Developer of Uniqueness

KNX Binary Input

Product Manual

PM2405200176BEN

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded versions by updates.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 ITR112-XXXX KNX Binary Input 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 version A of a product manual created on the date 17/10/18.

1. Content of The Document

This document contains Interra ITR112-XXXX coded KNX Binary Input 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 ITR112-XXXX KNX Binary Input. 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 KNX Binary Input 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

ITR112-XXXX series KNX Binary Input device is the newest product of Interra Technology. The Interra KNX Binary Inputs are designed for using at mainly in interior areas of buildings.

The Interra KNX Binary Input serves as interface for operation of KNX systems via conventional buttons/switches or coupling of binary signals (signal contacts). The devices feature a push-button for manual operation for each input. Input states can be simulated during manual operation, so that the conventional push buttons, switches or floating contacts do not need to be connected for commissioning purposes. The connection to the KNX Binary Input is established using the front-side bus connection terminal.

All versions have a rear connector with 12 digital inputs that can be connected to buttons and used for switch sensor, switch/dimming sensor, shutter sensor, value/forced operation, control scene, RGB colour control, RGBW control, mode selection and command sequence.

Interra KNX Binary Input has 5 logic function blocks and can be set to 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 KNX Binary Input.

Product Name	**KNX Binary Input**
Product Code	ITR112-XXXX
Power Supply	KNX Power Supply
Current Consumption	10 mA
Inputs	12
Cable Length	Maximum 100 m at 1.5 mm^2
Cable Cross-Section	0.25 – 1.5 mm^2
Cable Stripping	6 mm
Type of Inputs	Dry Contact Inputs
Mode of Commissioning	S-Mode
Type of Protection	IP 20
Temperature Range	Operation (-5°C...45°C) Storage (-25°C...55°C)
Colour	Light Grey
Dimensions	90 x 36 x 71 mm (H x W x D)
Certification	KNX Certified
Configuration	Configuration with ETS

2.2. Connection Features

The figure below shows the KNX Binary Input connectors. All of the ITR112-XXXX models have the same connection layout.

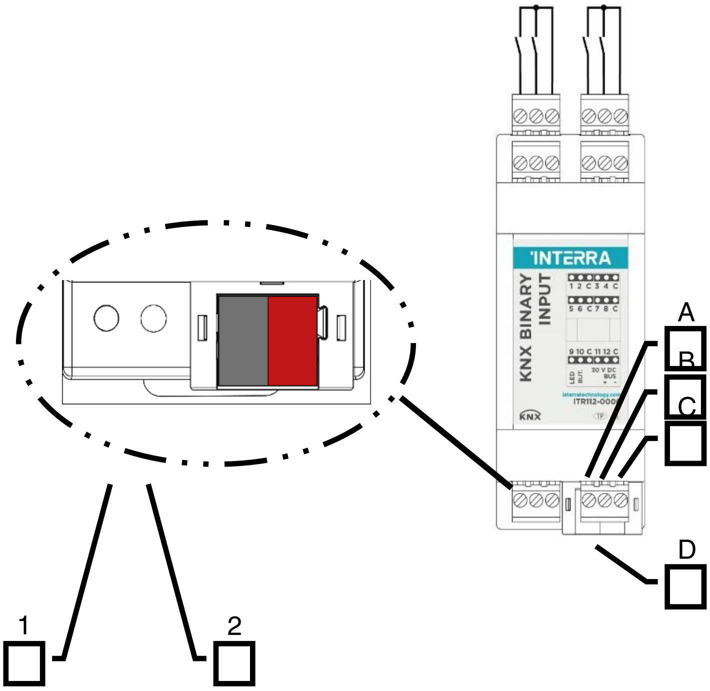

Fig. 1: Connection Features of KNX Binary Input

Letter	Feature
A	Input X
B	Input Y
C	Common
D	KNX Connector
1	Programming LED
2	Programming Button

Table 1: Connection Features Table

2.3. Additional Temperature Probe

Fig. 2: 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 2: Additional Temperature Probe Technical Information Table

2.4. Dimensions

All values given in the device dimensions are millimetres.

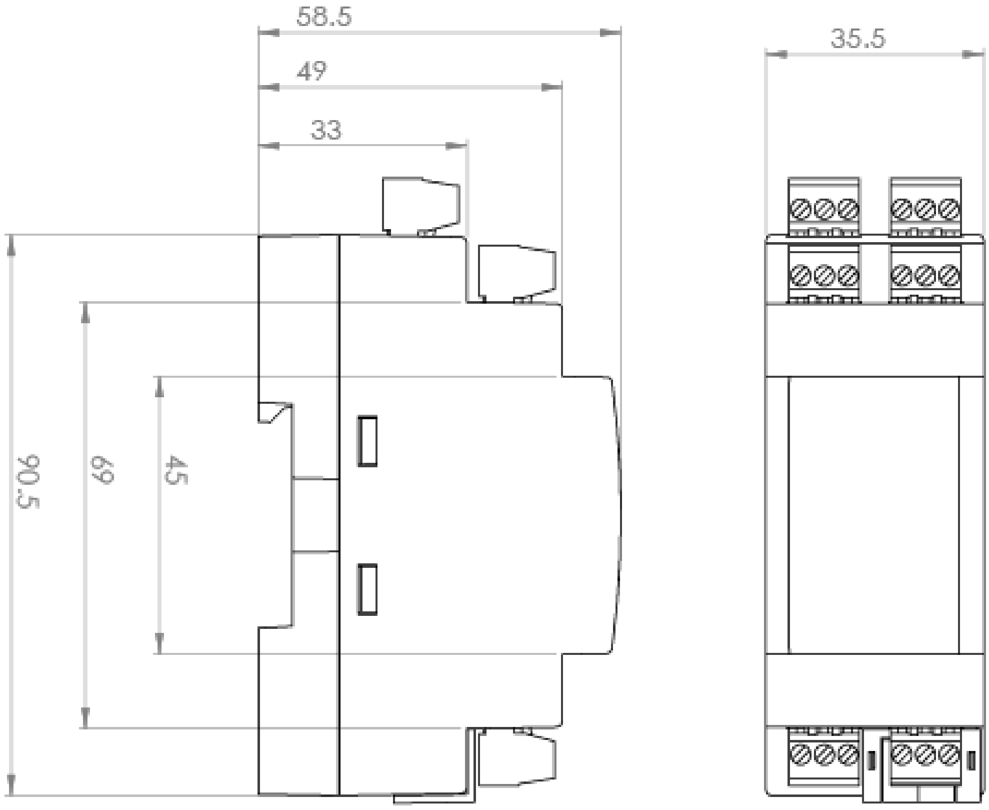

Fig. 3: Dimensions of KNX Binary Input

2.5. Functionality

The complete configuration of the device is performed via ETS5 or higher. Depending on ETS configuration and settings, the product features will be different. Available functions are:

Input Functions

- Switch Sensor
- Switch / Dimming Sensor
- Shutter Sensor
- Value / Forced Operation
- Control Scene
- RGB Colour Control
- RGBW Control
- Mode Selection
- Command Sequence
- Counter

Logic Functions

Internal Inputs (max. 12)

External Inputs

- Binary Value (adj. size) (max. 3 selectable)
- Movement
- Temperature
- Brightness

Output Types (max. 5 selectable)

- Switch
- Dim
- Shutter
- Alarm
- Percentage
- Sequence
- Scene Number
- String
- Threshold

Most functions only need one input, and therefore each input might be assigned a different function. However, some functions can also use two inputs, such as “Dimming with 2 buttons” and “Shutter/Blinds with 2 buttons”.

3. ETS Parameters & Descriptions

In this chapter, the ETS parameters of ITR112-XXXX KNX Binary Input devices are described using the parameter pages and options. The parameter page features are dynamic structures which means further parameters and parameter pages are enabled depending on the configuration.

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 ITR112-XXXX KNX Binary Input 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 appear as shown below. General settings for the devices are made in this window.

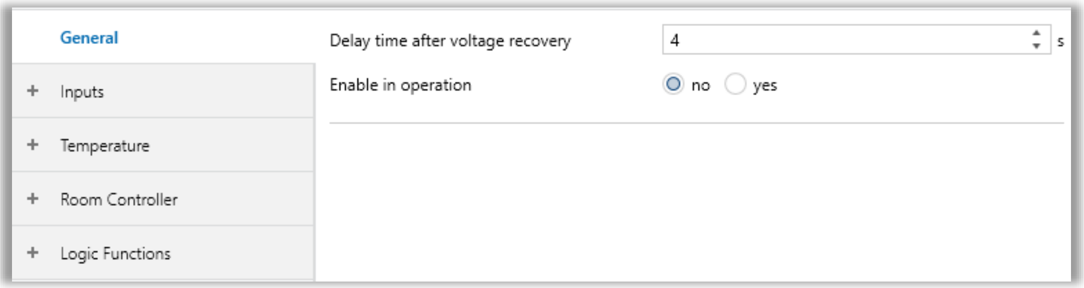

Fig. 4: General Configuration Page

3.1.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
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 KNX Binary Input 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...**4**…60
Enable In Operation	This parameter is used to determine the existence of the KNX Binary Input 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	This parameter is used to determine the send value of the "General - In operation" group object on the KNX bus line.	**Alive value '0'** Alive value '1'
-> In operation send interval (min)	This parameter is used to set the cyclically sending time interval value of the "General - In operation" group object.	1...**5**...255
Input 11 is	This parameter is used to determine whether the selected input is analog or digital.	**Digital Input** Analog Input
Input 12 is	This parameter is used to determine whether the selected input is analog or digital.	**Digital Input** Analog Input

3.2. Inputs

Interra KNX Binary Input has 12 digital inputs or 10 digital and 2 analog inputs. 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 Binary Input.

3.2.1. Input – Switch Sensor

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

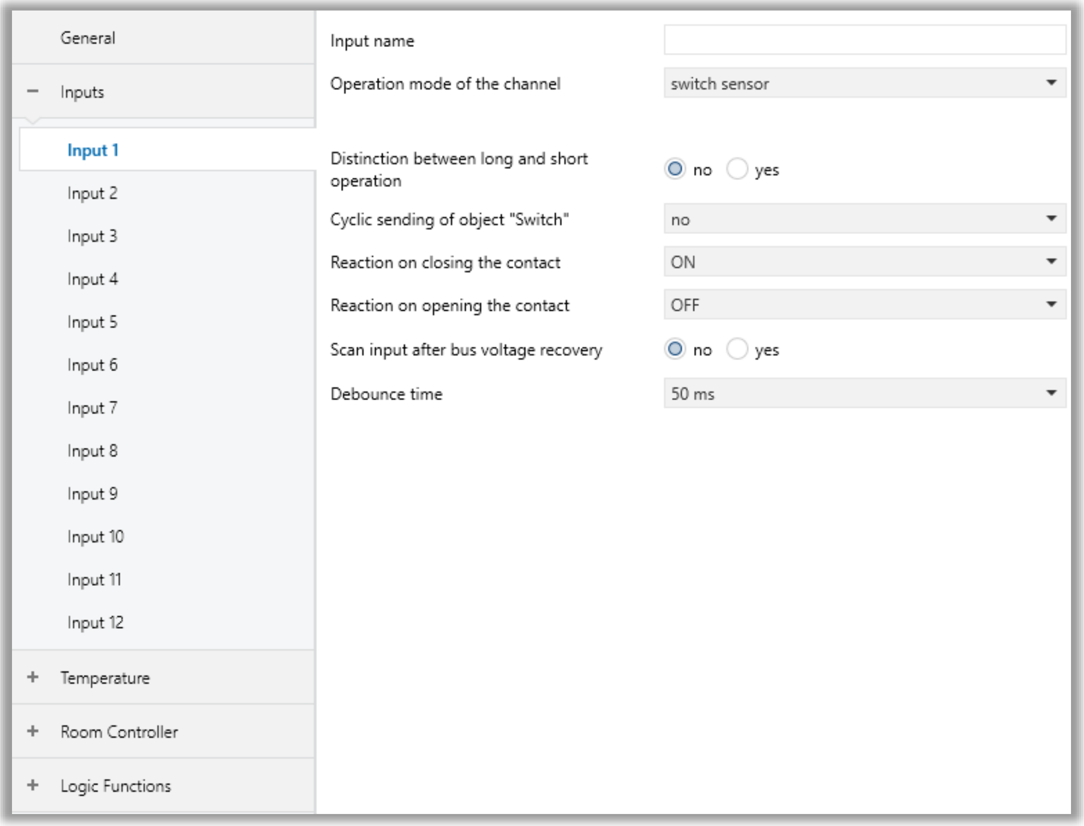

Fig. 5: Input – Switch Sensor Configuration Page

3.2.1.1. 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
-> Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Binary Input input x.	Normally closed **Normally open**
-> 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. 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
-> 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	00:00:005…**00:00:500** …01:05:535
-> 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
-> 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.	00:00.005…**00:00.500** …01:05.535
-> 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

3.2.2. Input – Switch / Dimming Sensor

In this section, it is explained how to control the unit of lighting unit through the KNX Binary Input, 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.

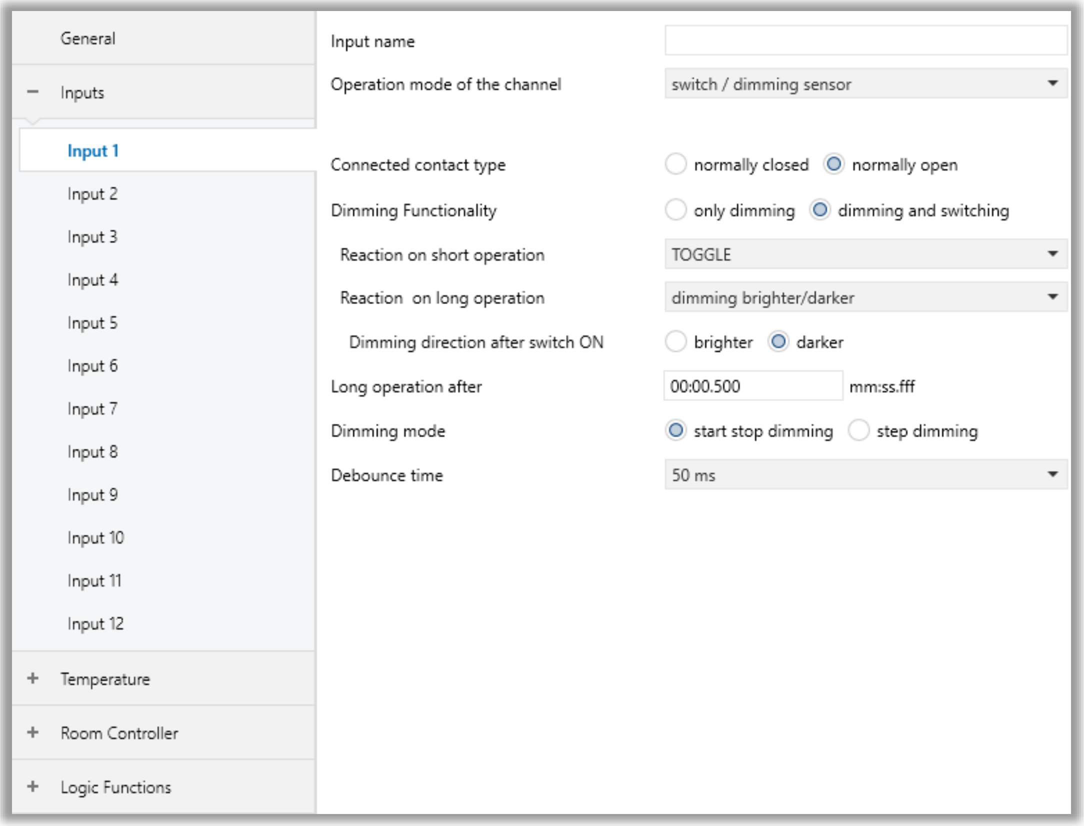

Fig. 6: Input – Switch / Dimming Sensor

3.2.2.1. 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 Binary 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 the "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**
-> Dimming direction after switch ON	This parameter is used to determine the dimming direction when the switch object is ON on long operation.	**Brighter** Darker
-> Long operation after	This parameter is used to determine long operation detection after the button press operation. For	00:00.005…**00:00.500** …01:05.535

	making a long operation, the button should be pressed at least the configured value.	
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** Step Dimming
-> 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 %3.125 %1.563
-> 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

3.2.3. 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 KNX Binary Input. Detailed information on the relevant parameter configurations is described in the table below.

