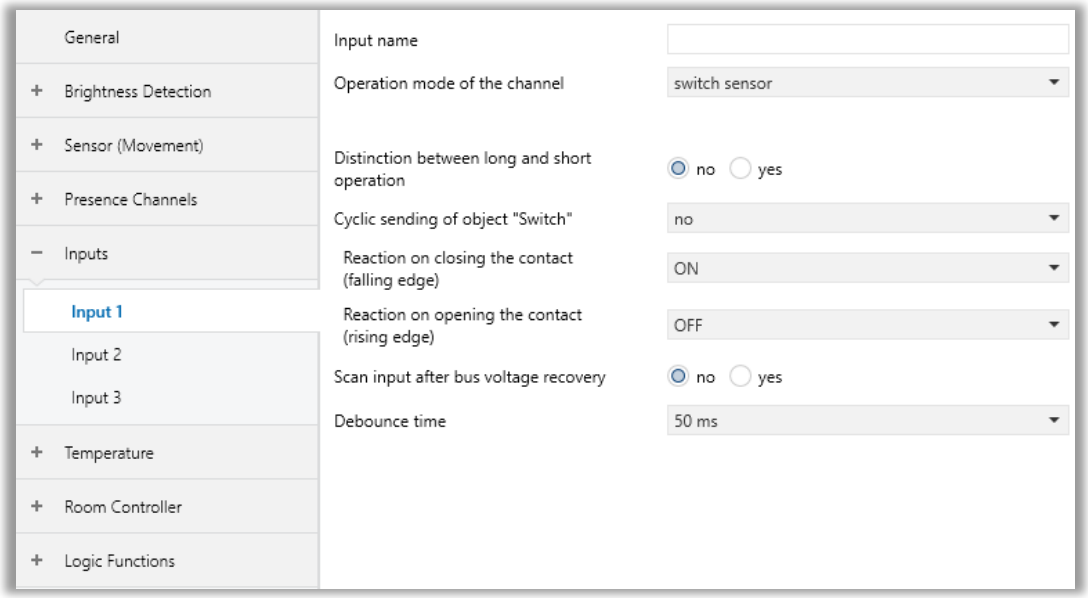


Fig. 24: Input – Input Switch Sensor Configuration Page

3.5.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function **Switch sensor** Switch/dimming sensor Shutter sensor Value/forced operation Control scene RGB colour control Mode selection Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can consist of up to 40 characters.	**40 bytes allowed**
Distinction between short and long operation	This parameter is used to set if the input differentiates between short and long operations. With the option “yes”, after opening/closing of the contact, it must, first of all, be ascertained if a short or long operation has occurred here. Only thereafter will a possible reaction be triggered.	**No** Yes
-> Cyclic sending of object “Switch”	This parameter is visible if there is no distinction between short and long operations. The communication object “Switch” can be sent cyclically. If the parameter “always” is set, the object sends cyclically on the bus, regardless of its value. Should the parameter value “if telegram switch = ON” or “if telegram switch = OFF” be set, the corresponding object value is sent cyclically.	**No** If “Switch” = OFF If “Switch” = ON always
-> Reaction on closing the contact (rising edge)	This parameter is visible if there is no distinction between short and long operations. For each edge, you can set if the object value is to be switched ON, OFF or TOGGLE, or if no reaction should occur. If cyclical sending has been parameterized, it is possible by setting the parameter value “terminate cyclic sending” with an operation of the input, to stop cyclic sending without a new object value being sent.	No reaction **ON** OFF TOGGLE
-> Reaction on opening the contact (Falling edge)	This parameter is visible if there is no distinction between short and long operations. For each edge, you can set if the object value is to be switched ON, OFF or TOGGLE, or if no reaction should occur.	No reaction ON **OFF** TOGGLE

	If cyclical sending has been parameterized, it is possible by setting the parameter value “terminate cyclic sending” with an operation of the input, to stop cyclic sending without a new object value being sent.	
-> Telegram is repeated every	This parameter is visible if the cyclical transmission is active. The send cycle time describes the time used between two cyclically transmitted telegrams	00:00:05…**00:00:30**…18:12:15
Scan input after bus voltage recovery	This parameter is used to determine the scanning of the inputs when the bus voltage has been recovered.	**No** Yes
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Reaction on short operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the short press operation sending the value of the input x.	No reaction **ON** OFF TOGGLE
Reaction on long operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the long-press operation sending the value of the input x.	No reaction **ON** OFF TOGGLE
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200…**00:00.500**…01:05.000
Number of object for short/long operation	This parameter is used to determine the object count to use for short and long operations. **1 object:** short and long operations will proceed with the same object. **2 object:** Short and long operations will proceed with 2 different objects.	**1 object** 2 object
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

3.5.2. Input – Switch / Dimming Sensor

In this section, it is explained how to control the unit of a lighting unit through the Multi Presence Detector, both by switching and dimming, via the buttons connected to the digital inputs. Detailed information on the relevant parameter configurations is described in the table below. Make sure that the lighting unit to be controlled has a dimming feature.

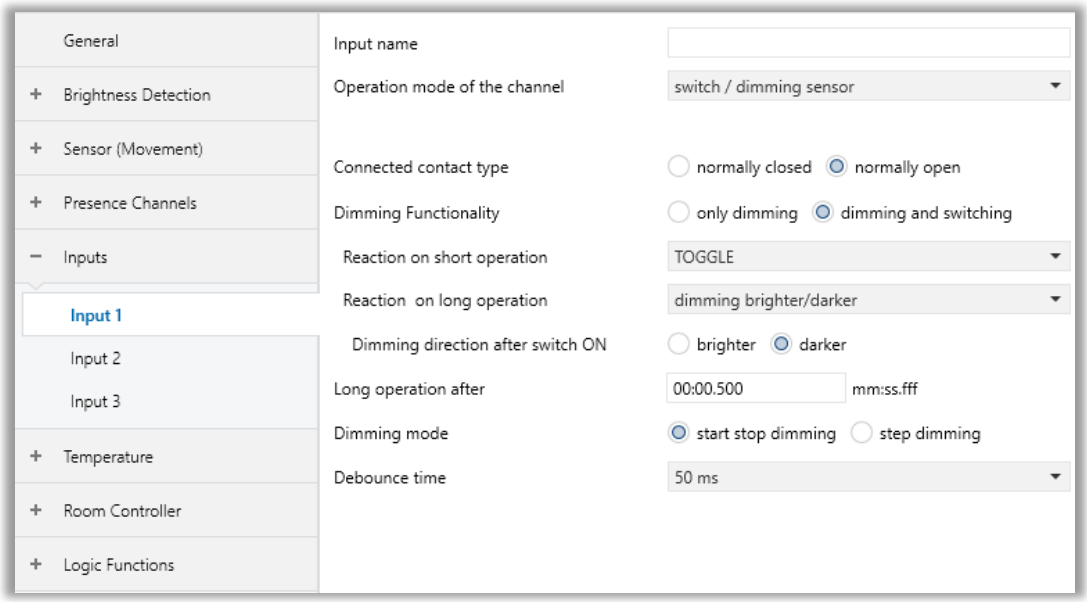


Fig. 25: Input – Input Switch / Dimming Sensor Configuration Page

3.5.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor **Switch/dimming sensor** Shutter sensor Value/forced operation Control scene RGB colour control Mode selection Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 bytes allowed**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Dimming functionality	This parameter is used to define if the lighting can only be dimmed “Only dimming” or if additional switching is also permitted “Dimming and switching”. In this case, a long button press dims and a short button push switch.	**Only dimming** Dimming and switching
Reaction on operation	This parameter is visible if “Only dimming” dimming functionality is set. A distinction is not made between short and long operations here.	Dimming brighter Dimming darker **Dimming brighter/darker**
Reaction on short operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the short press operation sending the value of the input x.	No reaction **ON** OFF TOGGLE
Reaction on long operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the long-press operation sending the value of the input x.	Dimming brighter Dimming darker **Dimming brighter/darker**

Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200…**00:00.500** …01:05.000
Dimming mode	This parameter is used to determine the dimming mode. Normal “Start-stop-dimming” starts the dimming process with a telegram BRIGHTER or DARKER and ends the dimming process with a STOP telegram. Cyclic sending of the telegram is not necessary in this case. With “Dimming steps”, the dimming telegram is sent cyclically during a long operation. The STOP telegram ends the dimming process at the end of the operation.	**Start-stop dimming** Dimming step
Brightness change on every sent telegram	This parameter is only visible with “Dimming steps”. This parameter is set to change the brightness (in per cent), which is cyclically sent with every dimming telegram.	**%100** %50 %25 %12.5 %6.25 %3.13 %1.56
Sending cycle time: Telegram is repeated every	This parameter is used to determine the sending cycle time. The dimming telegram is sent cyclically during a long operation if “Dimming steps” is set. The cycle time for sending corresponds with the time interval between two telegrams during cyclical sending.	0.3s, 0.4s, **0.5s**, 0.6s, 0.8s, 1s, 1.2s, 1.5s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 10s,
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

3.5.3. Input – Shutter Sensor

In this section, it is explained how to control a shutter/blind unit via the buttons connected to the digital inputs via the Multi Presence Detector. Detailed information on the relevant parameter configurations is described in the table below.

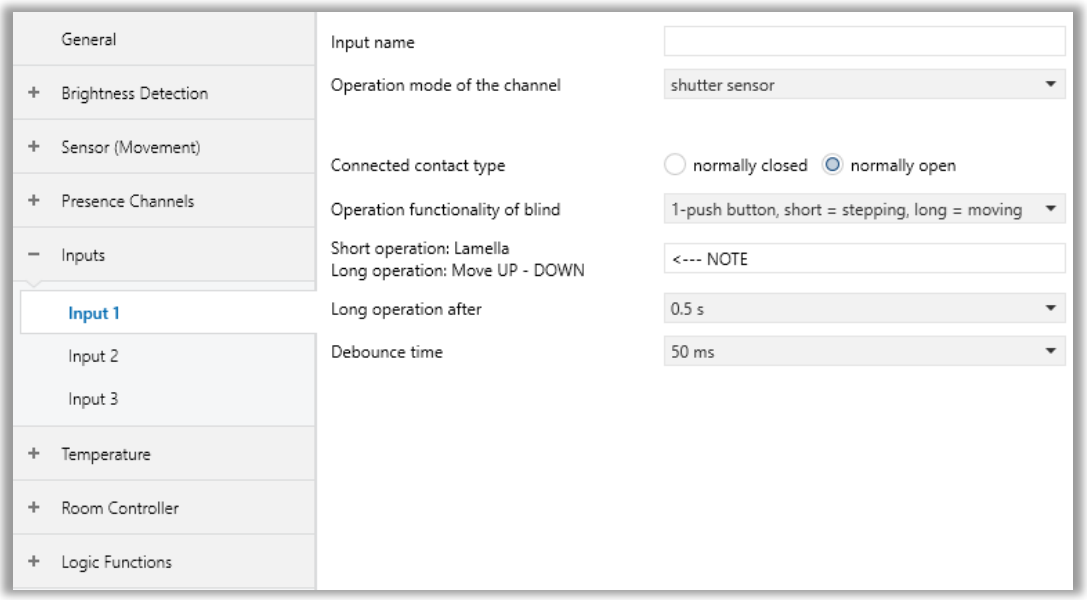


Fig. 26: Input – Input Shutter Sensor Configuration Page

3.5.3.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor **Shutter sensor** Value/forced operation Control scene RGB colour control Mode selection Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 bytes allowed**
Operation Functionality of blind	This parameter is used to define the type of blind operation. An overview of the operating modes is described below.	**1-push-button, short = stepping, long = moving** 1-push-button, short = moving, long = stepping 1-push-button-operation 1-switch button operation 2-push-button, standard 2-switch-operation, moving 2-push-button, moving 2-push-button, stepping
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
1-push-button, short = stepping, long = moving		
Short Operation: Lamella	NOTE	NOTE

Long Operation: Move UP / DOWN		
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	0.3s, 0.4s, **0.5s**, 0.6s, 0.8s, 1s, 1.2s, 1.5s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 10s,
1-push-button, short = moving, long = stepping		
Short Operation: Move UP / DOWN **Long Operation: Lamella**	NOTE	NOTE
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	0.3s, 0.4s, **0.5s**, 0.6s, 0.8s, 1s, 1.2s, 1.5s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 10s,
"STOP/Lamella adj." is repeated every	This parameter is used to determine the time between two telegrams is set. This parameter is visible in operations in which the object "STOP/lamella adjustment" is sent cyclically on the bus during a long operation.	0.3s, **0.4s**, 0.5s, 0.6s, 0.8s, 1s, 1.2s, 1.5s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 10s,
1-push button operation		
On Every operation in success: **UP – STOP–DOWN-STOP**	NOTE	NOTE
1-switch button operation		
On operation: UP–DOWN **End of operation: STOP**	NOTE	NOTE
2-push button, standard		
Short Operation: STOP – Lamella UP / DOWN **Long Operation: Move UP / DOWN**	NOTE	NOTE
Reaction on short operation	This parameter is visible if there is a distinction between short and long operations. It is used to	**Stop / lamella up** Stop / lamella down

	determine the short press operation sending the value of the input x.	
Reaction on long operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the long-press operation sending the value of the input x.	**Move up** Move down
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	0.3s, 0.4s, **0.5s**, 0.6s, 0.8s, 1s, 1.2s, 1.5s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 10s,
2-switch operation, moving		
On Operation: Moving **End of Operation: STOP**	NOTE	NOTE
Reaction on operation	This parameter is used to determine the reaction when an operation occurs. A distinction is not made between short and long operations here.	**Move up** Move down
2-push button operation, moving		
On Operation: Moving **End of Operation: STOP**	NOTE	NOTE
Reaction on operation	This parameter is used to determine the reaction when an operation occurs. A distinction is not made between short and long operations here.	**Move up** Move down
2-push-button, stepping		
On Operation: Stepping	NOTE	NOTE
Reaction on operation	This parameter is used to determine the reaction when an operation occurs. A distinction is not made between short and long operations here.	**Stop / Lamella up** Stop / Lamella down
“STOP/Lamella adj.” is repeated every	This parameter is used to determine the time between two telegrams is set. This parameter is visible in operations in which the object “STOP/lamella adjustment” is sent cyclically on the bus during a long operation.	0.3s, **0.4s**, 0.5s, 0.6s, 0.8s, 1s, 1.2s, 1.5s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 10s,
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms 100 ms, 150 ms

3.5.3.2. The Functionality of Each Function

1 push button: Short Press = stepping, Long Press = moving	
Short Operation	Stop/ Lamella Adjustment
Long Operation	Toggle between “Move Up” and “Move Down”
1 push button: Short Press = moving, Long Press = stepping	
Short Operation	Toggle between “Move Up” and “Move Down”
Long Operation	Stop/Lamella Adjustment (Sent Cyclically as the button is kept pressed)
1 push button: Press: moving, Long Press Disabled	
On Operation	Following signals are sent in order on each press. → Move UP → Stop/Lamella Adj. Up → Move Down → Stop/Lamella Adj. Down →
1 switch Operation: Moving, Long Press Disabled	
Press Operation	Toggle between “Move Up” and “Move Down”
Release Operation	Stop/Lamella Adjustment
2 Push Button Operation: Standard	
Short Operation	“Stop/Lamella Adj. Down” or Stop/Lamella Adj. Up (Whichever is chosen as the parameter)
Long Operation	“Move Up” or “Move Down” (Whichever is chosen as the parameter)
2 Switch Operation: Moving, Long Press Disabled	
Press Operation	“Move Up” or “Move Down” (Whichever is chosen as the parameter)
Release Operation	“Stop/Lamella Adj. Down” or “Stop/Lamella Adj. Up” (Whichever is chosen)
2 Push Button Operation: Moving, Long Press Disabled	
On Operation	Whichever sequence is selected as the parameter; “ → Move Up → Stop/Lamella Adj. Up → “ or “ → Move Down → Stop/Lamella Adj. Down → “
2 Push Button Operation: Stepping, Long Press Disabled	
On Operation	Whichever signal is selected as the parameter, is sent cyclically as the button is kept pressed; “Stop/Lamella Adj. Up” or “Stop/Lamella Adj. Down”

3.5.4. Input – Value / Forced Operation

In this section, it is explained how to control an automation unit via Multi Presence Detector via a value/forced via buttons connected to digital inputs. Detailed information on the relevant parameter configurations is described in the table below.

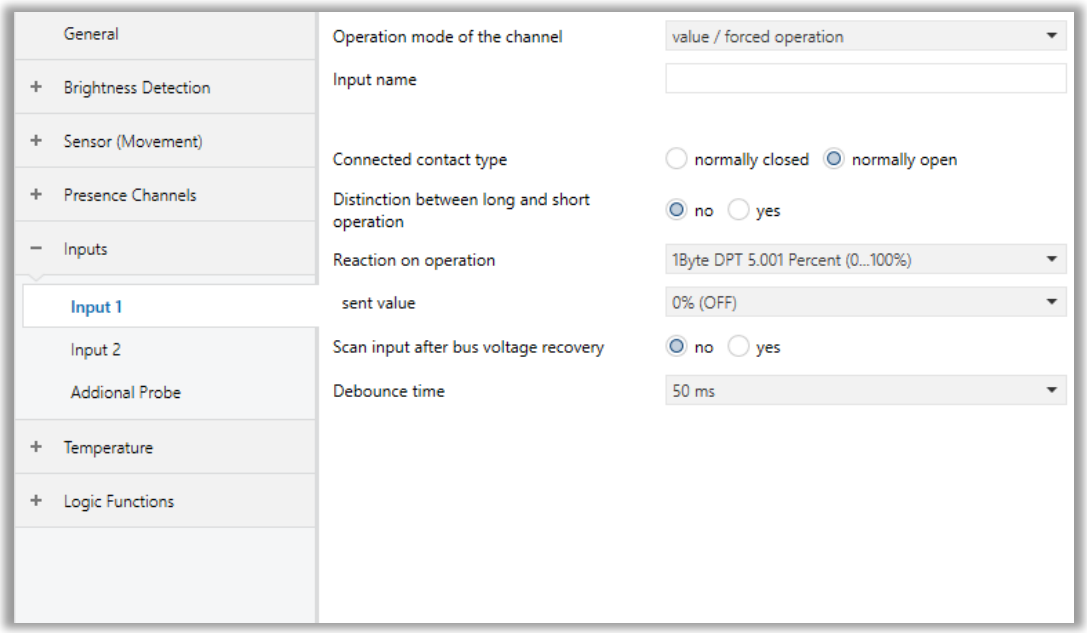


Fig. 27: Input – Input Value / Forced Operation Configuration Page

3.5.4.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor Shutter sensor **Value/forced operation** Control scene RGB colour control Mode selection Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 bytes allowed**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Distinction between short and long operation	This parameter is used to set if the input differentiates between short and long operations. With the option “yes”, after opening/closing of the contact, it must, first of all, be ascertained if a short or long operation has occurred here. Only thereafter will a possible reaction be triggered.	**No** Yes
Reaction on short operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the short press operation sending the value of the input x.	2-bit DPT 2.001 Switch Control **1-byte DPT 5.001 Percent (0…100%)** 1-byte DPT 5.005 Decimal factor (0…255) 1-byte DPT 17.001 Scene Number 2-byte DPT 7.600 Color temperature(Kelvin) 2-byte DPT 9.001 Color temperature (˚C) 2-byte DPT 9.004 Brightness (Lux) 3-byte DPT 232.600 RGB value 3x (0…255)

-> sent value	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depends on DPT selection.
Reaction on long operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the long-press operation sending the value of the input x.	2-bit DPT 2.001 Switch Control **1-byte DPT 5.001 Percent (0…100%)** 1-byte DPT 5.005 Decimal factor (0…255) 1-byte DPT 17.001 Scene Number 2-byte DPT 7.600 Color temperature(Kelvin) 2-byte DPT 9.001 Color temperature (˚C) 2-byte DPT 9.004 Brightness (Lux) 3-byte DPT 232.600 RGB value 3x (0…255)
-> sent value	This parameter is used to determine the sending value to the bus when a long operation occurs.	Values depends on DPT selection.
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200…**00:00.400**…01:05.000
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

3.5.5. Input – Control Scene

In this section, it is explained how to control the related automation unit via the Multi Presence Detector by triggering a scenario via buttons connected to digital inputs. Detailed information on the relevant parameter configurations is described in the table below.

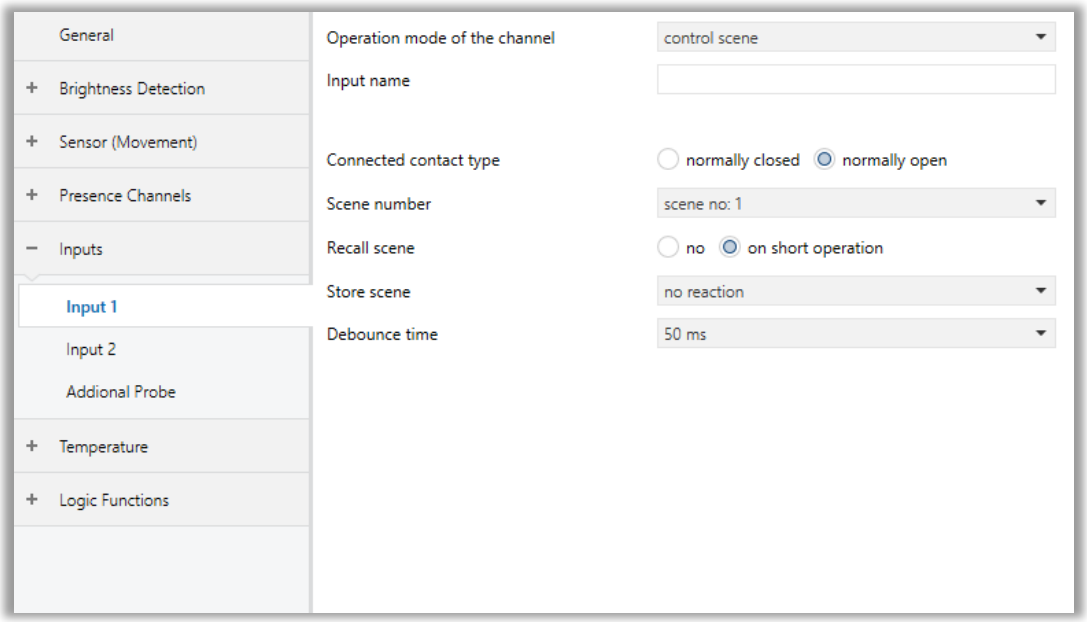


Fig. 28: Input – Control Scene Configuration Page

3.5.5.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor Shutter sensor Value/forced operation **Control scene** RGB colour control Mode selection Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 bytes allowed**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Scene Number	This parameter is used to configure the scene number to send to the KNX when a short press operation occurs.	**Scene no.1**…Scene no.64
Recall scene	This parameter is used to determine the recalling of the scene. If this parameter is selected as “on short operation” the configured scene number will be called.	No **On short operation**
Store Scene	This parameter is used to determine to store or not to store the related scene. **On long operation:** The scene will be stored after a long operation. **With “Store scene” obj. value = 1:** The scene will be stored on operation if the Store scene object value is 1. **On long operation (“Store scene” obj. value=1):** The scene will be stored on long operation if the Store scene object is 1.	**No reaction** On long operation With “Store scene” obj value = 1 On long operation (“Store scene” obj value = 1)
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200…**00:00.500**…01:05.000
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

3.5.6. Input – RGB Colour Control

In this section, it is explained how to control an RGB LED device through the buttons connected to the digital inputs via the Multi Presence Detector. Detailed information on the relevant parameter configurations is described in the table below.

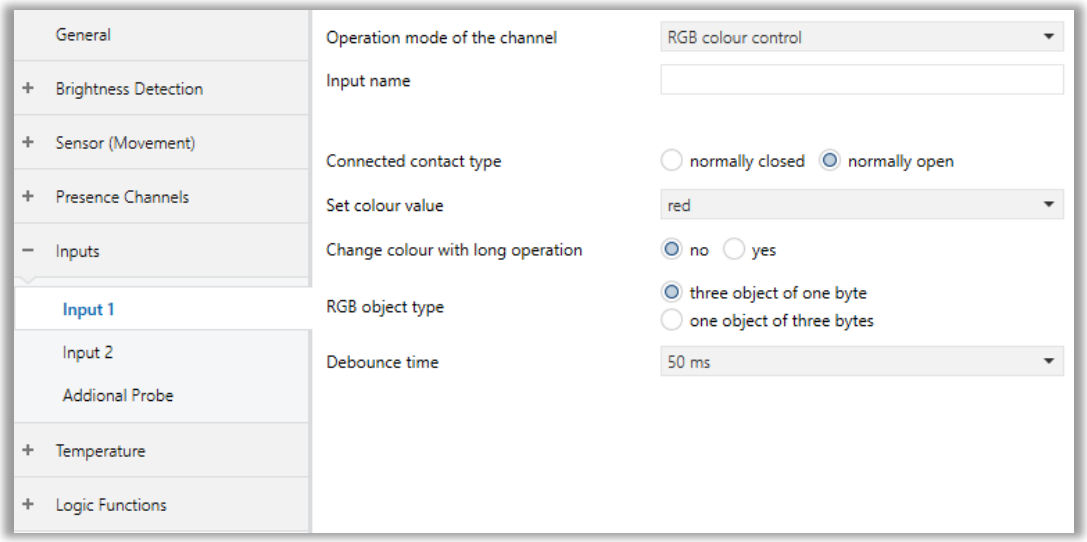


Fig. 29: Input – RGB Colour Control Configuration Page

3.5.6.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor Shutter sensor Value/forced operation Control scene **RGB colour control** Mode selection Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 bytes allowed**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Set colour value	This parameter is used to set RGB colours according to the configured values.	**Red,** Orange, Yellow, Green-yellow, Green, Green-cyan, Cyan, Blue-cyan, Blue, Blue-magenta Red-magenta, White
Change colour with long operation	This parameter is used to enable or disable the colour changing with long press operation.	**No** Yes
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200…**00:00.500** …01:05.000
RGB object type	This parameter is used to determine the RGB colour object type.	**Three object of one byte** one object of three bytes
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms,150 ms

3.5.7. Input – Mode Selection

In this section, it is explained how to control the operating modes of an HVAC unit via the buttons connected to the digital inputs via the Multi Presence Detector. Detailed information on the relevant parameter configurations is described in the table below.

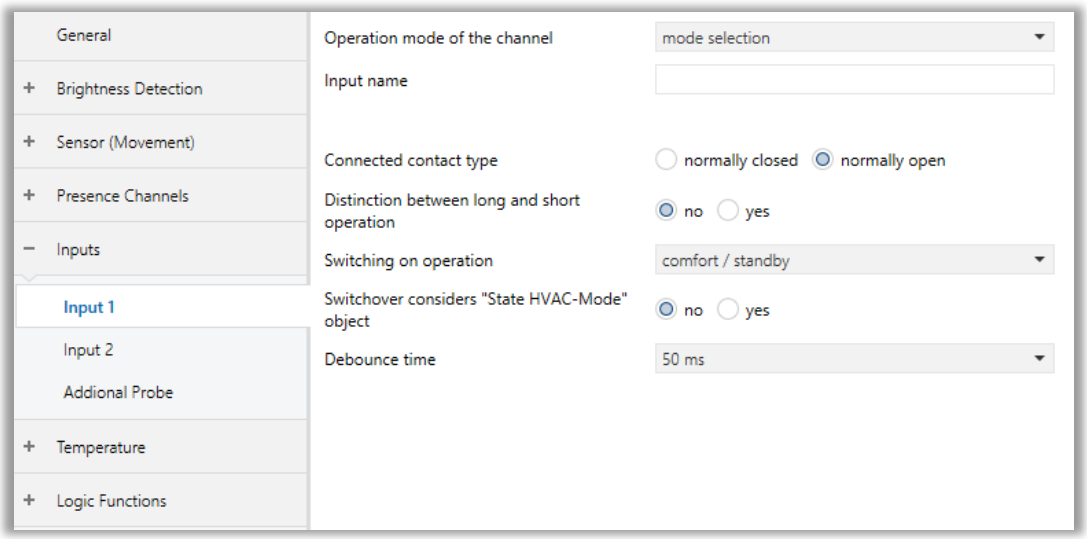


Fig. 30: Input – Mode Selection Configuration Page

3.5.7.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Operation Mode of the channel	This parameter is used to determine the input x operation mode. If no function is selected, the input x will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor Shutter sensor Value/forced operation Control scene RGB colour control **Mode selection** Command sequence Counter RGBW control
Input Name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 bytes allowed**
Distinction between short and long operation	This parameter is used to set if the input differentiates between short and long operations. With the option “yes”, after opening/closing of the contact, it must, first of all, be ascertained if a short or long operation has occurred here. Only thereafter will a possible reaction be triggered.	**No** Yes
->Switching on short operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the short press operation sending the value of the input x.	**Comfort / standby** Comfort / economy Comfort / standby / economy Comfort / standby / economy / frost
-> Reaction on long operation	This parameter is visible if there is a distinction between short and long operations. It is used to determine the long-press operation sending the value of the input x.	**Comfort / standby** Comfort / economy Comfort / standby / economy Comfort / standby / economy / frost
Long operation after	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200…**00:00.500** …01:05.000
Switchover considers “State HVAC-Mode” object	This parameter is used to enable the HVAC-Mode state object to change the current HVAC mode via KNX.	**No** Yes
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

3.5.8. Input – Command Sequence

In this section, it is explained how the command sequence function works. Up to 4 commands are attainable with either 1-bit, 1-byte (percentage) or 1-byte (0...255) objects. Each press event toggles through the used commands (Object A, B, C, D) via the assigned buttons. Detailed information on the relevant parameter configurations is described in the table below.

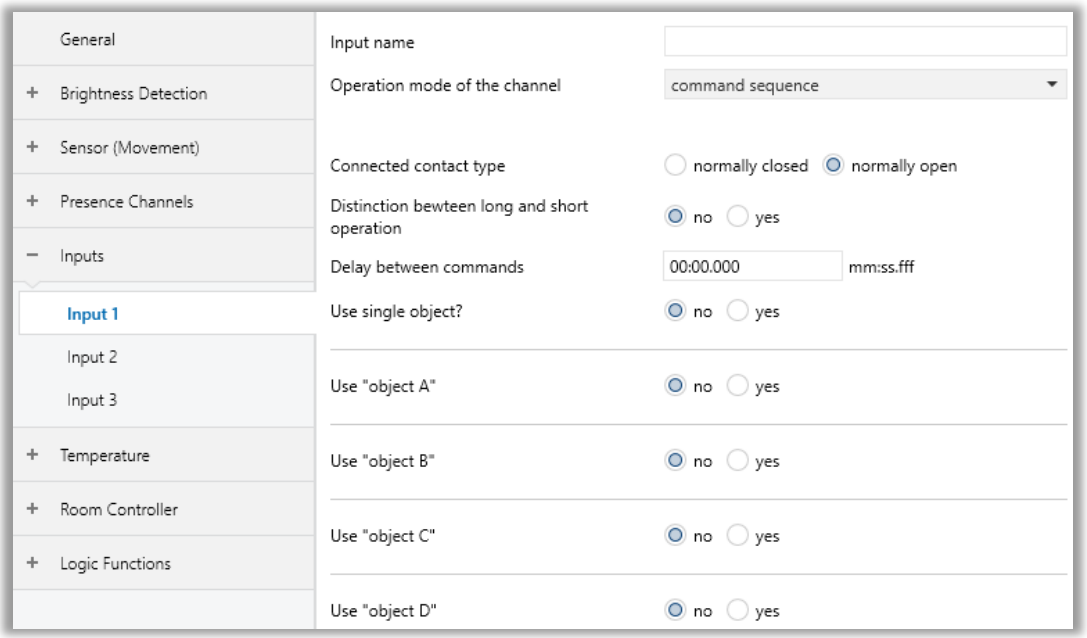


Fig. 31: Input – Command Sequence Configuration Page

3.5.8.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Input name	This parameter is used to type an input name. The name can be consisting of 40 characters	**40 Bytes allowed**
Operation Mode of the channel	This parameter is used to determine the Input X operation mode. If no function is selected, Input X will not be used. For other choices, all functionalities are configured separately.	No function Switch Switch/dimming Shutter/blinds Value/forced operation Scene control Mode selection **Command sequence** Counter RGB colour control RGBW control Thermostat Extension
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Distinction between long and short press	This parameter is used to set if the input differentiates between short and long operations. With the option “yes”, after opening/closing of the contract, it must, first of all, be ascertained if a short or long operation has occurred here. Only thereafter will a possible reaction be triggered.	**No** Yes
-> Long press after[1]	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	00:00.200… **00:00.500** … 01:05.535
Delay between commands	This parameter is used to determine the delay between sending the value of the sequence	**00:00.000**…00:20.000
Use single object?	This parameter decides whether each sequence is sent to a single object or multiple objects.	**No** Yes
-> Use “object X”[2]	This parameter is used to enable each command object when they are set to yes.	**No** Yes
-> Data type[2]	This parameter is used to determine the sending data type to the bus when an operation occurs.	1 bit 1 byte (0...255)

		1 byte (0...100%) HVAC mode
-> Value 'X'[2]	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depend on DPT selection.
-> Value 'X' for long press[3]	This parameter is used to determine the sending value to the bus when a long operation occurs.	Values depend on DPT selection.
-> Value amount[4]	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the button, e.g., due to bouncing of the contact.	2 **3** 4
-> Data type[4]	This parameter is used to determine the sending value to the bus when a short operation occurs.	1 bit 1 byte (0...255) 1 byte (0...100%) HVAC mode
-> Value 'X'[4]	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depend on DPT selection.
-> Value 'X' for long press[5]	This parameter is used to determine the sending value to the bus when a long operation occurs.	Values depend on DPT selection.
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

[*1] This parameter is visible when the parameter "Distinction between long and short press" is set to "Yes".
[*2] This parameter is visible when the parameter "Use single object?" is set to "No".
[*3] This parameter is visible when the parameter "Distinction between long and short press" is set to "Yes" and the parameter "Use single object?" is set to "No".
[*4] This parameter is visible when the parameter "Use single object?" is set to "Yes".
[*5] This parameter is visible when the parameters "Distinction between long and short press" and "Use single object?" are set to "Yes".

3.5.9. Input – Counter

In this section, it is explained how to count input pulses on the Multi Presence Detector. Detailed information on the relevant parameter configurations is described in the table below.

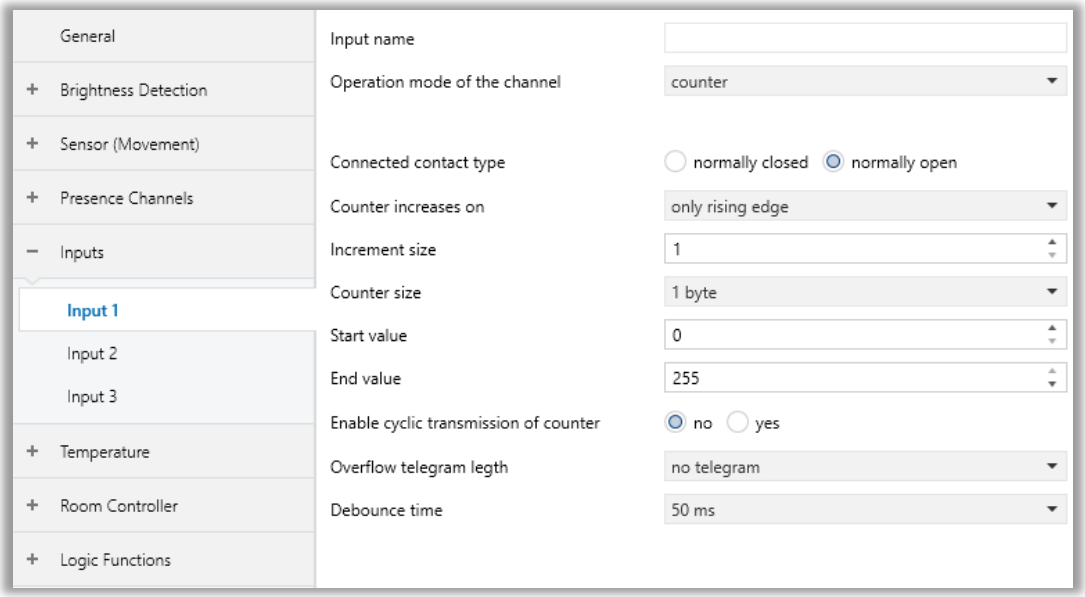


Fig. 32: Counter Function Configuration Page

3.5.9.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Input name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 Bytes allowed**
Operation Mode of the channel	This parameter is used to determine the Input function. If no function is selected, Input X will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor Shutter sensor Value/forced operation Control scene RGB colour control Mode selection Command sequence **Counter** RGBW control
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Counter increase on	This parameter is used to set how the input pulse is to be generated.	**Only when pressed** Only when released Both when pressed and released
Increment size	This parameter is used to assign the increment size when a press event occurs.	**1**..255
Counter size	This parameter is used to determine long operation detection after the button press operation. For making a long operation, the button should be pressed at least the configured value.	**1 byte** 2 bytes 4 bytes
Start value	This parameter is used to set the initial value of the counter after a reset or failure.	Values depend on DPT selection.
End value	This parameter is used to set the end value of the counter.	Values depend on DPT selection.
Enable cyclic transmission of counter	This parameter is used to determine if the counter value is sent cyclically on the bus.	**No** Yes
-> Repeated transmit cycle period[1]	This parameter is used to determine the sending value to the bus when a short operation occurs.	00:00:00…**00:00:30** …18:12:15
Overflow telegram length	This parameter is used to set the length of the overflow telegram which will be sent to the bus when	**No telegram** 1 bit

	the counter value exceeds the end value set in the parameter list.	1 byte
-> Overflow telegram value[2]	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depend on DPT selection.
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

[*1] This parameter is visible when the parameter “Enable cyclic transmission of counter” is set to “Yes”.
[*2] This parameter is visible when the parameter “Overflow telegram length” is set to “1 bit” or “1 byte”.

3.5.10. Input – RGBW Control

This section, it is explained how to control an RGBW device through the buttons connected to the Multi Presence Detector. Detailed information on the relevant parameter configurations is described in the table below.

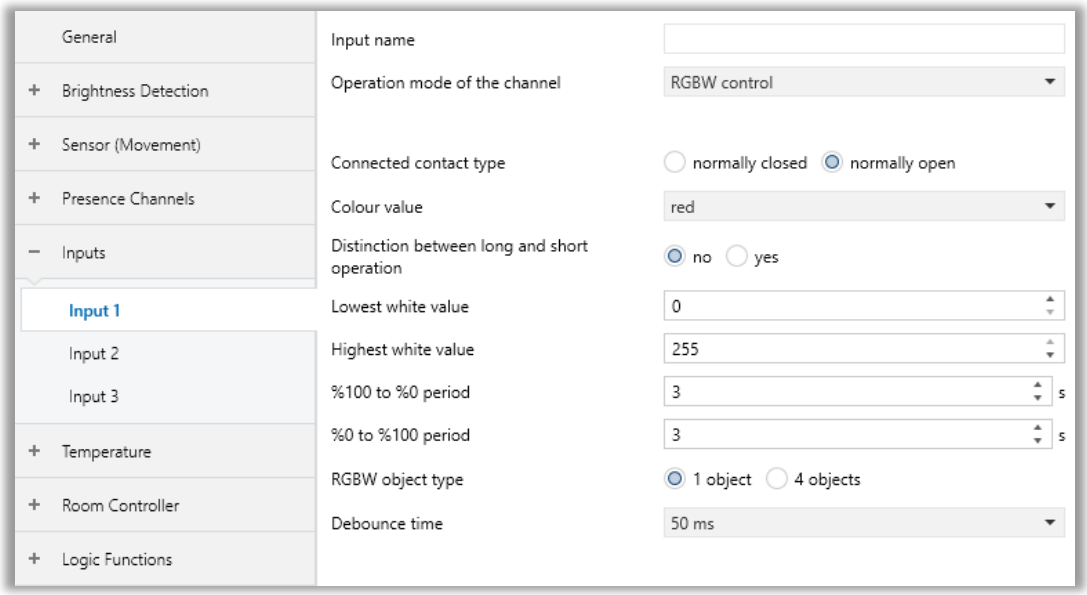


Fig. 33: RGBW Control Configuration Page

3.5.10.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Input name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 Bytes allowed**
Operation Mode of the channel	This parameter is used to determine the input function. If no function is selected, Input X will not be used. For other choices, all functionalities are configured separately.	No function Switch sensor Switch/dimming sensor Shutter sensor Value/forced operation Control scene RGB colour control Mode selection Command sequence Counter **RGBW control**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Sensor input x.	Normally closed **Normally open**
Colour value	This parameter is used to set RGBW colours according to the configured values.	**Red** Orange Yellow Green-yellow Green Green-cyan Cyan Blue-cyan Blue Blue-magenta Magenta Red-magenta White
Distinction between long and short press	This parameter is used to enable or disable the colour changing with long press operation.	**No** Yes
-> Long press after [1]	This parameter is used to determine long operation detection after the button press	00:00.200…**00:00.500** …01:05.535

	operation. For making a long operation, the button should be pressed at least the configured value.	
Lowest white value	This parameter is set to the lowest white value.	**0**…254
Highest white value	This parameter is set to the highest white value.	1…**255**
%100 to %0 period	This parameter is used to set how long it takes to go from 100% to 0%.	1s…**3s**…10s
%0 to %100 period	This parameter is used to set how long it takes to go from 0% to 100%.	1s…**3s**…10s
RGBW object type	This parameter is used to determine the RGBW colour object type.	**1 object** 4 object
Debounce time	This parameter is used to determine the debounce time. Debouncing prevents unwanted multiple operations of the input, e.g., due to bouncing of the contact.	10 ms, 20 ms, 30 ms, 40 ms, **50 ms,** 70 ms, 100 ms, 150 ms

*1 This parameter is visible when the parameter “Distinction between long and short press” is set to “Yes”.

3.5.11. Input – Additional Probe

This section describes how to configure a parameter for an NTC sensor that can be connected to the analog input of the Multi Presence Detector. After obtaining the necessary information about the NTC sensor to be connected from the relevant document, you should configure it.

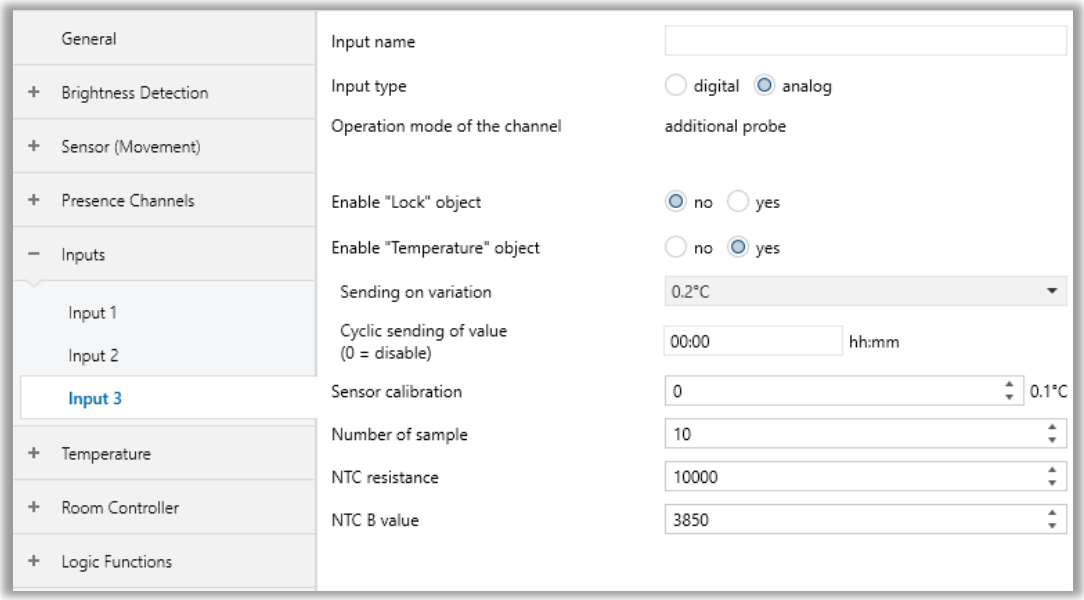


Fig. 34: Input – Additional Probe Configuration Page

3.5.11.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Input name	This parameter is used to type an input name. The name can be consisting of 40 characters.	**40 Bytes allowed**
Input type	This parameter is only available in Input 3. This parameter is used to specify whether the relevant channel will operate digital or analog.	**Digital** analog
Operation Mode of the channel	This parameter is used to determine the input function.	**Additional probe**
Enable "Lock" Object	This parameter is used to enable or disable the lock object.	**No** Yes
-> Lock initial state[1]	This parameter is used to determine the initial state of the lock status. If it is selected as enable, the initial status of the lock will be enabled.	**Disable** Enable
-> Lock Activation Telegram[1]	This parameter is used to determine the telegram value to activate the lock. E.g., if it is selected as telegram 1, a "1" value telegram will lock and a "0" value telegram will unlock. Vice versa also applies.	Telegram 0 **Telegram 1**
Enable "Temperature" Object	This parameter is used to enable or disable the Temperature object to send the measured temperature values of the ambient to the bus.	**No** Yes
-> Sending On Variation[2]	This parameter is used to determine the temperature variation value. If it is selected as never, the current value will be sent.	Never, 0.1 ˚C, **0.2 ˚C** 0.3 ˚C, 0.4 ˚C 0.5 ˚C, 0.6 ˚C 0.7 ˚C, 0.8 ˚C 0.9 ˚C, 1.0 ˚C 1.1 ˚C, 1.2 ˚C 1.3 ˚C, 1.4 ˚C 1.5 ˚C 1.4 ˚C 1.5 ˚C
-> Cyclic sending of value (0 = disable)[2]	This parameter is used to determine the cyclic sending period value of the measured temperature value.	**00:00**…23.59
Sensor Calibration	This parameter is used to determine the calibration value of the analogue probe. E.g., the Measured value is 25 ˚C, and the calibration value is selected as 10.	**-**100…**0**…100

	The calibrated value is 25 – (10 x 0.1) = 24 ˚C. 0.1 is a constant factor value.	
Number of Sample	This parameter is used to determine the number of samples of the temperature values to calculate the ambient temperature.	0…**10**…255
NTC resistance	This parameter is used to determine the NTC resistance value that will be connected to the analogue input of the KNX sensor.	0…**10000**…65535
NTC B value	This parameter is used to determine the NTC B value that will be connected to the analogue input of the KNX sensor.	0…**3850**…65535

*1 This parameter is visible when the parameter “Enable “Lock” Object” is set to “Yes”.
*2 This parameter is visible when the parameter “Enable “Temperature” Object” is set to “Yes”.

3.6. Temperature

Temperature is a quantity that should always be measured for a comfortable life in building automation systems. Temperature measurements can be made from a variety of sources. Interra KNX valence sensor temperature measurement can be made with the internal temperature sensor, an external NTC sensor that can be connected to its analog input, or the values obtained over the KNX bus line.

3.6.1. Temperature Info

This section provides information on configuring temperature parameters and what they mean. Detailed information about the parameters is given in the table below.

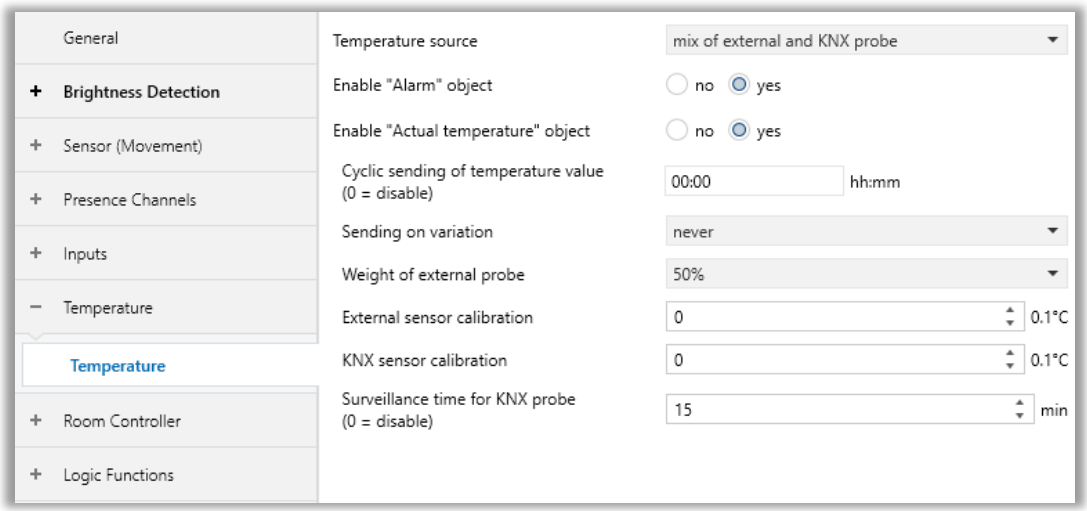


Fig. 35: Temperature Configuration Page

3.6.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Temperature source	This parameter is used to determine the temperature source for measuring the ambient temperature. **Several options can be made** : you can choose a single source or also a mix of 2 different sources according to needs.	**Internal probe** External probe Mix of internal and external probe KNX probe Mix of internal and KNX probe Mix of external and KNX probe
Enable "Alarm" Object	This parameter is used to enable the "Alarm" object to define a threshold value for alarm information.	**No** Yes
Enable "Actual Temperature" Object	This parameter is used to enable the "Actual Temperature" object to send the actual ambient temperature value to the bus.	**No** Yes
-> Cyclic sending of temperature value (0 = disable)[6]	This parameter is used to determine the cyclic sending period time of the current temperature value. If it is selected as 00:00, the cyclic sending will be disabled.	**00:00**…23.59
-> Sending on variation[6]	This parameter is used to determine the temperature variation value. If it is selected as never, the current value will be sent.	**Never** 0.1˚C,0.2˚C...1.5 ˚C
Weight Of Internal Probe[3]	This parameter is used to determine the weight of the external probe. E.g., the temperature source is selected as Mix of the internal and external probe. The external probe weight is selected as %50. So, the calculated temperature value will be: Calculated Temperature: Internal Temperature * 0.5 + External Temperature * 0.5.	%10, %20, %30, %40, **%50**, %60, %70, %80, %90
Weight Of External Probe[5]	This parameter is used to determine the weight of the external probe. E.g., the temperature source is selected as Mix of the internal and external probe. The external probe weight is selected as %50. So, the calculated temperature value will be : Calculated Temperature : Internal Temperature * 0.5 + External Temperature * 0.5.	%10, %20, %30, %40, **%50**, %60, %70, %80, %90

Internal Sensor Calibration[1]	This parameter is used to determine the calibration value of the internal sensor. E.g., the Measured value is 23 ˚C, and the calibration value is selected as -10. The calibrated value is 23 – (10 x 0.1) = 22 ˚C. 0.1 is a constant factor value.	-100…**0**…100
External Sensor Calibration[2]	This parameter is used to determine the calibration value of the external sensor. E.g., the Measured value is 26 ˚C, and the calibration value is selected as -20. The calibrated value is 26 – (20 x 0.1) = 24 ˚C.	-100…**0**…100
KNX Sensor Calibration[4]	This parameter is used to determine the calibration value is received from the KNX Probe temperature object. E.g., the Measured value is 20 ˚C, and the calibration value is selected as 20. The calibrated value is 20 + (20 x 0.1) = 22 ˚C.	-100…**0**…100
Surveillance time for KNX probe[4] **(0 = disable)**	This parameter is used to determine the surveillance time for the KNX probe. E.g., if this parameter is configured as 10. Every 10 min the received value from KNX is taken into account for temperature calculation.	0…**15**…255

[*1] This parameter is visible when the parameter “Temperature source” is set to “internal probe” or “mix of internal and external probe” or “mix of internal and KNX probe”.
[*2] This parameter is visible when the parameter “Temperature source” is set to “external probe” or “mix of internal and external probe” or “mix of external and KNX probe”.
[*3] This parameter is visible when the parameter “Temperature source” is set to “mix of internal and external probe” or “mix of internal and KNX probe”.
[*4] This parameter is visible when the parameter “Temperature source” is set to “KNX probe” or “mix of internal and KNX probe” or “mix of external and KNX probe”.
[*5] This parameter is visible when the parameter “Temperature source” is set to “mix of external and KNX probe”.
[*6] This parameter is visible when the parameter “Enable “Actual temperature” object” is set to “Yes”.

3.7. Room Controller - Thermostat

All configurations related to thermostat control on the Multi Presence Detector are described in the sections of this chapter. This parameter page will be shown when it is enabled in the “General” parameter page section. The information about the “General” parameter configuration section is described after the theoretical control type expressions that are given below.

- 2 points/Proportional fan controller that can be used by main and additional heating/cooling systems.
- Thermostat weekly program.
- Energy saving function for thermostat functions.
- Temperature limitation for thermostat functions.

3.7.1. Control Types Theoretical Explanations

The room controller device can be used for only heating, only cooling or heating and cooling. If the room controller is in heating and cooling mode, the transition from heating to cooling or vice versa can occur automatically. The thermostat measures the actual temperature of the ambient air and continuously compares it to the set temperature, and the controller automatically calculates whether to send a control signal for heating or cooling.

The control algorithm based on the difference between the desired setpoint temperature values and the measured actual temperature values processes a command value that can be either percentage or ON / OFF. The command, periodically or depending on the event, is transmitted to a KNX actuator device via a bus line with communication objects.

3.7.1.1. 2-Points Control

This control algorithm, also known as ON / OFF, is the most classic and popular one. The algorithm follows a hysteresis cycle, allowing the system to switch ON / OFF. Hence, 2 switching levels are considered for switching.

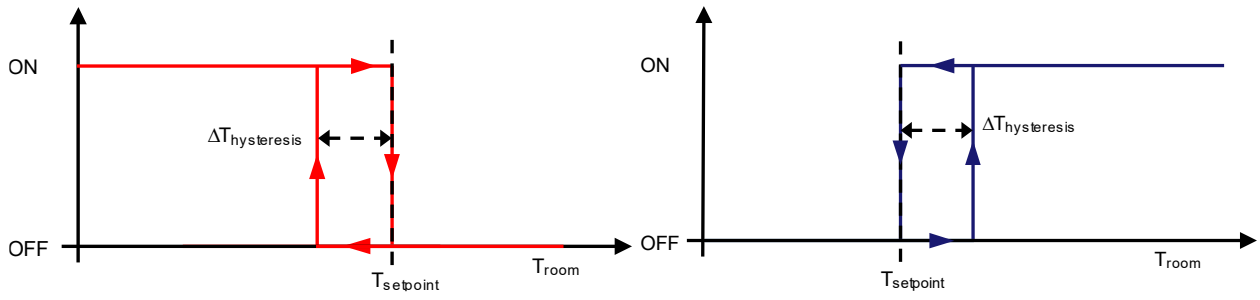


Fig. 36: 2 – Points Control Hysteresis Cycle

Heating mode

When the measured temperature is lower than the difference between the setpoint and the hysteresis value ($T_{setpoint} - \Delta T_{hysteresis}$), the device activates the heating system by sending the KNX command to the actuator that controls the heating system via connected to a related group address. When the measured temperature reaches the setpoint temperature, the device sends a related command and deactivates the heating system. In this way, there are 2 decision thresholds to activate and deactivate the heating system. The first one is the temperature at which the device activates the system ($T_{setpoint} - \Delta T_{hysteresis}$), and the second one is the temperature at which the device deactivates the heating system ($T_{setpoint}$).

Cooling mode

When the measured temperature is higher than the difference between the setpoint and the hysteresis value ($T_{setpoint} - \Delta T_{hysteresis}$), the device activates the heating system by sending the KNX command to the actuator that controls the cooling system via connected to a related group address. When the measured temperature reaches the setpoint temperature, the device sends a related command and deactivates the cooling system. In this way, there are 2 decision thresholds to activate and deactivate the cooling system. The first one is the temperature at which the device activates the system ($T_{setpoint} + \Delta T_{hysteresis}$), and the second one is the temperature at which the device deactivates the heating system ($T_{setpoint}$). There are 2 different parameters for heating and cooling hysteresis values in the ETS programme. Values differ depending on the system type.

3.7.1.2. Continuous (PI) Control

Proportional – Integral control (PI control) is explained by the relationship shown below:

$$\boldsymbol{control\ variable(t)\ =\ Kp\ \times\ error(t)\ +\ Ki\ \times \int_0^t error(t)\ dt}$$

whereby:

$$error(t)\ =\ (Setpoint\ -Measured\ temperature)\ in\ heating$$
$$error(t)\ =\ (Measured\ temperature\ -Setpoint)\ in\ cooling$$
$$Kp\ =\ proportional\ constant$$
$$Ki\ =\ integral\ constant$$

The control variable contains integral and proportional (Ki and Kp) constants to eliminate errors. In practice, intuitively generated values are generally used.

Ex 1:

$$\boldsymbol{Proportional\ band\ BP\ [K]\ =\ 100\ /\ Kp} \qquad \boldsymbol{Integral\ time\ Ti\ [min]\ =\ Kp\ /\ Ki}$$

The proportional band is the error value that determines the maximum deflection output as 100%.

For example, a regulator with a proportional band of 5 K provides a 100% control output when the Setpoint = 20°C and the measured temperature is ≤ 15°C in heating; in the cooling conduction mode, it provides a 100% control output when the Setpoint = 24°C and the measured temperature is ≥ 29°C. As shown in the figure, a regulator with a small proportional band tends to provide higher values of the control variable for small errors than a regulator with a higher proportional band.

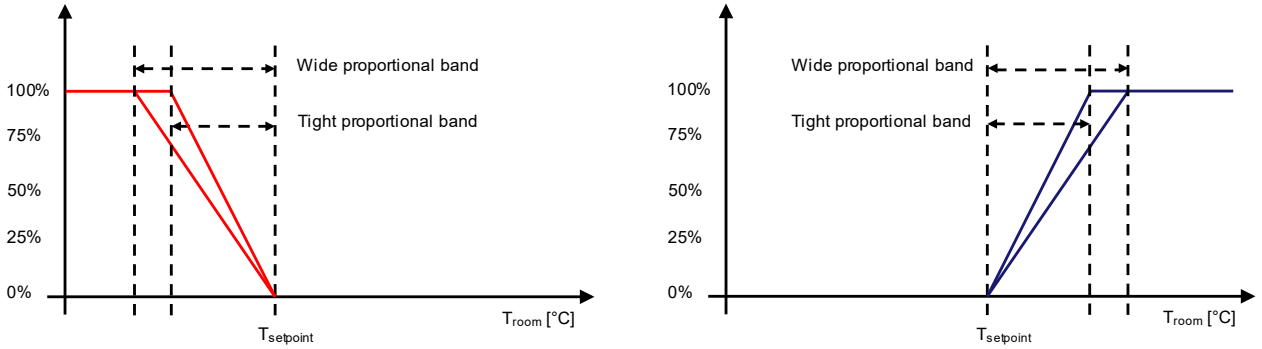


Fig. 37: Continuous PI Control Proportional Band Widths

The integral time is the time required to repeat the value of the control variable of a purely proportional regulator when the error remains constant in time.

Ex 2:

For example, with a purely proportional controller in heating and with a value of proportional band of 4 K, if the setpoint is = 20°C and the measured temperature is = 18°C, the control variable assumes the value of 50%. With an integral time = 60 minutes, if the error remains constant, the control variable will take the value = 100% after 1 hour, i.e., a contribution equal to the value given by only proportional contribution will be added to the control variable. In heating and air conditioning systems, a purely proportional controller is not able to guarantee the achievement of the setpoint. You should always introduce an integrated action for achieving the Setpoint: that is why the integral action is also called automatic reset.

3.7.1.3. PWM (PI) Control

The PWM (Pulse Width Modulation) proportional-integral controller allows the digital output to be set to ON and OFF by sampling an analogue control variable within a specified period. The controller runs periodically through a cycle and keeps its output ON for each period in proportion to the value of the control variable. As shown in the below figure, by varying the ratio between the “ON” time and the “OFF” time, the average activation time of the output changes, and as a result, the average heating or cooling power supplied by the room changes.

The cycle time for the control value for the PWM signal calculated from the PI controller's control value is specified. Depending on the control value, the selected cycle time is divided into an ON and OFF signal. Therefore, a control value output of 50 % with a PWM cycle of 12 min signifies an ON phase of 6 min. and an OFF phase of 6 min.

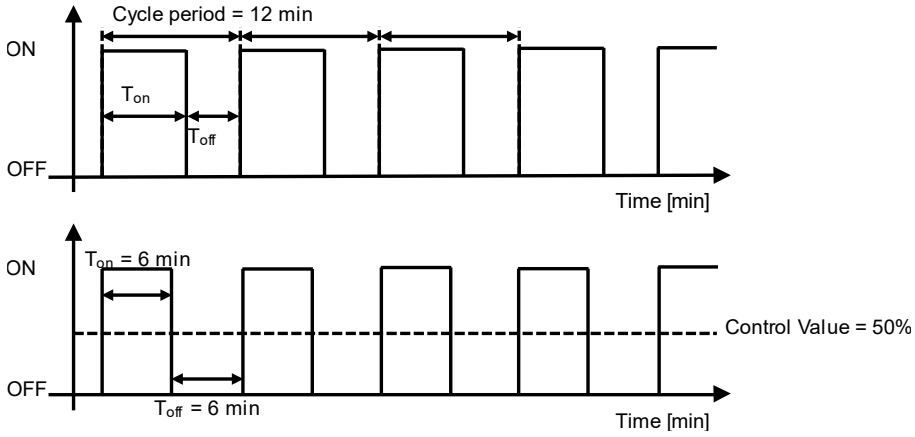


Fig. 38: PWM Control Sampling

This type of control is well suited for use with ON / OFF actuators, such as electrothermal actuators and drives for zone valves, which are less expensive than proportional actuators.

A distinctive advantage of this type of control is that it eliminates the inertia of the system: it allows significant energy savings because unnecessary interventions on the system introduced by the 2-point control with hysteresis are avoided and only the power is required to compensate for the losses.

Every time the changes the desired temperature setpoint is, the cycle time is interrupted, the control output is reprocessed and the PWM restarts with a new cycle: this allows the system to reach its steady state more quickly.

Terminal Type	Proportional Ban [K]	Integral Time [min]	Cycle Period [min]
Radiators	5	150	15-20
Electrical heaters	4	100	15-20
Fan-coil	4	90	15-20
Floor radiant panels	5	240	15-20
Ceiling radiant panels	5	100	15-20

Table 11: Guidelines for choosing the proper parameters of a PMW PI controller

Guidelines for choosing the proper parameters of a PMW Proportional-Integral controller:

- Cycle time: for low-inertial systems such as heating and air conditioning systems, short cycle times must be chosen (10-15 minutes) to avoid oscillations of the room temperature.
- Narrow proportional band: wide and continuous oscillations of the room temperature, short setpoint settling time.
- Wide proportional band: small or no oscillations of the room temperature, long setpoint settling time.
- Short integral time: short setpoint settling time, continuous oscillations of the room temperature.
- Long integral time: long setpoint settling time, no oscillations of the room temperature.

3.7.2. Thermostat X

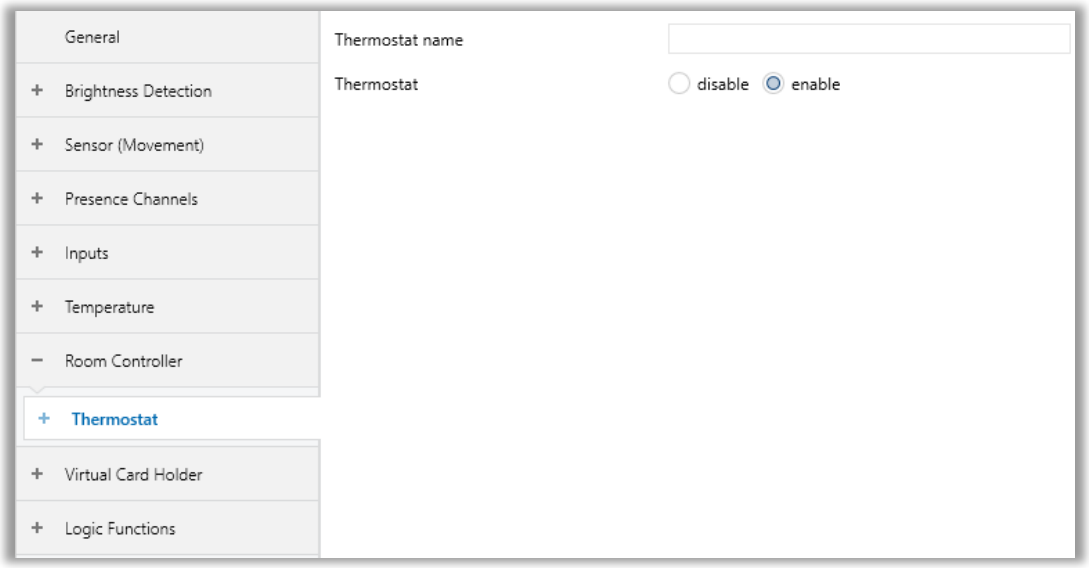


Fig. 39: Room Controller Thermostat Configuration Page

3.7.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Thermostat name	This parameter is used to type a Thermostat name. The name can be consisting of 40 characters.	**40 Bytes allowed**
Thermostat	This parameter is used to control the thermostat features.	**Disable** Enable

3.7.3. Thermostat - General

The thermostat function can be selected as the “master” controller or “slave” controller in the configuration settings in this section. When the selection is made as to the “master” controller, configuration sections and the communication objects are opened to define the thermostat functions. When the selection is made as to the “slave” controller, some configuration sections related to the thermostat functions are disabled. The slave controller must be connected to the master controller with the KNX communication object as it will operate as a dependent controller with commutations object. In thermostat slave mode, setpoint adjustment, thermostat activation control, heating/cooling switchover and operation mode control can be made. Also, LCD can be used as fan indicator in slave operation or fan controller isn’t used for thermostat.