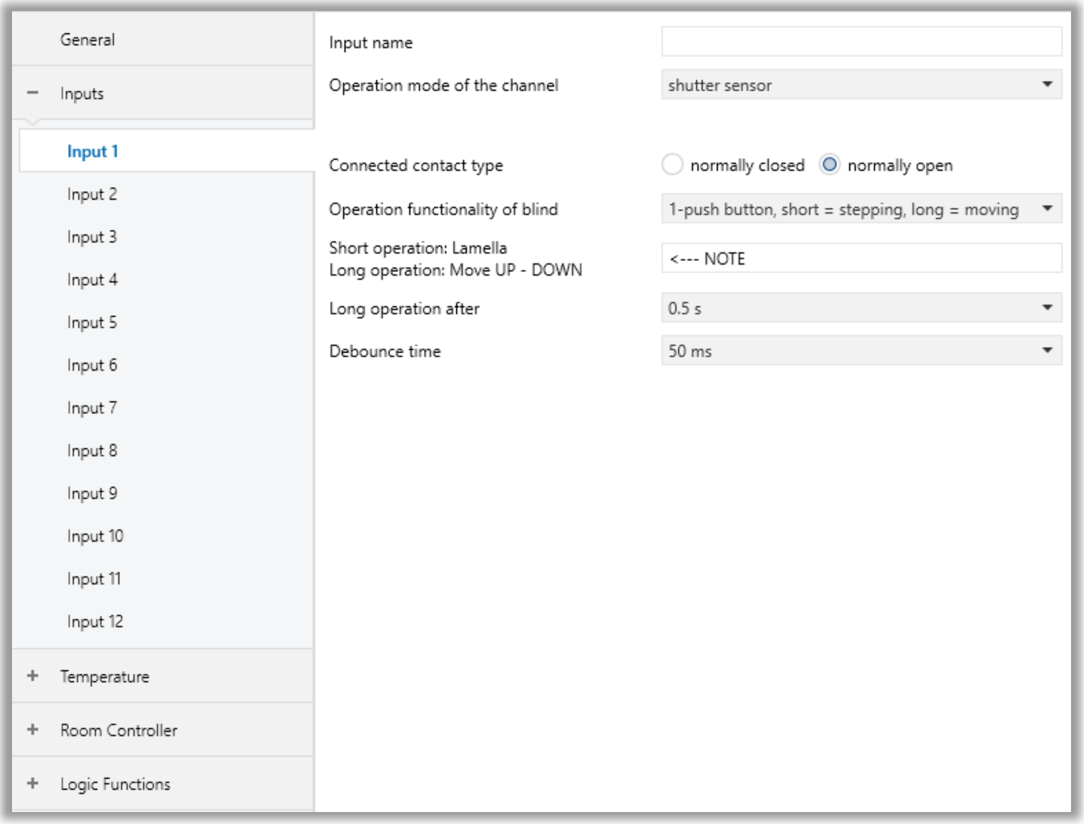

Fig. 7: Input – Shutter Sensor

3.2.3.1. 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**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX binary input x.	Normally closed **Normally open**
Operation Functionality of blind	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-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
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 operation, 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 determine the short press operation sending the value of the input x.	**Stop / lamella up** Stop / lamella down

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 operation, 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
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

3.2.3.2. 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 operation: 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”

3.2.4. Input Value / Forced Operation

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

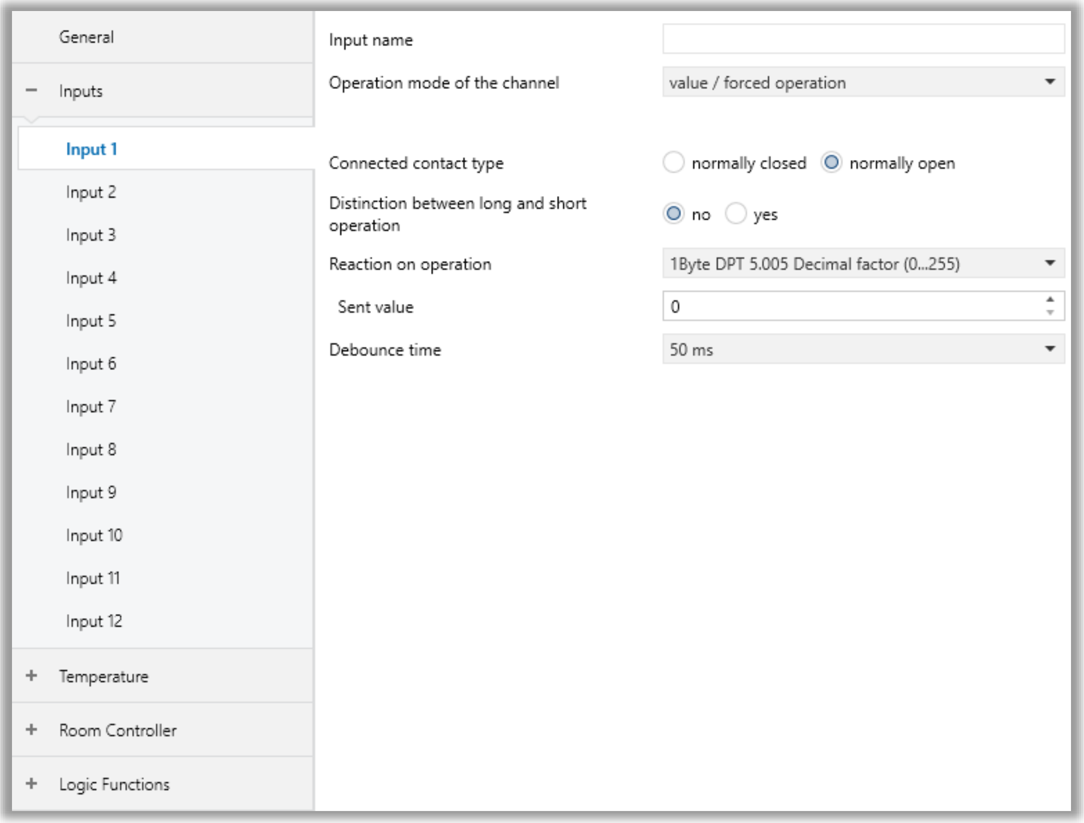

Fig. 8: Input – Value / Forced Operation

3.2.4.1. 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 Binary 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 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 Colour temperature(Kelvin) 2-byte DPT 9.001 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 depend 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
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.
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

3.2.5. Input – Control Scene

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

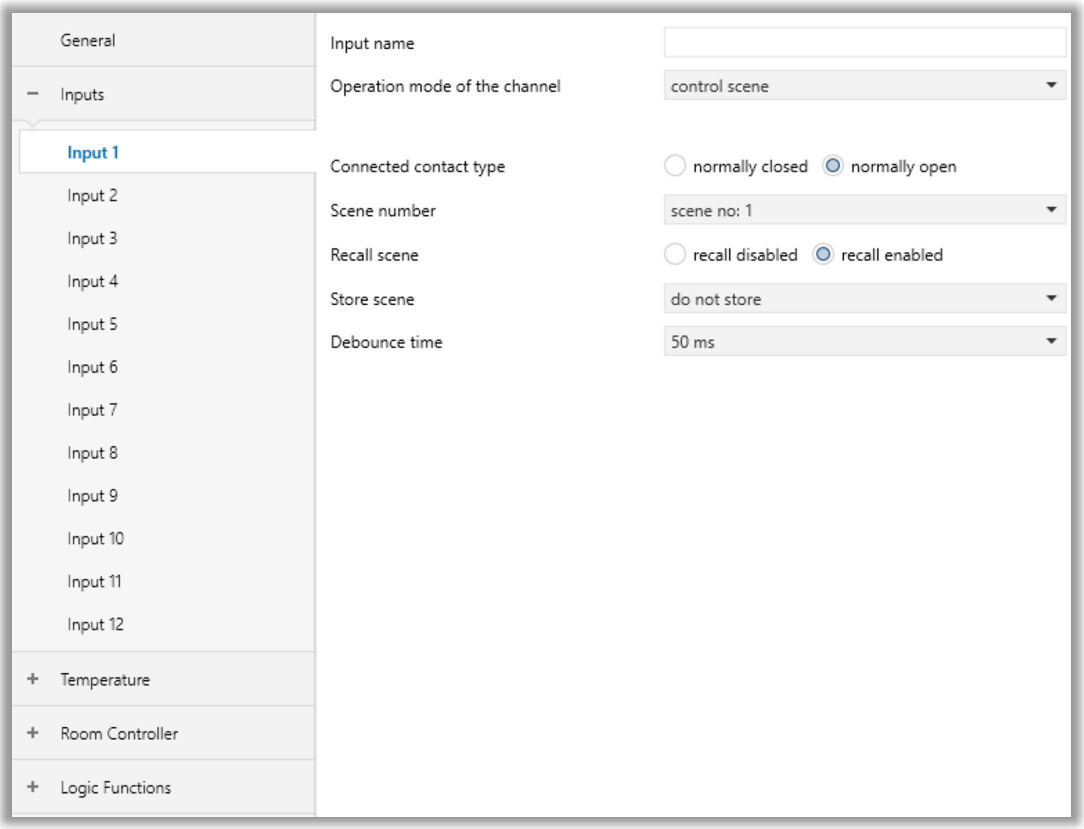

Fig. 9: Input – Control Scene

3.2.5.1. 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 Binary 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 recall of the scene. If this parameter is selected as “recall enabled” the configured scene number will be called.	Recall disabled **Recalled enabled**
Store Scene	This parameter is used to determine whether to store or not 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.	**Do not store** 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.	00:00.005…**00:00.500**…01:05.535
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

3.2.6. 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 KNX Binary Input. Detailed information on the relevant parameter configurations is described in the table below.

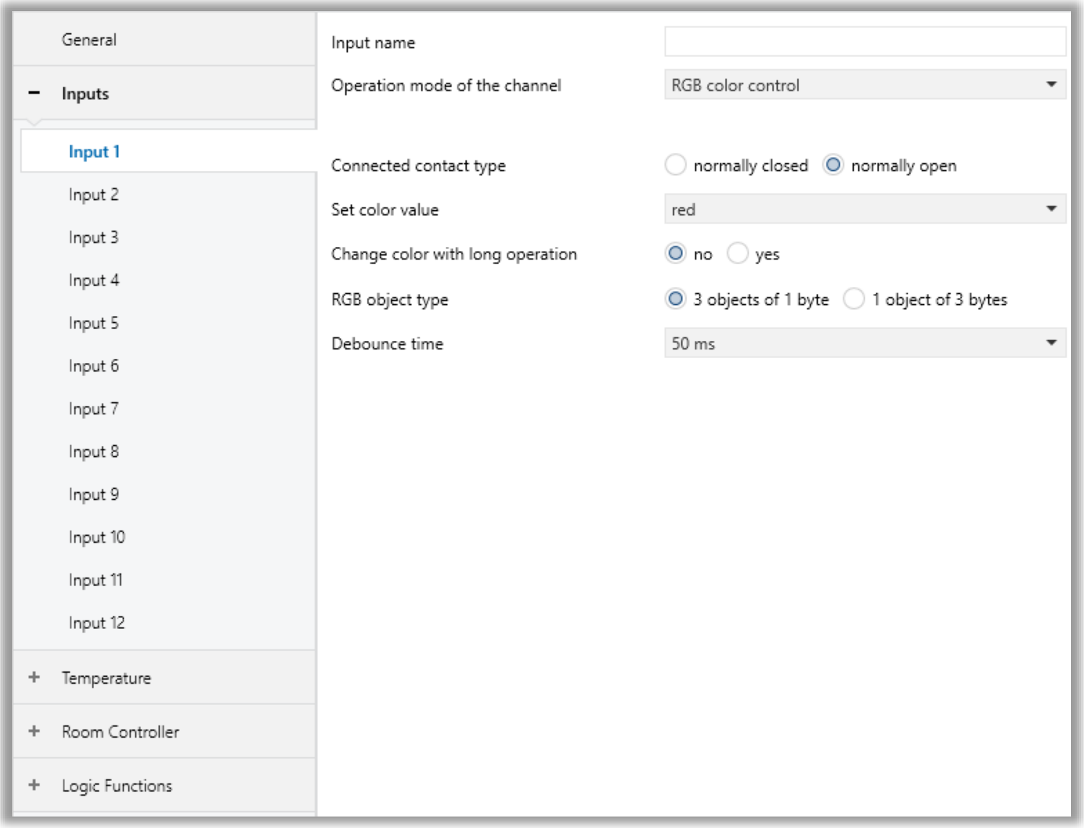

Fig. 10: Input – RGB Colour Control

3.2.6.1. 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 Binary 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.	00:00.005…**00:00.500** …01:05.535
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

3.2.7. 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 KNX Binary Input. Detailed information on the relevant parameter configurations is described in the table below.

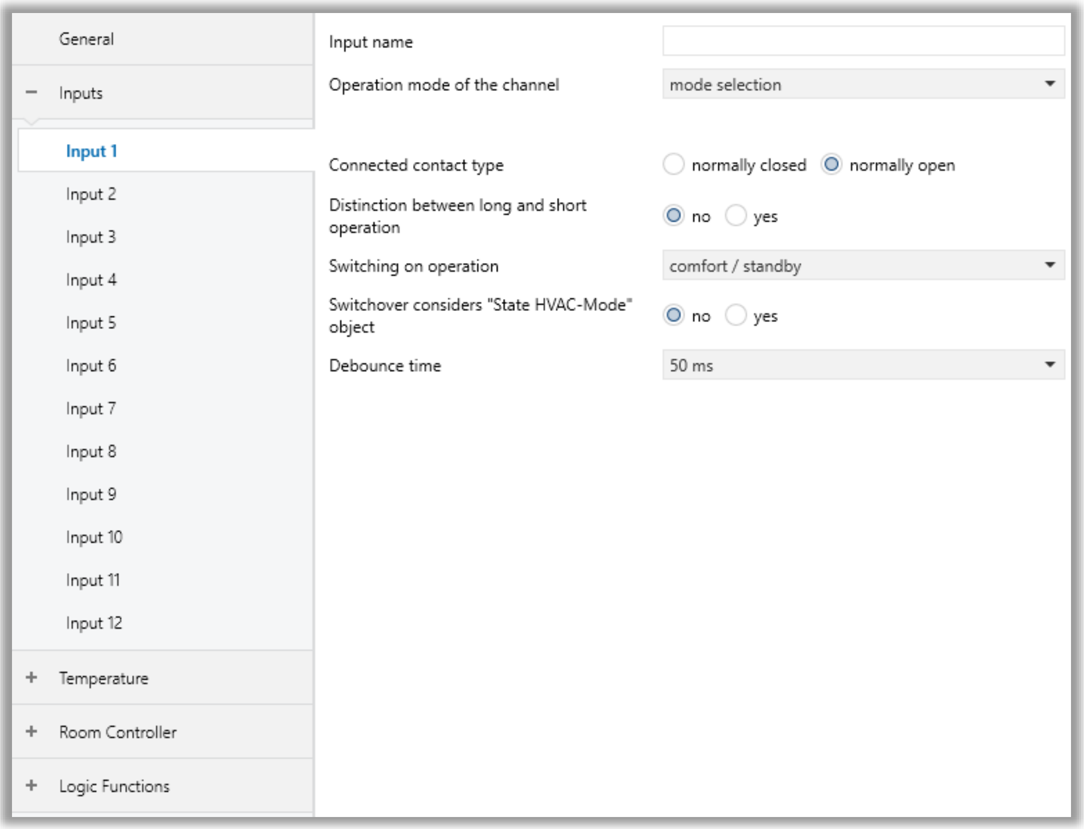

Fig. 11: Input – Mode Selection

3.2.7.1. 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**
Connected contact type	This parameter is used to specify the contact type that is connected to the KNX Binary 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 **Switching on 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.	00:00.005…**00:00.500** …01:05.535

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

3.2.8. 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.