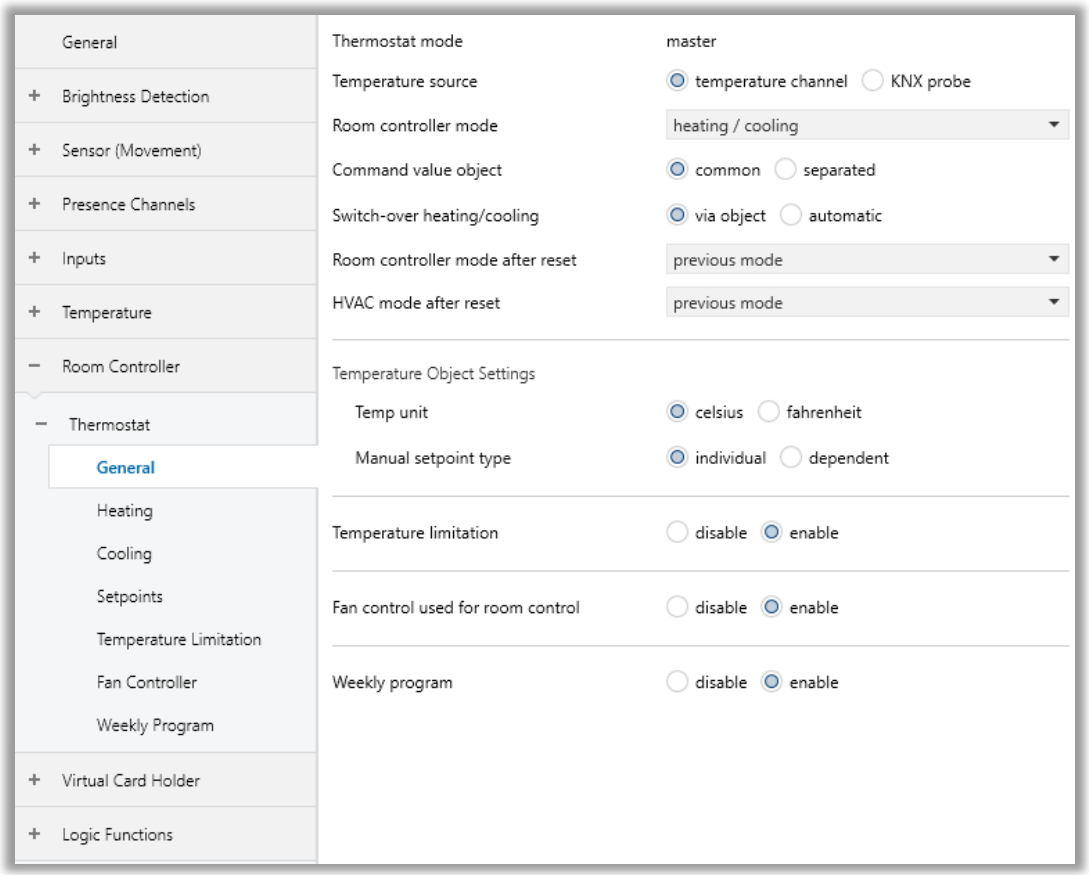


Fig. 40: Room Controller Thermostat General Configuration Page

3.7.3.1. Parameters List

PARAMETER	DESCRIPTION	VALUES
Thermostat mode	The thermostat function's operating type is determined with this parameter.	**Master**
Temperature source	This parameter determines the temperature source of room controller. If thermostat temperature value is higher/lower than the setpoints of the protection mode's setpoint max/min limit values, the active operation mode is changed as Protection mode. After that the end-users can change the operation mode again.	**Temperature channel** KNX probe
Room controller mode[1]	Room controller mode is determined with this parameter.	**Heating** Cooling Heating & Cooling
HVAC mode after reset[1]	This parameter determines the operating mode of the room controller after a reset occurs. **Ex:** When a power failure occurs.	**Previous value** Comfort Standby Economy Protection
Command value object[2]	The object types of temperature command values for heating and cooling mode are determined with this parameter.	**Common** Separated
Switch-over heating / cooling[2]	This parameter determines how the heating/cooling transition is made. If heating/cooling switch-over mode isn't Automatic, the user can be configured heating or cooling setpoint. If heating/cooling switch-over mode is Automatic, the user can't be configured that the cooling setpoint is higher than the heating setpoint. In automatic mode the cooling setpoint is equal the heating setpoint at least. If an input value that is higher than heating setpoint, is received over "Cooling [Operation Mode] Setpoint Temperature" object, received telegram is ignored.	**Via communication object** Automatic

Room controller mode after reset[3]	This parameter determines the room controller mode of the room controller after a reset occurs. **Ex:** When a power failure occurs.	Heating Cooling **Previous mode**
Temp Unit	The temperature unit type to be used by thermostat objects is defined by this parameter.	**Celsius** Fahrenheit
Manual setpoint type	The desired temperature value can be controlled with individual or dependent setpoints by this parameter. **Individual setpoint:** The input value must be the desired setpoint. **Dependent setpoint:** The input value must be the difference of desired setpoint according to base setpoint.	**Individual** Dependent
Temperature limitation	This parameter enables temperature limitation function of thermostat.	**Disable** Enable
Fan control used for room control[1]	This parameter determines the fan controls that are used inside or outside of the thermostat function. If the it is selected to use outside of the thermostat function, just the fan states will be displayed on the device as fan indicators.	**Disable** Enable
Weekly program	This parameter enables weekly program of thermostat.	**Disable** Enable

*1 This parameter is visible when the parameter “Thermostat mode” is set to “Master”.

*2 This parameter is visible when the parameter “Room controller mode” is set to “Heating / cooling”.

*3 This parameter is visible when the parameter “Switch-over heating/cooling” is set to “Via object”.

3.7.4. Thermostat - Heating

The device's operation principle of the heating feature is as follows: When the measured temperature is lower than the setpoint temperature, the device activates the heating system by sending a KNX command to the actuator that controls the heating system via connected to the related group address. When the measured temperature reaches the setpoint temperature, the device sends a related command and deactivates the heating system. The heating feature can be controlled with different types of configuration settings. These configuration settings are as follows;

Selection of the "Heating 2 – Points Control" parameter, 1-bit / 1-byte on/off control can be selected.

Selection of the "Heating PWM Control" parameter, 1-bit / 1-byte on/off control can be selected.

Selection of the "Heating Continuous Control" parameter, 1-byte proportional-integral control.

3.7.4.1. Heating 2 – Points Control

When the measured temperature is lower than the difference between the setpoint and the hysteresis value ($T_{setpoint} - \Delta T_{hysteresis}$), the device activates the heating system by sending a KNX command to the actuator that controls the heating system via connected to a related group address. When the measured temperature reaches the setpoint temperature, the device sends a related command and deactivates the heating system. In this way, there are 2 decision thresholds to activate and deactivate the heating system. The first one is the temperature at which the device activates the system ($T_{setpoint} - \Delta T_{hysteresis}$), and the second one is the temperature at which the device deactivates the heating system ($T_{setpoint}$).

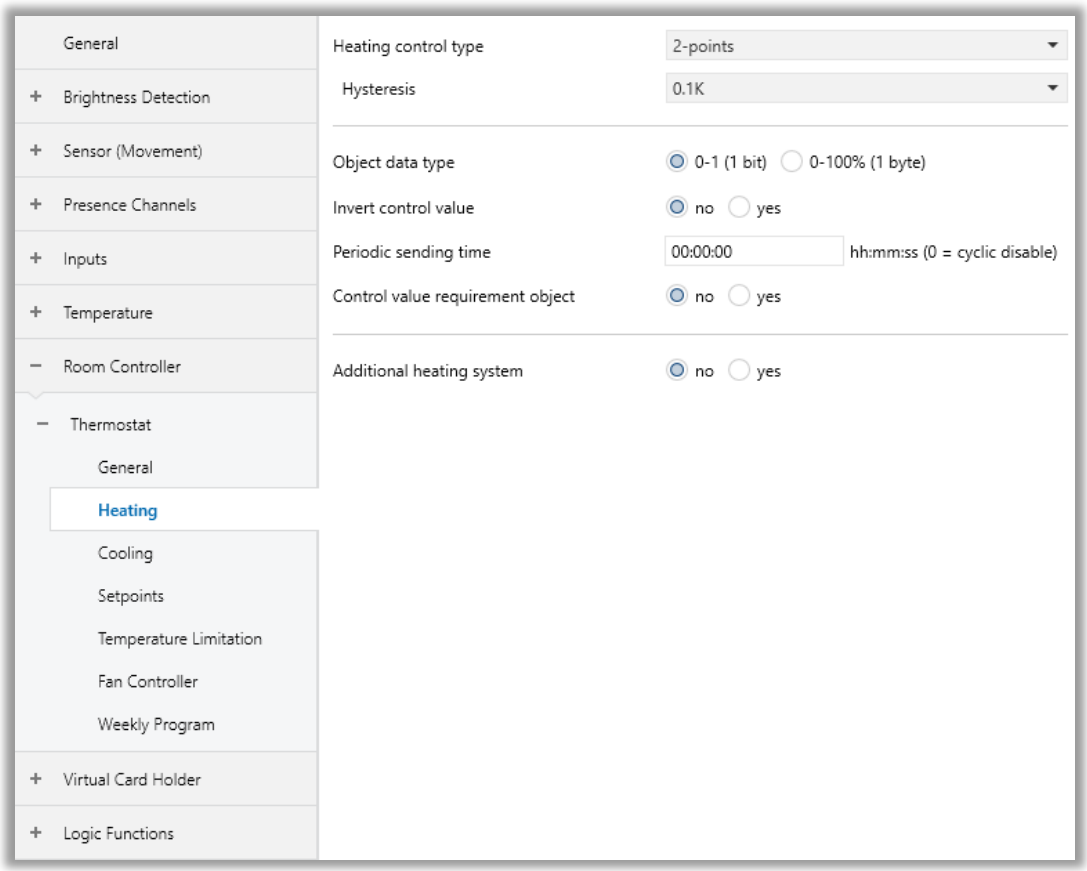


Fig. 41: Heating 2-Points Control Configuration Page

3.7.4.2. Parameters List

PARAMETER	DESCRIPTION	VALUES
Heating control type	This parameter determines the heating control type.	**2 – points** PWM Continuous
Hysteresis	This parameter determines the hysteresis value.	0.1K…2.0K
Object data type	This parameter is used to determine data type of control value object.	**0-1 (1 bit)** 0-100% (1 byte)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the heating system.	**No** Yes
Additional heating system	This parameter activates the additional heating system.	**No** Yes

3.7.4.3. Heating PWM Control

The PWM (Pulse Width Modulation) proportional-integral controller allows the digital output to be set to ON and OFF by sampling an analogue control variable within a specified period. The controller runs periodically through a cycle and keeps its output ON for each period in proportion to the value of the control variable. By varying the ratio between the “ON” time and the “OFF” time of the heating system, the average activation time of the output changes, and as a result, the average heating power supplied by the room changes.

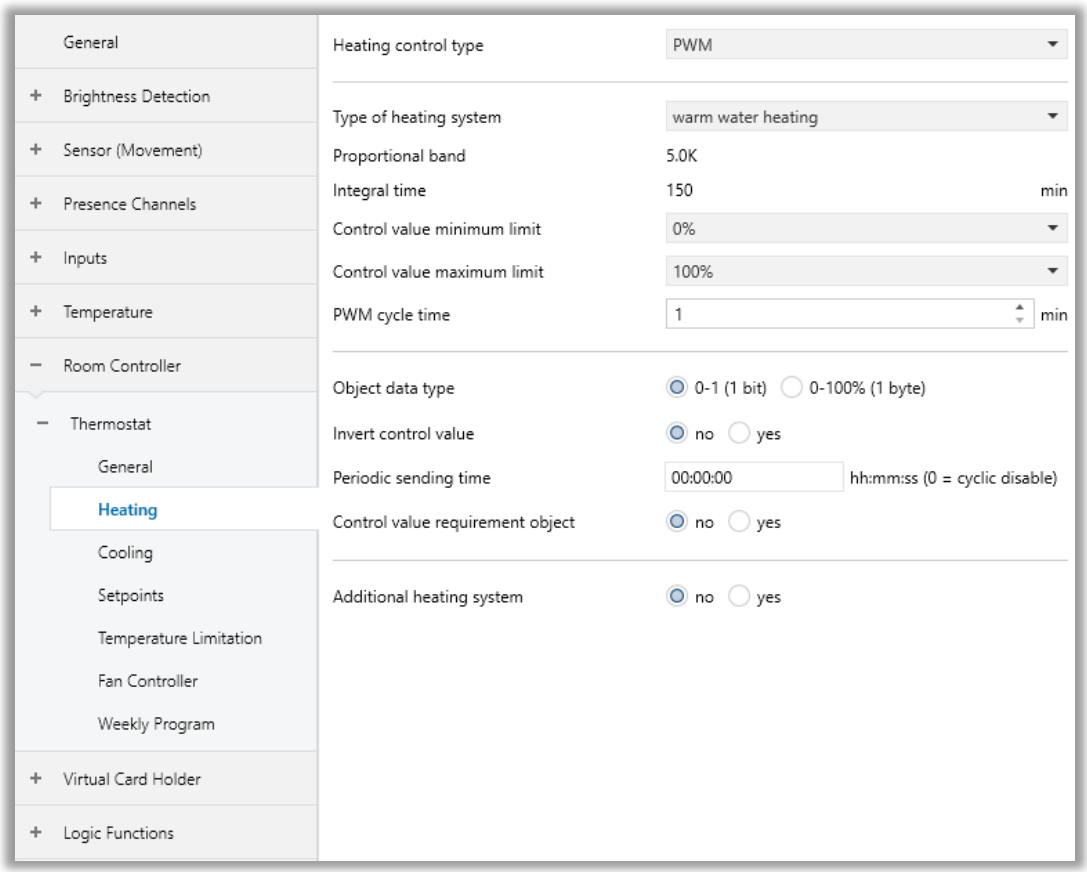


Fig. 42: Heating PWM Control Configuration Page

3.7.4.4. Parameters List

PARAMETER	DESCRIPTION	VALUES
Type of heating system	This parameter determines the heating system to be controlled.	**Warm water heating** Electric heating Floor heating Split unit Fan coil User defined
Proportional band (K)	This parameter determines the proportional band.	**5.0K** (0.5K…10.0K)
Integral time (min)	This parameter determines the integral time.	**150** (0…255)
Control value minimum (%)	This parameter determines the output object's minimum control value.	**0%** (0%, 5%, 10%, 15%, 20%, 25%, 30%)
Control value maximum (%)	This parameter determines the output object's maximum control value.	**100%** (70%, 75%, 80%, 85%, 90%, %95, 100%)
PWM cycle time (min)	This parameter determines the PWM cycle time.	**1**…255
Object data type	This parameter is used to determine data type of control value object.	**0-1 (1 bit)** 0-100% (1 byte)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the heating system.	**No** Yes

3.7.4.5. Heating Continuous Control

Proportional – Integral control (PI control) is explained by the relationship shown below:

$$\boldsymbol{control\ variable(t)\ =\ Kp\ \times\ error(t)\ +\ Ki\ \times \int_0^t error(t)\ dt}$$

whereby:

$$error(t)\ =\ (Setpoint\ -Measured\ temperature)\ in\ heating$$
$$error(t)\ =\ (Measured\ temperature\ -Setpoint)\ in\ cooling$$
$$Kp\ =\ proportional\ constan$$
$$Ki\ =\ integral\ constant$$

The control variable contains integral and proportional (Ki and Kp) constants to eliminate errors. In practice, intuitively generated values are generally used.

Ex 1:

$$\boldsymbol{Proportional\ band\ BP\ [K] = \frac{100}{Kp}}$$
$$\boldsymbol{Integral\ time\ Ti\ [min]\ =\ Kp\ /\ Ki}$$

The proportional band is the error value that determines the maximum deflection output as 100%.

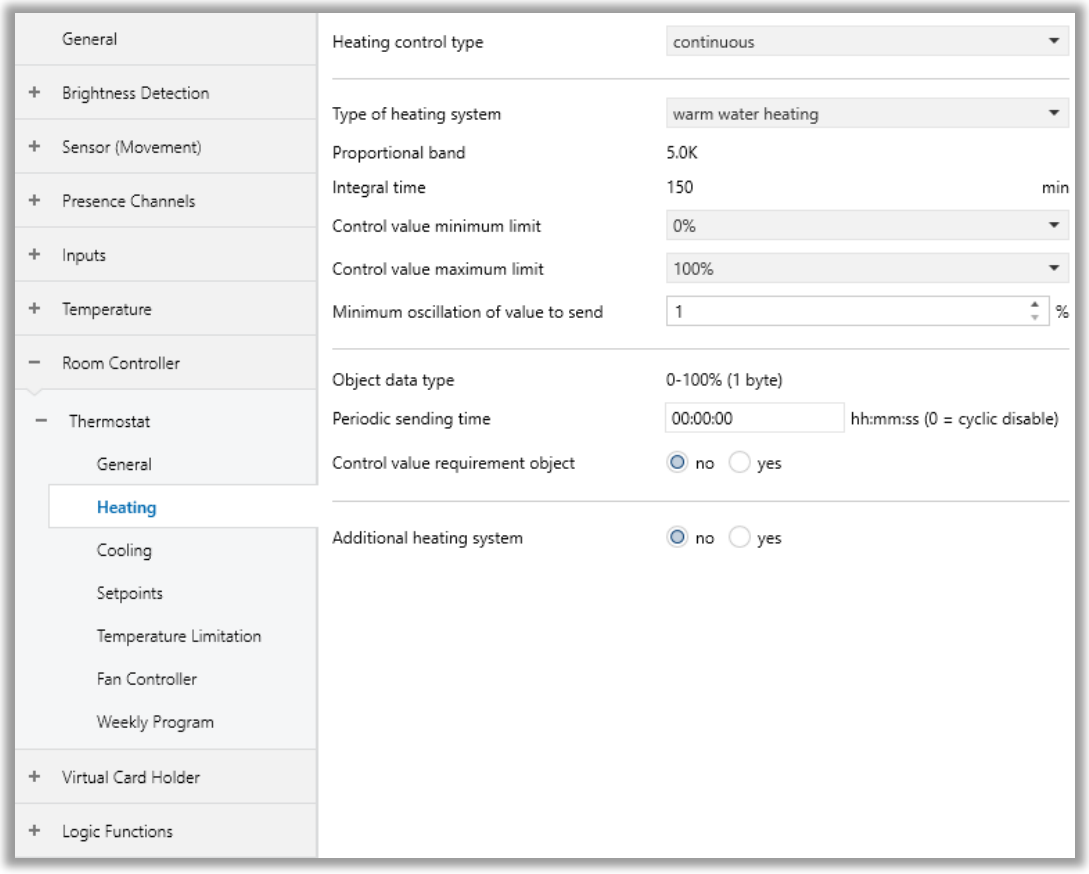


Fig. 43: Heating Continuous Control Configuration Page

3.7.4.6. Parameters List

PARAMETER	DESCRIPTION	VALUES
Type of heating system	This parameter determines the heating system to be controlled.	**Warm water heating** Electric heating Floor heating Split unit Fan coil User defined
Proportional band (K)	This parameter determines the proportional band.	**5.0K** (0.5K … 10.0K)
Integral time (min)	This parameter determines the integral time.	**150** (0 … 255)
Control value minimum (%)	This parameter determines the output object's minimum control value.	**0%** (0%, 5%, 10%, 15%, 20%, 25%, 30%)
Control value maximum (%)	This parameter determines the output object's maximum control value.	**100%** (70%, 75%, 80%, 85%, 90%, 95%, 100%)
Minimum oscillation of value to send (%)	This parameter determines the minimum oscillation value for the output object to send a value.	**3** (0…100)
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the heating system.	**No** Yes

3.7.4.7. Additional Heating System

All types of heating controls (2-points, PWM and continuous control) have additional heating system options. The additional heating system works in all control types with the same characteristics. The system activates itself according to the offset configuration. If ($T_{setpoint} - \Delta T_{offset}$) is lower than the ambient room temperature, the additional heating system will be activated according to controller type.

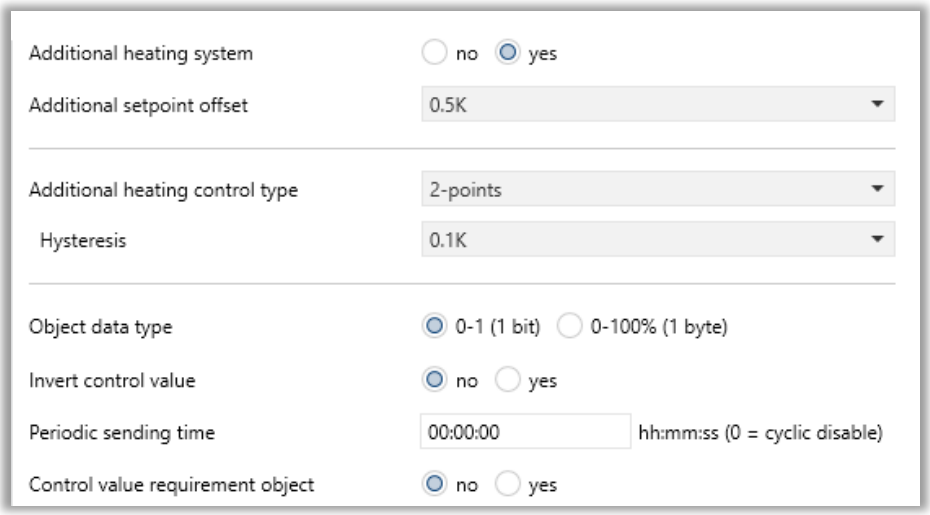


Fig. 44: Additional Heating System Configuration Page

In additional heating control, 2 - Points and PI Continuous controller heat the room until the difference between ($T_{setpoint} - T_{room}$) is equal to “Additional setpoint offset” parameter.

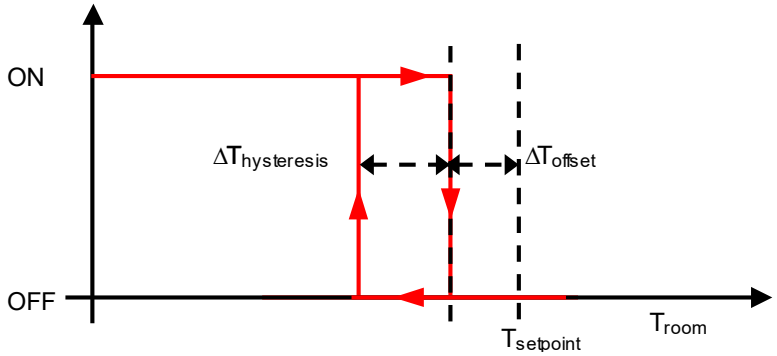


Fig. 45: 2 – Points Hysteresis Cycle for Additional Heating Control

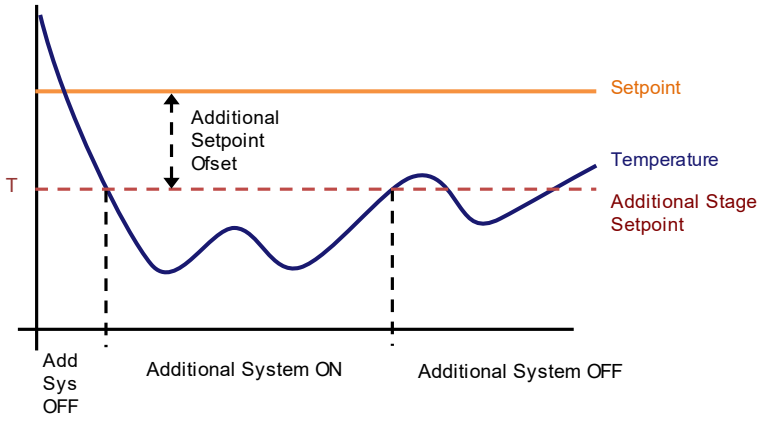


Fig. 46: PI Continuous Graph for Additional Heating Control

3.7.4.8. Parameters List

PARAMETER	DESCRIPTION	VALUES
Additional heating system	This parameter activates the additional heating system.	**No** Yes
Additional setpoint offset	This parameter determines the difference between the setpoint temperature value and the additional heating system's setpoint temperature value.	**0.5K** … 5.0K (°C) **0.9K** … 9.0K (°F)
Additional heating control type	This parameter determines the additional heating system's control object type.	**2 – points** PWM Continuous
Additional heating control type: 2-points		
Hysteresis Value	This parameter determines the hysteresis value.	**0.1K**…2.0K (°C) **0.18K**…3.6K (°F)
Object type	This parameter is used to determine the data type of the control value object.	**0-1 (1 bit)** 0-100% (1 byte)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter determines the time of control value to be sent periodically.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the additional heating system.	**No** Yes
Additional heating control type: PWM		
Type of additional heating system	This parameter determines the heating system to be controlled.	**Warm water heating** Electric heating Floor heating Split unit Fan coil User defined

Proportional band	This parameter determines the proportional band.	0.5K… **5.0K** … 10.0K (°C) 0.9K… **9.0K** … 18.0K (°F)
Integral time	This parameter determines the integral time.	0 … **90** … 255
Control value minimum limit	This parameter determines the output object's minimum control value.	**0%**, 5%, 10%, 15%, 20%, 25%, 30%)
Control value maximum limit	This parameter determines the output object's maximum control value.	70%, 75%, 80%, 85%, 90%, %95, **100%**
PWM cycle time (min)	This parameter determines the PWM cycle time.	**1**…255
Object data type	This parameter is used to determine data type of control value object.	**0-1 (1 bit)** 0-100% (1 byte)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the additional heating system.	**No** Yes
Additional heating control type: Continuous		
Type of additional heating system	This parameter determines the heating system to be controlled.	Warm water heating Electric heating Floor heating Split unit **Fan coil** User defined
Proportional band	This parameter determines the proportional band.	0.5K… **5.0K** … 10.0K (°C) 0.9K… **9.0K** … 18.0K (°F)
Integral time	This parameter determines the integral time.	0 … **90** … 255
Control value minimum limit	This parameter determines the output object's minimum control value.	**0%** (0%, 5%, 10%, 15%, 20%, 25%, 30%)

Control value maximum limit	This parameter determines the output object's maximum control value.	**100%** (70%, 75%, 80%, 85%, 90%, %95, 100%)
Minimum oscillation of value to send	This parameter determines the minimum oscillation value for the output object to send a value.	**1** … 100
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the additional heating system.	**No** Yes

3.7.5. Thermostat - Cooling

The device's operation principle of cooling feature is as follows: When the measured temperature is higher than the setpoint temperature, the device activates the cooling system by sending a KNX command to the actuator that controls the cooling system via connection to the related group address. When the measured temperature reaches the setpoint temperature, the device sends a related command and deactivates the cooling system. The cooling feature can be controlled with different types of configuration settings. These configuration settings are as follows;

Selection of the "Cooling 2 – Points Control" parameter, 1-bit / 1-byte on/off control can be selected.

Selection of the "Cooling PWM Control" parameter, 1-bit / 1-byte on/off control can be selected.

Selection of the "Cooling Continuous Control" parameter, 1-byte proportional-integral control.

3.7.5.1. Cooling 2 – Points Control

When the measured temperature is higher than the difference between the setpoint and the hysteresis value ($T_{setpoint} + \Delta T_{hysteresis}$), the device activates the cooling system by sending a KNX command to the actuator that controls the cooling system via connected to a related group address. When the measured temperature reaches the setpoint temperature, the device sends a related command and deactivates the cooling system. In this way, there are 2 decision thresholds to activate and deactivate the cooling system. The first one is the temperature at which the device activates the cooling system ($T_{setpoint} + \Delta T_{hysteresis}$), and the second one is the temperature at which the device deactivates the cooling system ($T_{setpoint}$).

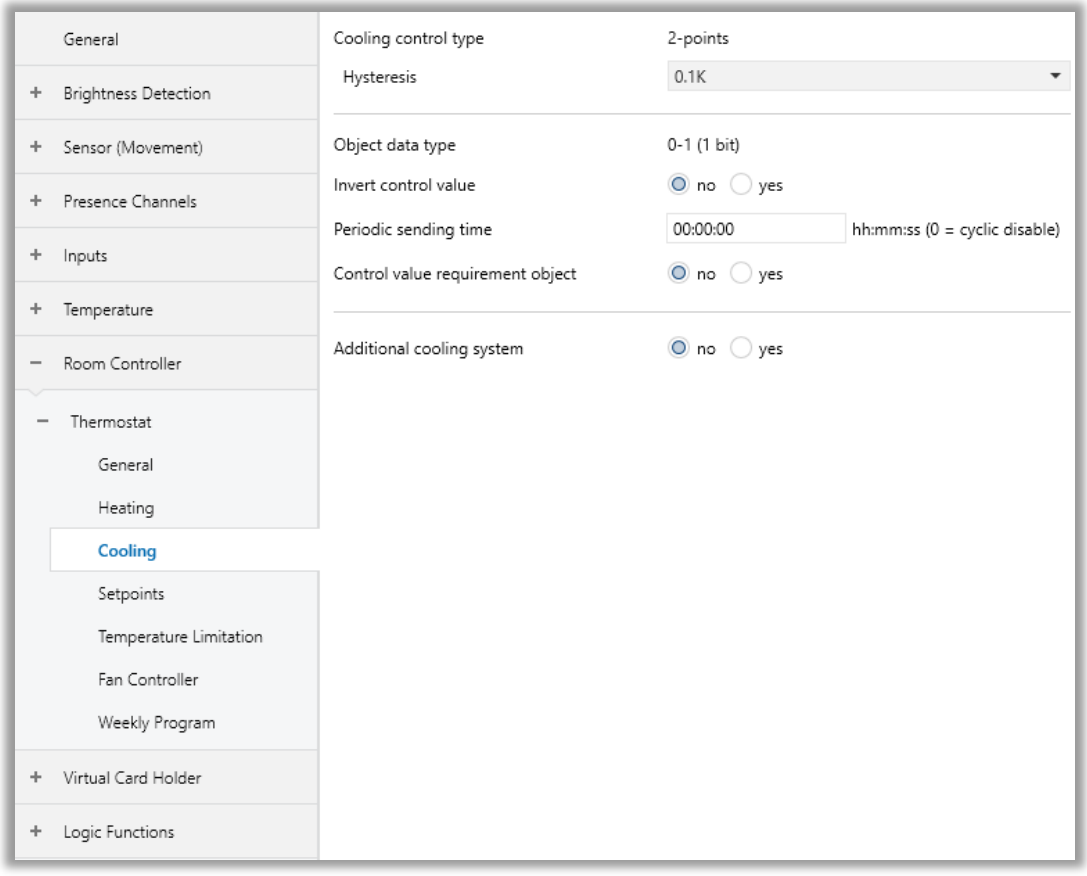


Fig. 47: Cooling 2-Points Control Configuration Page

3.7.5.2. Parameters List

PARAMETER	DESCRIPTION	VALUES
Cooling control type	This parameter determines the cooling control type.	**2 – points** PWM Continuous
Hysteresis	This parameter determines the hysteresis value.	**0.1K**…2.0K (°C) **0.18K**…3.6K (°F)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the cooling system.	**No** Yes
Additional cooling system	This parameter activates the additional cooling system.	**No** Yes

3.7.5.3. Cooling PWM Control

The PWM (Pulse Width Modulation) proportional-integral controller allows the digital output to be set to On and Off by sampling an analogue control variable within a specified time. The controller runs periodically through a cycle and keeps its output ON for each period in proportion to the value of the control variable. By varying the ratio between the “ON” time and the “OFF” time of the heating system, the average activation time of the output changes, and as a result, the average heating power supplied by the room changes.

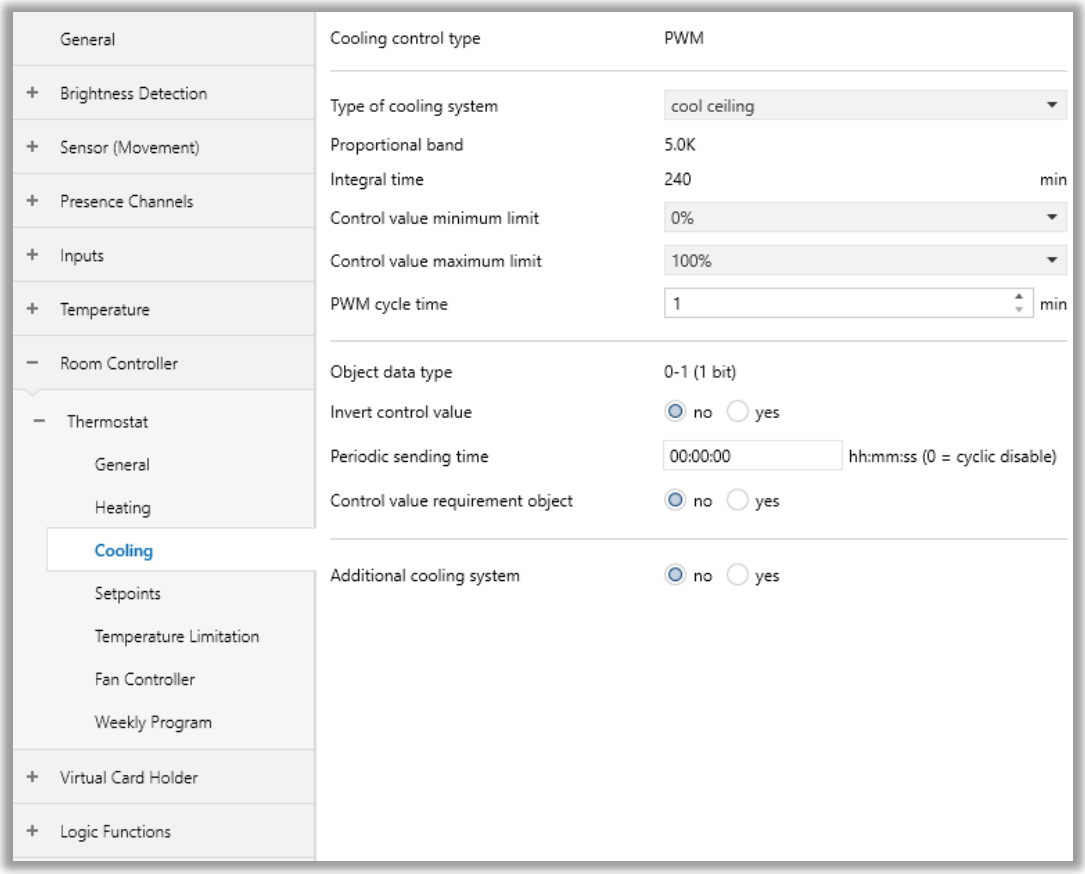


Fig. 48: Cooling PWM Control Configuration Page

3.7.5.4. Parameters List

PARAMETER	DESCRIPTION	VALUES
Type of cooling system	This parameter determines the cooling system to be controlled.	**Cool ceiling** Split unit Fan coil User defined
Proportional band (K)	This parameter determines the proportional band.	0.5K… **4.0K** … 10.0K (°C) 0.9K… **7.2K** … 18.0K (°F)
Integral time (min)	This parameter determines the integral time.	0…**90**…255
Control value minimum (%)	This parameter determines the output object's minimum control value.	**0%** (0%, 5%, 10%, 15%, 20%, 25%, 30%)
Control value maximum (%)	This parameter determines the output object's maximum control value.	**100%** (70%, 75%, 80%, 85%, 90%, %95, 100%)
PWM cycle time (min)	This parameter determines the PWM cycle time.	**1**…255
Object data type	This parameter is used to determine data type of control value object.	**0-1 (1 bit)** 0-100% (1 byte)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the cooling system.	**No** Yes
Additional cooling system	This parameter activates the additional cooling system.	**No** Yes

3.7.5.5. Cooling Continuous Control

Proportional–integral control (PI control) is explained by the relationship shown below:

$$\boldsymbol{control\ variable(t)} = \boldsymbol{Kp} \times \boldsymbol{error(t)} + \boldsymbol{Ki} \times \int_0^t \boldsymbol{error(t)\ dt}$$

whereby:

$$error(t) = (Setpoint - Measured\ temperature)\ in\ heating$$
$$error(t) = (Measured\ temperature - Setpoint)\ in\ cooling$$
$$Kp = proportional\ constant$$
$$Ki = integral\ constant$$

The control variable contains integral and proportional (Ki and Kp) constants to eliminate errors. In practice, intuitively generated values are generally used.

Ex 1:

$$\boldsymbol{Proportional\ band\ BP\ [K]} = \frac{\boldsymbol{100}}{\boldsymbol{Kp}}$$
$$\boldsymbol{Integral\ time\ Ti\ [min]} = \frac{\boldsymbol{Kp}}{\boldsymbol{Ki}}$$

The proportional band is the error value that determines the maximum deflection output as 100%.

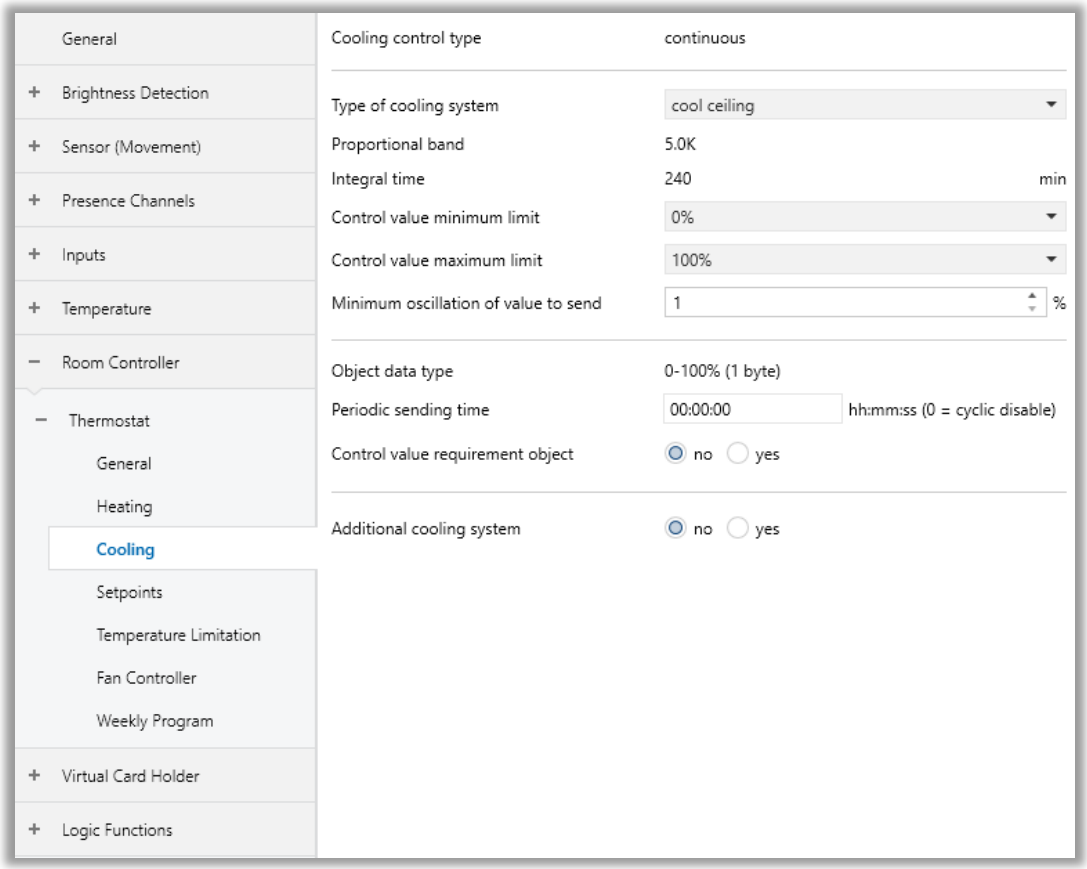


Fig. 49: Cooling Continuous Control Configuration Page

3.7.5.6. Parameters List

PARAMETER	DESCRIPTION	VALUES
Type of cooling system	This parameter determines the cooling system to be controlled.	**Cool ceiling** Split unit Fan coil User defined
Proportional band (K)	This parameter determines the proportional band.	0.5K… **5.0K** … 10.0K (°C) 0.9K… **9.0K** … 18.0K (°F)
Integral time (min)	This parameter determines the integral time.	0 … **90** … 255
Control value minimum (%)	This parameter determines the output object's minimum control value.	**0%** (0%, 5%, 10%, 15%, 20%, 25%, 30%)
Control value maximum (%)	This parameter determines the output object's maximum control value.	**100%** (70%, 75%, 80%, 85%, 90%, 95%, 100%)
Minimum oscillation of value to send (%)	This parameter determines the minimum oscillation value for the output object to send a value.	**1** …100
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the cooling system.	**No** Yes

3.7.5.7. Additional Cooling System

All types of cooling controls (2-points, PWM and continuous control) have additional cooling system options. The additional cooling system works in all control types with the same characteristics. The system activates itself according to the offset configuration. If ($T_{setpoint}$ + ΔT_{offset}) is higher than the ambient room temperature, the additional cooling system will be activated according to controller type.

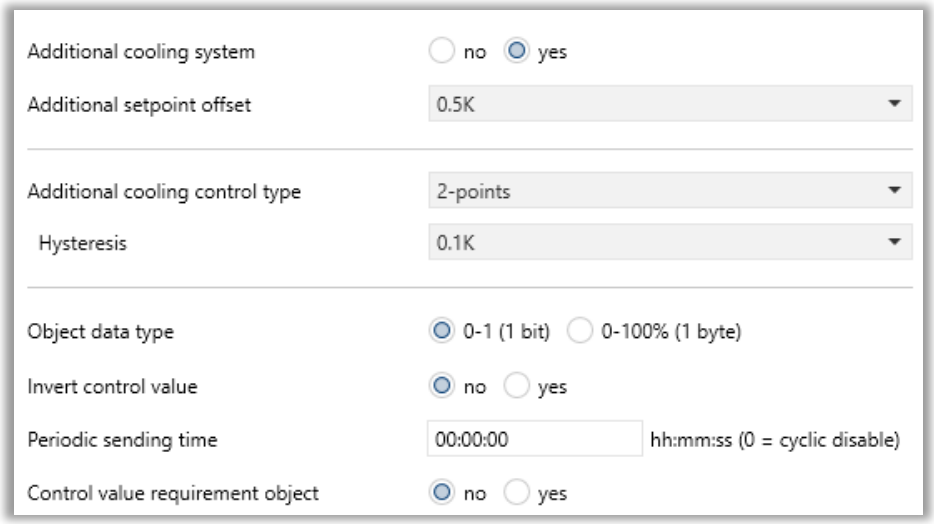


Fig. 50: Additional Cooling System Configuration Page

In additional cooling control, 2 - Points and PI Continuous controller cool the room until the difference between (T_{room} – $T_{setpoint}$) is equal to “Additional setpoint offset” parameter.

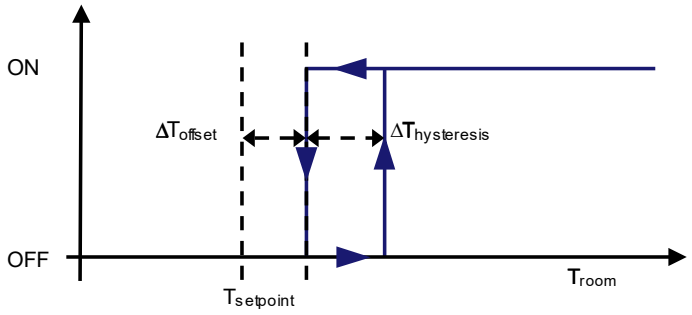


Fig. 51: 2 – Points Hysteresis Cycle for Additional Cooling Control

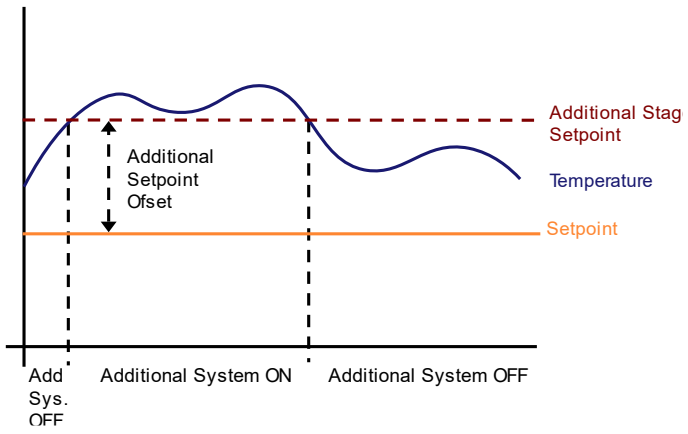


Fig. 52: PI Continuous Graph for Additional Cooling Control

3.7.5.8. Parameters List

PARAMETER	DESCRIPTION	VALUES
Additional setpoint offset	This parameter determines the difference between the setpoint temperature value and the additional cooling system's setpoint temperature value.	**0.5K** … 5.0K (°C) **0.9K** … 9.0K (°F)
Additional cooling control type	This parameter determines the additional cooling system's control object type.	**2 – points** PWM Continuous
Additional cooling control type: 2-points		
Hysteresis Value	This parameter determines the hysteresis value.	**0.1K**…2.0K (°C) **0.18K**…3.6K (°F)
Object type	This parameter determines the additional cooling system's object type.	**0-2 (1 bit)** 0-100% (1 byte)
Invert control value	This parameter is used to invert control output.	**No** Yes
Periodic sending time	This parameter determines the time of control value to be sent periodically.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the additional cooling system.	**No** Yes
Additional cooling control type: PWM		
Type of additional cooling system	This parameter determines the cooling system to be controlled.	**Cool ceiling** Split unit Fan coil User defined
Proportional band	This parameter determines the proportional band.	0.5K… **5.0K** … 10.0K (°C) 0.9K… **9.0K** … 18.0K (°F)
Integral time	This parameter determines the integral time.	0 … **240** … 255
Control value minimum limit	This parameter determines the output object's minimum control value.	**0%**, 5%, 10%, 15%, 20%, 25%, 30%

Control value maximum limit	This parameter determines the output object's maximum control value.	70%, 75%, 80%, 85%, 90%, %95, **100%**
PWM cycle time (min)	This parameter determines the PWM cycle time.	**1**…255
Object data type	This parameter is used to determine data type of control value object.	**0-2 (1 bit)** 0-100% (1 byte)
Invert control value	It is used to invert control output.	**No** Yes
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the additional cooling system.	**No** Yes
Additional cooling control type: Continuous		
Type of additional cooling system	This parameter determines the cooling system to be controlled.	Cool ceiling Split unit **Fan coil** User defined
Proportional band	This parameter determines the proportional band.	0.5K… **5.0K** … 10.0K (°C) 0.9K… **9.0K** … 18.0K (°F)
Integral time	This parameter determines the integral time.	0 … **240** … 255
Control value minimum limit	This parameter determines the output object's minimum control value.	**0%**, 5%, 10%, 15%, 20%, 25%, 30%
Control value maximum limit	This parameter determines the output object's maximum control value.	70%, 75%, 80%, 85%, 90%, %95, **100%**
Minimum oscillation of value to send	This parameter determines the minimum oscillation value for the output object to send a value.	**1** … 100
Periodic sending time	This parameter is used to periodically send the commands to the bus line.	**00:00:00** … 18:12:15
Control value requirement object	This parameter is used to send status information about the controller value of the additional cooling system.	**No** Yes

3.7.6. Thermostat - Heating & Cooling

Heating & Cooling mode is generally used when there are 2 different heating and cooling sources or only 1 source that has both heating and cooling ability together. If the heating/cooling sources are different, the command value object parameter should be selected as “2 separated objects”. However, if heating and cooling are obtained from the same source, the command value object parameter should be selected as “1 common object”. Additionally, in this mode, the distinction is made whether the switch-over between heating and cooling is to be affected automatically or in a controlled way through the communication object.

In the automatic switch-over option: for the heating, the controller will turn on the heating when the room temperature has fallen below a preset dead band limit. As soon as the room temperature is exceeding the heating setpoint, the control will turn off the heating in the heating & cooling mode. For the cooling, the controller will turn on the cooling system when the room temperature has exceeded a preset dead band limit. As soon as the room temperature is reaching above the cooling setpoint, the control will turn off the cooling system in the heating & cooling mode.

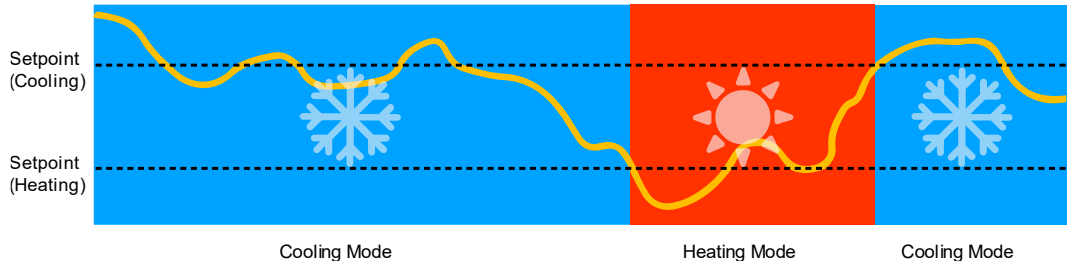


Fig. 53: Automatic Heating & Cooling Mode Switch

For a proper behaviour of the automatic switch function, the setpoint of the Cooling mode is required to be higher than that of the Heating mode.

In via communication object option: In this option, there is no dead band concept compared to the automatic option. The main difference between automatic and communication object options; the mode switch-over between modes is made manually.

3.7.6.1. Parameters List

In heating & cooling mode, cooling configurations and heating configurations can be made separately mentioned before. In this section, only extra parameters for this mode are described below.

PARAMETER	DESCRIPTION	VALUES
Thermostat mode	The thermostat mode's operating type is determined with this parameter.	**Master** Slave
Temperature source	This parameter determines whether the temperature source is external or internal.	**Temperature channel** KNX probe
Room controller mode	Room controller mode is determined with this parameter.	**Heating** Cooling Heating & Cooling
Command value object	The object types of temperature command values for heating and cooling mode are determined with this parameter.	**Common** Separated
Switch-over heating/cooling	This parameter determines how the heating/cooling transition is made.	Via object **Automatic**
Room controller mode after reset	This parameter determines the room controller mode after the device restarts.	**Previous mode** Heating Cooling

3.7.7. Thermostat - Set Points

Temperature setpoints for heating or cooling modes are configured in this section. The operation modes such as comfort, standby, night and frost protection of “heating”, “cooling” and “heating & cooling” modes can be separately specified from this section. The temperature setpoint value can be configured to send to the KNX bus line with 4 different settings such as “Disable”, “Periodically”, “On change” and “Periodically and on change”. Besides, how much the maximum bandwidth setting will be configured for that increasing or decreasing the temperature value manually can be determined. Moreover, it is possible to set which setpoint values will be used when there is a power failure.

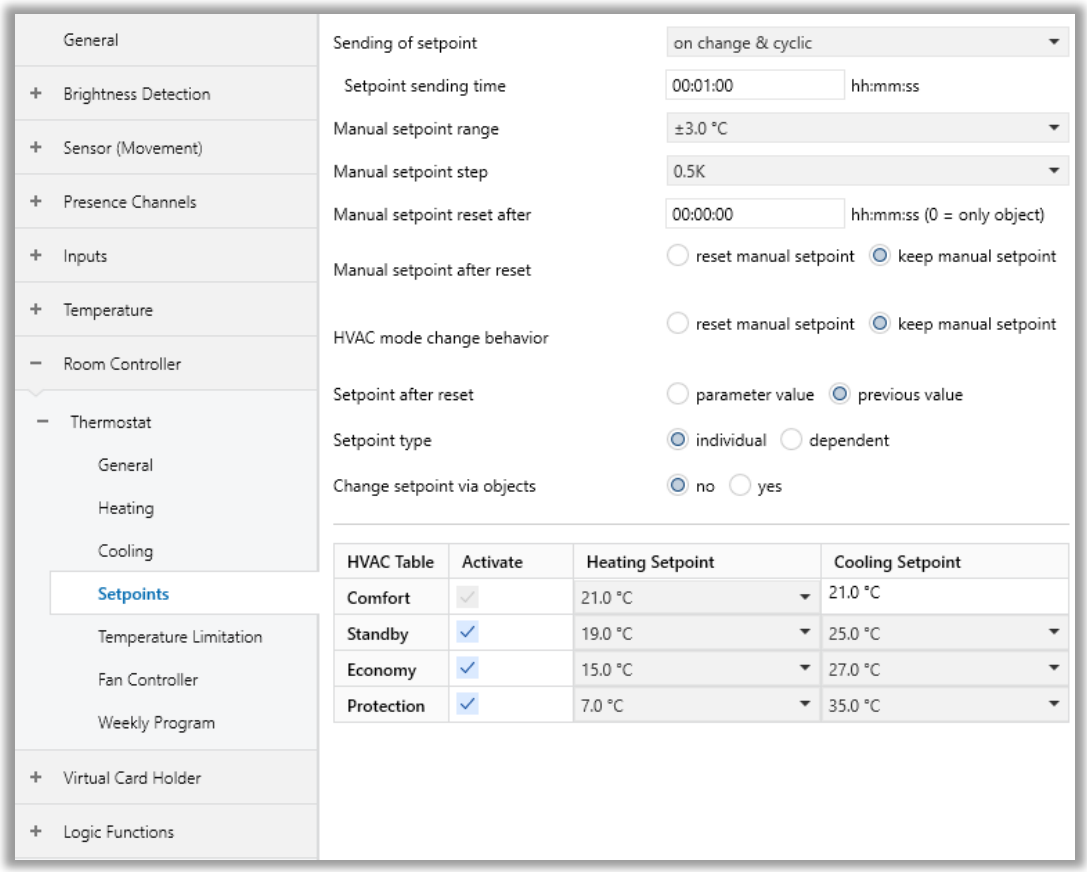


Fig. 54: Set Points Configuration Page

Note: If Heating/Cooling automatic mode is used HVAC mode setpoints must be in the range of manual setpoint. Otherwise, shifts in setpoints may occur in automatic heating-cooling transitions.

Note: Heating and Cooling setpoints limited with 10°C to 40°C for Comfort, Standby and Economy modes, 0°C to 15.5°C for frost protection mode and 25°C to 45°C for heat protection mode. User can change setpoint temperature bases with this ranges. If an attempt is made to apply a setpoint base other than the limits from the setpoint base objects, the limit value becomes valid.

3.7.7.1. Parameters List

PARAMETER	DESCRIPTION	VALUES
Sending of setpoint	This parameter allows sending the setpoint temperature value information. **On change:** The Temperature value information is sent when the setpoint temperature value changes by 1 K. **Periodically:** The Temperature value information is sent periodically. **Periodically and on change:** The Temperature value information is sent periodically or when the setpoint temperature value changed 1 K.	Disable **On change** Cyclic On change & cyclic
Setpoint sending time[1]	This parameter determines the time of the setpoint temperature value to be sent periodically.	
Manual setpoint range	This parameter configures the maximum and minimum limit values for the setpoint temperature value.	±1.0 ... **±3.0** ... ±10.0 (˚C) ±1.8 ... **±5.4** ... ±22.5 (˚F)
Manual setpoint step	This parameter configures the maximum and minimum limit values for the setpoint temperature value.	0.1K … **0.5K** … 3.5K (˚C) 0.18K … **0.9K** … 6.3K (˚F)
Manual setpoint reset after	This parameter determines the time of value to be sent setpoint reset after.	**00:00:00** … 18:12:15
Manual setpoint after reset	This parameter determines the behaviour of the manual setpoint's value after device reset. **Reset manual setpoint:** The manual setpoint is reset after device reset. **Keep manual setpoint:** The manual setpoint is continued after device reset.	Reset manual setpoint **Keep manual setpoint**
HVAC mode change behaviour	This parameter determines the behaviour of the manual setpoint's value after receiving the new set mode. **Reset manual setpoint:** The manual setpoint is reset after the new setting mode is received with this option.	Reset manual setpoint **Keep manual setpoint**

	Keep manual setpoint: The manual setpoint is continued after the new setting mode is received with this option.	
Setpoint after reset	This parameter determines the setpoint temperature after a reset for any reason, such as power failure.	Parameter value **Previous value**
Setpoint type	The desired temperature value can be controlled with individual or dependent setpoints by this parameter. If dependent mode is selected the setpoints of comfort and protect can be configured as individual setpoint. Standby and economy mode's setpoints can be configured as dependent setpoint. Even dependent mode is selected, all of the operation mode's setpoints can be change via object separately. So, if the comfort's setpoint is changed economy or standby's setpoints aren't updated according to comfort setpoint.	**Individual** Dependent
Change setpoint via objects	With this parameter, setpoint objects for all operation mode are visible.	**No** Yes
Comfort Mode Activate	This parameter is used to determine the activation of comfort mode. If this parameter is checked, comfort mode can be useable.	**Checked** Unchecked
Comfort Mode Heating Setpoint (˚C)	The desired temperature value for comfort mode is configured with this parameter.	10.0 … **21.0** … 40 (˚C) 50.0 … **69.8** … 104 (˚F)
Comfort Mode Cooling Setpoint (˚C)	The desired temperature value for comfort mode is configured with this parameter.	10.0 … **21.0** … 40 (˚C) 50.0 … **69.8** … 104 (˚F)
Standby Mode Activate	This parameter is used to determine the activation of standby mode. If this parameter is checked, standby mode can be useable.	**Checked** Unchecked
Standby Mode Heating Setpoint (˚C)	The desired temperature value of heating for standby mode is configured with this parameter.	10.0 … **19.0** … 40 (˚C) 50.0 … **66.2** … 104 (˚F)
Standby Mode Cooling Setpoint (˚C)	The desired temperature value for standby mode is configured with this parameter.	10.0 … **25.0** … 40 (˚C) 50.0 … **77.0** … 104 (˚F)

Economy Mode Activate	This parameter is used to determine the activation of economy mode. If this parameter is checked, economy mode can be useable.	**Checked** Unchecked
Economy Mode Heating Setpoint (˚C)	The desired temperature value of heating for economy mode is configured with this parameter.	10.0 … **15.0** … 40 (˚C) 50.0 … **59.0** … 104 (˚F)
Economy Mode Cooling Setpoint (˚C)	The desired temperature value of cooling for economy mode is configured with this parameter	10.0 … **27.0** … 40 (˚C) 50.0 … **80.6** … 104 (˚F)
Protection Mode Activate	This parameter is used to determine the activation of protection mode. If this parameter is checked, protection mode can be useable.	**Checked** Unchecked
Protection Mode Heating Setpoint (˚C)	The desired temperature value of heating for protection mode is configured with this parameter.	0.0 … **7.0** … 15.5 (˚C) 32.0…**44.6** … 59.9 (˚F)
Protection Mode Heating Setpoint (˚C)	The desired temperature value of cooling for protection mode is configured with this parameter	25.0…**35.0**…45.0 (˚C) 77.0…**95.0**…113.0 (˚F)

*1 This parameter is visible when the parameter “Sending of setpoint” is set to “Periodically” or “periodically and on change”.

3.7.8. Thermostat – Temperature Limitation

Using the limit temperature, the controller's control value for this stage can be set to 0 on reaching a parameterized temperature. In this way, exceeding (heating) or dropping below (cooling) this temperature can be prevented. An example of the usage of the limit temperature is floor heating, where exceeding a specific temperature must be prevented to protect the material of the floor.

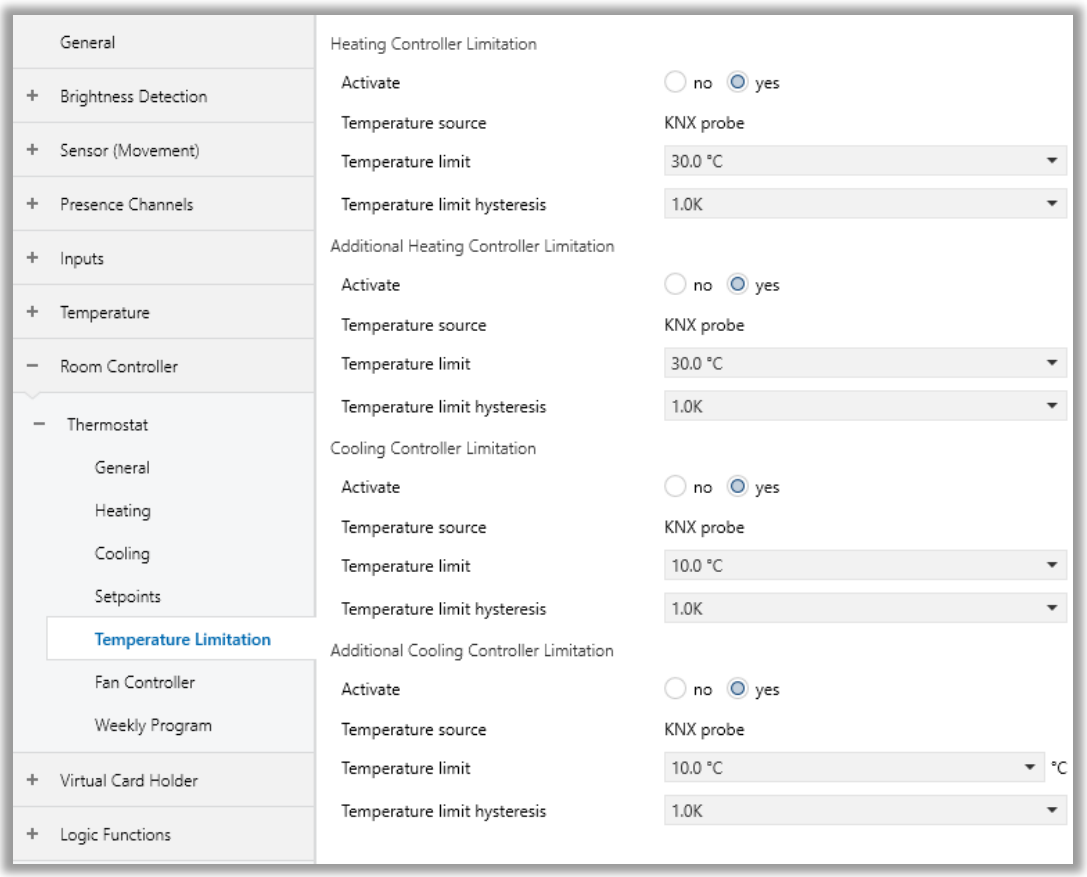


Fig. 55: Temperature Limitation Configuration Page

3.7.8.1. Parameters List

PARAMETER	DESCRIPTION	VALUES
Heating Controller Limitation Activate	This parameter is used to activate limit temperature for heating controller.	**No** Yes
Heating Controller Limitation Activate: Yes		
Temperature Source	This parameter is used to determine the source of temperature for limitation function. It is not suitable to use the same temperature sensor for the measurement of the room temperature and for the measurement of the limit temperature.	**KNX probe**
Temperature Limit	This parameter is used to determine the limit temperature that is not allowed to be exceeded (heating). If the temperature reaches this value, the control value is immediately set to 0.	1… **30** …60 (°C) 32… **86** …140 (°F)
Temperature Limit Hysteresis	This parameter is used to determine the hysteresis on the limit temperature specifies the value by which the limit temperature must be dropped below again (heating) before the controller becomes active again.	0.5K … **1K** … 5K (°C) 0.9K … **1.8K** … 9K (°F)
Integral on temperature limitation[1]	This parameter is used to decide what is to happen to the I-proportion on reaching the limit temperature. **Freeze:** Keeps the current accumulated error caused by I-proportion. **Reset:** Resets the accumulated error caused by I-proportion.	**Freeze** Reset
Additional Heating Controller Limitation Activate	This parameter is used to activate limit temperature for additional heating controller.	**No** Yes
Additional Heating Controller Limitation Activate: Yes		
Temperature Source	This parameter is used to determine the source of temperature for limitation function. It is not suitable to use the same temperature sensor for the measurement of the room temperature and for the measurement of the limit temperature.	**KNX probe**
Temperature Limit	This parameter is used to determine the hysteresis on the limit temperature specifies the value by which the limit temperature must be dropped below again	1… **30** …60 (°C) 32… **86** …140 (°F)

	(heating) before the controller becomes active again.	
Temperature Limit Hysteresis	This parameter is used to determine the hysteresis on the limit temperature specifies the value by which the limit temperature must be dropped below again (heating) before the controller becomes active again.	0.5K … **1K** … 5K (°C) 0.9K … **1.8K** … 9K (°F)
Integral on temperature limitation[2]	This parameter is used to decide what is to happen to the I-proportion on reaching the limit temperature. **Freeze:** Keeps the current accumulated error caused by I-proportion. **Reset:** Resets the accumulated error caused by I-proportion.	**Freeze** Reset
Cooling Controller Limitation Activate	This parameter is used to activate limit temperature for cooling controller.	**No** Yes
Cooling Controller Limitation Activate: Yes		
Temperature Source	This parameter is used to determine the source of temperature for limitation function. It is not suitable to use the same temperature sensor for the measurement of the room temperature and for the measurement of the limit temperature.	**KNX probe**
Temperature Limit	This parameter is used to determine the limit temperature that is not allowed to be dropped below (cooling). If the temperature reaches this value, the control value is immediately set to 0.	1… **10** …60 (°C) 32… **50** …140 (°F)
Temperature Limit Hysteresis	This parameter is used to determine the hysteresis on the limit temperature specifies the value by which the limit temperature must be exceeded (cooling) before the controller becomes active again.	0.5K … **1K** … 5K (°C) 0.9K … **1.8K** … 9K (°F)
Integral on temperature limitation[3]	This parameter is used to decide what is to happen to the I-proportion on reaching the limit temperature. **Freeze:** Keeps the current accumulated error caused by I-proportion. **Reset:** Resets the accumulated error caused by I-proportion.	**Freeze** Reset
Additional Cooling Controller Limitation Activate	This parameter is used to activate limit temperature for additional cooling controller.	**No** Yes

Additional Cooling Controller Limitation Activate: Yes		
Temperature Source	This parameter is used to determine the source of temperature for limitation function. It is not suitable to use the same temperature sensor for the measurement of the room temperature and for the measurement of the limit temperature.	Internal temperature **Temperature object** Calculation 1…6
Temperature Limit	This parameter is used to determine the limit temperature that is not allowed to be dropped below (cooling). If the temperature reaches this value, the control value is immediately set to 0.	1… **10** …60 (°C) 32… **50** …140 (°F)
Temperature Limit Hysteresis	This parameter is used to determine the hysteresis on the limit temperature specifies the value by which the limit temperature must be exceeded (cooling) before the controller becomes active again.	0.5K … **1K** … 5K (°C) 0.9K … **1.8K** … 9K (°F)
Integral on temperature limitation[4]	This parameter is used to decide what is to happen to the I-proportion on reaching the limit temperature. **Freeze:** Keeps the current accumulated error caused by I-proportion. **Reset:** Resets the accumulated error caused by I-proportion.	**Freeze** Reset

[*1] This parameter is visible when heating controller type is set to “PWM” or “Continuous”.