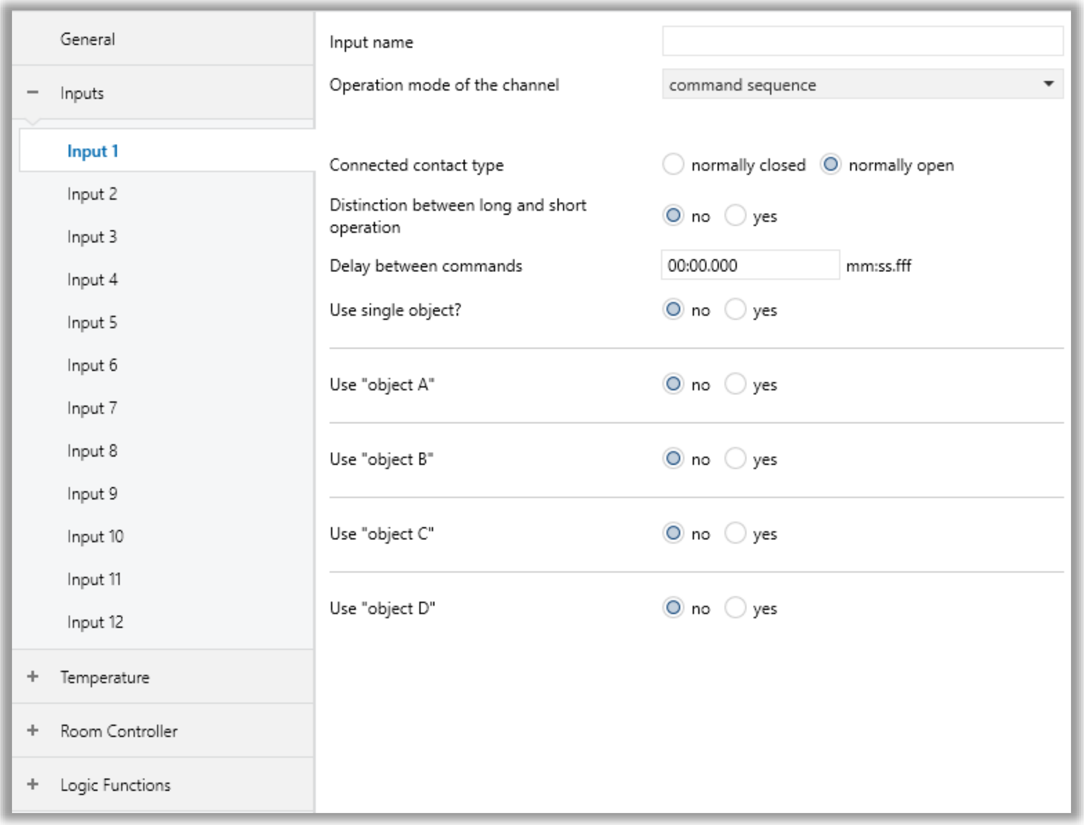

Fig. 12: Input – Command sequence

3.2.8.1. 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 Binary 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
Delay between commands	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.	**00:00.000**…00:20.000
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.	00:00.005…**00:00.500** …01:05.535
Use single object?	This parameter decides whether each object is sent to a single object or objects assigned to each command.	**No** Yes
-> Value Amount	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.	2 **3** 4

-> Data type	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depends on DPT selection.
Use "object X"	This parameter is used to enable each command object when they are set to yes.	**No** Yes
-> Data type	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depends on DPT selection.

3.2.9. Input – Counter

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

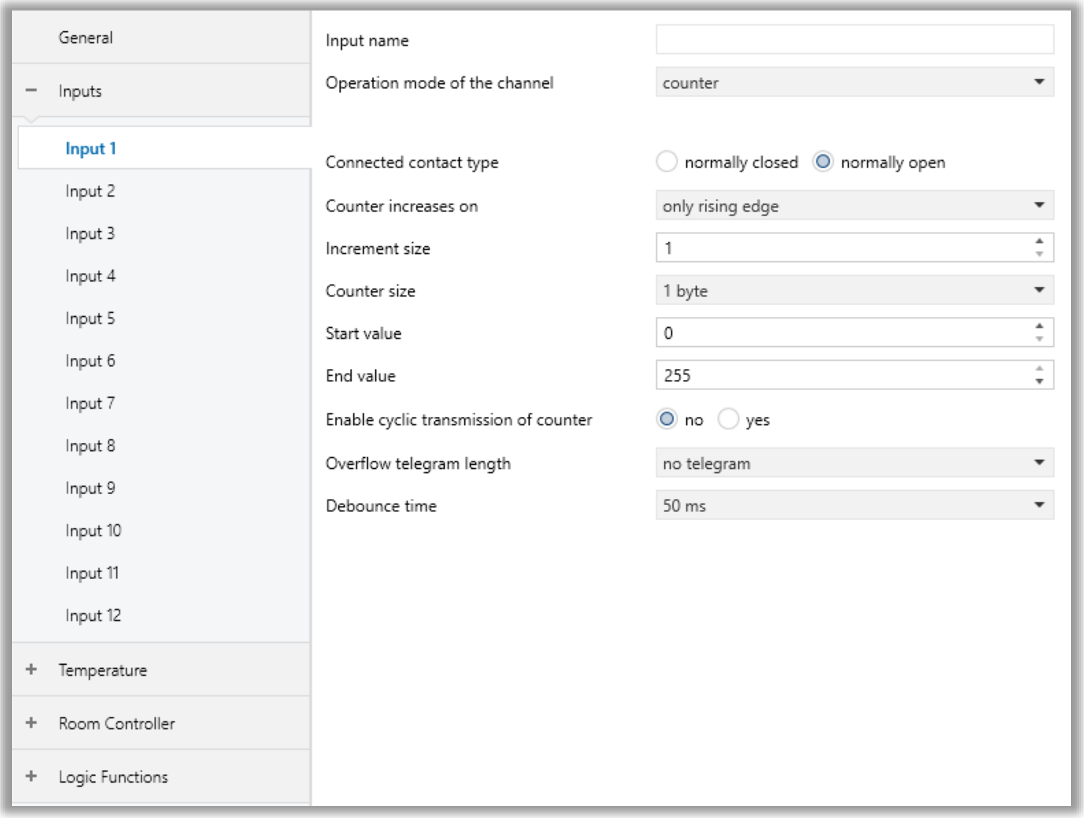

Fig. 13: Input – Counter

3.2.9.1. 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 Binary input x.	Normally closed **Normally open**
Counter increases on	This parameter is used to set how the input pulse is to be generated.	**Only rising edge** Only falling edge Both edges
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 byte** 4 byte
Start Value	This parameter is used to set the initial value of the counter after a reset or failure.	Values depends on DPT selection.
End Value	This parameter is used to set the end value of the counter.	Values depends 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	This parameter is used to determine the sending value to the bus when a short operation occurs.	00:00.005…**00:00.500** …01:05.535

Overflow telegram length	This parameter is used to set the length of the overflow telegram which will be sent to the bus when the counter value exceeds the end value set in the parameter list.	**No telegram** 1 bit 1 byte
-> Overflow telegram value	This parameter is used to determine the sending value to the bus when a short operation occurs.	Values depends 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

3.2.10. Input – RGBW control

In this section, it is explained how to control an RGBW device through the buttons connected to the digital inputs via the KNX Binary Input. Detailed information on the relevant parameter configurations is described in the table below.

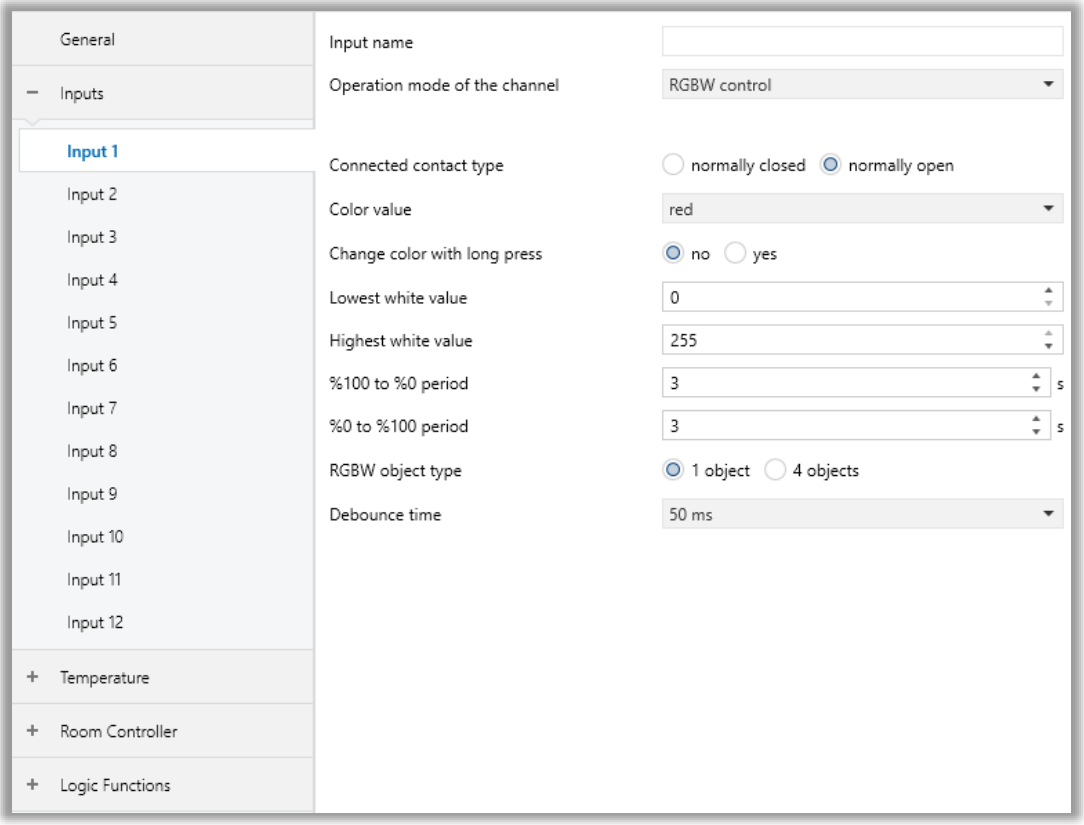

Fig. 14: Input – RGBW Control

3.2.10.1. 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 Binary 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.	00:00.005…**00:00.500** …01:05.535
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
Object type	This parameter is used to determine the RGB colour object type.	**1 object** 4 objects
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

3.3. 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 binary input temperature measurement can be made with an external NTC sensor that can be connected to its analog input, or the values obtained over the KNX bus line.

3.3.1. 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.

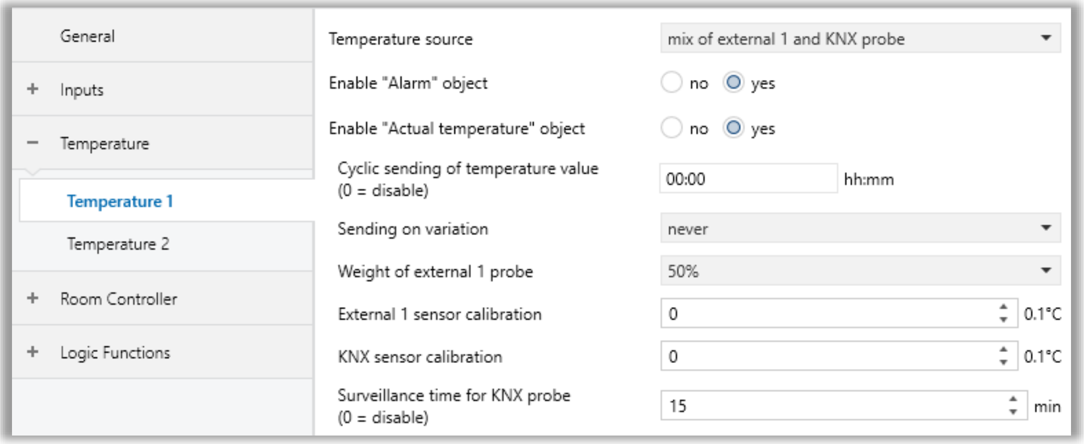

Fig. 15: Temperature Configuration Page

3.3.1.1. 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.	**External probe 1** External probe 2 Mix of external probe 1 and external probe 2 KNX probe Mix of external 1 and KNX probe Mix of external 2 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)[1]	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[1]	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 External 1/2 Probe[2]	This parameter is used to determine the weight of the external probe. E.g., the temperature source is selected as Mix of the external probe 1 and external probe 2. The external probe weight is selected as %50. So, the calculated temperature value will be: Calculated Temperature: external 1 Temperature * 0.5 + External 2 Temperature * 0.5.	%10, %20, %30, %40, **%50**, %60, %70, %80, %90
External 1/2 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[3]	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[3] **(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 “Enable “Actual temperature” object” is set to “Yes”.
[*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 “KNX probe” or “mix of external 1 and KNX probe” or “mix of external 2 and KNX probe”.

3.4. Room Controller - Thermostat

All configurations related to thermostat control on the KNX Binary Input 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.

3.4.1. 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.

3.4.1.1. 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.

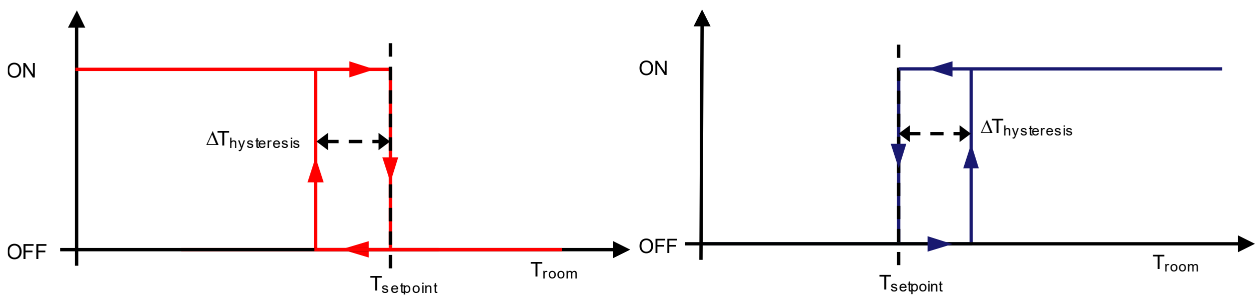

Fig. 16: 2 – Points Control Hysteresis Cycle

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 program. Values differ depending on the system type.

3.4.1.2. 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.