[*2] This parameter is visible when additional heating controller type is set to “PWM” or “Continuous”.

[*3] This parameter is visible when cooling controller type is set to “PWM” or “Continuous”.

[*4] This parameter is visible when additional cooling controller type is set to “PWM” or “Continuous”.

3.7.9. Thermostat – Fan Controller

If the parameter “Fan control used for room control” is set to “Enabled” from the “General” parameter page, the configuration page that is related to fan controller is now opened as “Fan Controller” under the “Room Controller” parameter page instead of the “LCD” parameter page.

The configuration settings in this section are configured such as, the selection of the fan speed level of the device to be used, the fan speed transitions in regard to the percentage value to be changed, the fan controller type selection, delay time for starting and delay time for stopping the fan and other arrangements related to fan control.

3.7.9.1. Fan 2-Points Control

This type of fan control is similar to the 2 points control with hysteresis: the fan speed is activated/deactivated according to the difference between the desired temperature and the measured temperature. The relevant difference with the 2 points algorithms with hysteresis is that, in this case, there is not a single stage on which the hysteresis loop is executed, by setting the thresholds for switching on and off of the speed, but five stages may exist.

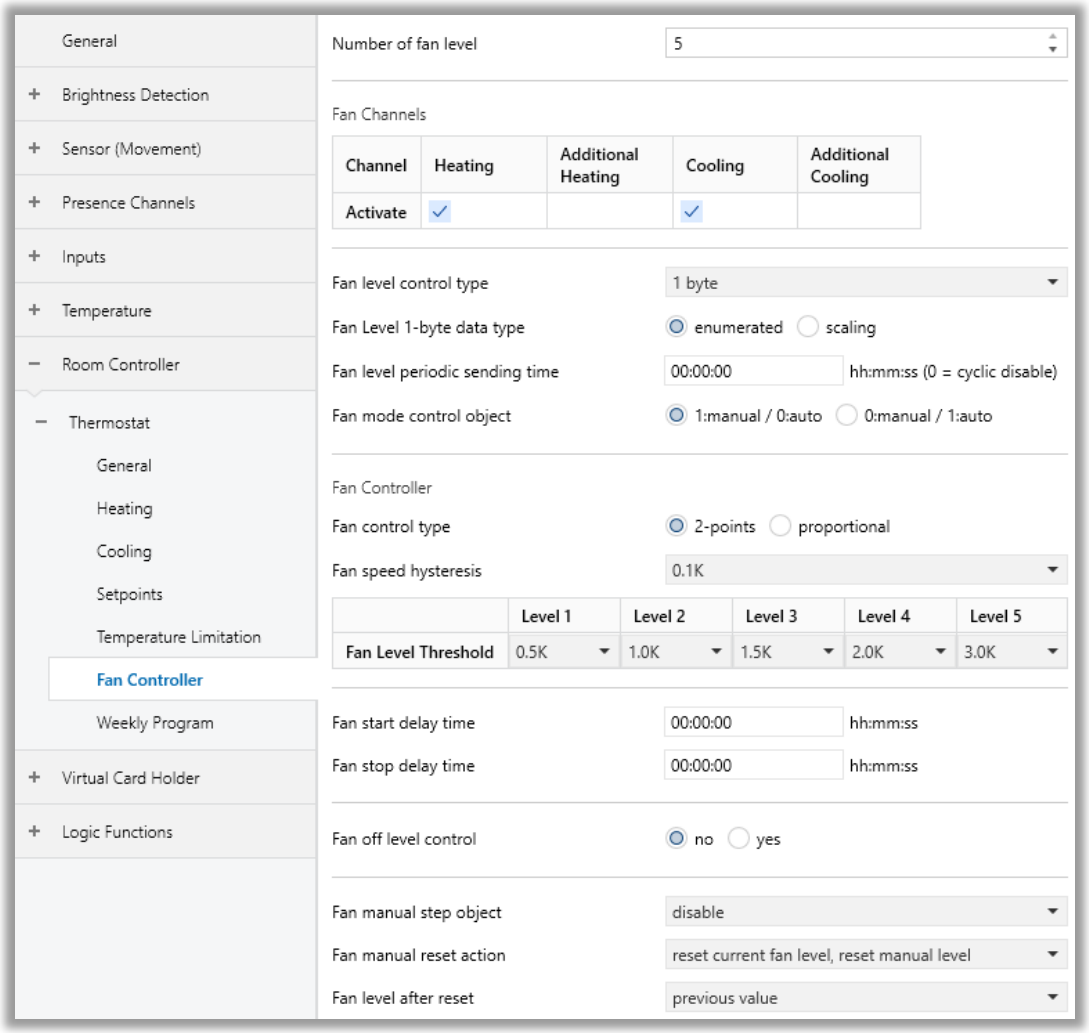


Fig. 56: Fan Controller 2-Points Control Configuration Page

This means that a speed level corresponds to each stage and when the difference between the measured temperature and the desired temperature causes the activation of a further speed.

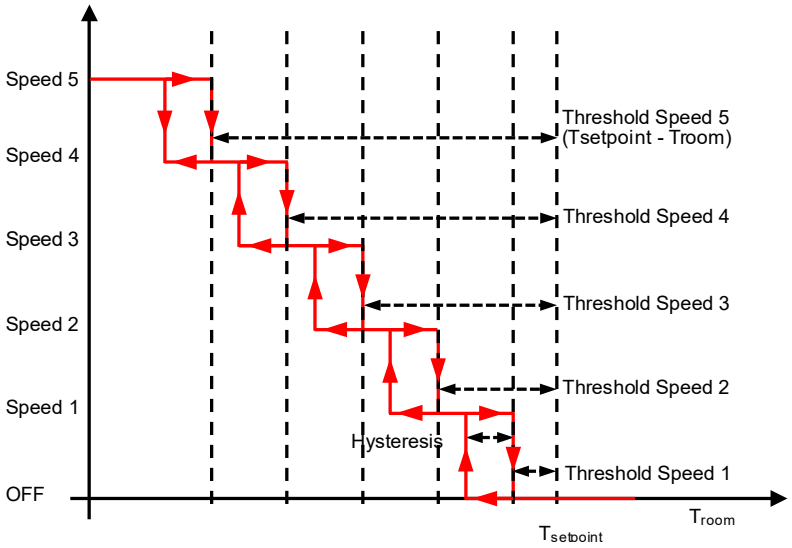


Fig. 57: Fan Controller 2-Points Control Cycle for Heating

The figure in the above graph refers to the speed control of the fan with three operating stages as regards the heating. Looking at the graph, it has to be noted that for each stage there is a hysteresis loop, as well as at any speed are assigned two thresholds which determine the activation and deactivation. The thresholds are determined by the values set in the application program and can be summarized as follows:

- Speed 1 (1st stage) – The speed is turned ON when the value of the room temperature is lower than the value (T_{set} – Threshold Speed1 – hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} – Threshold Speed1); the first speed is also switched OFF when a higher speed must be turned ON. The default value for the parameter Threshold Speed1 = 0 K.
- Speed 2 (2nd stage) – The speed is turned ON when the value of the room temperature is lower than the value (T_{set} – Threshold Speed2 – hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} – Threshold Speed 2); the second speed is also switched OFF when Speed 3 must be turned ON.
- Speed 3 (3rd stage) – The speed is turned ON when the value of the room temperature is lower than the value (T_{set} – Threshold Speed3 – hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} – Threshold Speed 3).
- Speed 4 (4rd stage) – The speed is turned ON when the value of the room temperature is lower than the value (T_{set} – Threshold Speed 4 – hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} – Threshold Speed 4).
- Speed 5 (5rd stage) – The speed is turned ON when the value of the room temperature is lower than the value (T_{set} – Threshold Speed 5 – hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} – Threshold Speed 5).

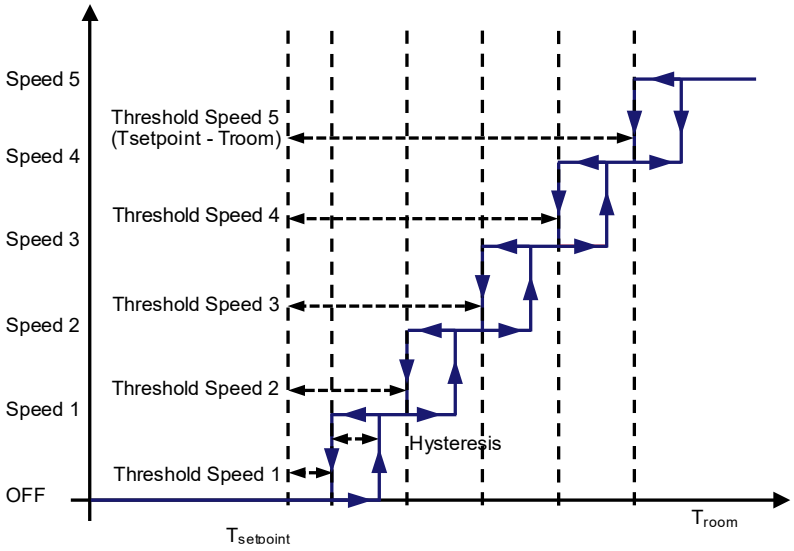


Fig. 58: Fan Controller 2-Points Control Cycle for Cooling

The figure in the above graph refers to the speed control of the fan with three operating stages as regards the cooling. Looking at the graph, it has to be noted that for each stage there is a hysteresis loop, as well as at any speed are assigned two thresholds which determine the activation and deactivation. The thresholds are determined by the values set in the application program and can be summarized as follows:

- Speed 1 (1st stage) – The speed is turned ON when the value of the room temperature is higher than the value (T_{set} + Threshold Speed1 + hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} + Threshold Speed1); the first speed is also switched OFF when a higher speed must be turned ON. The default value for the parameter Threshold Speed1 = 0 K.
- Speed 2 (2nd stage) – The speed is turned ON when the value of the room temperature is higher than the value (T_{set} + Threshold Speed2 + hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} + Threshold Speed2); the second speed is also switched OFF when Speed 3 must be turned ON.
- Speed 3 (3rd stage) – The speed is turned ON when the value of the room temperature is higher than the value (T_{set} + Threshold Speed3 + hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} + Threshold Speed3).
- Speed 4 (4rd stage) – The speed is turned ON when the value of the room temperature is higher than the value (T_{set} + Threshold Speed 4 + hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} + Threshold Speed 4)
- Speed 5 (5rd stage) – The speed is turned ON when the value of the room temperature is higher than the value (T_{set} + Threshold Speed 5 + hysteresis) and turned OFF when the room temperature value reaches the value (T_{set} + Threshold Speed 5)

If “Fan level 1-byte data type” is selected as “Enumerated”, what fan speed calculated according to above graph, is sent over 1 byte object. For example; If fan speed was calculated as speed 2, 2 is sent over fan speed object.

If “Fan level 1-byte data type” is selected as “Scaling”, fan level scaling value is sent according to fan level limits table. For example; if “Fan level 2 threshold value” is 40% and fan speed was calculated as speed 2, %40 value is sent over fan speed object.

3.7.9.2. Fan Proportional Control

Proportional – Integral control (PI control) is explained by the relationship shown below:

$$\boldsymbol{control\ variable(t)\ =\ Kp\ \times\ error(t)}$$

whereby:

$$error(t)\ =\ (Setpoint\ -\ Measured\ temperature)\ in\ heating$$

$$error(t)\ =\ (Measured\ temperature\ -\ Setpoint)\ in\ cooling$$

$$Kp\ =\ proportional\ constant$$

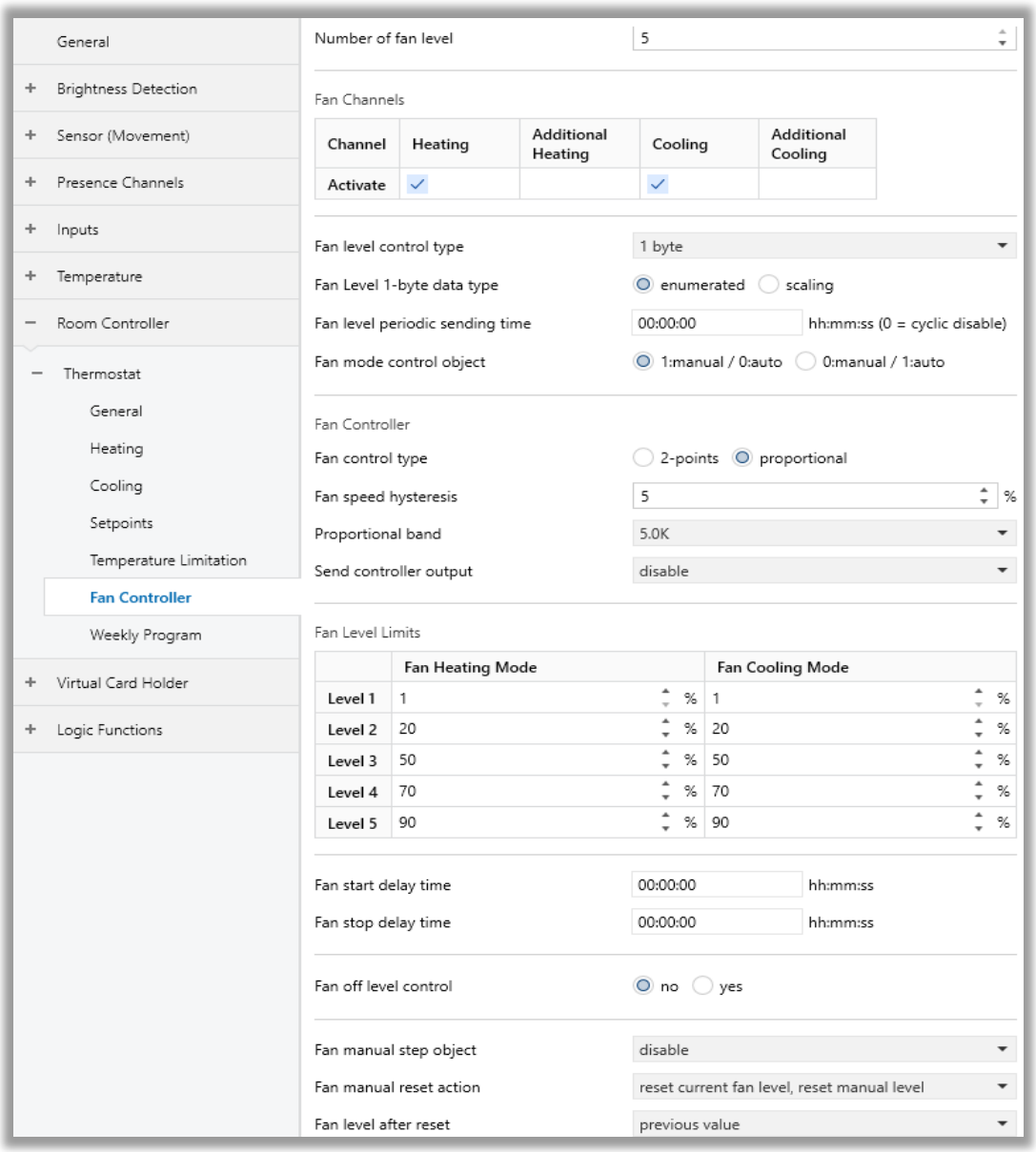


Fig. 59: Fan Controller Proportional Control Configuration Page

The control variable contains proportional (Kp) constants to eliminate errors. In practice, intuitively generated values are generally used.

$$\boldsymbol{Proportional\ band\ BP\ [K]\ =\ 100\ /\ Kp}$$

The proportional band is the error value that determines the maximum deflection output as 100%.

For example, a regulator with a proportional band of 5 K provides a 100% control output when the Setpoint = 20°C and the measured temperature is ≤ 15°C in heating; in the cooling conduction mode, it provides a 100% control output when the Setpoint = 24°C and the measured temperature is ≥ 29°C. As shown in the figure, a regulator with a small proportional band tends to provide higher values of the control variable for small errors than a regulator with a higher proportional band.

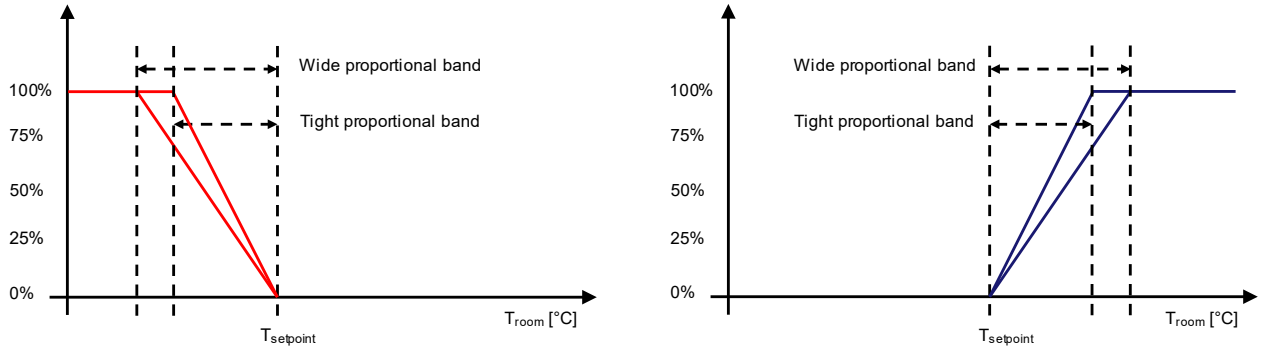


Fig. 60: Fan Controller Proportional Control

The control output is compared to the limit value of fan speed. The fan speed is assigned according to whether the limit values is exceeded or below.

For example, fan level limits are assigned subsequently as 1, 20, 50, 70 and 90 for heating or cooling mode. Assume that the current working mode is Heating and the fan proportional controller generates %65 control value. The control value is compared to fan level limits and as seen the %65 control value is higher than the limits value of levels 1, 2 and 3. So, the fan level is assigned to Level 3.

Note: Fan controller have feedback objects for syncing with controlled device. These objects are not for changing fan level but showing actual value of controlled device. For changing fan level manually manual fan level objects should be used.

3.7.9.3. Parameters List

PARAMETER	DESCRIPTION	VALUES
Number of fan level	The number of fan levels is determined with this parameter.	1…**5**
Channel Heating Activate	This parameter allows the fan controls to work with the heating system. If the heating system is checked, the fan can't connect to the additional heating system at the same time.	**Checked** Unchecked
Channel Additional Heating Activate	This parameter allows the fan controls to work with the additional heating system. If the additional heating system is checked, the fan can't connect to the heating system at the same time.	Checked **Unchecked**
Channel Cooling Activate	This parameter allows the fan controls to work with the cooling system. If the cooling system is checked, the fan can't connect to the additional cooling system at the same time.	**Checked** Unchecked
Channel Additional Cooling Activate	This parameter allows the fan controls to work with the cooling system. If the additional cooling system is checked, the fan can't connect to the cooling system at the same time.	Checked **Unchecked**
Fan level control object	This parameter allows the control of the fan speed with 1-bit individual or 1 byte or 1 bit /1 byte object.	1 bit 1 byte **1 bit / 1 byte**
-> Fan level control data type[1]	This parameter is used to determine with which data type the fan level is sent to the bus. **Enumerated:** 0~5 value is sent. **Scaling:** The percentage equivalent of the fan level value in the fan level limits table.	**Enumerated** Scaling
Fan level periodic sending time	This parameter determines the time of the fan level value to be sent periodically.	**00:00:00**…18:12:15
Fan mode control object	Manual or automatic fan speed control is selected with this parameter.	1: manual / 0: auto **0: manual / 1: auto**
Fan control type	This parameter determines the fan controller type.	**2-points** Proportional
-> Fan speed hysteresis[2]	This parameter determines the fan speed hysteresis value at which switchover to the next fan speed occurs. Using hysteresis avoids continual switching between the fan speeds caused by fluctuating input signals around the limit value.	Values depend on fan controller type

-> Fan Level X Threshold[2]	This parameter determines the fan level X threshold value.	0.5K…5.0K (°C) 0.9K…18.0K (°F)
-> Proportional band[3]	This parameter determines the proportional band of the fan controller.	0.5K… **5K** …10.0K (°C) 0.9K… 9**K** …18.0K (°F)
Fan Heating Mode Level [1…5]	The lower limit value of the 1…5 speed is determined with this parameter.	1…100
Fan Cooling Mode Level X	The lower limit value of the 1…5 speed is determined with this parameter.	1…100
Fan start delay time	This parameter is used to determine the delay time for switching to a higher fan speed than zero.	**00:00:00**…18:12:15
Fan stop delay time	This parameter is used to determine the delay time for switching to zero fan speed.	**00:00:00**…18:12:15
Fan off level control	This parameter is used to enable fan off level control.	**No** Yes
-> Fan off level[4]	This parameter determines the speed of the fan off state.	Values depend on number of fan level.
Fan manual step object	This parameter allows the control of the fan speed with 1 – bit object	**Disable** Increase/decrease (1.007) Up/down (1.008)
Fan manual reset action	This parameter is used to determine what the action is after the value of controller that is connected to fan, is zero in fan manual mode. **No action:** Do nothing, continue to work. **Reset current fan level, hold manual level:** Current manual fan level resets but the previous manual level saves in memory. When the controller value is higher than zero again or manual fan level is changed with the object or thermostat extension of the push button, the manual fan level begins with the value in memory. **Reset current fan level, reset manual level:** Manual fan levels that are current and saved in memory, reset.	No action **Reset current fan level, hold manual level** Reset current fan level, reset manual level
Fan level after reset	The desired fan level after a power failure is determined with this object.	**Previous value** Off Level 1…5 Auto

[*1] This parameter is visible when the parameter “Fan level control object” is set to “1 byte” or “1 bit / 1 byte”.
[*2] This parameter is visible when the parameter “Fan control type” is set to “2-points”.
[*3] This parameter is visible when the parameter “Fan control type” is set to “Proportional”.
[*4] This parameter is visible when the parameter “Fan off level control” is set to “Yes”.

3.7.10. Thermostat – Weekly Program

Weekly Thermostat Program can be configured over the device. The weekly program works with if HVAC mode is Auto. If HVAC mode is set over object as Auto but the “Thermostat Time” object hasn’t been received yet and until the “Thermostat Time” object is received, weekly program doesn’t work. During the weekly program runs, the users can change the HVAC mode anytime.

If “Weekly program” parameter is selected as “enable” and “Thermostat Time” object was received, thermostat runs according to weekly program table. If weekly program is active, but any time zone isn’t configured, Auto HVAC mode is ended and the HVAC mode switches Comfort mode.

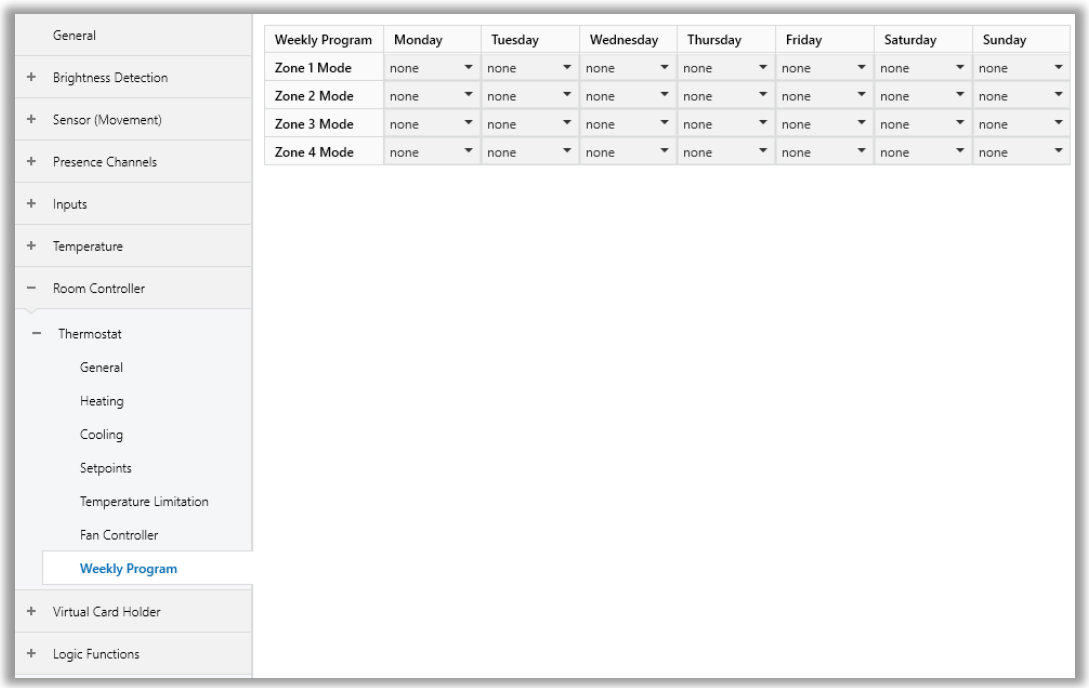


Fig. 61: Weekly Program Configuration Page

3.7.10.1. Parameters List

PARAMETER	DESCRIPTION	VALUES
Zone X Mode	This parameter is used to determine which HVAC mode will be active according to selected day, hour and minute.	**None** Comfort Standby Economy Protection
=> Zone X Hour	This parameter is used to determine the hour that the HVAC mode will be active.	**0** … 23
=> Zone X Minute	This parameter is used to determine the minute that the HVAC mode will be active.	**0** … 59
Auto switch-over HVAC modes	If this parameter is enabled, HVAC mode is changed according to the weekly program table.	**Disable** Enable

3.8. Virtual Card Holder

In this section, the 'virtual card holder' feature, which is intended to be provided by the Multi Presence Detector and which is the equivalent of the card holder systems used in hotels, is explained. In addition, scenarios such as 'welcome scenario' customers when the relevant hotel room is taken and 'good bye scenario' scenarios, when they leave the room, can also be realized with the 'virtual card holder' feature. With this feature, you can remove the Holder for access control. This logical module provides a set of parameters and communication objects that, suitably configured, allow you to set up whether or not a person is occupying the room.

3.8.1. Virtual Card Holder – General

In this section, the general parameters of the 'virtual card holder' feature, which is the equivalent of the card holder systems used especially in hotels, are explained.

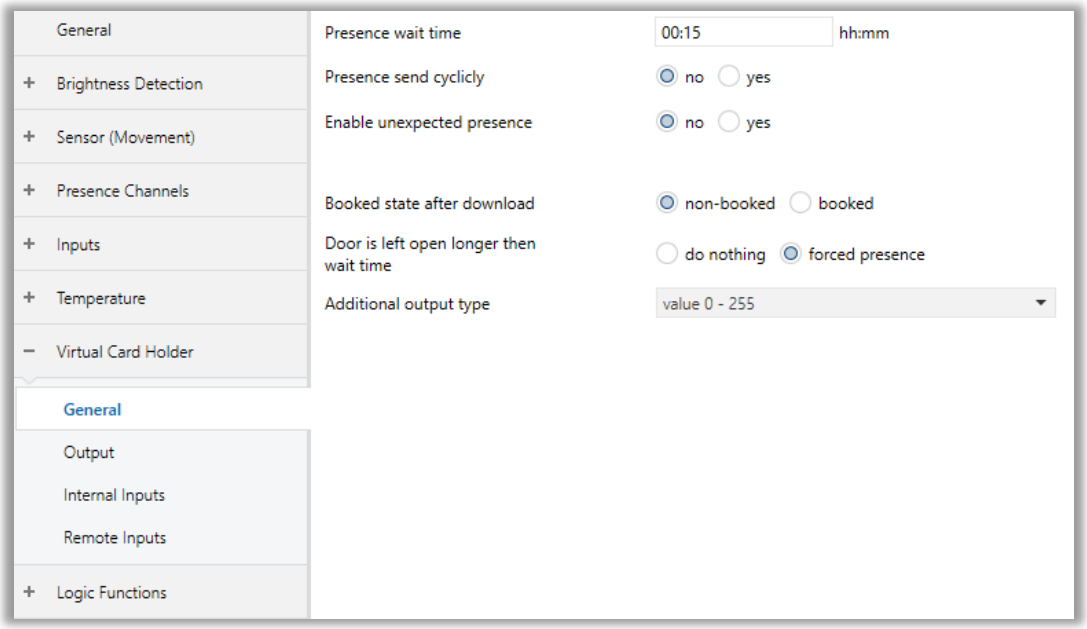


Fig. 62: Virtual Card Holder – General Configuration Page

3.8.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Presence wait time	This parameter is used to set the “wait time” for the virtual holder module. The “wait time” is the time triggered by the opening and consequent closing of the door. During this time the room is in “wait mode and after this can go in “occupied” or “non occupied” status.	00:05…00:15…06:00
Presence send cyclically	This parameter is used to determine the sending the presence status or not.	**No** Yes
-> Cyclical Send time	This parameter is used to determine the cyclical sending time of the virtual holder presence output.	1…**5**…60
Enable unexpected presence	This parameter enables the sensor to sense the unexpected presence	**no** yes
->Send welcome on unexpected presence	This parameter is used to define the behaviour when the virtual holder module detects a presence inside the room and is in “not occupied” status 'unexpected presence'. It’s possible to send or not the welcome event	**Contact closed with door closed** Contact closed with door open
->Enable Unexpected Presence Object	This parameter creates the “unexpected presence” object, which gives the value “1” when the sensor senses the unexpected presence	**No** Yes
Booked state after download	This parameter is used to set the initial value for the object virtual holder room booked	**Non-booked** Booked
Door is left open longer than wait time	When the parameter is set to the forced presence and if the door is kept open longer than the “Presence waits time”, the virtual cardholder will automatically act as if it has detected presence.	Do Nothing **Forced Presence**
Additional output type	This parameter is used to enable an additional object to transmit on the bus a command linked to presence or absence events	**Value 0 – 255** Value 0 – 100% scene

3.8.2. Virtual Card Holder – Output

In this section, the output parameters of the 'virtual card holder' feature, which is the equivalent of the cardholder systems used especially in hotels, are explained.