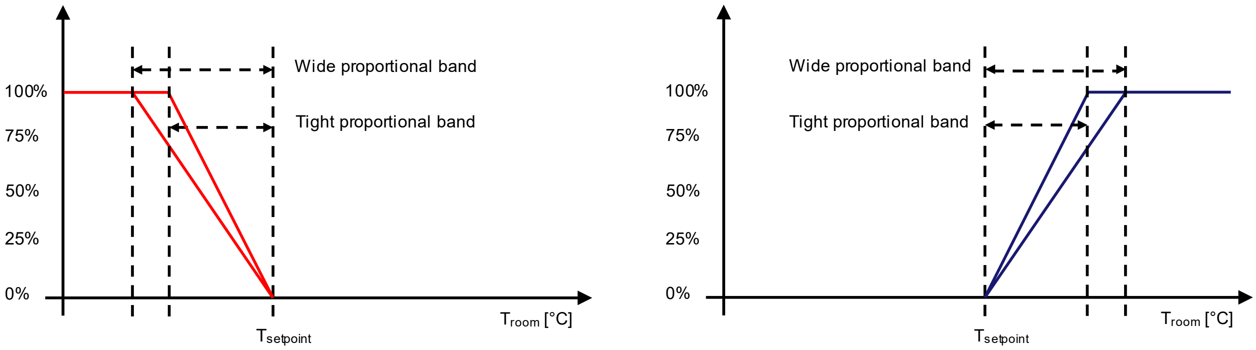

Fig. 17: Continuous PI Control Proportional Band Widths

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.

3.4.1.3. 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.

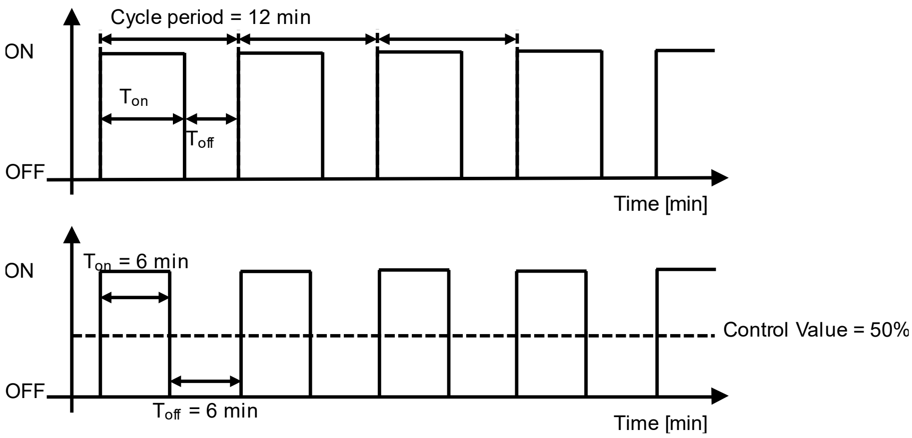

Fig. 18: PWM Control Sampling

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 3: 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.

3.4.2. Thermostat X

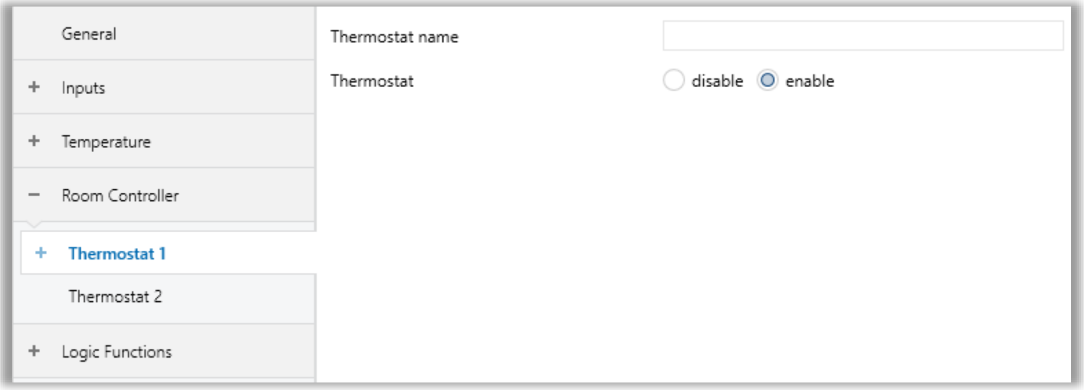

Fig. 19: Room Controller Thermostat Configuration Page

3.4.2.1. 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

3.4.3. 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.

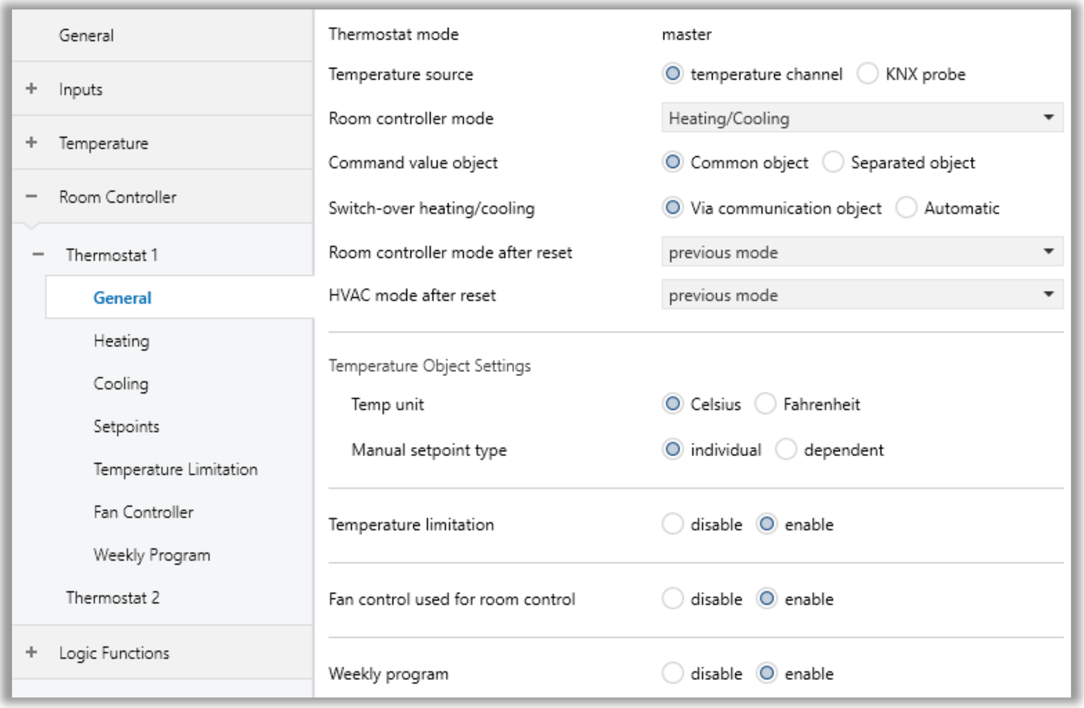

Fig. 20: Room Controller Thermostat General Configuration Page

3.4.3.1. 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 mode** Auto 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 object** Separated object
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”.

3.4.4. 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 ($T_{setpoint}$).

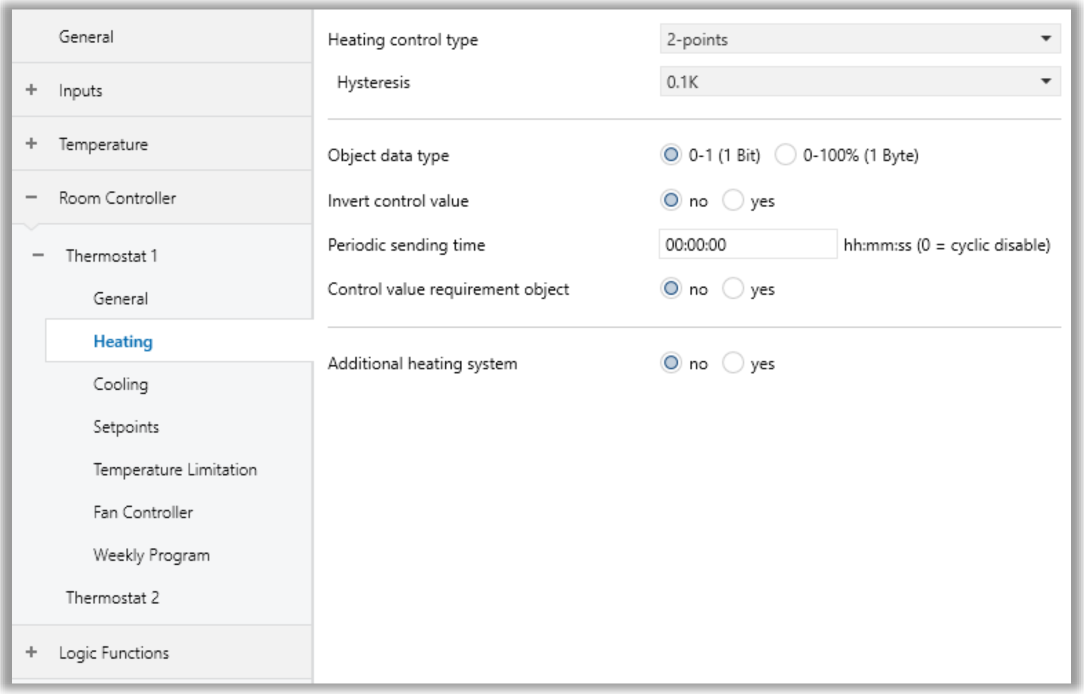

Fig. 21: Heating 2-Points Control Configuration Page

3.4.4.2. 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

3.4.4.3. 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.

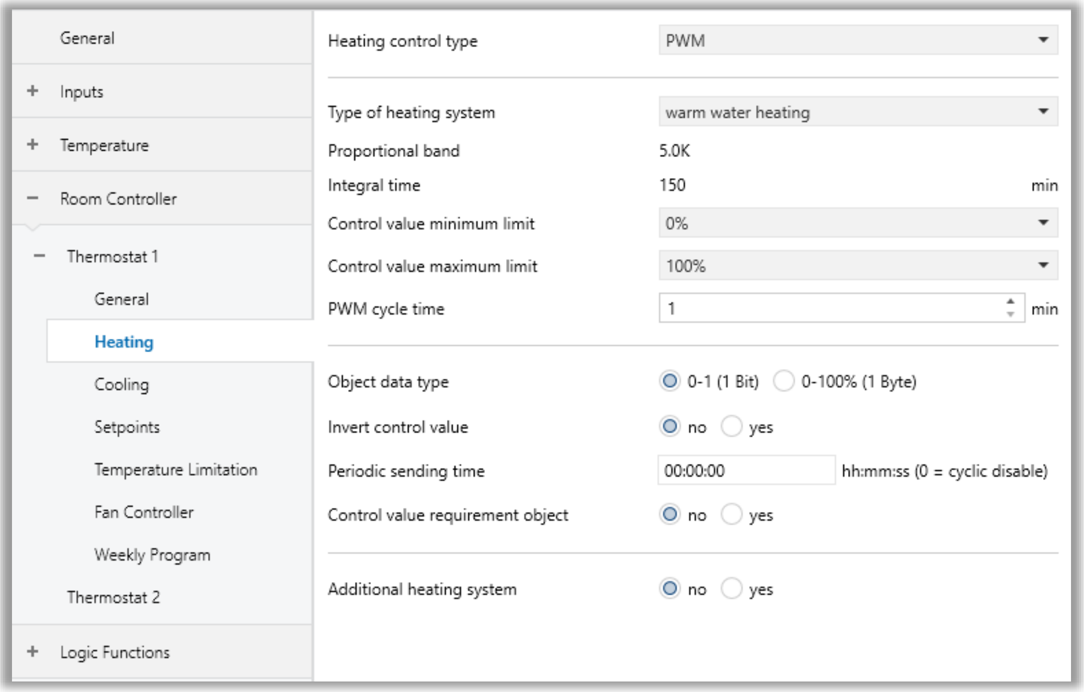

Fig. 22: Heating PWM Control Configuration Page

3.4.4.4. 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

3.4.4.5. 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%.

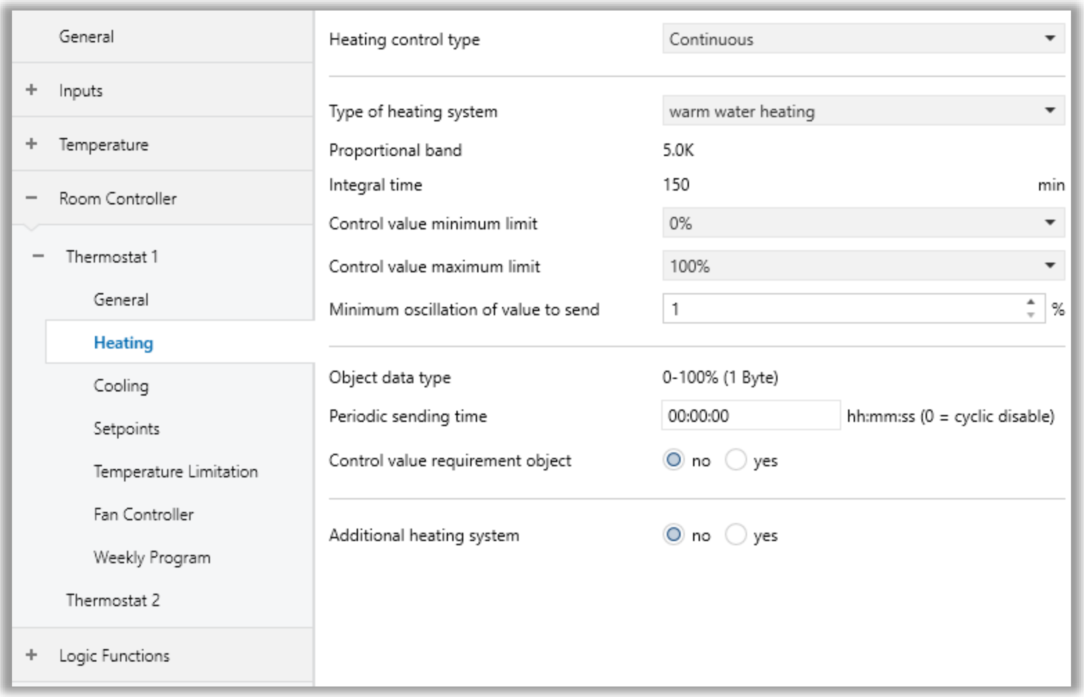

Fig. 23: Heating Continuous Control Configuration Page

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 ($T_{setpoint}$ – ΔT_{offset}) is lower than the ambient room temperature, the additional heating system will be activated according to controller type.

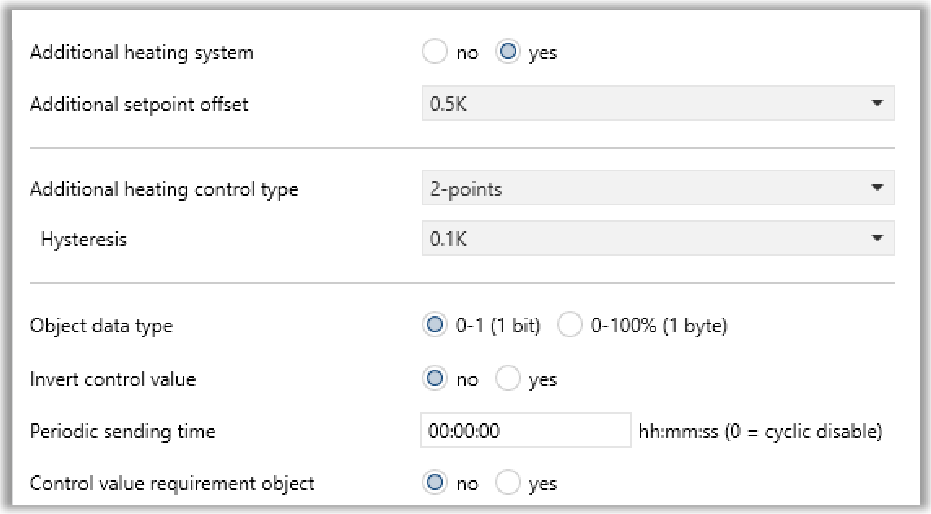

Fig. 24: Additional Heating System Configuration Page

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.