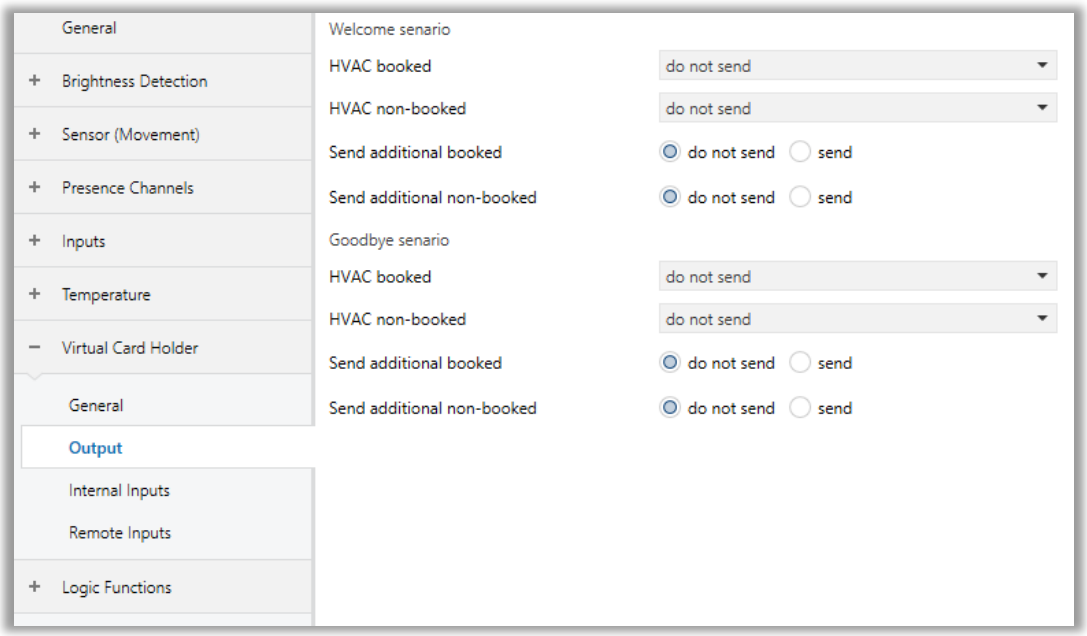


Fig. 63: Virtual Card Holder – Output Configuration Page

3.8.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Welcome scenario		
HVAC booked	This parameter is used to determine the HVAC status of the welcome scenario when the room is booked.	**Do not send** Comfort Standby Economy Building Protection
HVAC Non-booked	This parameter is used to determine the HVAC status of the welcome scenario when the room is non-booked.	**Do not send** Comfort Standby Economy Building Protection
Send additional booked	This parameter is used to determine sending or not sending a value within the welcome scenario to the bus while the room is additionally booked.	**Do not send** send
-> Value additional booked	This parameter is used to determine the sending value within the welcome scenario if the room is additionally booked.	**%0**…%100
Send additional non-booked	This parameter is used to determine sending or not sending a value within the welcome scenario to the bus while the room is additional non-booked.	**Do not send** send
-> Value additional non-booked	This parameter is used to determine the sending value within the welcome scenario if the room is additional non-booked.	**%0**…%100
Goodbye scenario		
HVAC booked	This parameter is used to determine the HVAC status of the goodbye scenario when the room is booked.	**Do not send** Comfort Standby Economy Building Protection
HVAC Non-booked	This parameter is used to determine the HVAC status of the goodbye scenario when the room is non-booked.	**Do not send** Comfort Standby Economy Building Protection

Send additional booked	This parameter is used to determine sending or not sending a value within the goodbye scenario to the bus while the room is additionally booked.	**Do not send** send
-> Value additional booked	This parameter is used to determine the sending value within the goodbye scenario if the room is additionally booked.	**%0**…%100
Send additional non-booked	This parameter is used to determine sending or not sending a value within the goodbye scenario to the bus while the room is additional non-booked.	**Do not send** send
-> Value additional non-booked	This parameter is used to determine the sending value within the goodbye scenario if the room is additional non-booked.	**%0**…%100

3.8.3. Virtual Card Holder – Internal Inputs

In this section, the internal input parameters used to realize the 'virtual card holder' feature, which is the equivalent of the cardholder systems used in hotels, are mentioned. Internal inputs are the inputs on the Multi Presence Detector.

Note: At least one of the input types of total inputs must be selected as the door for sake of the functionality.

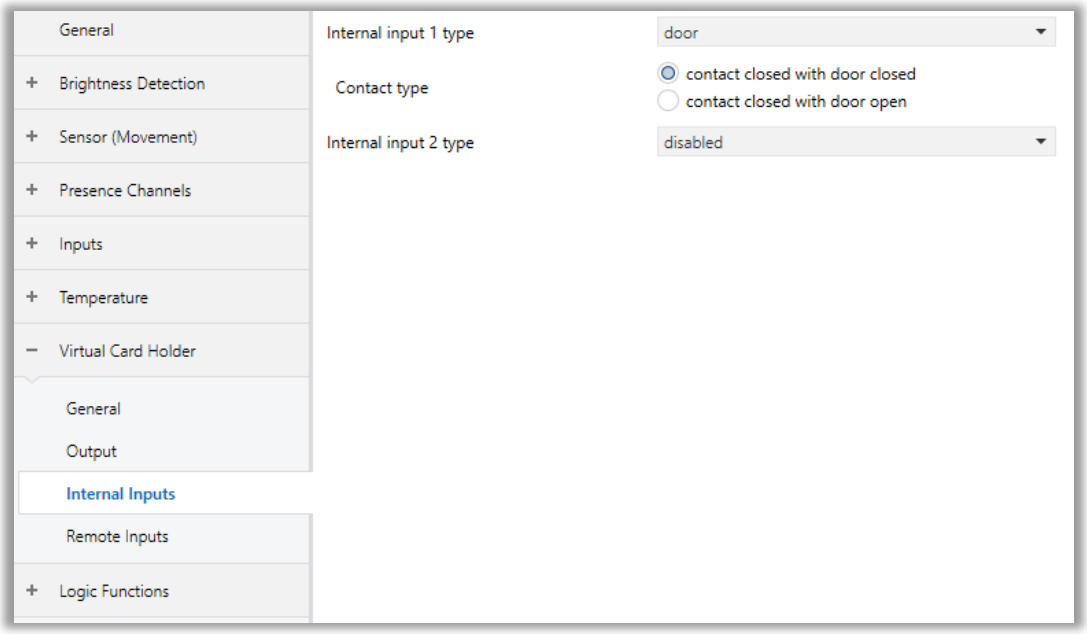


Fig. 64: Virtual Card Holder – Internal Inputs Configuration Page

3.8.3.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Internal input type 1	This parameter is used to connect the internal inputs of the Multi Presence Detector to the door or conventional presence sensor. **Door** : Internal input is connected to a door contact **Presence Sensor** : Internal input is connected to a conventional presence detector.	Disabled **Door** Presence Sensor
-> Contact type	This parameter is used to determine the contact type that will the door contact is closed with door open or close.	**Contact closed with door closed** Contact closed with door open
-> Contact type	This parameter is used to determine the connected contact type as normally closed or normally open.	**Normally closed** Normally open
Internal input type 2	This parameter is used to connect the internal inputs of the Multi Presence Detector to the door or conventional presence sensor. **Door**: Internal input is connected to a door contact **Presence Sensor**: Internal input is connected to a conventional presence detector.	**Disabled** Door Presence Sensor
-> Contact type	This parameter is used to determine the contact type that will the door contact is closed with door open or close.	**Contact closed with door closed** Contact closed with door open
-> Contact type	This parameter is used to determine the connected contact type as normally closed or normally open.	**Normally closed** Normally open

3.8.4. Virtual Card Holder – Remote Input

In this section, the remote input parameters used to realize the 'virtual card holder' feature, which is the equivalent of the cardholder systems used in hotels, are mentioned. When you need to consider 2 adjacent rooms as separated or joined as if they were a single room (suite room) you can use remote inputs. To handle this situation both "Virtual Holder" modules of the 2 single rooms must be activated and configured. The principle is to connect the sensors (door or presence) of room 1 to room 2 and vice versa; however, the "Subordinate to Remote Inputs Enable" parameter for every single Remote Input should be properly configured.

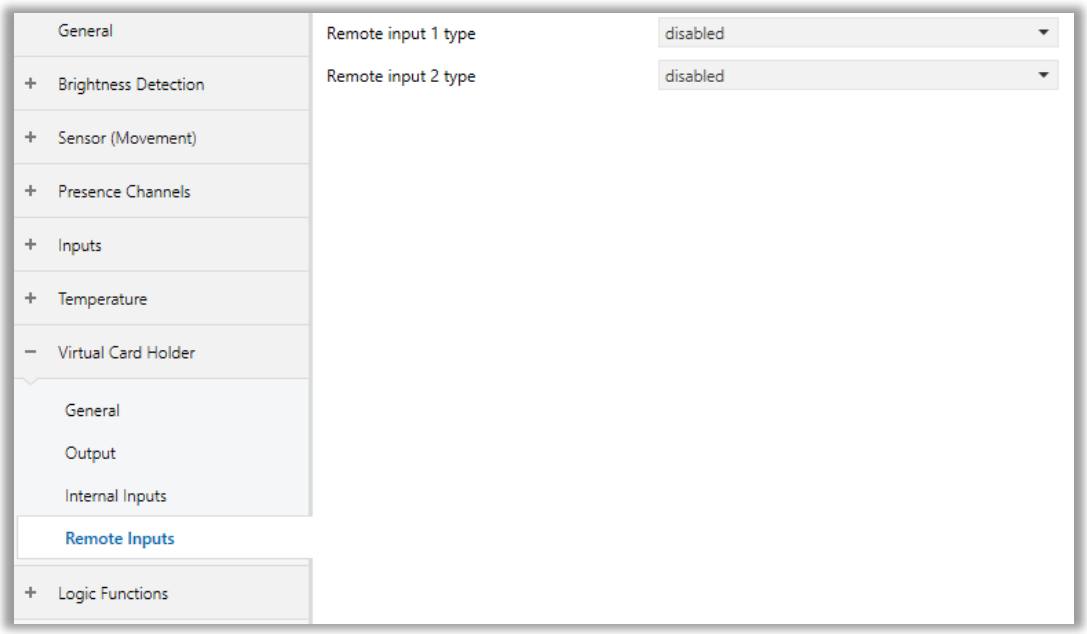


Fig. 65: Virtual Card Holder – Remote Input Configuration Page

3.8.4.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Remote input type 1	This parameter is used to enable or disable the remote input type 1. If it is selected as disabled, there will be no input 1. **Door:** The remote input is connected to door contact. **Presence Sensor:** The remote input is connected to the presence sensor.	**Disabled** Door Presence Sensor
-> Subordinate to remote input enables	This parameter is used to determine the subordinate or not subordinate relation of the related room. For sensors located in Room 1, this parameter must be set to "not subordinate" on room 1 "Virtual Holder" while should be “subordinated” to room 2 Virtual Holder The same principle must be applied to room 2 sensors that are "subordinate" only for the connections to room 1.	**Not subordinate** Subordinate
Remote input type 2	This parameter is used to enable or disable the remote input type 2. If it is selected as disabled, there will be no input 2. **Door:** The remote input is connected to door contact. **Presence Sensor:** The remote input is connected to the presence sensor.	**Disabled** Door Presence Sensor
-> Subordinate to remote input enables	This parameter is used to determine the subordinate or not subordinate relation of the related room. For sensors located in Room 1, this parameter must be set to "not subordinate" on room 1 "Virtual Holder" while should be “subordinated” to room 2 Virtual Holder The same principle must be applied to room 2 sensors that are "subordinate" only for the connections to room 1.	**Not subordinate** Subordinate

3.9. Logic Functions

This section describes the logical function modules of the Interra Multi Presence Detector. With the logical function blocks on the Multi Presence Detector, a logical expression can be created with the ambient temperature, the brightness level of the environment, whether there is a presence detection in the environment, the data coming through the local digital inputs or external inputs, and various 'TRUE' or 'FALSE' results can be obtained. actions can be taken and scenarios can be triggered.

3.9.1. Logic Functions – General

This section describes the general parameters of the logical association module of the Interra Multi Presence Detector. Parameters must be configured separately for each logic block.

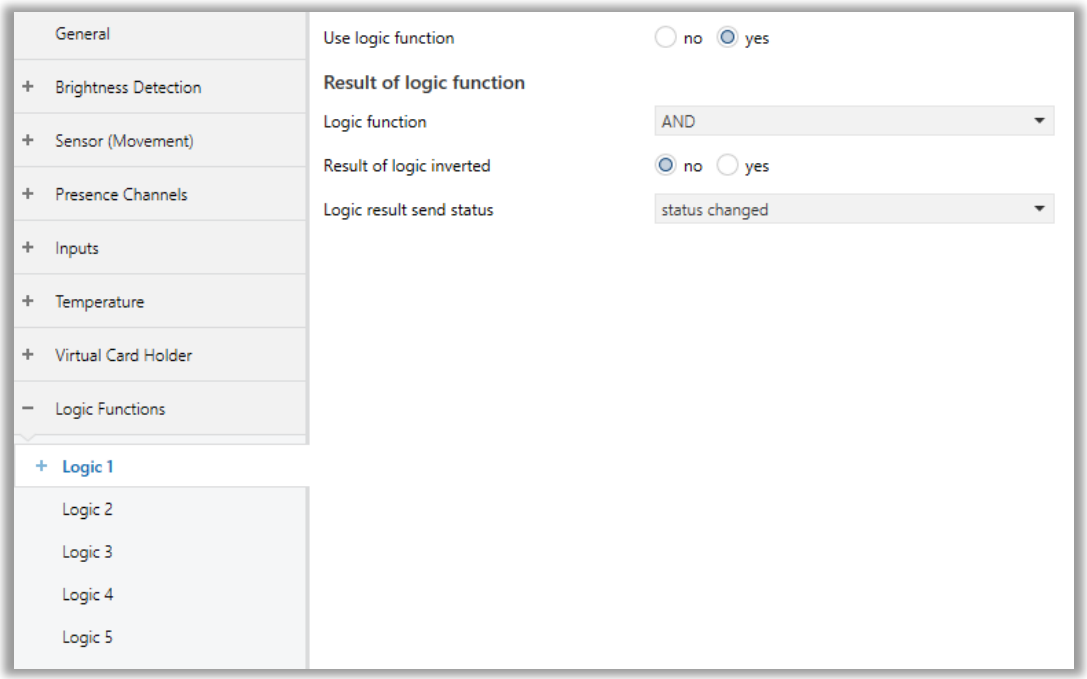


Fig. 66: Logic Functions – General Configuration Page

3.9.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Use Logic Function	This parameter is used to enable or disable the related logic function gate.	**No** Yes
Logic Function	This parameter is used to determine the logical relation of the parameterized logic inputs. **AND:** All inputs are put into the 'AND' operation. **OR:** All inputs are put into the 'OR' operation. **XOR:** All inputs are put into the 'XOR' operation.	**AND** OR XOR
Result of Logic Inverted	This parameter is used to invert or not invert the calculated logic function block. If it is selected as yes for example, when the logic function gate output is 'TRUE', the output will be 'FALSE'. Vice versa also applies.	**No** Yes
Logic result send status	This parameter is used to determine the logic function block result sending status to the KNX bus.	**Status changed** Status is TRUE Status is FALSE Status changed and periodically Status is TRUE periodically Status is FALSE periodically

3.9.2. Logic Functions - Input

This section describes the input parameters of the logical association module of the Interra Multi Presence Detector. Parameters must be configured separately for each logic block.

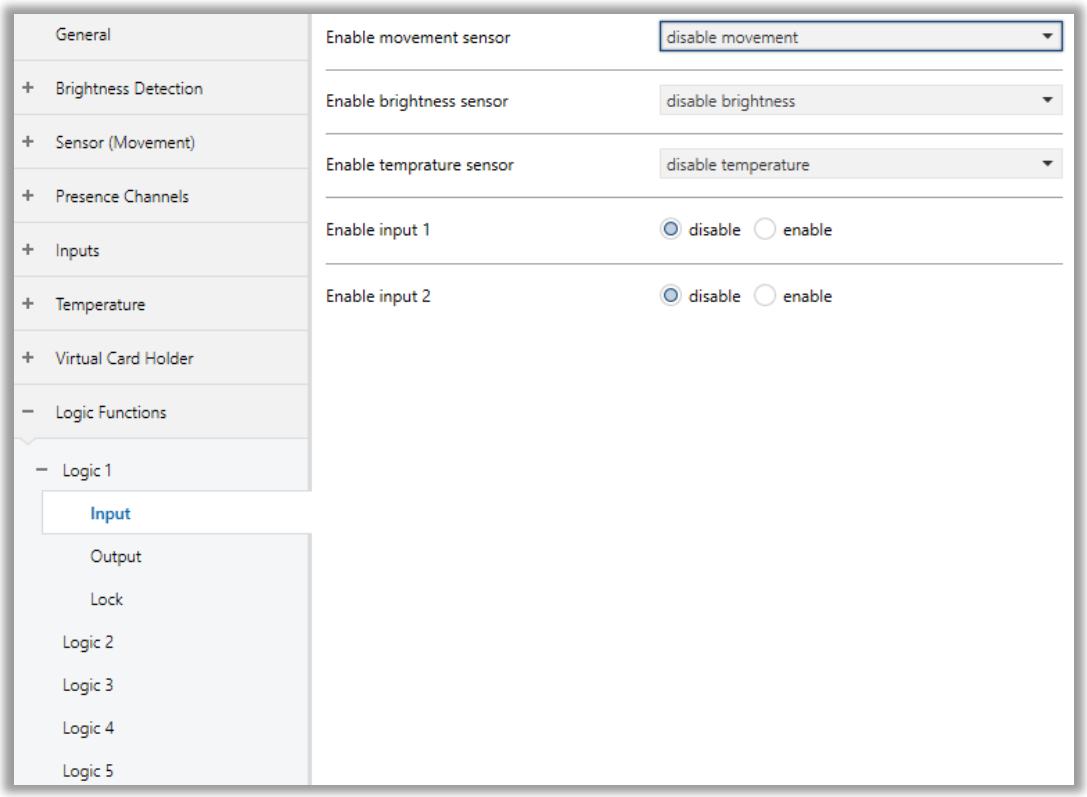


Fig. 67: Logic Functions – Input Configuration Page

3.9.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Enable Movement Sensor	This parameter is used to enable or disable the movement sensor: **Internal movement:** The internal movement sensor will be used as movement logic input. **External movement:** The external movement information will be used for movement detection.	**Disable movement** Internal movement External movement
-> Internal Movement Sensor Status	This parameter is used to determine when the internal movement sensor detects a movement is accounted as TRUE or FALSE.	Movement sensor detected is FALSE else is TRUE **Movement sensor detected is TRUE else is FALSE**
Enable Brightness Sensor	This parameter is used to enable or disable the brightness sensor. **Internal Brightness:** The internal brightness sensor will be used as brightness logic input. **External Brightness:** The external brightness sensor will be used as brightness logic input.	**Disable Brightness** Internal Brightness External Brightness
-> Threshold brightness lower	This parameter is used to determine the lower threshold brightness value.	1…**100**…1200
-> Threshold brightness upper	This parameter is used to determine the upper threshold brightness value.	1…**300**…1200
-> Brightness status	This parameter is used to determine when the ambient brightness value is accounted as TRUE or FALSE.	**In range is TRUE, else FALSE** Out range is TRUE, else FALSE Under lower is TRUE, above upper is FALSE Under lower is FALSE, above upper is TRUE
-> Change brightness threshold via bus	This parameter is used to change the brightness threshold value via a KNX bus object.	**No** yes

Enable Temperature Sensor	This parameter is used to enable or disable the temperature sensor. **Temperature:** The temperature chosen in the sensor's temperature function. **KNX temperature:** The external temperature sensor will be used as temperature logic input.	**Disable Temperature** Internal temperature External temperature
-> Threshold temperature upper	This parameter is used to determine the lower threshold temperature value.	-300…**260**…700
-> Threshold temperature lower	This parameter is used to determine the upper threshold temperature value.	-300…**220**…700
-> Temperature status	This parameter is used to determine when the ambient temperature value is accounted as TRUE or FALSE.	**In range is TRUE, else FALSE** Out range is TRUE, else FALSE Under lower is TRUE, above upper is FALSE Under lower is FALSE, above upper is TRUE
-> Change temperature threshold via bus	This parameter is used to change the temperature threshold value via a KNX bus object.	**No** yes
Enable Input 1	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
-> Select Input Source	This parameter is used to determine the input source as an internal or external object.	**External object** Internal object
->> Contact Input Status	This parameter is used to determine when a press occurs on the local input is accounted as TRUE or FALSE.	**Pressed TRUE else FALSE** Pressed FALSE else TRUE
->> External input type	This parameter is used to determine the external input type of the enabled input 1 object.	**1-bit value('1'/'0')** 1-byte value(0...255) 2-byte threshold(0…65535)

		2-byte float threshold(-50C …100C) 4-byte threshold(0…4294967295)
->> External input status	This parameter is used to determine the input status as TRUE or FALSE according to the value. (This is visible if the input is selected as 1 bit)	**‘1’ is TRUE, ‘0’ is FALSE** ‘1’ is FALSE, ‘0’ is TRUE
->> External Input value	This parameter is used to determine the external input threshold value to evaluate the input status as TRUE or FALSE.	0…**100**…255 0…**1000**…65535 -500…**0**…1000 0…**10000**…4294967295
->> External input status	This parameter is used to determine the input status as TRUE or FALSE according to the value. (This is visible if the input is not selected as 1 bit)	**TRUE if input value >= threshold else FALSE** TRUE if input value <= threshold else FALSE
->> Default status after bus voltage recovery	This parameter is used to determine the logic status after bus voltage recovery.	**False** True recovery
Enable Input 2	This parameter is used to enable or disable input 2 for logic function block as input	**Disable** enable
-> Select Input Source	This parameter is used to determine the input source as an internal or external object.	**External object** Internal object
->> Contact Input Status	This parameter is used to determine when a press occurs on the local input is accounted as TRUE or FALSE.	**Pressed TRUE else FALSE** Pressed FALSE else TRUE
->> External input type	This parameter is used to determine the external input type of the enabled input 2 object.	**1-bit value(‘1’/’0’)** 1-byte value(0..255) 2-byte threshold(0…65535)

		2-byte float threshold(-50C …100C) 4-byte threshold(0…4294967295)
->> External input status	This parameter is used to determine the input status as TRUE or FALSE according to the value. (This is visible if the input is selected as 1 bit)	**‘1’ is TRUE, ‘0’ is FALSE** ‘1’ is FALSE, ‘0’ is TRUE
->> External Input value	This parameter is used to determine the external input threshold value to evaluate the input status as TRUE or FALSE.	0…**100**…255 0…**1000**…65535 -500…**0**…1000 0…**10000**…4294967295
->> External input status	This parameter is used to determine the input status as TRUE or FALSE according to the value. (This is visible if the input is not selected as 1 bit)	**TRUE if input value >= threshold else FALSE** TRUE if input value <= threshold else FALSE
->> Default status after bus voltage recovery	This parameter is used to determine the logic status after bus voltage recovery.	**False** True recovery

3.9.3. Logic Functions – Output General

This section describes the general parameters of the logic output functions. The property of each respective output channel is set by configuring the parameters in this section. Also, repetitive sending of output values can be set here.

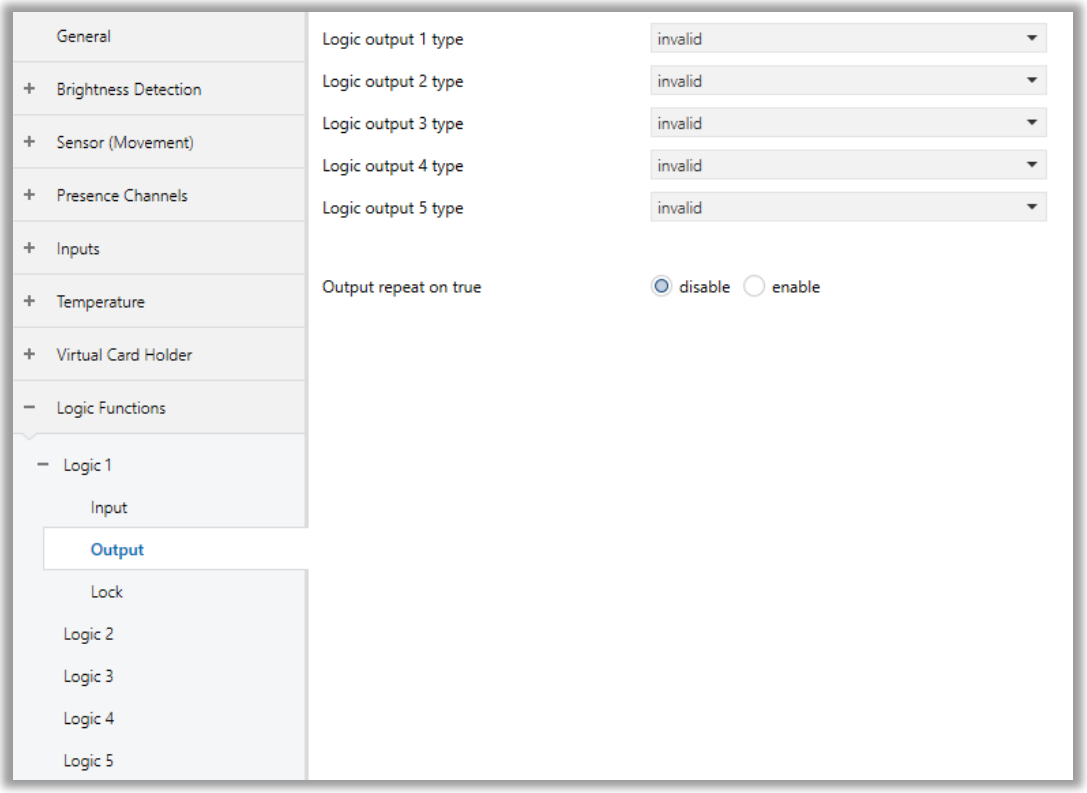


Fig. 68: Logic Functions – Output General Configuration Page

3.9.3.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Logic Output X type (1...5)	This parameter is used to specify the related logic output x channel functionality. If this parameter is selected as invalid, the related output channel will not be used. Other selected options will be configured separately.	**Invalid** Switch controller Absolute dimming controller Shutter controller Alarm controller Percentage control. Sequence control. Scene controller String controller Threshold controller
Output repeat on true	This parameter is used to enable or disable the output repeating time for all output channels when the logic gate state is true.	**On telegram** Off telegram
-> Repeated time interval	This parameter is used to determine the repeated time for all enabled output channels to send output channel values when the logic gate state is true.	0…**120**…65535

3.9.4. Logic Functions – Outputs 1-5

This section describes parameter configurations for each logic output channel. Although the working principle is the same for all output channels, only the type of values to be sent changes depending on the selected output functionality. For this reason, parameters are described in a common table about only one feature.

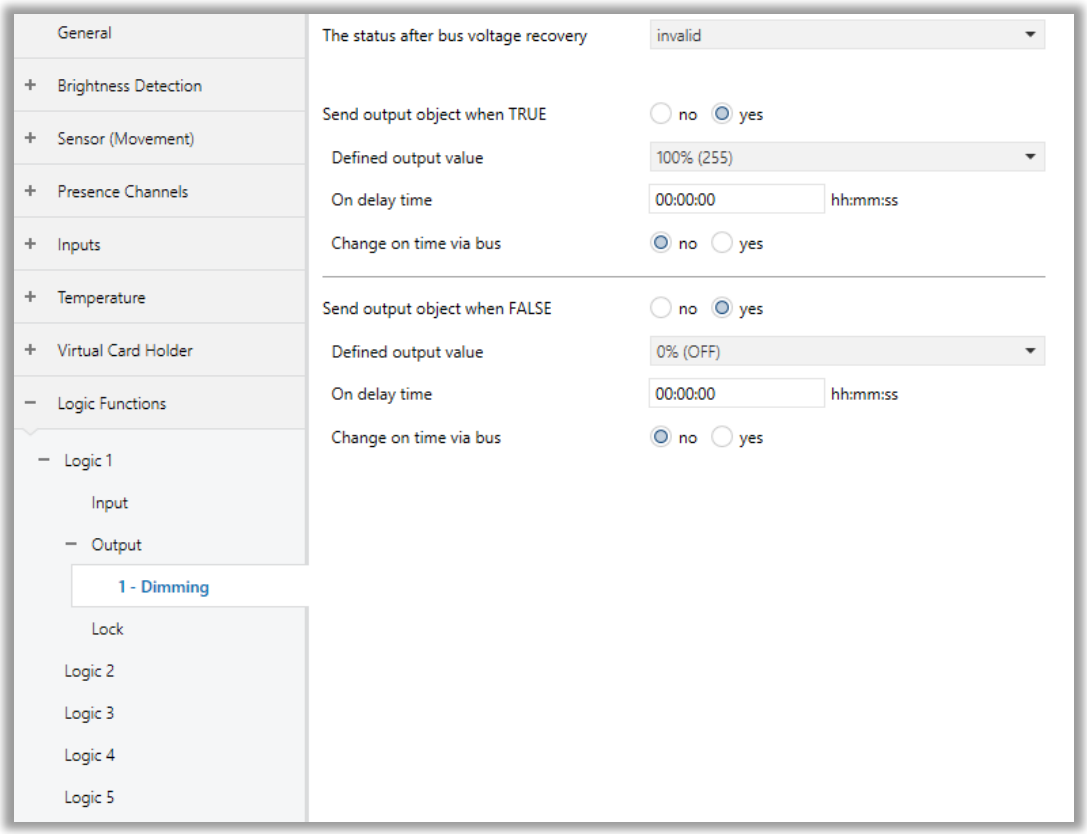


Fig. 69: Logic Functions – Output: Dimming Configuration Page

3.9.4.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
The status after bus voltage recovery	This parameter is used to determine the logic output channel x status after bus voltage recovery.	**invalid** Recovery Defined
-> Recovery Defined Value	This parameter is used to determine the output channel x value when the bus voltage has been recovered.	**On**…Off **%0**…%100 **Up**…Down **No alarm**…alarm **Stop**…start **Scene** **no.1**…scene no64 14 bytes string 0…**1000**…65535
Send output object when TRUE	This parameter is used to enable or disable the sending output object when the logic gate is true.	**No** yes
-> Defined Output Value	This parameter is used to determine the logic output channel x defined value when the logic gate is true.	**On**…Off **%0**…%100 **Up**…Down **No alarm**…alarm **Stop**…start **Scene** **no.1**…scene no64 14 bytes string 0…**1000**…65535
-> On Delay Time	This parameter is used to determine the on-delay time of the related logic output channel x when the logic gate is true.	**00:00:00**…18:12:15
-> Change on Time Via Bus	This parameter is used to enable or disable the on-delay time object for changing the delay time on the true state.	**No** yes
Send output object when FALSE	This parameter is used to enable or disable the sending output object when the logic gate is false.	**No** yes
-> Defined Output Value	This parameter is used to determine the logic output channel x defined value when the logic gate is false.	**On**…Off **%0**…%100 **Up**…Down **No alarm**…alarm **Stop**…start **Scene** **no.1**…scene no64 14 bytes string 0…**1000**…65535

-> On Delay Time	This parameter is used to determine the on-delay time of the related logic output channel x when the logic gate is false.	**00:00:00**…18:12:15
-> Change on Time Via Bus	This parameter is used to enable or disable the on-delay time object for changing the delay time on the false state.	**No** yes

3.9.5. Logic Functions – Lock

In this section, the locking feature of the logic functions is mentioned. The locking feature is for each logic function gate and is configured separately. Since there are 5 different logic function gates in the KNX sensor device, a separate configuration is required for each. Since the parameter page for each section is the same, only 1 is explained in this section.