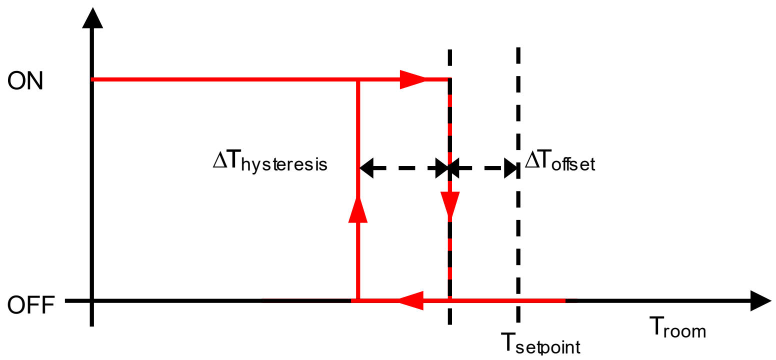

Fig. 25: 2 – Points Hysteresis Cycle for Additional Heating Control

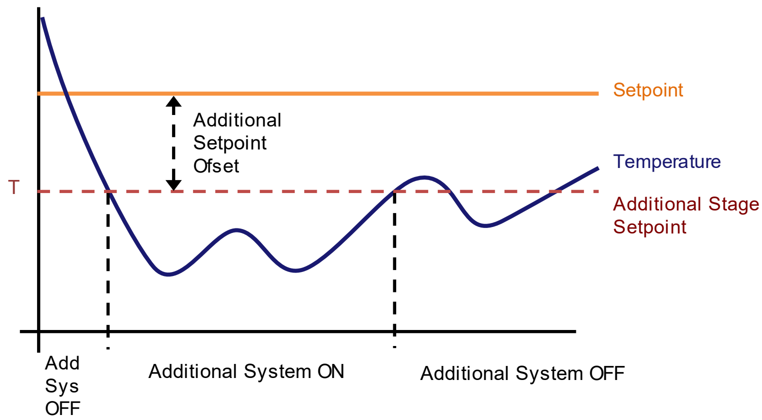

Fig. 26: PI Continuous Graph for Additional Heating Control

3.4.4.8. 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

3.4.5. 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.

3.4.5.1. 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 ($T_{setpoint}$).

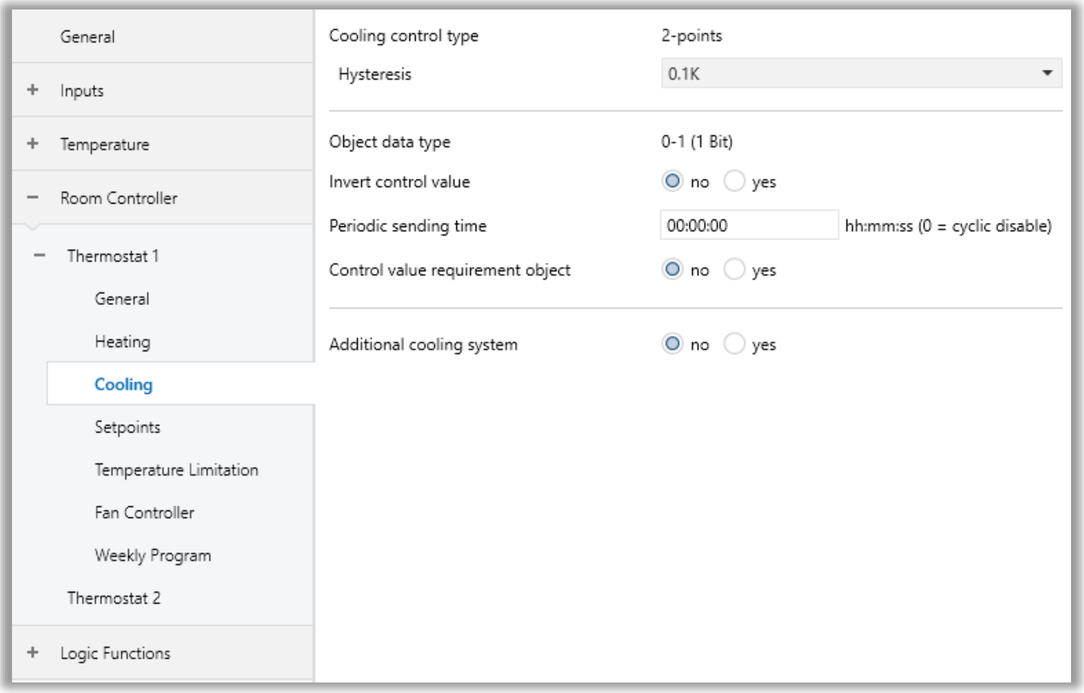

Fig. 27: Cooling 2-Points Control Configuration Page

3.4.5.2. 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

3.4.5.3. 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.

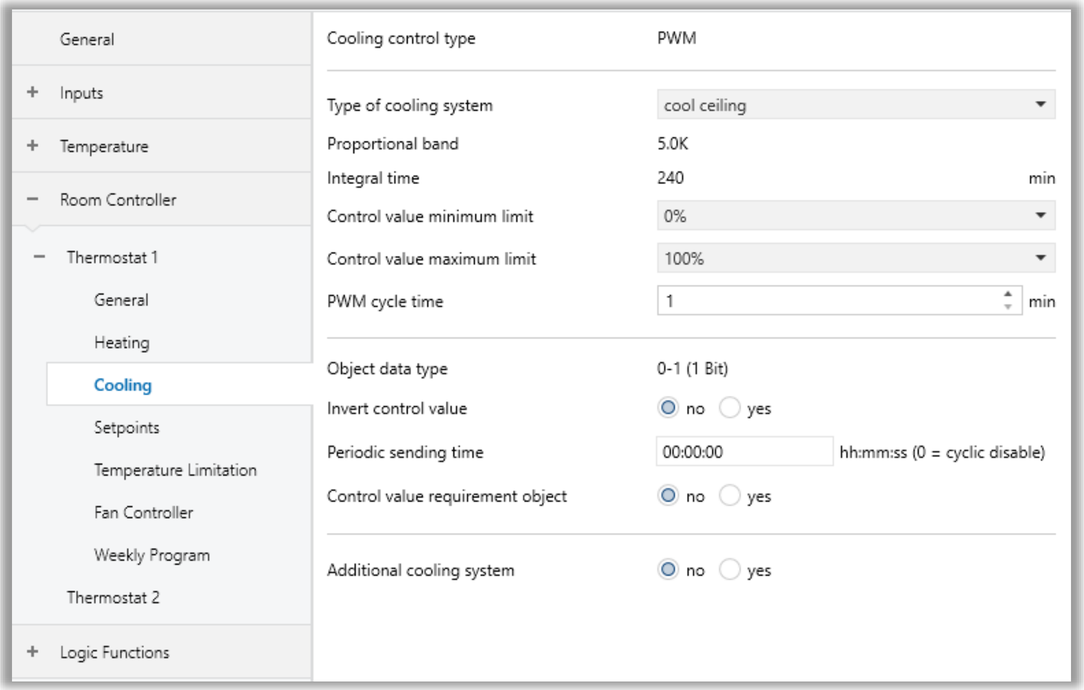

Fig. 28: Cooling PWM Control Configuration Page

3.4.5.4. 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

3.4.5.5. 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%.

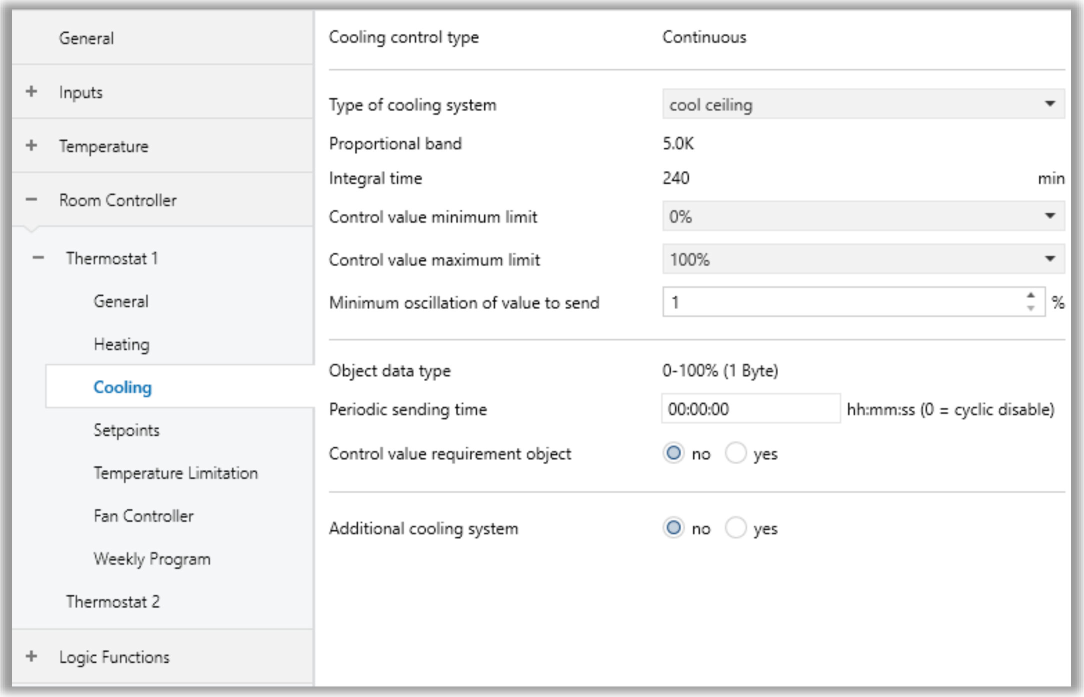

Fig. 29: Cooling Continuous Control Configuration Page

3.4.5.6. 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

3.4.5.7. 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.

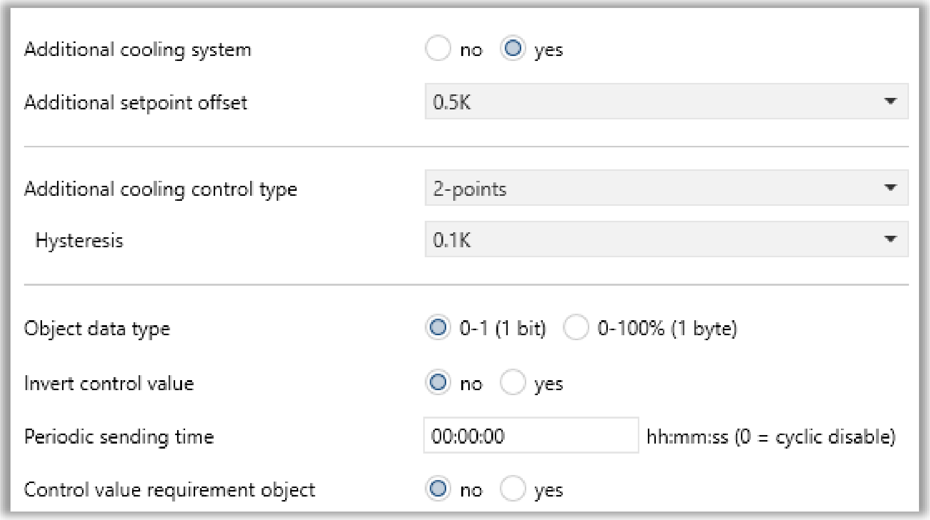

Fig. 30: Additional Cooling System Configuration Page

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.

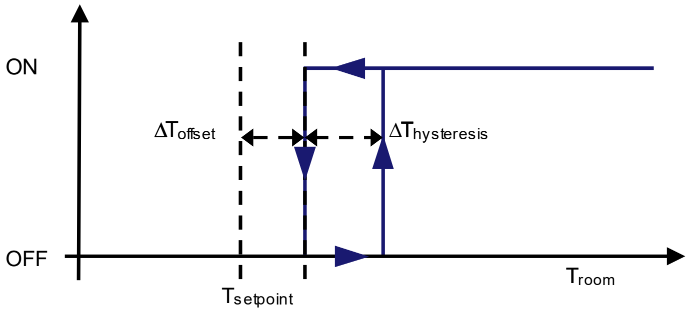

Fig. 31: 2 – Points Hysteresis Cycle for Additional Cooling Control

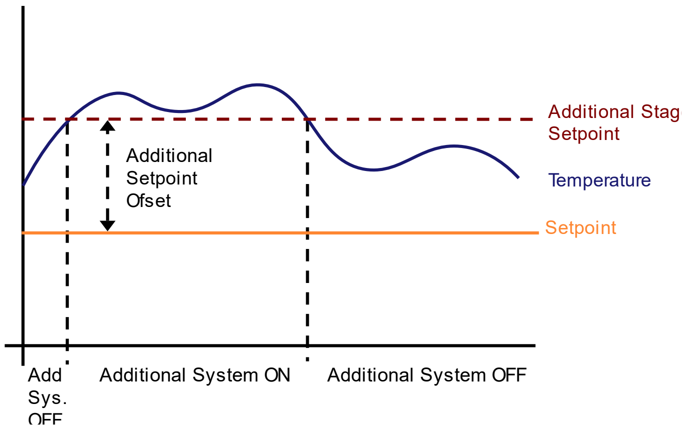

Fig. 32: PI Continuous Graph for Additional Cooling Control

3.4.5.8. 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

3.4.6. 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.

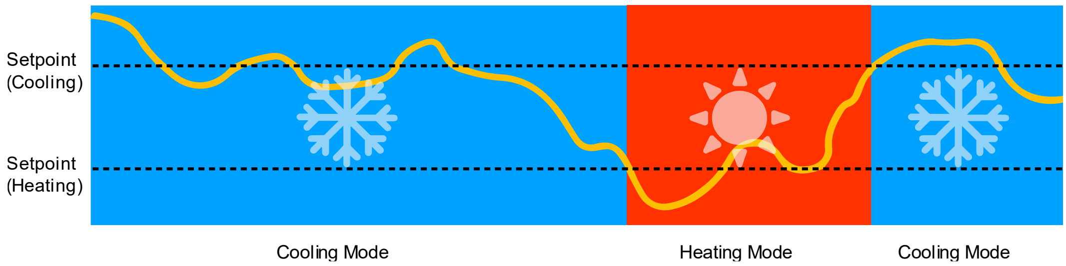

Fig. 33: Automatic Heating & Cooling Mode Switch

For a proper behavior 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.

3.4.6.1. 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

3.4.7. 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.