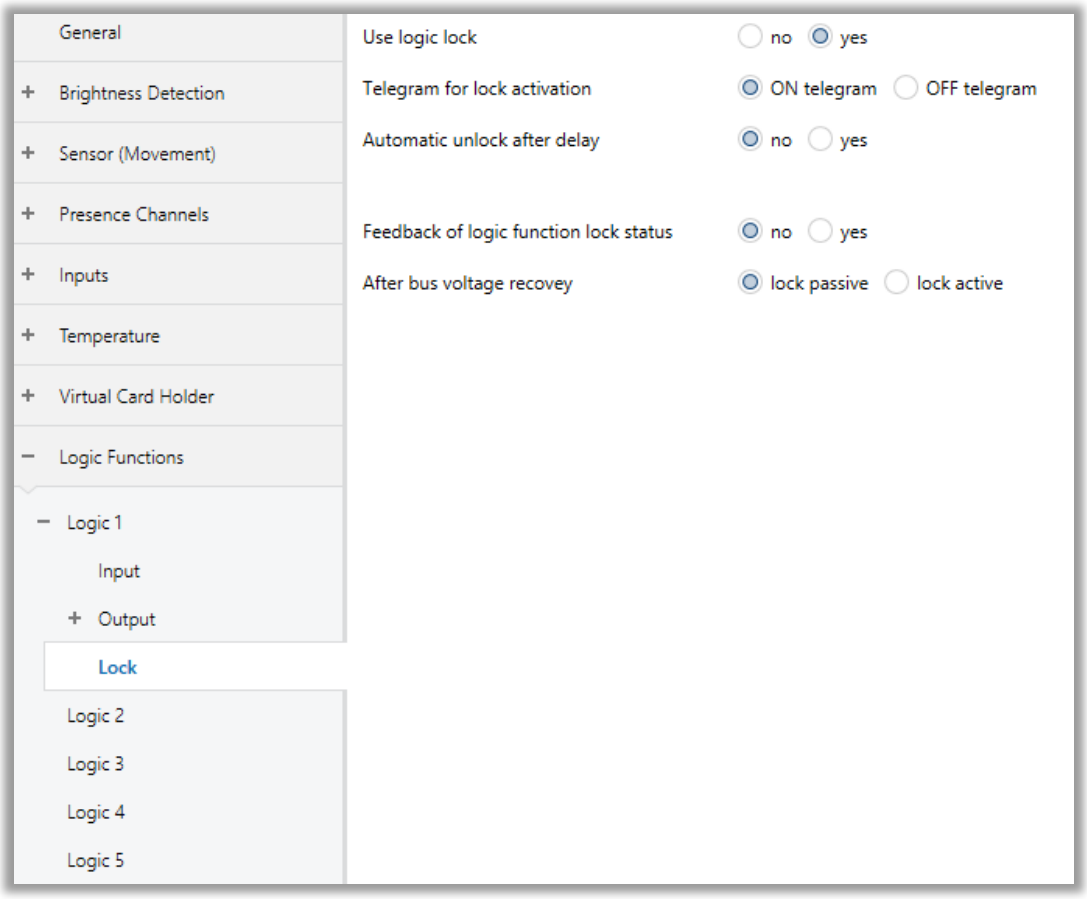


Fig. 70: Logic Functions – Lock Configuration Page

3.9.5.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Use Logic Lock	This parameter is used to lock the related logic function gate.	**no** yes
Telegram for Lock Activation	This parameter is used to determine the telegram value that locks the related logic function gate.	**On telegram** Off telegram
Automatic Unlock After Delay	This parameter is used to enable or disable the automatic unlock to unlock the logic gate after a while.	**no** yes
Automatic unlock time	This parameter is used to determine the automatically unlock period to unlock the logic function gate.	00:00:05…**00:00:30** ...18:12:15
Feedback of logic function lock status	This parameter is used to enable or disable the feedback of the logic lock status object.	**No** yes
After Bus Voltage Recovery	This parameter is used to determine the logic function gate lock status after the bus voltage recovery.	**Lock Passive** Lock Active

4. ETS Objects List & Descriptions

The Interra Multi Presence Detectors can communicate via the KNX bus line. In this section, the group objects of the Interra Multi Presence Detectors are described. All of the communication objects listed below are available to the presence detector. Which of these group objects are visible and capable of being linked with group addresses are explained in sub-sections.

No	Name	Function	DTP Type	Length	Flags				
					C	R	W	T	U
1	General	In operation	1.002	1 bit	X			X	
2	General	LED indicator	1.003	1 bit	X	X	X	X	
3	General	Movement Sensor 1	1.001	1 bit	X	X		X	
4	General	Movement Sensor 2	1.001	1 bit	X	X		X	
5	General	Movement Sensor 3	1.001	1 bit	X	X		X	
6	General	Microwave sensor boost	5.001	1 byte	X	X	X	X	
7	General	Microwave sensor sensitivity	5.001	1 byte	X	X	X	X	
10	Input 1	Block	1.003	1 bit	X		X		
11	Input 1	Switch	1.001	1 bit	X		X	X	
		Shutter Up/Down	1.008	1 bit	X		X	X	
		Forced Operation – Switch	2.001	2 bit	X			X	
		Forced Operation – Percent	5.001	1 byte	X			X	
		Forced Operation – Decimal	5.005	1 byte	X			X	
		Forced Operation – Scene	17.001	1 byte	X			X	
		Forced Operation – Colour	7.600	2 bytes	X			X	
		Forced Operation – Temperature	9.001	2 bytes	X			X	
		Forced Operation – Brightness	9.004	2 bytes	X			X	
		Forced Operation – RGB	232.600	3 bytes	X			X	
		8-bit Scene	18.001	1 byte	X			X	
		Red colour	5.010	1 byte	X	X		X	
		RGB colour	232.600	3 bytes	X	X		X	
		Mode selection	20.102	1 byte	X	X		X	
		Sequence	1.001	1 bit	X	X		X	
			5.010	1 byte	X	X		X	
			5.001	1 byte	X	X		X	
			20.102	1 byte	X	X		X	
		Sequence A	1.001	1 bit	X	X		X	
		Sequence A (0…255)	5.010	1 byte	X	X		X	
		Sequence A (0…100%)	5.001	1 byte	X	X		X	
		Sequence A HVAC	20.102	1 byte	X	X		X	
		Counter value	5.010	1 byte	X	X		X	
			7.001	2 bytes	X	X		X	
			12.001	4 bytes	X	X		X	
		RGBW Colour	251.600	6 bytes	X			X	

		RGBW – Red Colour	5.010	1 byte	X	X		X	
12	Input 1	Switch - long	1.001	1 bit	X			X	
		Dimming	3.007	4 bits	X			X	
		STOP / Lamella Adjustment	1.007	1 bit	X			X	
		Forced Operation – Switch	2.001	2 bits	X			X	
		Forced Operation – Percent	5.001	1 byte	X			X	
		Forced Operation – Decimal	5.005	1 byte	X			X	
		Forced Operation – Scene	17.001	1 byte	X			X	
		Forced Operation – Colour	7.600	2 bytes	X			X	
		Forced Operation – Temperature	9.001	2 bytes	X			X	
		Forced Operation – Brightness	9.004	2 bytes	X			X	
		Forced operation – RGB	232.600	3 bytes	X			X	
		Scene Store	1.003	1 bit	X	X	X		
		HVAC-Mode State	20.102	1 byte	X		X		
		Sequence B	1.001	1 bit	X	X		X	
		Sequence B (0…255)	5.010	1 byte	X	X		X	
		Sequence B (0…100%)	5.001	1 byte	X	X		X	
		Sequence B HVAC	20.102	1 byte	X	X		X	
		Reset Counter	1.001	1 bit	X	X	X	X	
		RGB – Gree Colour	5.010	1 byte	X			X	
		RGBW – Green Colour	5.010	1 byte	X			X	
13	Input 1	Upper limit position	1.002	1 bit	X		X		
		RGB – Blue Colour	5.010	1 byte	X			X	
		RGBW – Blue Colour	5.010	1 byte	X			X	
		Sequence C	1.001	1 bit	X	X		X	
		Sequence C (0…255)	5.010	1 byte	X	X		X	
		Sequence C (0…100%)	5.001	1 byte	X	X		X	
		Sequence C HVAC	20.102	1 byte	X	X		X	
		Overflow	1.001	1 bit	X	X	X	X	
			5.010	1 byte	X	X	X	X	
14	Input 1	Lower limit position	1.002	1 bit	X		X		
		Sequence D	1.001	1 bit	X	X		X	
		Sequence D (0…255)	5.010	1 byte	X	X		X	
		Sequence D (0…100%)	5.001	1 byte	X	X		X	
		Sequence D HVAC	20.102	1 byte	X	X		X	
		RGBW – White Colour	5.010	1 byte	X			X	
21	Input 3 - Analog	Temperature	9.001	2 bytes	X	X		X	
15…24	**Input 2…3**								
25	Brightness	Measured value	9.004	2 bytes	X	X		X	
26	Brightness	Calibration value	9.004	2 bytes	X		X		
27	Sensor	Block	1.003	1 bit	X		X		
28	Sensor	Presence output	1.001	1 bit	X	X		X	

			5.001	1 byte	X	X		X	
29	Sensor	Light-on time	7.005	2 bytes	X		X		X
30	Sensor	External movement	1.001	1 bit	X		X		X
31	Sensor	External brightness	9.004	2 bytes	X		X		X
32	Sensor	Brightness-value threshold	9.004	2 bytes	X	X	X	X	X
33	Presence 1	Lock	1.003	1 bit	X		X		X
34	Presence 1	Presence output	1.001	1 bit	X	X		X	
			5.001	1 byte	X	X		X	
35	Presence 1	Light-on time	7.005	2 bytes	X		X		X
36	Presence 1	External movement	1.001	1 bit	X		X		
37	Presence 1	External brightness	9.004	2 bytes	X		X		
38	Presence 1	Brightness-value threshold	9.004	2 bytes	X	X	X	X	X
39	Presence 1	Manual on/off	1.001	1 bit	X		X		X
40	Presence 1	Relative dimming	3.007	4 bit	X		X		X
41	Presence 1	Absolute dimming	5.001	1 byte	X		X		X
42	Presence 1	Status of control	1.001	1 bit	X	X		X	
43	Presence 1	Forced operation	1.010	1 bit	X	X	X		X
			2.001	2 bit	X	X	X		X
44	Presence 1	Forced operation value	5.001	1 byte	X	X	X		X
45…84	**Presence 2...4**								
85	Temperature	Alarm	1.005	1 bit	X	X		X	
86	Temperature	Actual Temperature	9.001	2 bytes	X	X		X	
87	Temperature	KNX Probe Temperature	9.001	2 bytes	X		X		
88	Thermostat	Disabling	1.003	1 bit	X		X		
		Disabling	1.003	1 bit	X	X		X	
89	Thermostat	Status	1.003	1 bit	X	X		X	
		Status	1.003	1 bit	X		X		
92	Thermostat	Operation Mode	20.102	1 byte	X		X		
		Operation Mode	20.102	1 byte	X	X		X	
93	Thermostat	Operation Mode Forced	20.102	1 byte	X		X		
94	Thermostat	Operation Mode Status	20.102	1 byte	X	X		X	
		Operation Mode Feedback	20.102	1 byte	X		X		
95	Thermostat	Operation Mode [Comfort]	1.001	1 bit	X	X	X		
96	Thermostat	Operation Mode [Standby]	1.001	1 bit	X	X	X		
97	Thermostat	Operation Mode [Economy]	1.001	1 bit	X	X	X		
98	Thermostat	Operation Mode [Protection]	1.001	1 bit	X	X	X		
99	Thermostat	Heating/Cooling Switchover	1.100	1 bit	X		X		
		Heating/Cooling Switchover	1.100	1 bit	X	X		X	
100	Thermostat	Heating/Cooling Status	1.100	1 bit	X	X		X	
		Heating/Cooling Feedback	1.100	1 bit	X		X		
101	Thermostat	Heating Control Disabling	1.001	1 bit	X		X		
102	Thermostat	Heating Control Running	1.001	1 bit	X	X		X	

		Heating Control Running	1.001	1 bit	X		X		
103	Thermostat	Heating Value (1-bit)	1.001	1 bit	X	X		X	
		Heating Value (1-byte)	5.001	1 byte	X	X		X	
		Heating/Cooling Value (1-bit)	1.001	1 bit	X	X		X	
		Heating/Cooling Value (1-byte)	5.001	1 byte	X	X		X	
104	Thermostat	Heating Value Request	1.016	1 bit	X		X		
		Heating/Cooling Value Request	1.016	1 bit	X		X		
105	Thermostat	Cooling Control Disabling	1.001	1 bit	X		X		
106	Thermostat	Cooling Control Running	1.001	1 bit	X	X		X	
		Cooling Control Running	1.001	1 bit	X		X		
107	Thermostat	Cooling Value (1-bit)	1.001	1 bit	X	X		X	
		Cooling Value (1-byte)	5.001	1 byte	X	X		X	
108	Thermostat	Cooling Value Request	1.016	1 bit	X		X		
109	Thermostat	Additional Heating Control Disabling	1.001	1 bit	X		X		
110	Thermostat	Additional Heating Control Running	1.001	1 bit	X	X		X	
111	Thermostat	Additional Heating Value(1-Bit)	1.001	1 bit	X	X		X	
		Additional Heating Value(1-Byte)	5.001	1 byte	X	X		X	
112	Thermostat	Additional Heating Value Request	1.016	1 bit	X		X		
113	Thermostat	Additional Cooling Control Disabling	1.003	1 bit	X		X		
114	Thermostat	Additional Cooling Control Running	1.002	1 bit	X	X		X	
115	Thermostat	Additional Cooling Value (1-Bit)	1.001	1 bit	X	X		X	
		Additional Cooling Value (1-Byte)	5.001	1 byte	X	X		X	
116	Thermostat	Additional Cooling Value Request	1.016	1 bit	X		X		
117	Thermostat	Room Temperature Output -Celsius	9.001	2 bytes	X	X		X	
		Room Temperature Input -Celsius	9.001	2 bytes	X		X		
		Room Temperature Output - Fahrenheit	9.027	2 bytes	X	X		X	
		Room Temperature Input -Fahrenheit	9.027	2 bytes	X		X		
118	Thermostat	Actual Setpoint Output	9.001	2 bytes	X	X		X	
					X		X		
			9.002	2 bytes	X	X		X	
					X		X		
			9.027	2 bytes	X	X		X	
					X		X		
119	Thermostat	Manual Setpoint Input	9.001	2 bytes	X		X		
					X	X		X	
			9.002	2 bytes	X		X		
					X	X		X	
			9.027	2 bytes	X	X		X	
					X		X		
120	Thermostat	Manual Setpoint Reset	1.015	1 bit	X		X		

121	Thermostat	Heating Comfort Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
122	Thermostat	Heating Standby Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
123	Thermostat	Heating Economy Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
124	Thermostat	Heating Protection Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
125	Thermostat	Cooling Comfort Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
126	Thermostat	Cooling Standby Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
127	Thermostat	Cooling Economy Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
128	Thermostat	Cooling Protection Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
129	Thermostat	Fan Controller Disable	1.003	1 bit	X		X		
130	Thermostat	Fan Controller Status	1.003	1 bit	X	X		X	
131	Thermostat	Fan Controller Working Mode	1.001	1 bit	X		X		
132	Thermostat	Fan Controller Working Mode Status	1.001	1 bit	X	X		X	
133	Thermostat	Fan Controller Proportional Output	5.001	1 byte	X	X		X	
134	Thermostat	Fan Controller Manual Step	1.007	1 bit	X		X		
		Fan Controller Manual Up/Down	1.008	1 bit	X		X		
135	Thermostat	Fan Controller Manual Stage	5.100	1 byte	X		X		
136	Thermostat	Fan Controller Speed (1 Byte)	5.001	1 byte	X	X		X	
		Fan Controller Speed (1 Byte)	5.100	1 byte	X	X		X	
137	Thermostat	Fan Controller Speed Feedback (1 Byte)	5.001	1 byte	X		X		X
		Fan Controller Speed Feedback (1 Byte)	5.100	1 byte	X		X		X
138	Thermostat	Fan Level 1	1.001	1 bit	X	X		X	
139	Thermostat	Fan Level 2	1.001	1 bit	X	X		X	
140	Thermostat	Fan Level 3	1.001	1 bit	X	X		X	
141	Thermostat	Fan Level 4	1.001	1 bit	X	X		X	
142	Thermostat	Fan Level 5	1.001	1 bit	X	X		X	
143	Thermostat	Fan Level 1 Feedback Input	1.001	1 bit	X		X		X
144	Thermostat	Fan Level 2 Feedback Input	1.001	1 bit	X		X		X
145	Thermostat	Fan Level 3 Feedback Input	1.001	1 bit	X		X		X
146	Thermostat	Fan Level 4 Feedback Input	1.001	1 bit	X		X		X
147	Thermostat	Fan Level 5 Feedback Input	1.001	1 bit	X		X		X
154	Thermostat	Temperature Limit Heating Source	9.001	2 bytes	X		X		

			9.027	2 bytes	X		X		
155	Thermostat	Temperature Limit Cooling Source	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
156	Thermostat	Temperature Limit Additional Heating Source	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
157	Thermostat	Temperature Limit Additional Cooling Source	9.001	2 bytes	X		X		
158	Thermostat	Time	10.001	3 bytes	X		X		
159	Virtual Holder	Room booked	1.002	1 bit	X		X		
160	Virtual Holder: Output	Presence	1.002	1 bit	X	X		X	
161	Virtual Holder: Output	Unexpected presence	1.005	1 bit	X			X	
162	Virtual Holder: Output	HVAC	20.102	1 byte	X	X		X	
163	Virtual Holder: Output	Additional	5.001	1 byte	X			X	
			5.010	1 byte	X			X	
			17.001	1 byte	X			X	
164	Virtual Holder: Remote Input1	Presence	1.002	1 bit	X		X		
		Door	1.009	1 bit	X		X		
165	Virtual Holder: Remote Input2	Presence	1.002	1 bit	X		X		
		Door	1.009	1 bit	X		X		
166	Logic 1	Lock	1.003	1 bit	X		X		
167	Logic 1	Feedback of lock	1.003	1 bit	X	X		X	
168	Logic 1: Input	External Movement	1.001	1 bit	X		X		
169	Logic 1: Input	External Brightness	9.004	2 bytes	X		X	X	X
170	Logic 1: Input	Brightness Threshold Lower	9.004	2 bytes	X		X	X	X
171	Logic 1: Input	Brightness Threshold Upper	9.004	2 bytes	X		X	X	X
172	Logic 1: Input	External Temperature	9.001	2 bytes	X		X		
173	Logic 1: Input	Temperature Threshold Lower	9.001	2 bytes	X		X	X	X
174	Logic 1: Input	Temperature Threshold Upper	9.001	2 bytes	X		X	X	X
175, 176, 177	Logic 1: Input: 1…3	External Input 1 / 2 / 3	1.001	1 bit	X		X		X
			5.010	1 byte	X		X		X
			7.001	2 bytes	X		X		X
			9.001	2 bytes	X		X		X
			12.001	4 bytes	X		X		X
178	Logic 1: Output	Result Status	1.002	1 bit	X			X	
179, 182, 185, 188, 191	Logic 1: Output 1…5	Switching	1.001	1 bit	X	X		X	
		Absolute Dimming	5.001	1 byte	X	X		X	
		Shutter	1.008	1 bit	X	X		X	
		Alarm	1.005	1 bit	X	X		X	
		Sequence	1.010	1 bit	X	X		X	
		Scene	17.001	1 byte	X	X		X	

		String (14 byte)	16.000	14 bytes	X	X		X	
		Threshold	7.001	pulses	X	X		X	
180, 183, 186, 189, 192	Logic 1: Output 1…5	Delay Time on TRUE State	7.005	2 bytes	X		X	X	X
181, 184, 187, 190, 193	Logic 1: Output 1…5	Delay Time on FALSE State	7.005	2 bytes	X		X	X	X
194…277	**Logic 2…4**								

4.1. General Objects

This section describes the "general" group objects and their properties. General group objects, as the name suggests, indicate the general characteristics of the Multi Presence Detector.

Object Number	Object Name	Function	Type	Flags
1	General	In operation	1 bit	CT

This object is used to monitor the presence of the device on the KNX bus line regularly. However, monitoring telegrams can be sent cyclically on the KNX bus line.

DPT: 1.002 (boolean)

2	General	LED Indicator	1 bit	CRWT

This object is used in place of the blue LED used to indicate that motion detection has occurred. When a motion in the detector's sensing angle is detected by the sensor, a 1-bit telegram is sent over this object. This object is usually used during installation to quickly and easily determine the detection range of the sensor.

DPT: 1.003 (switch)

3, 4, 5	General	Movement Sensor 1/ 2/ 3	1 bit	CRT

This object enables the parameter-independent use of sensing data from the sensing elements included according to the sensor type. It is suitable for testing purposes.

DPT: 1.001 (switch)

6	General	Microwave Sensor Boost	1 byte	CRWT

This object enables the parameter-independent use of sensing data from the sensing elements included according to the sensor type. It is suitable for testing purposes.

DPT: 5.001 (percentage)

7	General	Microwave Sensor Sensitivity	1 byte	CRWT

This object enables the parameter-independent use of sensing data from the sensing elements included according to the sensor type. It is suitable for testing purposes.

DPT: 5.001 (percentage)

4.2. Input Objects

This section contains information about KNX objects and their properties related to the input channels. The types, flags and properties of the objects are explained in detail below. There are 3 digital inputs channels with the same functionality and an additional probe channel. In this section, digital inputs objects are described only for one channel due to the identical.

X: 1 … 3

Object Number	Object Name	Function	Type	Flags
10, 15, 20	**Input X**	**Block**	**1 bit**	**CW**

This object is used to set the Multi Presence Detector input X status. “Enabled” or “Disabled” telegram is received via this object.

For example, it will be disabled when an “Enabled” telegram is received from the KNX bus line, and when a “Disabled” telegram is received, the input X will continue working.

DPT: 1.003 (enable)

11, 16, 21	**Input X**	**Switch**	**1 bit**	**CWT**

This communication object changes in functionality depending on the selected input function. In accordance with the parameter setting, this communication object can be switched by actuation of the input to ON, OFF or TOGGLE.

DPT: 1.001 (switch)

11, 16, 21	**Input X**	**Shutter UP/Down**	**1 bit**	**CWT**

This communication object changes in functionality depending on the selected input function. This communication object sends a shutter motion telegram UP or DOWN on the bus. By receiving telegrams, the device also recognizes movement telegrams of another sensor, e.g. parallel operation.

DPT: 1.008 (up/down)

11, 16, 21	**Input X**	**Forced Operation**	**2 bit / 1 byte / 2 byte / 3 bytes**	**CT**

This communication object changes in functionality depending on the selected input function. This communication object sends a value on the bus with short operation when opening or closing of the contact. Depending on the configuration, the data type of this object changes. forced, percent value, decimal value, Scene number, temperature value, brightness value and percent value (RGB) can be performed on this object.

DPT: According to parameter selection, DPT changes.

11, 16, 21	**Input X**	**8-bit Scene**	**1 byte**	**CT**

This communication object stores the value of the active scene number (1 - 64).

DPT: 18.001 (scene control)

11, 16, 21	**Input X**	**Mode selection**	**1 byte**	**CRT**

This object keeps the active HVAC state that can be toggled through press events.

Note: There can be up to 4 different HVAC state (comfort, standby, economy, building protection) selected and each press event toggles through the HVAC states that are set as available in the parameter list.

DPT: 20.102 (HVAC mode)

11, 16, 21	**Input X**	**Sequence**	**1 bit / 1 byte**	**CRT**

This object keeps the current command that can be toggled through press events. Used for “Single Object” parameter selection.

Note: Each state (State A, B, C, D) holds a different value with adjustable data length. Each press event puts the next available state’s data to the “Sequence” object.

DPT: According to parameter selection, DPT changes.

11, 16, 21	**Input X**	**Sequence A**	**1 bit / 1 byte**	**CRT**

This object keeps the current command that can be toggled through press events. Used for “Multiple Object” parameter selection.

Note: Each object (Object A, B, C, D) holds a different value with adjustable data length. Each press event puts the next available state’s data to the “Sequence X” object and whichever object is holds the current state is sent to bus with its data.

DPT: According to parameter selection, DPT changes.

11, 16, 21	**Input X**	**Counter value**	**1 byte / 2 byte / 4 bytes**	**CRT**

This object keeps the current value of the press counter.

DPT: According to parameter selection, DPT changes.

11, 16, 21	**Input X**	**RGB Red Colour / RGB Colour**	**1 byte / 3 bytes**	**CRT**

This object either keeps the 1-Byte Red value of the RGB, or keeps the entire 3-Byte RGB value. Decision is made in the parameter list as either “1 object of 3 bytes” or 3 objects of 1 byte”.

DPT: 5.010 (counter pulses) / 232.600 (RGB value)

11, 16, 21	**Input X**	**RGBW Colour / RGBW Red Colour**	**6 bytes / 1 byte**	**CT / CRT**

If the “object type” is set to “1 object”, this object keeps the 6-Byte RGBW value, but, if the “object type” is set to “4 objects”, this object keeps the 1-Byte Red value of the RGBW.

DPT: 251.600 (RGBW value) / 5.010 (counter pulses)

21	**Input 3 - Analog**	**Temperature**	**2 bytes**	**CRT**

This object is the object used to send the temperature data read from the temperature probe plugged into the Input 3 port, located physically on the back of the device, to the KNX line.

DPT: 9.001 (temperature (°C))

12, 17, 22	**Input X**	**RGB Green Colour**	**1 byte**	**CT**

This object keeps the 1-Byte green value of RGB if “3 objects of 1 Byte” option is selected in the parameter list.

DPT: 5.010 (counter pulses)

12, 17, 22	**Input X**	**RGBW Green Colour**	**1 byte**	**CT**

If the “object type” is set to “4 objects”, this object keeps the 1-Byte Green value of the RGBW.

DPT: 5.010 (counter pulses)

12, 17, 22	**Input X**	**Switch - Long**	**1 bit**	**CRWTU**

This communication object changes in functionality depending on the selected input function. In accordance with the parameter setting, this communication object can be switched by actuation of the input to ON, OFF or TOGGLE.

DPT: 1.001 (switch)

12, 17, 22	**Input X**	**Dimming**	**4 bits**	**CT**

This communication object changes in functionality depending on the selected input function. In accordance with the parameter setting, A long operation at the input has the effect that BRIGHTER or DARKER dim telegrams are sent via this communication object on the bus. A STOP telegram is sent and the cyclic sending of dim telegrams is stopped at the end of actuation with START-STOP-DIMMING.

DPT: 3.007 (dimming control)

12, 17, 22	**Input X**	**STOP / Lamella Adjustment**	**1 bit**	**CT**

This communication object changes in functionality depending on the selected input function. This communication object sends a STOP telegram or slat adjustment.

DPT: 1.007 (step)

12, 17, 22	**Input X**	**Forced Operation – Long**	**1 bit / 1 byte / 2 bytes / 3 bytes**	**CT**

This communication object changes in functionality depending on the selected input function. This communication object sends a value on the bus with short operation when opening or closing of the contact. Depending on the configuration, the data type of this object changes. forced, percent value, decimal value, Scene number, temperature value, brightness value and percent value (RGB) can be performed on this object.

DPT: According to parameter selection, DPT changes.

12, 17, 22	**Input X**	**Scene Store**	**1 bit**	**CRW**

This communication object, when active, decides whether to call or store the preset 8-bit scene number in the parameter list. When the store scene object is enabled the preset scene number is stored, but, when disabled preset scene number is called to be active.

DPT: 1.003 (enable)

12, 17, 22	**Input X**	**HVAC-Mode State**	**1 byte**	**CW**

This object takes the HVAC state changed via the bus.

Note: Whenever this object is updated from the bus, the HVAC state that this object holds will be considered as the valid HVAC state and press events will act as if the last HVAC state is what this object is updated with.

DPT: 20.102 (HVAC mode)

12, 17, 22	**Input X**	**Sequence B**	**1 bit / 1 byte**	**CRT**

This object keeps the current command that can be toggled through press events. Used for “Multiple Object” parameter selection.

Note: Each object (Object A, B, C, D) holds a different value with adjustable data length. Each press event puts the next available state’s data to the “Sequence X” object and whichever object is holds the current state is sent to bus with its data.

DPT: According to parameter selection, DPT changes.

12, 17, 22	Input X	Reset counter	1 bit	CRWT

This object is used to reset the counter value to preset start value that can be set from parameter list.

DPT: According to parameter selection, DPT changes.

13, 18, 23	Input X	RGB Blue Colour	1 byte	CT

This object keeps the 1-Byte green value of RGB if “3 objects of 1 Byte” option is selected in the parameter list.

DPT: 5.010 (counter pulses)

13, 18, 23	Input X	RGBW Blue Colour	1 byte	CT

If the “object type” is set to “4 objects”, this object keeps the 1-Byte Green value of the RGBW.

DPT: 5.010 (counter pulses)

13, 18, 23	Input X	Upper Limit Position	1 bit	CW

This object is used for the shutter actuator indicates if it is in the lower limit position (“shutter/blind closed”). The object is intended for a 1-button operation. ‘0’ is no lower limit operation, ‘1’ lower end operation.

DPT: 1.002 (boolean)

13, 18, 23	Input X	Sequence C	1 bit / 1 byte	CRT

This object keeps the current command that can be toggled through press events. Used for “Multiple Object” parameter selection.

Note: Each object (Object A, B, C, D) holds a different value with adjustable data length. Each press event puts the next available state’s data to the “Sequence X” object and whichever object is holds the current state is sent to bus with its data.

DPT: According to parameter selection, DPT changes.

13, 18, 23	Input X	Overflow	1 bit / 1 byte	CRWT

This object is sent to bus with the preset value from the parameter list when the counter value exceeds the preset end value of the counter.

DPT: 1.001 (switch) / 5.010 (counter pulses)

14, 19, 24	Input X	Lower Limit Position	1 bit	CW

This object is used for the shutter actuator indicates if it is in the lower limit position (“shutter/blind closed”). The object is intended for a 1-button operation. ‘0’ is no lower limit operation, ‘1’ lower end operation.

DPT: 1.002 (boolean)

14, 19, 24	**Input X**	**Sequence D**	**1 bit / 1 byte**	**CRT**

This object keeps the current command that can be toggled through press events. Used for “Multiple Object” parameter selection.

Note: Each object (Object A, B, C, D) holds a different value with adjustable data length. Each press event puts the next available state’s data to the “Sequence X” object and whichever object is holds the current state is sent to bus with its data.

DPT: According to parameter selection, DPT changes.

14, 19, 24	**Input X**	**RGBW White Colour**	**1 byte**	**CT**

If the “object type” is set to “4 objects”, this object keeps the 1-Byte White value of the RGBW.

Note: White value is the colour temperature.

DPT: 5.010 (counter pulses)

4.3. Brightness Objects

This section describes the "brightness" group objects and their properties. Brightness group objects, as the name suggests, indicates the brightness configuration of the Multi Presence Detector. Brightness sensor objects are directly related to both the sensor and presence channels.

Object Number	Object Name	Function	Type	Flags
25	**Brightness**	**Measured value**	**2 bytes**	**CRT**

This object is used to send the values measured by the brightness sensor on the detector to the KNX bus line. Depending on the parameter configuration, the measured data can be sent to the bus line periodically or according to the amount of change.

Note: Brightness sensor calibration is required for the measurements to be healthier and more accurate.

DPT: 9.004 (lux (Lux))

26	**Brightness**	**Calibration value**	**2 bytes**	**CW**

This object is used for brightness sensor calibration. In general, sensor calibration is required to accurately measure the ambient brightness by the detector. In the environment where the sensor is installed, a suitable reference point is selected (for example, right under the sensor) and the ambient brightness is measured with a lux meter. Then, the measured value is sent to this calibration object as the value and the calibration process is completed.

DPT: 9.004 (lux (Lux))

4.4. Sensor

This section contains information about KNX objects and their properties related to the sensor channel. The types, flags and properties of the objects are explained in detail below.

Object Number	Object Name	Function	Type	Flags
27	**Sensor**	**Block**	**1 bit**	**CW**

This object is used to lock the sensor channel. It becomes visible when the "use sensor lock" parameter is set to yes. Depending on the parameter setting, when an ON or OFF telegram is sent to this object, the corresponding presence channel is locked.

For example, when "ON telegram" is selected in the parameter page for locking, it will be locked when an ON telegram is received from the KNX bus line, and when an OFF telegram is received, the sensor channel will be unlocked. Depending on the parameter configuration, an output value can also be sent when the locking operation is performed.

DPT: 1.003 (enable)

28	**Sensor**	**Presence output**	**1 bit / 1 byte**	**CRT**

This object is used to determine the output value of the sensor channel. The parameterized value during movement is sent to the actuator via the output when the parameterized brightness threshold including the hysteresis is exceeded or falls short. The value set in the "value for switch on" parameter is sent to the KNX bus line when detection occurs. After the detection process is finished (i.e., after the light-on time expires), the "value for switch off" value is sent to the KNX bus line.

If the "detection independent of brightness" parameter is set to yes, the output value is sent to the KNX bus line when a motion is detected regardless of the ambient brightness.

DPT: 1.001 (switch) / 5.001 (percentage)

29	**Sensor**	**Light-on time**	**2 bytes**	**CWU**

This object is used to change the duration of the sensor channel light-on time. The light-on time is the time between the end of motion detection and the sending of the telegram "Value for switch off". If movement is detected again within this period, the light-on time timer is started again. When a 2-byte time value is sent from the KNX bus line, this sent value will be used instead of the light-on time set via the parameter. Until a new time value is sent to this object, the last value sent will be used as the light-on period.

Note: The values which can be sent are between 10-65535 seconds. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 7.005 (time)

30	Sensor	External movement	1 bit	CWU

This object is used to inform the detector by external sources that a movement has occurred in the environment to be controlled by the detector. For this object to be visible, the "used movement detection" parameter must be selected "internal and external". When the ON telegram comes to this object from the KNX bus line, the detector determines motion detection and performs the relevant actions.

For instance, this object can be used when detectors are installed that will operate as a master-slave in the same environment. Typically, the output of sensors operating in slave mode is connected to this object of the sensor operating in master mode. Thus, the detection distance to be controlled is extended.

DPT: 1.001 (switch)

31	Sensor	External brightness	2 bytes	CWU

This object is used to send the brightness value to the detector's sensor channel via external sources. The value read on this object is compared with the brightness value threshold and operations are performed by the detector. For instance, if the brightness value is higher or lower than the hysteresis band, the detector will switch the lights.

Note: If the brightness value will be used over external sources, it is recommended to send the ambient brightness values to this object periodically so that the detector can make accurate evaluations.