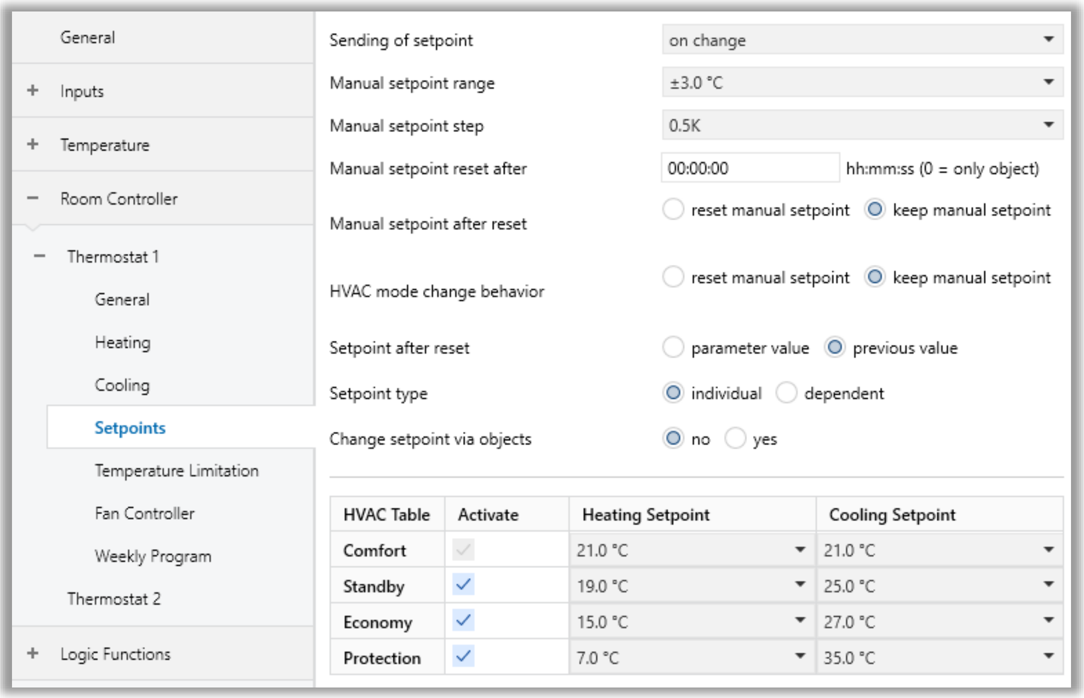

Fig. 34: Set Points Configuration Page

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.

3.4.7.1. 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”.

3.4.8. 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.

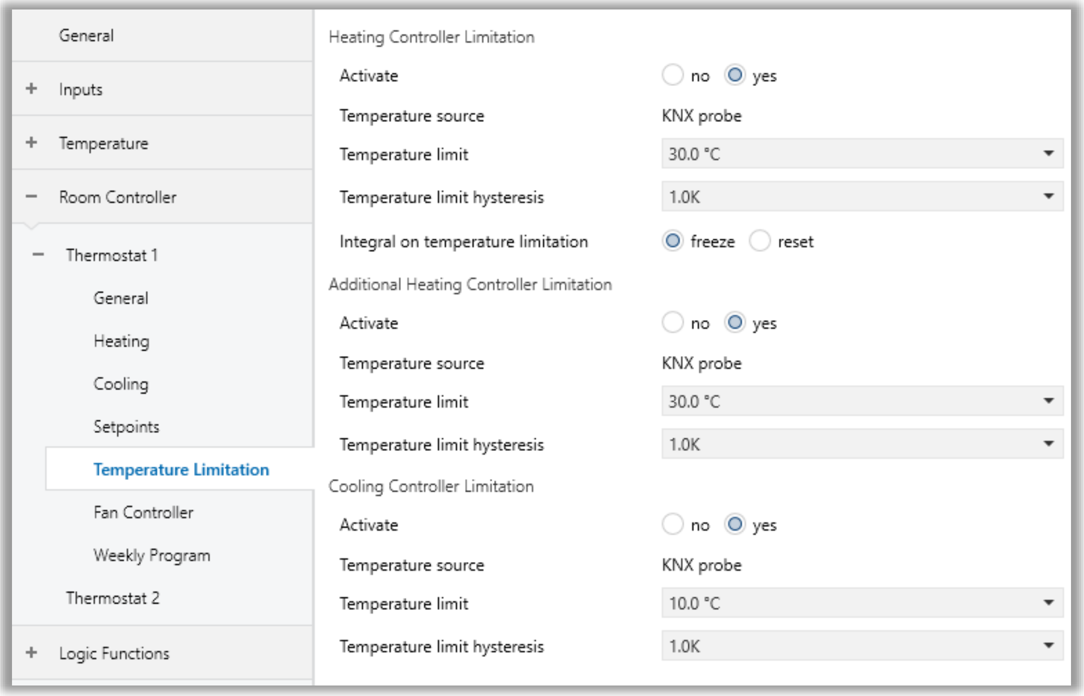

Fig. 35: Temperature Limitation Configuration Page

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".

3.4.9. 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.

3.4.9.1. 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.

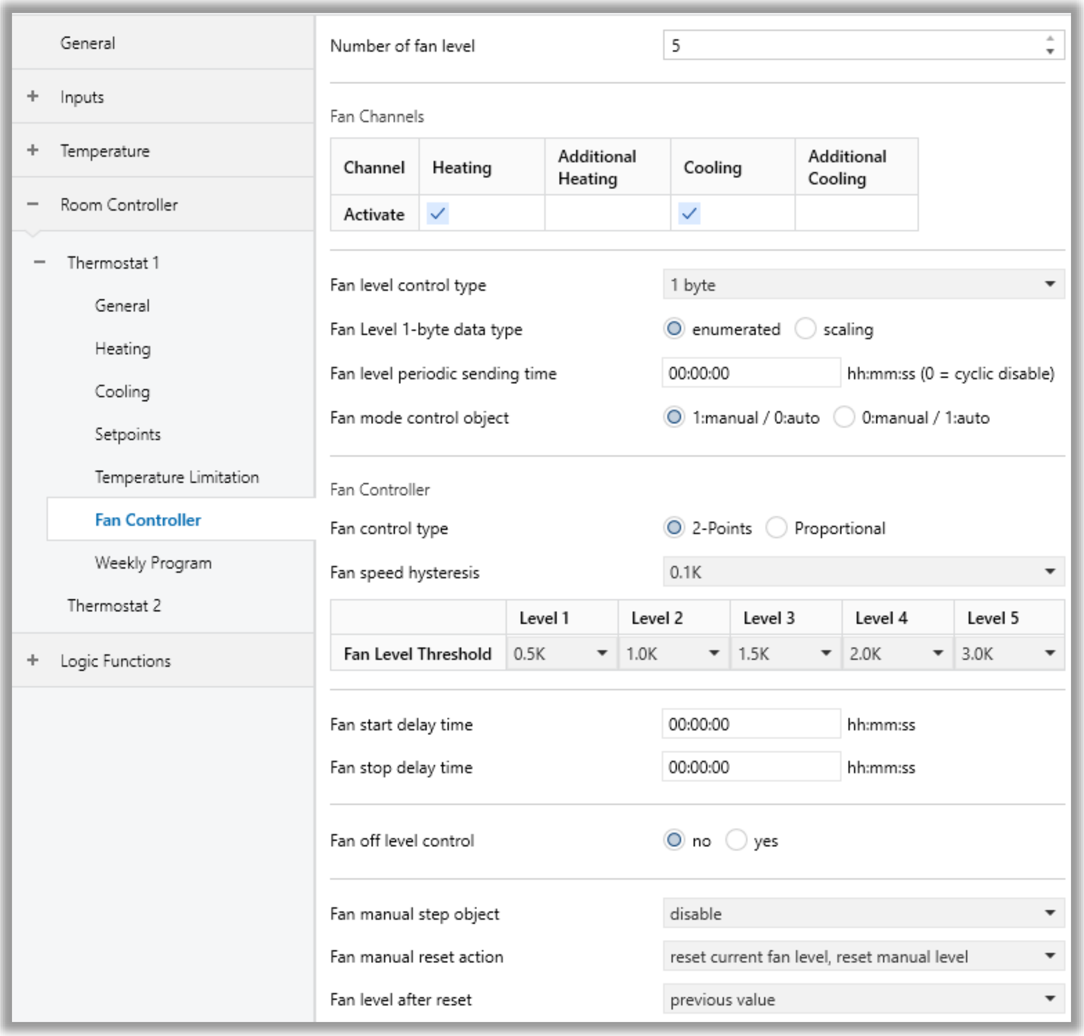

Fig. 36: Fan Controller 2-Points Control Configuration Page

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.

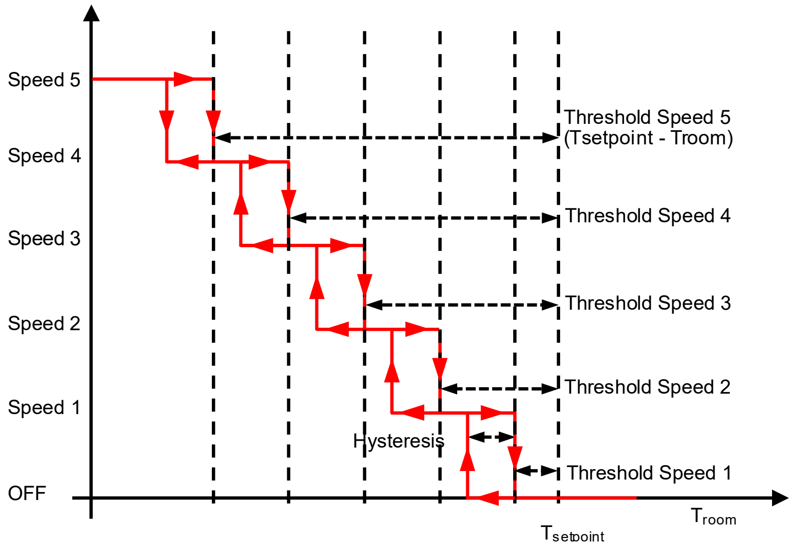

Fig. 37: Fan Controller 2-Points Control Cycle for Heating

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).

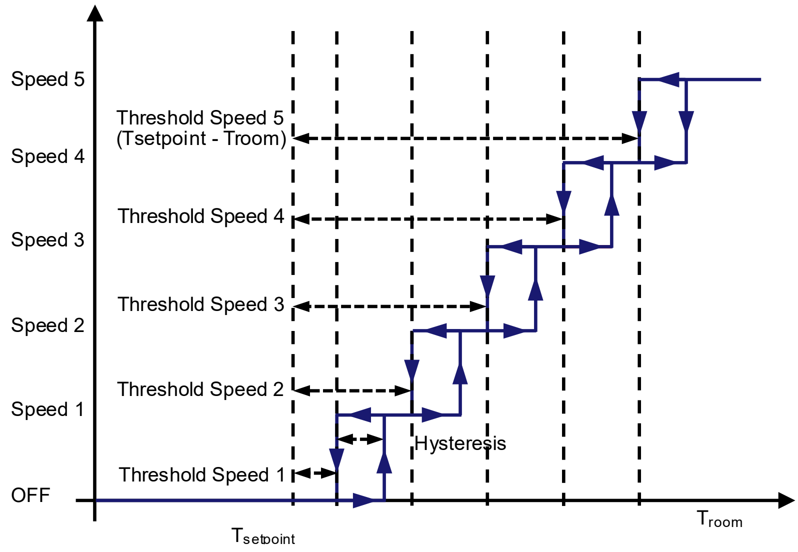

Fig. 38: Fan Controller 2-Points Control Cycle for Cooling

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.

3.4.9.2. Fan Proportional Control

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

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

whereby:

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

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

$$Kp = proportional\ constant$$

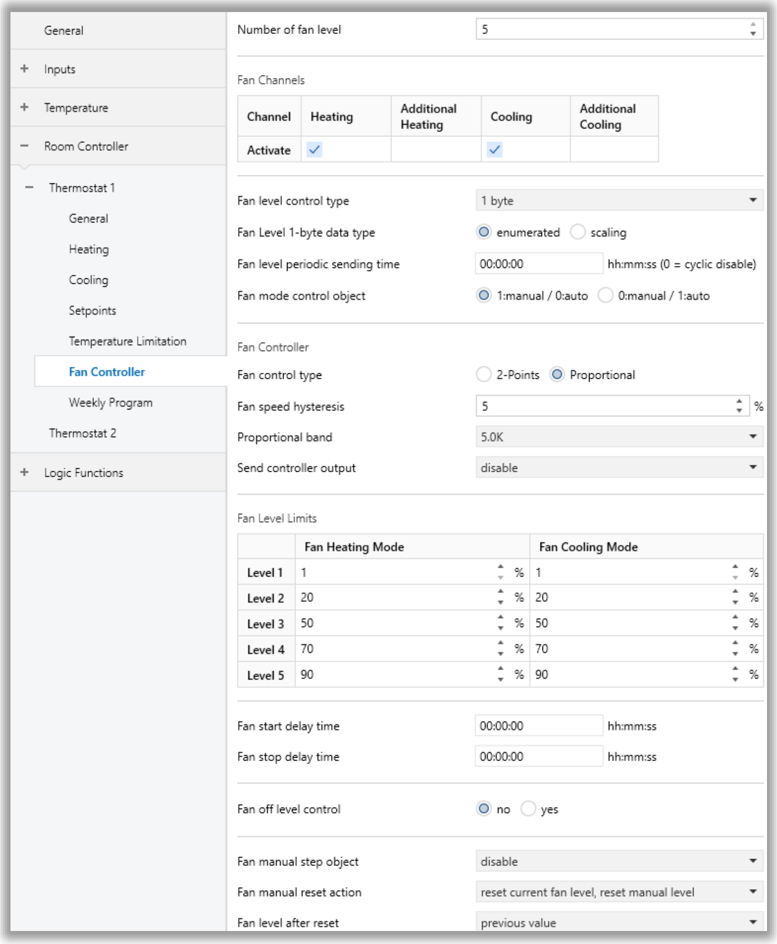

Fig. 39: Fan Controller Proportional Control Configuration Page

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

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

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.

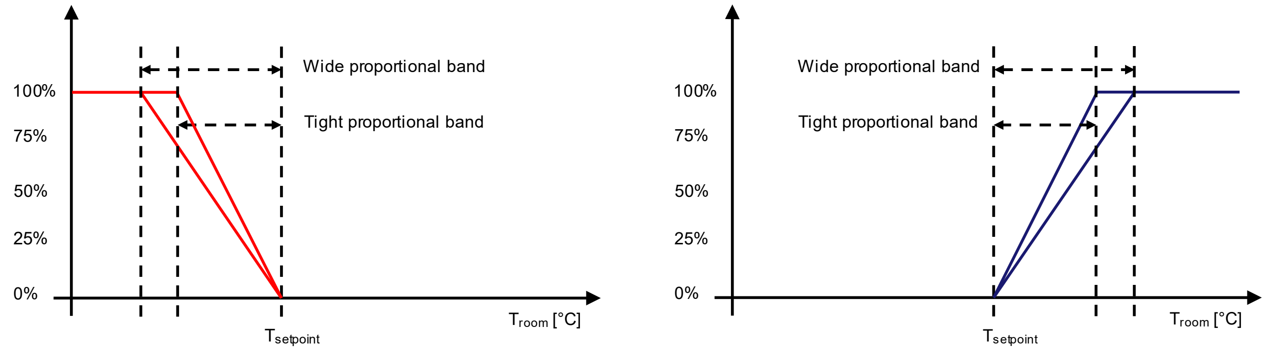

Fig. 40: Fan Controller Proportional Control

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.

3.4.9.3. 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”.

3.4.10. 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.

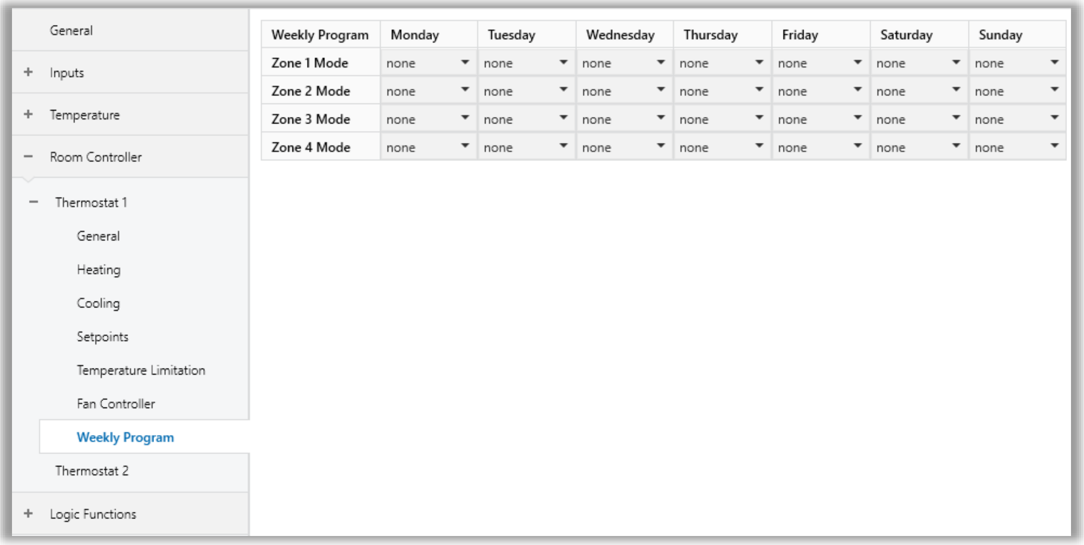

Fig. 41: Weekly Program Configuration Page

3.4.10.1. 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

3.5. Logic Channels

This section describes the logical function modules of the Interra KNX Binary Input. With the logical function blocks on the KNX Binary Input, 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.

3.5.1. Logic Channels – General

This section describes the general parameters of the logical association module of the Interra KNX Binary Input. Parameters must be configured separately for each logic block.

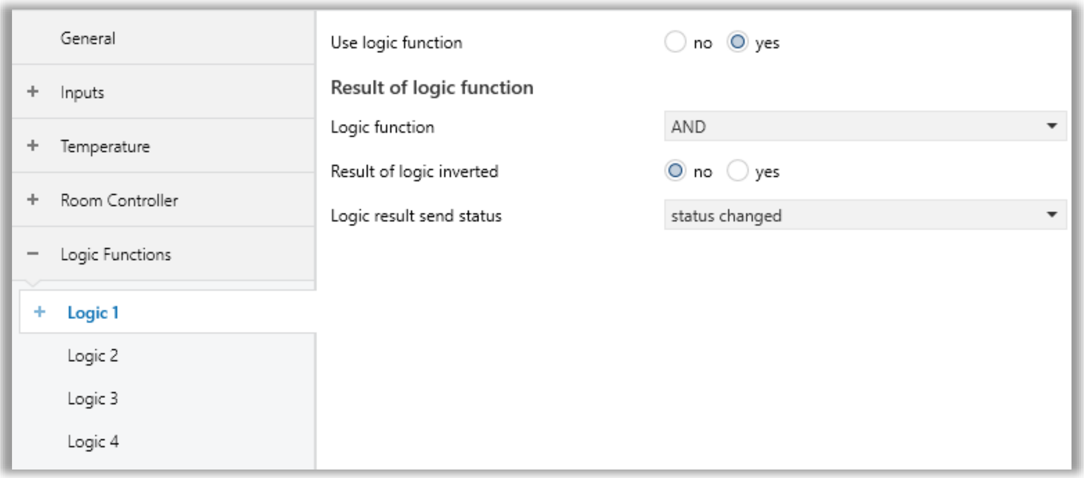

Fig. 42: Logic Functions – General

3.5.1.1. 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

3.5.2. Logic Functions – Internals Inputs

This section describes the input parameters of the logical association module of the Interra KNX Binary Input. Parameters must be configured separately for each logic block.

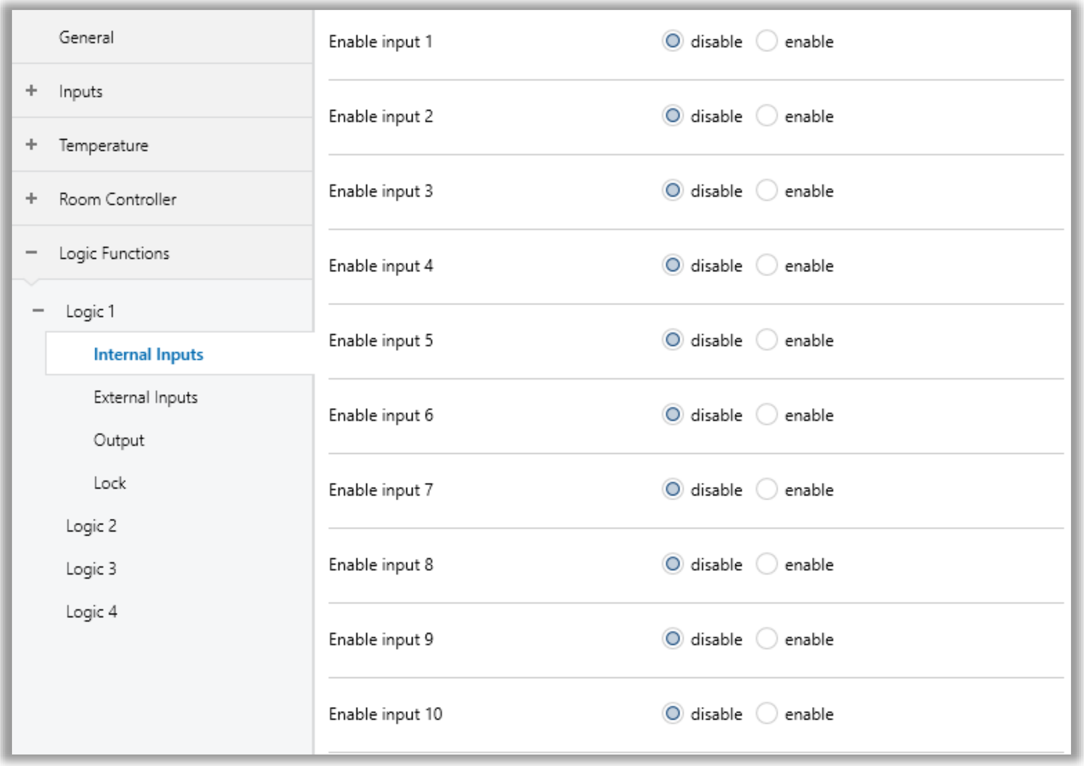

Fig. 43: Logic Functions – Internal Inputs

3.5.2.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Enable Input 1	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
->> 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
Enable Input 2	This parameter is used to enable or disable input 2 for logic function block as input	**Disable** enable
->> 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
Enable Input 3	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
->> 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
Enable Input 4	This parameter is used to enable or disable input 2 for logic function block as input	**Disable** enable
->> 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
Enable Input 5	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
->> 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
Enable Input 6	This parameter is used to enable or disable input 2 for logic function block as input	**Disable**

		enable
->> 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
Enable Input 7	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
->> 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
Enable Input 8	This parameter is used to enable or disable input 2 for logic function block as input	**Disable** enable
->> 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
Enable Input 9	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
->> 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
Enable Input 10	This parameter is used to enable or disable input 2 for logic function block as input	**Disable** enable
->> 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

3.5.3. Logic Functions – External Inputs

This section describes the external inputs parameters of the logical association module of the Interra KNX Binary Input. Parameters must be configured separately for each logic block.

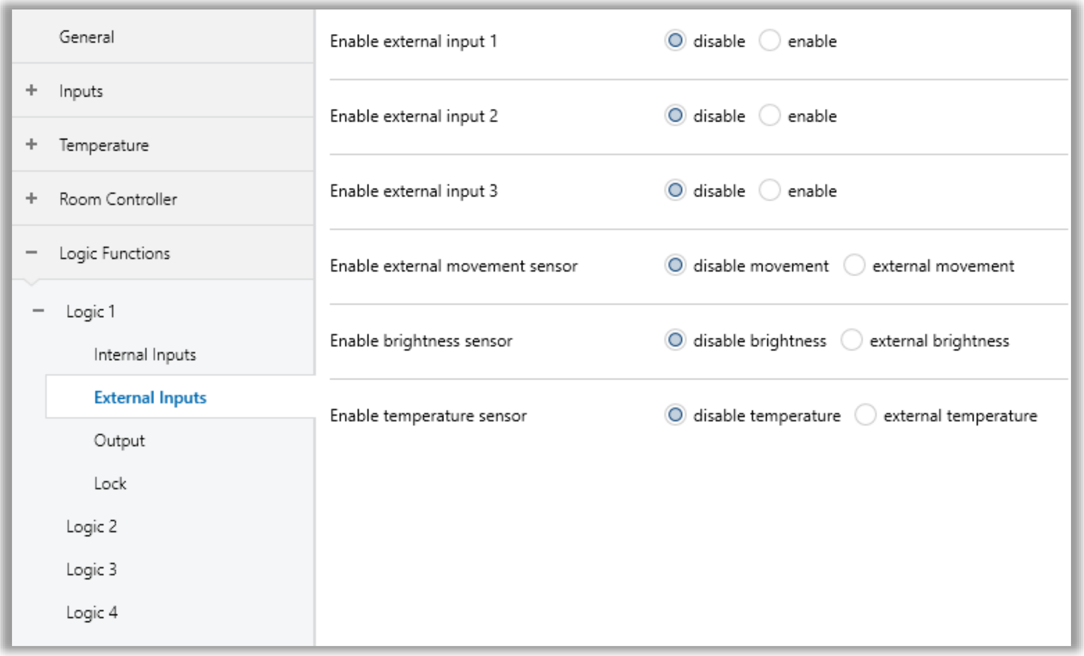

Fig. 44: Logic Functions – External Inputs

3.5.3.1. Parameters List

PARAMETERS	DESCRIPTION	VALUES
Enable External Input 1	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** Enable
->> 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 threshold (0..255) 2-byte threshold (0..65535) 2-byte float threshold (-50C..100C) 4-byte threshold (0..4294967295)
->> External Input Threshold Value	This parameter is used to determine the external input threshold value to evaluate the input status as TRUE or FALSE.	**0**…255 **0**…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** FALSE if input value <= threshold else TRUE
Enable Input 2	This parameter is used to enable or disable input 2 for logic function block as input	**Disable** enable
->> 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
Enable Input 3	This parameter is used to enable or disable input 1 for logic function block as input	**Disable** enable
->> 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

->> 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
Enable External movement	This parameter is used to enable or disable input 1 for logic function block as input	**Disable movement** Enable movement
->> External movement input is set to TRUE when received	This parameter is used to determine when a press occurs on the local input is accounted as TRUE or FALSE.	**Slave value '0'** Slave value '1'
Enable External Brightness	This parameter is used to enable or disable input 2 for logic function block as input	**Disable brightness** Enable 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 is FALSE** Out range is TRUE, else is FALSE Under lower is TRUE, above upper is FALSE Under lower is FALSE, above upper is TRUE
->> Change brightness via bus	This parameter is used to determine when a press occurs on the local input is accounted as YES or NO.	**No** Yes
Enable External Temperature	This parameter is used to enable or disable input 2 for logic function block as input	**Disable temperature** Enable temperature
->> Threshold temperature lower	This parameter is used to determine the lower threshold temperature value.	-300…**220**…700°C
->> Threshold temperature upper	This parameter is used to determine the upper threshold temperature value.	-300…**260**…700°C

->> Temperature Status	This parameter is used to determine when the ambient temperature value is accounted as TRUE or FALSE.	**In range is TRUE, else is FALSE** Out range is TRUE, else is 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 determine when a press occurs on the local input is accounted as YES or NO.	**No** Yes

3.5.4. 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.

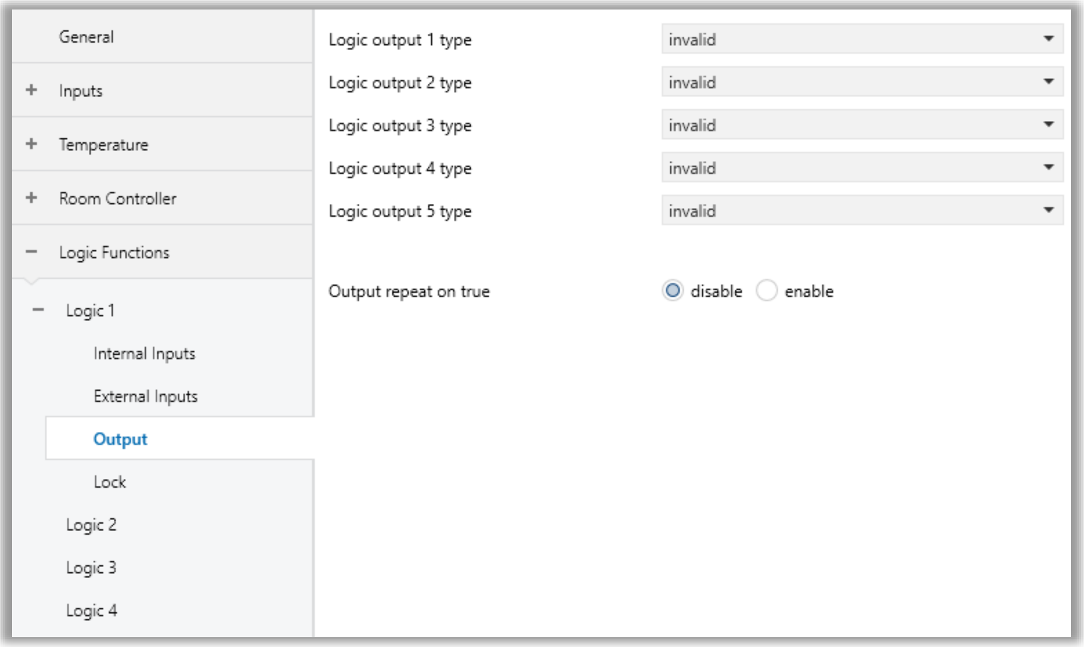

Fig. 45: Logic Functions – Output General

3.5.4.1. 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 Dim 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.	**Disable** Enable
-> 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**…65535

3.5.5. 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.

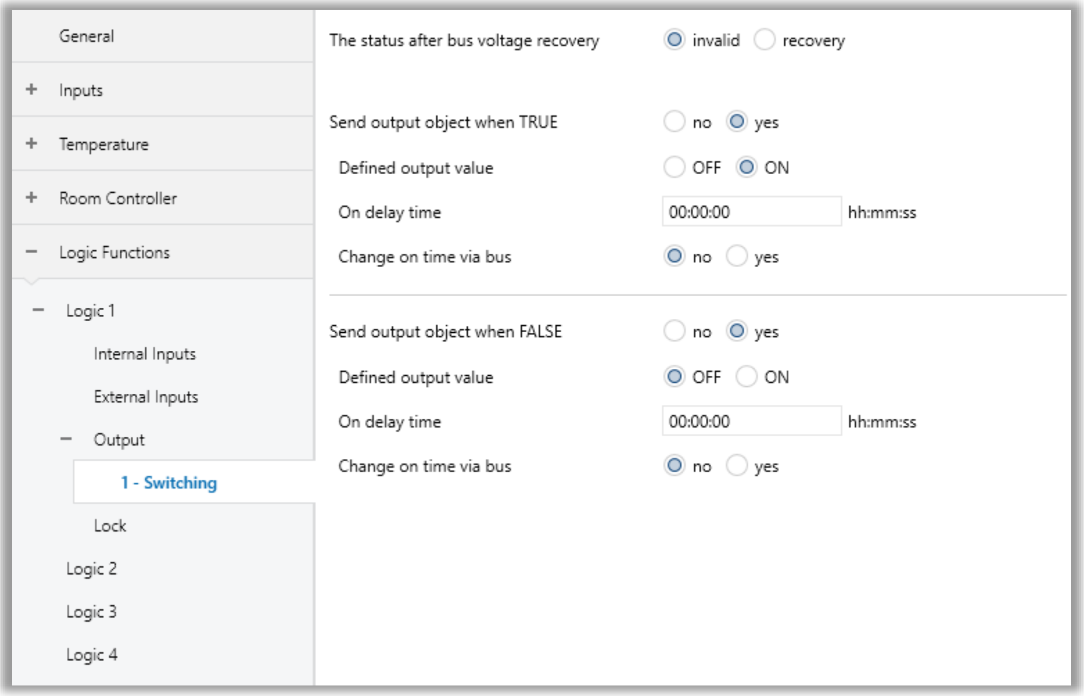

Fig. 46: Logic Functions – Output: Dimming

3.5.5.1. 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** Defined Recovery
-> 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 no.64 14 bytes string **0**…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…**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**…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

3.5.6. 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 binary input 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.