DPT: 9.004 (lux)

32	Sensor	Brightness-value threshold	2 bytes	CRWTU

This object is used to change the switching threshold for the sensor channel, at which the detector is activated. The value is sent to this object in lux. This value is used as the new switching threshold. The current switching threshold can be read via this communication object. Until a new brightness threshold value is sent to this object, the last value sent will be used as the brightness-value threshold.

Note: The values which can be sent are between 1-1200 lux. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 9.004 (lux)

4.5. Presence

This section contains information about KNX objects and their properties related to the sensor channel. The types, flags and properties of the objects are explained in detail below. There are 4 presence channels with identical features in the Multi Presence Detector. For this reason, only objects for presence 1 channel are described in this section. These objects and their descriptions can be referenced when the other channels will be used.

X: 1..4

Object Number	Object Name	Function	Type	Flags
33, 46, 59, 72	**Presence X**	**Lock**	**1 bit**	**CWU**

This object is used to lock the presence channel. It becomes visible when the "use presence lock" parameter is set to yes. Depending on the parameter setting, when an ON or OFF telegram is sent to this object, the corresponding presence channel is locked.

For example, when "ON telegram" is selected in the parameter page for locking, it will be locked when an ON telegram is received from the KNX bus line, and when an OFF telegram is received, the presence channel will be unlocked. Depending on the parameter configuration, an output value can also be sent when the locking operation is performed.

DPT: 1.003 (enable)

34, 47, 60, 73	**Presence X**	**Presence output**	**1 bit / 1 byte**	**CRT**

This object is used to determine the output value of the present channel. The parameterized value during movement is sent to the actuator via the output when the parameterized brightness threshold including the hysteresis is exceeded or falls short. The value set in the "value for switch on" parameter is sent to the KNX bus line when detection occurs. After the detection process is finished (i.e., after the light-on time expires), the "value for switch off" value is sent to the KNX bus line.

If the application is selected as Constant Light Switch and If the "detection independent of brightness" parameter is set to yes, the output value is sent to the KNX bus line when a motion is detected regardless of the ambient brightness.

If the Application Constant Light Controller is selected and the “Presence depending” parameter is set to no, the output value according to the brightness threshold is sent to the KNX bus line regardless of motion detection.

DPT: 1.001 (switch) / 5.001 (percentage)

35, 48, 61, 74	Presence X	Light-on time	2 bytes	CWU

This object is used to change the duration of the presence channel light-on time. The light-on time is the period between the end of motion detection and the sending of the telegram "Value for switch off". If movement is detected again within this period, the light-on time timer is started again. When a 2-byte time value is sent from the KNX bus line, this sent value will be used instead of the light-on time set via the parameter. Until a new time value is sent to this object, the last value sent will be used as the light-on period.

Note: The values which can be sent are between **10-65535** seconds. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 7.005 (time)

36, 49, 62, 75	Presence X	External movement	1 bit	CW

This object is used to inform the detector by external sources that a movement has occurred in the environment to be controlled by the detector. For this object to be visible, the "used movement detection" parameter must be selected "internal and external". When the ON telegram comes to this object from the KNX bus line, the detector determines motion detection and performs the relevant actions.

For instance, this object can be used when detectors are installed that will operate as a master-slave in the same environment. Typically, the output of sensors operating in slave mode is connected to this object of the sensor operating in master mode. Thus, the detection distance to be controlled is extended.

DPT: 1.001 (switch)

37, 50, 63, 76	Presence X	External brightness	2 bytes	CW

This object is used to send the brightness value to the detector's related presence channel via external sources. The value read on this object is compared with the brightness value threshold and operations are performed by the detector. For instance, if the brightness value is higher or lower than the hysteresis band, the detector will be dim or switch the lights (according to application: constant light control or constant light switch).

Note: If the brightness value will be used over external sources, it is recommended to send the ambient brightness values to this object periodically so that the detector can make accurate evaluations.

DPT: 9.004 (lux (Lux))

38, 51, 64, 77	Presence X	Brightness-value threshold	2 bytes	CRWTU

This object is used to change the switching threshold for the related presence channel, at which the detector is activated. The value is sent to this object in lux. This value is used as the new switching threshold. The current switching threshold can be read via this communication object. Until a new brightness threshold value is sent to this object, the last value sent will be used as the brightness-value threshold.

Note: The values which can be sent are between **1-1200** lux. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 9.004 (lux (Lux))

39, 52, 65, 78	Presence X	Manual on/off	1 bit	CWU

This object is used to dim the lights manually to on (value for switch on) and off (value for switch off) values. The object becomes visible when the "Enable 'Manual on/off' object" parameter in the input section of the presence page is selected as yes. When a telegram is sent to this object via the KNX bus, the constant light control will be disabled until the manual dim mode time expires. After the detector exits the manual mode, if there is no presence detection in the relevant environment, the value set in the "value for switch off" parameter is sent to the KNX bus line through the output object.

Note: The dimming operation will proceed according to the “Dimming time for manual from 0…%100 (0 = immediately)” parameter.

DPT: 1.001 (switch)

40, 53, 66, 79	Presence X	Relative dimming	4 bit	CWU

This object is used to dim manually with the relative dim method. The object becomes visible when the "Enable 'Relative dimming' object" parameter in the input section of the presence page is selected as yes. When a telegram is sent to this object via the KNX bus, the constant light control will be disabled until the manual dim mode time expires. After the detector exits the manual mode, if there is no presence detection in the relevant environment, the value set in the "value for switch off" parameter is sent to the KNX bus line through the output object.

Possible object values : %1, %3, %6, %12, %25, %50, %100 (increase or decrease direction)

Note: The dimming operation will proceed according to the “Dimming time for manual from 0…%100 (0 = immediately)” parameter.

DPT: 3.007 (dimming control)

41, 54, 67, 80	Presence X	Absolute dimming	1 byte	CWU

This object is used to dim manually with the absolute dim method. The object becomes visible when the "Enable 'Absolute dimming' object" parameter in the input section of the presence page is selected as yes. When a telegram is sent to this object via the KNX bus, the constant light control will be disabled until the manual dim mode time expires. After the detector exits the manual mode, if there is no presence detection in the relevant environment, the value set in the "value for switch off" parameter is sent to the KNX bus line through the output object.

Possible object values : %0…%100

Note: The dimming operation will proceed according to the “Dimming time for manual from 0…%100 (0 = immediately)” parameter.

DPT: 5.001 (percentage)

42, 55, 68, 81	Presence X	Status of control	1 bit	CRT

This object is used to detect whether the detector is operating in manual or automatic mode. During the mode transitions, the detector sends a telegram to the KNX bus line through this object. Sending an ON telegram to the KNX bus line means that the detector is in automatic mode, sending an OFF telegram means that the detector is in manual mode.

DPT: 1.001 (switch)

43, 56, 69, 82	**Presence X**	**Forced operation**	**1 bit / 2 bit**	**CRWU**

This object is used to force the detector operating in constant light control mode out of this mode and to stay it at a constant output value. For example, when an ON telegram is sent to the object selected as 1 bit via KNX, the detector will exit the constant light control mode and transmit the value determined through the "Brightness value when forced activated" parameter to the output object and the detector switches to forced status. An OFF telegram must be sent to this object to activate the constant light control again and remove the detector from this forced state.

DPT: 1.010 (start/stop) / 2.001 (switch control)

44, 57, 70, 83	**Presence X**	**Forced operation value**	**1 byte**	**CRWU**

This object is used to determine what the fixed output value will be when the detector is forced out of constant light control mode and remains at a constant output value. When a telegram is sent to this object via KNX, the new forcing state brightness value will now be this value instead of the value in the "Brightness value when forced activated" parameter. Until a new value is sent, the last sent value is used as the forced status constant output value.

DPT: 5.001 (percentage)

4.6. Temperature

This section contains information about KNX objects and their properties related to the temperature channel. The types, flags and properties of the objects are explained in detail below.

Object Number	Object Name	Function	Type	Flags
85	**Temperature**	**Alarm**	**1 bit**	**CRT**
This object is used to send the alarm temperature value calculated by the Multi Presence Detector to the KNX bus line. Also, temperature measuring sources (internal, external and KNX) can be configured via ETS parameters. **Note:** Temperature sensor calibration is required for the measurements to be healthier and more accurate. DPT: 1.005 (alarm)				
86	**Temperature**	**Actual Temperature**	**2 bytes**	**CRT**
This object is used to send the actual temperature value calculated by the Multi Presence Detector to the KNX bus line. Depending on the parameter configuration, the measured data can be sent to the bus line periodically or according to the amount of change. Also, temperature measuring sources can be configured via ETS parameters. **Note:** Temperature sensor calibration is required for the measurements to be healthier and more accurate. DPT: 9.001 (temperature (°C)				
87	**Temperature**	**KNX Probe Temperature**	**2 bytes**	**CW**
This object is used to receive the temperature value from the KNX bus line. This value can be used as a single temperature source or mixing part for the temperature calculation by the Multi Presence Detector. DPT: 9.001 (temperature (°C)				

4.7. Thermostat Objects

In this section, Thermostat objects are described in the table below. In the first column name of the object, in the second column function name, the third column data type and fourth column the objects flags, information is given.

Object Number	Object Name	Function	Type	Flags
88	**Thermostat**	**Thermostat Disabling**	**1 bit**	**CW / CRT***

This object is used to set the Multi Presence Detector thermostat status. “Enabled” or “Disabled” telegram is received via this object.

For example, it will be disabled when an “Enabled” telegram is received from the KNX bus line, and when a “Disabled” telegram is received, the Multi Presence Detector thermostat will continue working.

*This object is used as feedback object in thermostat slave mode.

DPT: 1.003 (enable)

89	**Thermostat**	**Thermostat Status**	**1 bit**	**CRT / CW***

This object is used to watch thermostat status. “Enabled” or “Disabled” telegram is transmitted to KNX bus via this object when thermostat status is changed over device.

*This object is used as input object in thermostat slave mode.

DPT: 1.003 (enable)

92	**Thermostat**	**Thermostat Operation Mode**	**1 byte**	**CW / CRT***

This object switches over the operating modes with a 1-byte value.

*This object is used as feedback object in thermostat slave mode.

DPT: 20.102 (HVAC mode)

93	**Thermostat**	**Thermostat Operation Mode Forced**	**1 byte**	**CW**

This object is used to set operation mode of thermostat. Its priority is highest including thermostat energy saving functions except window contact and the mode cannot be changed until “Auto” is received via this object. If “Auto” is received, the operation mode is back the HVAC mode that before enter the forced operation mode.

DPT: 20.102 (HVAC mode)

94	Thermostat	Thermostat Operation Mode Status	1 byte	CRT / CWU*

This object indicates the status of the operating mode with a 1-byte value.

*This object is used as input object in thermostat slave mode.

DPT: 20.102 (HVAC mode)

95	Thermostat	Operation Mode [Comfort]	1 bit	CRW

The Comfort mode activation command is sent via this object. If “On” telegram is received via this object, operation mode is changed as Comfort. If active operation mode is Comfort and “Off” telegram is received via this object, the operating mode is changed as Auto. If weekly program isn’t active, the operating mode isn’t changed and keep current state.

DPT: 1.001 (switch)

96	Thermostat	Operation Mode [Standby]	1 bit	CRW

The Standby mode activation command is sent via this object. If “On” telegram is received via this object, operation mode is changed as Standby. If active operation mode is Standby and “Off” telegram is received via this object, the operating mode is changed as Auto. If weekly program isn’t active, the operating mode isn’t changed and keep current state.

DPT: 1.001 (switch)

97	Thermostat	Operation Mode [Economy]	1 bit	CRW

The Economy mode activation command is sent via this object. If “On” telegram is received via this object, operation mode is changed as Economy. If active operation mode is Economy and “Off” telegram is received via this object, the operating mode is changed as Auto. If weekly program isn’t active, the operating mode isn’t changed and keep current state.

DPT: 1.001 (switch)

98	Thermostat	Operation Mode [Protection]	1 bit	CRW

The Protection mode activation command is sent via this object. If “On” telegram is received via this object, operation mode is changed as Protection. If active operation mode is Protection and “Off” telegram is received via this object, the operating mode is changed as Auto. If weekly program isn’t active, the operating mode isn’t changed and keep current state.

DPT: 1.001 (switch)

99	Thermostat	Thermostat Heating/Cooling Switchover	1 bit	CW / CRT*

This object is used to change over the heating/cooling modes.

*This object is used as feedback object in thermostat slave mode.

DPT: 1.100 (cooling/heating)

100	**Thermostat**	**Thermostat Heating/Cooling Status**	**1 bit**	**CRT / CW***

Heating/cooling status information is indicated via this object.

*This object is used as input object in thermostat slave mode.

DPT: 1.100 (cooling/heating)

101	**Thermostat**	**Thermostat Heating Control Disabling**	**1 bit**	**CW**

This object activates or deactivates the heating system.

DPT: 1.001 (switch)

102	**Thermostat**	**Thermostat Heating Control Running**	**1 bit**	**CRT / CW***

This object is used to inform about the heating control. If the heating control is active and the control value is higher than zero, ON telegram is transmitted to KNX bus. If the heating control is not active and the control value is zero, OFF telegram is transmitted to KNX bus.

*This object is used as input object in thermostat slave mode.

DPT: 1.001 (switch)

103	**Thermostat**	**Thermostat Heating Value - Thermostat Heating/Cooling Value**	**1 bit / 1 byte**	**CRT**

The output value of thermostat control is transmitted via the object.

DPT: 1.001 (switch) / 5.001 (percentage)

104	**Thermostat**	**Thermostat Heating Value Request - Thermostat Heating/Cooling Request**	**1 bit**	**CW**

This object is used to get the output value of heating controller. If “Acknowledge command” telegram is received via this object, current value of the heating controller is transmitted to KNX bus.

DPT: 1.016 (acknowledge)

105	**Thermostat**	**Thermostat Cooling Control Disabling**	**1 bit**	**CW**

This object activates or deactivates the cooling system.

DPT: 1.003 (enable)

106	Thermostat	Thermostat Cooling Control Running	1 bit	CRT / CW*

This object is used to inform about the cooling control. If the cooling control is active and the control value is higher than zero, ON telegram is transmitted to KNX bus. If the cooling control is not active and the control value is zero, OFF telegram is transmitted to KNX bus.

*This object is used as input object in thermostat slave mode.

DPT: 1.002 (boolean)

107	Thermostat	Thermostat Cooling Value	1 bit / 1 byte	CRT

The output value of thermostat cooling control is transmitted via the object.

DPT: 1.001 (switch) / 5.004 (percentage (0...255%))

108	Thermostat	Thermostat Cooling Value Request	1 bit	CW

This object is used to get the output value of cooling controller. If “Trigger” telegram is received via this object, current value of the heating controller is transmitted to KNX bus.

DPT: 1.016 (acknowledge)

109	Thermostat	Thermostat Additional Heating Control Disabling	1 bit	CW

This object activates or deactivates the additional heating system.

DPT: 1.001 (switch)

110	Thermostat	Thermostat Additional Heating Control Running	1 bit	CRT

This object is used to inform about the additional heating control. If the additional heating control is active and the control value is higher than zero, ON telegram is transmitted to KNX bus. If the additional heating control is not active and the control value is zero, OFF telegram is transmitted to KNX bus.

DPT: 1.001 (switch)

111	Thermostat	Thermostat Additional Heating Value	1 bit / 1 byte	CRT

The output value of thermostat additional heating control is transmitted via the object.

DPT: 1.001 (switch) / 5.001 (percentage)

112	**Thermostat**	**Thermostat Additional Heating Value Request**	**1 bit**	**CW**

This object is used to get the output value of additional heating controller. If “Trigger” telegram is received via this object, current value of the heating controller is transmitted to KNX bus.

DPT: 1.016 (acknowledge)

113	**Thermostat**	**Thermostat Additional Cooling Control Disabling**	**1 bit**	**CW**

This object activates or deactivates the additional cooling system.

DPT: 1.003 (enable)

114	**Thermostat**	**Thermostat Additional Cooling Control Running**	**1 bit**	**CRT**

This object is used to inform about the additional cooling control. If the additional cooling control is active and the control value is higher than zero, ON telegram is transmitted to KNX bus. If the additional cooling control is not active and the control value is zero, OFF telegram is transmitted to KNX bus.

DPT: 1.002 (boolean)

115	**Thermostat**	**Thermostat Additional Cooling Value**	**1 bit / 1 byte**	**CRT**

The output value of thermostat additional cooling control is transmitted via the object.

DPT: 1.001 (switch) / 5.004 (percentage (0...255%))

116	**Thermostat**	**Thermostat Additional Cooling Value Request**	**1 bit**	**CW**

This object is used to get the output value of additional cooling controller. If “Trigger” telegram is received via this object, current value of the heating controller is transmitted to KNX bus.

DPT: 1.016 (acknowledge)

117	**Thermostat**	**Room Temperature Output (C) - Room Temperature Output (F)**	**1 bit**	**CRT / CW***

This object is used to inform about the temperature value that room controller uses.

*This object is used as input object if thermostat temperature source is selected as “Temperature object”.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature difference (K))

117	Thermostat	Room Temperature Input (C) - Room Temperature Input (F)	1 bit	CRT / CW*

This object is used to inform about the temperature value that room controller uses.

*This object is used as input object if thermostat temperature source is selected as “Temperature object”.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature difference (K))

118	Thermostat	Actual Setpoint Output	2 bytes	CRT / CW*

The pre-configured setpoint temperature is obtained with this object.

*This object is used as input object in thermostat slave mode.

DPT: According to parameter selection, DPT changes.

119	Thermostat	Manual Setpoint Input	2 bytes	CW / CRT*

The setpoint temperature is configured manually with this object. If HVAC mode is Build Protection, the setpoint can’t be changed via this object.

If the difference between the active setpoint and received value is higher than the “Manual setpoint range” parameter, Manual Setpoint value is set maximum or minimum limit value according to “Manual setpoint range” parameter.

*This object is used as feedback object in thermostat slave mode.

DPT: According to parameter selection, DPT changes.

120	Thermostat	Manual Setpoint Reset	1 bit	CW

The setpoint temperature that is desired to configure manually can be reset with this object.

DPT: 1.015 (reset)

121	Thermostat	Heating Comfort Setpoint Temperature	2 bytes	CW

The setpoint temperature value for heating comfort mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

122	Thermostat	Heating Standby Setpoint Temperature	2 bytes	CW

The setpoint temperature value for heating standby mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

123	Thermostat	Heating Economy Setpoint Temperature	2 bytes	CW

The setpoint temperature value for heating economy mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

124	Thermostat	Heating Protection Setpoint Temperature	2 bytes	CW

The setpoint temperature value for heating protection mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

125	Thermostat	Cooling Comfort Setpoint Temperature	2 bytes	CW

The setpoint temperature value for cooling comfort mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

126	Thermostat	Cooling Standby Setpoint Temperature	2 bytes	CW

The setpoint temperature value for cooling standby mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

127	Thermostat	Cooling Economy Setpoint Temperature	2 bytes	CW

The setpoint temperature value for cooling economy mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

128	Thermostat	Cooling Protection Setpoint Temperature	2 bytes	CW

The setpoint temperature value for cooling protection mode is configured with this object.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

129	Thermostat	Fan Controller Disable	1 bit	CW / CRT*

This object is used to set the Multi Presence Detector fan controller status. “Enabled” or “Disabled” telegram is received via this object.

For example, it will be disabled when an “Enabled” telegram is received from the KNX bus line, and when a “Disabled” telegram is received, the Multi Presence Detector fan controller will continue working.

*This object is used as feedback object in thermostat slave mode.

DPT: 1.003 (enable)

130	Thermostat	Fan Controller Status	1 bit	CRT

This object is used to watch fan controller status. “Enabled” or “Disabled” telegram is transmitted to KNX bus via this object when fan controller status is changed over device.

DPT: 1.003 (enable)

131	Thermostat	Fan Controller Working Mode	1 bit	CW

This object is used to switch over to automatic or manual fan speed control mode.

DPT: 1.001 (switch)

132	Thermostat	Fan Controller Working Mode Status	1 bit	CRT

This object indicates the manual / automatic fan operating mode with 1 bit value.

DPT: 1.001 (switch)

133	Thermostat	Fan Controller Proportional Output	1 byte	CRT

This object is used to send the output value of the fan proportional controller.

DPT: 5.001 (percentage (0...100%))

134	Thermostat	Fan Controller Manual Step - Fan Controller Manual Up/Down	1 bit	CW

This object is used to increase or decrease the fan speed

DPT: 1.007 (step) / 1.008 (up/down)

135	Thermostat	Fan Controller Manual Stage	1 byte	CW

This object allows the manual fan speed to be controlled with 1-byte value.

DPT: 5.100 (fan stage (0...255))

136	Thermostat	Fan Controller Speed (1 Byte)	1 byte	CRT

This object allows the fan speed to be controlled with 1-byte value.

DPT: 5.001 (percentage (0..100%) / 5.100 (fan stage (0...255))

137	Thermostat	Fan Controller Speed Feedback Input (1 Byte)	1 byte	CWU

This object waits the fan speed feedback with a 1-byte value.

DPT: 5.001 (percentage (0...100%))

138, 139, 140, 141, 142	**Thermostat**	**Fan Level 1…5**	**1 bit**	**CRT**

This object indicates the Fan Level Y value with a 1-bit value.

DPT: 1.001 (switch)

143, 144, 145, 146, 147	**Thermosta**	**Fan Level 1…5 Feedback Input**	**1 bit**	**CWU**

This object indicates the Fan Level X status with a 1-bit value.

DPT: 1.001 (switch)

154	**Thermostat**	**Temperature Limit Heating Source**	**2 bytes**	**CW**

This group object receives the limit temperature for heating stage. The temperature value received here is used to evaluate the limit temperature. The limit becomes active when the temperature set in the parameter is exceeded.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

155	**Thermostat**	**Temperature Limit Cooling Source**	**2 bytes**	**CW**

This group object receives the limit temperature for cooling stage. The temperature value received here is used to evaluate the limit temperature. The limit becomes active when the temperature set in the parameter is fallen below.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

156	**Thermostat**	**Temperature Limit Additional Heating Source**	**2 bytes**	**CW**

This group object receives the limit temperature for additional heating stage. The temperature value received here is used to evaluate the limit temperature. The limit becomes active when the temperature set in the parameter is exceeded.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

157	**Thermostat**	**Temperature Limit Additional Cooling Source**	**2 bytes**	**CW**

This group object receives the limit temperature for additional cooling stage. The temperature value received here is used to evaluate the limit temperature. The limit becomes active when the temperature set in the parameter is fallen below.

DPT: 9.001 (temperature (°C)) / 9.027 (temperature (°F))

4.8. Virtual Card Holder

This section contains information about KNX objects and their properties related to the virtual cardholder channel. The types, flags and properties of the objects are explained in detail below.

Object Number	Object Name	Function	Type	Flags
159	**Virtual Holder**	**Room Booked**	**1 bit**	**CW**
This object is used to receive the room booked information. By setting this communication object to “1” the status of the virtual holder is set to “room booked”; when set to “0” the status is “room not booked”. The virtual holder module can have different behaviour if the room is booked or not; the default value for this status can be set using the “Booked state after download” parameter. DPT: 1.002 (boolean)				
160	**Virtual Holder: Output**	**Presence Output**	**1 bit**	**CRT**
This object is used to receive the room booked information. By setting this communication object to “1” the status of the virtual holder is set to “room booked”; when set to “0” the status is “room not booked”. The virtual holder module can have different behaviour if the room is booked or not; the default value for this status can be set using the “Booked state after download” parameter. DPT: 1.002 (boolean)				
161	**Virtual Holder: Output**	**Unexpected Presence**	**1 bit**	**CT**
This object is used to send “1” when the sensor detects an unexpected Presence. DPT: 1.005 (alarm)				
162	**Virtual Holder: Output**	**HVAC**	**1 byte**	**CRT**
This object is the object that gives output according to the "HVAC booked / non-booked" values in the "Welcome" and "Goodbye" scenarios selected on the "Virtual Card Holder" - "Output" screen. DPT: 20.102 (HVAC mode)				
163	**Virtual Holder: Output**	**Additional**	**1 byte**	**CT**
This object is used to send commands in event of presence and absence. Commands can be set different if the room is booked or not and if the person who enters the room is a guest, service or maintenance. DPT: 5.001 (percentage (0..100%) / 5.010 (counter pulses (0..255)) / 17.001 (scene number)				

164	**Virtual Holder: Remote Input 1**	**Door / Presence**	**1 bit**	**CW**

This object is used when you need to consider 2 adjacent rooms as separated or joined as if they were a single room (suite room). This possibility must be considered during installation, so communication objects must be connected as described below if you want to switch runtime from one configuration (2 single rooms) to another (one double room) and vice versa.

Note: A Remote Input communication object (type door) must be connected only to one single door contact on/off a telegram.

A Remote Input communication object (type presence) must be connected only to one single device with a presence on/off a telegram.

DPT: 1.002 (boolean)

165	**Virtual Holder: Remote Input 2**	**Door / Presence**	**1 bit**	**CW**

This object is used when you need to consider 2 adjacent rooms as separated or joined as if they were a single room (suite room). This possibility must be considered during installation, so communication objects must be connected as described below if you want to switch runtime from one configuration (2 single rooms) to another (one double room) and vice versa.

Note: A Remote Input communication object (type door) must be connected only to one single door contact on/off a telegram.

A Remote Input communication object (type presence) must be connected only to one single device with a presence on/off a telegram.

DPT: 1.002 (boolean)

4.9. Logic Function

This section contains information about KNX objects and their properties related to the logic function channels. The types, flags and properties of the objects are explained in detail below. There are 5 identical logic channels in the Multi Presence Detector, so only one logical channel is described here. The X values can be between 1…4 and Y values also can be 1…5. Please do not forget to take this into account.

Object Number	Object Name	Function	Type	Flags
166, 194, 222, 250	**Logic X:**	**Lock**	**1 bit**	**CW**

This object is used to lock the related logic channel x. It becomes visible when the "use logic lock" parameter is set to yes. Depending on the parameter setting, when an ON or OFF telegram is sent to this object, the corresponding logical channel is locked.

For example, when "ON telegram" is selected in the parameter page for locking, it will be locked when an ON telegram is received from the KNX bus line, and when an OFF telegram is received, the logic channel will be unlocked.

DPT: 1.003 (enable)

167, 195, 223, 251	**Logic X:**	**Feedback of Lock**	**1 bit**	**CRT**

This object is used to send feedback on the lock status for the related logic channel x. It becomes visible when the "use logic lock" parameter is set to yes.

If a status change occurs on the lock function, the changed statue value will be sent from this object.

DPT: 1.003 (enable)

168, 196, 224, 252	**Logic X: Input**	**External movement**	**1 bit**	**CW**

This object is used to receive movement information from the KNX bus line. According to the ETS parameter configuration, the ‘0’ or ‘1’ value is accounted as there is a movement detection occurs.

DPT: 1.001 (switch)

169, 197, 225, 253	**Logic X: Input**	**External brightness**	**2 bytes**	**CWTU**

This object is used to obtain a brightness value from the KNX bus line. The received brightness value will be used to evaluate the input status according to the brightness thresholds.

DPT: 9.004 (lux (Lux))

170, 198, 226, 254	Logic X: Input	Brightness threshold lower	2 bytes	CWTU

This object is used to receive the brightness threshold lower value from the KNX bus line. The value read on this object is will be used as a new brightness threshold lower value. This object becomes visible when the "Change brightness threshold via bus" parameter is set to yes

Note: The values which can be sent are between **1-1200** lux. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 9.004 (lux (Lux))

171, 199, 227, 255	Logic X: Input	Brightness threshold upper	2 bytes	CWTU

This object is used to receive the brightness threshold upper value from the KNX bus line. The value read on this object is will be used as a new brightness threshold upper value. This object becomes visible when the "Change brightness threshold via bus" parameter is set to yes

Note: The values which can be sent are between **1-1200** lux. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 9.004 (lux (Lux))

172, 200, 228, 256	Logic X: Input	External temperature	2 bytes	CW

This object is used to obtain temperature values from the KNX bus line. The received temperature value will be used to evaluate the input status according to the temperature thresholds.

DPT: 9.001 (temperature (°C)

173, 201, 229, 256	Logic X: Input	Temperature threshold lower	2 bytes	CWTU

This object is used to receive the temperature threshold lower value from the KNX bus line. The value read on this object is will be used as a new temperature threshold lower value. This object becomes visible when the "Change temperature via bus" parameter is set to yes

Note: The values which can be sent are between **-30 ˚C - 70 ˚C**. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 9.001 (temperature (°C)

174, 202, 230, 257	**Logic X: Input**	**Temperature threshold upper**	**2 bytes**	**CWTU**

This object is used to receive the temperature threshold upper value from the KNX bus line. The value read on this object is will be used as a new temperature threshold upper value. This object becomes visible when the "Change temperature via bus" parameter is set to yes

Note: The values which can be sent are between **-30 ˚C - 70 ˚C**. If a value that is too small or too large is sent, the value is automatically adjusted to the limit value.

DPT: 9.001 (temperature (°C)

175, 203, 231, 258	**Logic X: Input**	**External input-1**	**1 bit / 1 byte / 2 byte / 4 byte**	**CWU**

This object is used to obtain external input 1 information from the KNX bus line. According to the ETS parameter configuration, the received values are accounted as TRUE or FALSE for this external input. For 1 bit configuration, there is only ‘1’ or ‘0’ values for calculating the input status. But for other input (such as 1 byte, etc.) the received value is compared to the external input value parameter.

DPT: According to parameter selection, DPT changes.

176, 204, 232, 259	**Logic X: Input**	**External input-2**	**1 bit / 1 byte / 2 byte / 4 byte**	**CWU**

This object is used to obtain external input 2 information from the KNX bus line. According to the ETS parameter configuration, the received values are accounted as TRUE or FALSE for this external input. For 1 bit configuration, there is only ‘1’ or ‘0’ values for calculating the input status. But for other input (such as 1 byte, etc.) the received value is compared to the external input value parameter.

DPT: According to parameter selection, DPT changes.

177, 205, 233, 260	**Logic X: Input**	**External input-3**	**1 bit / 1 byte / 2 byte / 4 byte**	**CWU**

This object is used to obtain external input 3 information from the KNX bus line. According to the ETS parameter configuration, the received values are accounted as TRUE or FALSE for this external input. For 1 bit configuration, there is only ‘1’ or ‘0’ values for calculating the input status. But for other input (such as 1 byte, etc.) the received value is compared to the external input value parameter.

DPT: According to parameter selection, DPT changes.

178, 206, 234, 261	**Logic X: Output**	**Result status**	**1 bit**	**CT**

This object is used to send the related logic function block's result status to the KNX bus line. According to the ETS parameter configuration, this value can be sent periodically, on change or only configured value(TRUE or FALSE).

DPT: 1.002 (boolean)

179, 182, 185, 188, 191/ 207, 210, 213, 216, 219/ 235, 238, 241, 244, 247/ 259, 262, 265, 268, 271	**Logic X: Output: Y**	**Switching – Absolute dimming – Shutter – Alarm – Sequence – Scene – String – Threshold – Alarm – Sequence**	**1 bit / 1 byte / 2 byte**	**CRT**

The output value of thermostat control is transmitted via the object.

DPT: 1.001 (switch) / 5.001 (percentage)

180, 183, 186, 189, 192/ 208, 211, 214, 217, 220/ 236, 239, 242, 245, 248/ 264, 267, 270, 273, 276	**Logic X: Output: Y**	**Delay time on the TRUE state**	**2 bytes**	**CWT**

This object is used to receive the 'delay time on TRUE state' value from the KNX bus line. When a new value is received from this object, the received value is used as the output on delay time for the TRUE state value. The configured parameter value will not be used anymore. This object becomes visible when the "Change on time via bus" parameter is set to yes

DPT: 7.005 (time (s))

181, 184, 187, 190, 193/ 209, 212, 215, 218, 221/ 237, 240, 243, 246, 249/ 265, 268, 271, 274, 277	**Logic X: Output: Y**	**Delay time on FALSE state**	**2 bytes**	**CWT**

This object is used to receive the 'delay time on FALSE state' value from the KNX bus line. When a new value is received from this object, the received value is used as the output on delay time for the FALSE state value. The configured parameter value will not be used anymore. This object becomes visible when the "Change on time via bus" parameter is set to yes

DPT: 7.005 (time (s))

CONTACT INFORMATION

THE INTERRA WEBSITE

Interra provides documentatıon support via our web site www.interratechnology.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favourite Internet browser, the website contains the following information:

- Information about our products and projects.
- Overview of Interra company and values.
- Product Support: Datasheets, product manuals, application descriptions, latest software releases, ETS databases and archived software.

EUROPE, Turkey

Interra
Cumhuriyet Mah. Kartal Cad. Interra R&D Centre No:95/1 Kartal/İstanbul
Tel: +90 (216) 326 26 40 Fax: +90 (216) 324 25 03
Web adress: http://www.interratechnology.com