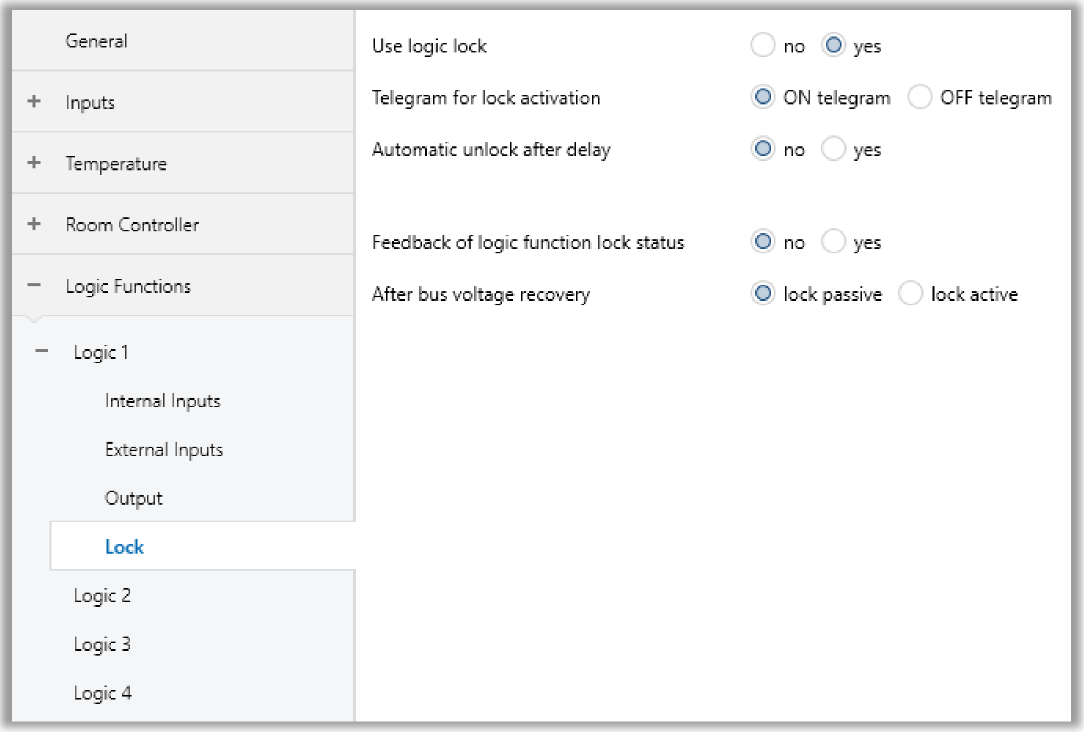

Fig. 47: Logic Functions – Lock

3.5.6.1. 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.	00:00:00…**00:00:05**... 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 KNX Binary Input can communicate via the KNX bus line. In this section, the group objects of the Interra KNX Binary Inputs are described. All of the communication objects listed below are available to the Universal Interface. Which of these group objects are visible and capable of being linked with group addresses are explained in sub-sections.

X: 1…12 / **Y:** 1…6 /**Z:** 1…2 / **T:** 6…12

No	Name	Function	DTP Type	Length	Flags				
					C	R	W	T	U
1	General	In operation	1.002	1 bit	X			X	
2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57	Input X	Block	1.003	1 bit	X		X		
3 , 8 , 13 18 , 23 , 28 33 , 38 , 43 48 , 53 , 58	Input X:	Switch	1.001	1 bit	X		X	X	
		Shutter UP/DOWN	1.008	1 bit	X		X	X	
		Forced	2.001	2 bit	X			X	
		Percent value	5.001	1 byte	X			X	
		Decimal value	5.005	1 byte	X			X	
		Scene number	17.001	1 byte	X			X	
		Colour Temperature	7.600	2 bytes	X			X	
		Temperature value	9.001	2 bytes	X			X	
		Brightness value	9.004	2 bytes	X			X	
		Percent value (RGB)	232.600	3 bytes	X			X	
		8-bit Scene	18.001	1 byte	X			X	
		RGB Colour	232.600	3 bytes	X	X		X	
		Red Colour	5.010	1 byte	X			X	
		Mode Selection	20.102	1 byte	X			X	
		Sequence	1.001	1 bit	X	X		X	
			5.001	1 byte	X	X		X	
			5.010	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)	5.001	1 byte	X	X		X	
		Sequence A (0…100%)	5.010	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	
		Percent Value (RGBW)	251.600	6 bytes	X			X	
		Red colour	5.010	1 byte	X	X		X	
	Input X	Switch - long	1.001	1 bit	X		X	X	
		Dimming	3.007	4 bit	X			X	

4 , 9 , 14 19 , 24 , 29 34 , 39 , 44 49 , 54 , 59		STOP/lamella adjustment	1.007	1 bit	X			X	
	Input X: Value/Forced op.	Forced – long	2.001	2 bit	X			X	
		Percent value – long	5.001	1 byte	X			X	
		Decimal value - long	5.005	1 byte	X			X	
		Scene number – long	17.001	1 byte	X			X	
		Colour Temperature – long	7.600	2 bytes	X			X	
		Temperature value – long	9.001	2 bytes	X			X	
		Brightness value – long	9.004	2 bytes	X			X	
		Percent value (RGB) – long	232.600	3 bytes	X			X	
	Input X	Store scene	1.003	1 bit	X			X	
		Green colour	5.010	1 byte	X			X	
		HVAC-Mode State	20.102	1 byte	X			X	
		Sequence B	1.001	1 bit	X	X		X	
		Sequence B (0…255)	5.001	1 byte	X	X		X	
		Sequence B (0…100%)	5.010	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	
		Green colour	5.010	1 byte	X			X	
5 , 10 , 15 20 , 25 , 30 35 , 40 , 45 50 , 55 , 60	Input X	Upper limit position	1.002	1 bit	X		X		
		Blue colour	5.010	1 byte	X			X	
		Sequence C	1.001	1 bit	X	X		X	
		Sequence C (0…255)	5.001	1 byte	X	X		X	
		Sequence C (0…100%)	5.010	1 byte	X	X		X	
		Sequence C HVAC	20.102	1 byte	X	X		X	
		Overflow – 1 bit	1.001	1 bit	X	X	X	X	
		Overflow – 1 byte	5.010	1 byte	X	X	X	X	
		Blue colour	5.010	1 byte	X			X	
6 , 11 , 16 21 , 26 , 31 36 , 41 , 46 51 , 56 , 61	Input X	Lower limit operation	1.002	1 bit	X		X		
		Sequence D	1.001	1 bit	X	X		X	
		Sequence D (0…255)	5.001	1 byte	X	X		X	
		Sequence D (0…100%)	5.010	1 byte	X	X		X	
		Sequence D HVAC	20.102	1 byte	X	X		X	
		White colour	5.010	1 byte	X			X	
33, 38, 43, 48, 53, 58	Input T - Analog	Temperature	9.001	2 bytes	X	X		X	
62, 65, 68, 71, 74, 77	Temperature Y	Alarm	1.005	1 bit	X	X		X	
63, 66, 69, 72, 75, 78	Temperature Y	Actual Temperature	9.001	2 bytes	X	X		X	
64, 67, 70, 73, 76, 79	Temperature Y	KNX Probe Temperature	9.001	2 bytes	X		X		
80, 151, 222, 293,364,435	Thermostat Y	Disabling	1.003	1 bit	X		X		
		Disabling	1.003	1 bit	X	X		X	

81, 152, 223, 294,365,436	Thermostat Y	Status	1.003	1 bit	X	X		X	
		Status	1.003	1 bit	X		X		
84, 155, 226, 297,368,439	Thermostat Y	Operation Mode	20.102	1 byte	X		X		
		Operation Mode	20.102	1 byte	X	X		X	
85, 156, 227, 298,369,440	Thermostat Y	Operation Mode Forced	20.102	1 byte	X		X		
86, 157, 228, 299,370,441	Thermostat Y	Operation Mode Status	20.102	1 byte	X	X		X	
		Operation Mode Feedback	20.102	1 byte	X		X		
87, 158, 229, 300,371,442	Thermostat Y	Operation Mode [Comfort]	1.001	1 bit	X	X	X		
88, 159, 230, 301,372,443	Thermostat Y	Operation Mode [Standby]	1.001	1 bit	X	X	X		
89, 160, 231, 302,373,444	Thermostat Y	Operation Mode [Economy]	1.001	1 bit	X	X	X		
90, 161, 232, 303,374,445	Thermostat Y	Operation Mode [Protection]	1.001	1 bit	X	X	X		
91, 162, 233, 304,375,446	Thermostat Y	Heating/Cooling Switchover	1.100	1 bit	X		X		
		Heating/Cooling Switchover	1.100	1 bit	X	X		X	
92, 163, 234, 305,376,447	Thermostat Y	Heating/Cooling Status	1.100	1 bit	X	X		X	
		Heating/Cooling Feedback	1.100	1 bit	X		X		
93, 164, 235, 306,377,448	Thermostat Y	Heating Control Disabling	1.001	1 bit	X		X		
94, 135, 236, 307,378,449	Thermostat Y	Heating Control Running	1.001	1 bit	X	X		X	
		Heating Control Running	1.001	1 bit	X		X		
95, 136, 237, 308,379,450	Thermostat Y	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	
96, 137, 238, 309,380,451	Thermostat Y	Heating Value Request	1.016	1 bit	X		X		
		Heating/Cooling Value Request	1.016	1 bit	X		X		
97, 138, 239, 310,381,452	Thermostat Y	Cooling Control Disabling	1.001	1 bit	X		X		
98, 139, 240, 311,382,453	Thermostat Y	Cooling Control Running	1.001	1 bit	X	X		X	
		Cooling Control Running	1.001	1 bit	X		X		
99, 140, 241, 312,383,454	Thermostat Y	Cooling Value (1-bit)	1.001	1 bit	X	X		X	
		Cooling Value (1-byte)	5.001	1 byte	X	X		X	
100,141,242 313,383,455	Thermostat Y	Cooling Value Request	1.016	1 bit	X		X		
101,142,243 314,384,456	Thermostat Y	Additional Heating Control Disabling	1.001	1 bit	X		X		
102,143,244 315,385,457	Thermostat Y	Additional Heating Control Running	1.001	1 bit	X	X		X	
103,144,245 316,386,458	Thermostat Y	Additional Heating Value(1-Bit)	1.001	1 bit	X	X		X	
		Additional Heating Value(1-Byte)	5.001	1 byte	X	X		X	

104,145,246 317,387,459	Thermostat Y	Additional Heating Value Request	1.016	1 bit	X		X		
105,146,247 318,388,460	Thermostat Y	Additional Cooling Control Disabling	1.003	1 bit	X		X		
106,147,248 319,389,461	Thermostat Y	Additional Cooling Control Running	1.002	1 bit	X	X		X	
107,148,249 320,390,462	Thermostat Y	Additional Cooling Value (1-Bit)	1.001	1 bit	X	X		X	
		Additional Cooling Value (1-Byte)	5.001	1 byte	X	X		X	
108,149,250 321,391,463	Thermostat Y	Additional Cooling Value Request	1.016	1 bit	X		X		
109,150,251 322,392,464	Thermostat Y	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		
110,181,252 323,394,465	Thermostat Y	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		
111,182,253 324,395,466	Thermostat Y	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		
112,183,254 325,396,467	Thermostat Y	Manual Setpoint Reset	1.015	1 bit	X		X		
113,184,255 326,397,468	Thermostat Y	Heating Comfort Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
114,185,256 327,398,469	Thermostat Y	Heating Standby Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
115,186,257 328,399,470	Thermostat Y	Heating Economy Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
116,187,258 329,400,471	Thermostat Y	Heating Protection Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
117,188,259 330,401,472	Thermostat Y	Cooling Comfort Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
118,189,260 331,402,473	Thermostat Y	Cooling Standby Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
119,190,261	Thermostat Y		9.001	2 bytes	X		X		

332,403,474		Cooling Economy Setpoint Temperature	9.027	2 bytes	X		X		
120,191,262 333,404,475	Thermostat Y	Cooling Protection Setpoint Temperature	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
121,192,263 334,405,476	Thermostat Y	Fan Controller Disable	1.003	1 bit	X		X		
122,193,264 335,406,477	Thermostat Y	Fan Controller Status	1.003	1 bit	X	X		X	
123,194,265 336,407,478	Thermostat Y	Fan Controller Working Mode	1.001	1 bit	X		X		
124,195,266 337,408,479	Thermostat Y	Fan Controller Working Mode Status	1.001	1 bit	X	X		X	
125,196,267 338,409,480	Thermostat Y	Fan Controller Proportional Output	5.001	1 byte	X	X		X	
126,197,268 339,410,481	Thermostat Y	Fan Controller Manual Step	1.007	1 bit	X		X		
127,198,269 340,411,482	Thermostat Y	Fan Controller Manual Up/Down	1.008	1 bit	X		X		
		Fan Controller Manual Stage	5.100	1 byte	X		X		
128,199,270 341,412,483	Thermostat Y	Fan Controller Speed (1 Byte)	5.001	1 byte	X	X		X	
		Fan Controller Speed (1 Byte)	5.100	1 byte	X	X		X	
129,200,271 342,413,484	Thermostat Y	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
130,201,272 343,414,485	Thermostat Y	Fan Level 1	1.001	1 bit	X	X		X	
131,202,273 344,415,486	Thermostat Y	Fan Level 2	1.001	1 bit	X	X		X	
132,203,274 345,416,487	Thermostat Y	Fan Level 3	1.001	1 bit	X	X		X	
133,204,275 346,417,488	Thermostat Y	Fan Level 4	1.001	1 bit	X	X		X	
134,205,276 347,418,489	Thermostat Y	Fan Level 5	1.001	1 bit	X	X		X	
135,206,277 348,419,490	Thermostat Y	Fan Level 1 Feedback Input	1.001	1 bit	X		X		X
136,207,278 349,420,491	Thermostat Y	Fan Level 2 Feedback Input	1.001	1 bit	X		X		X
137,208,279 350,421,492	Thermostat Y	Fan Level 3 Feedback Input	1.001	1 bit	X		X		X
138,209,280 351,422,493	Thermostat Y	Fan Level 4 Feedback Input	1.001	1 bit	X		X		X
139,210,281 352,423,494	Thermostat Y	Fan Level 5 Feedback Input	1.001	1 bit	X		X		X
146,217,288	Thermostat Y	Temperature Limit Heating Source	9.001	2 bytes	X		X		

359,430,501			9.027	2 bytes	X		X		
147,218,289 360,431,502	Thermostat Y	Temperature Limit Cooling Source	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
148,219,290 361,432,503	Thermostat Y	Temperature Limit Additional Heating Source	9.001	2 bytes	X		X		
			9.027	2 bytes	X		X		
149,220,291 362,433,504	Thermostat Y	Temperature Limit Additional Cooling Source	9.001	2 bytes	X		X		
150,221,292 363,434,505	Thermostat Y	Time	10.001	3 bytes	X		X		
506,534	Logic Z	Lock	1.003	1 bit	X		X		
507,535	Logic Z	Feedback of lock	1.003	1 bit	X	X		X	
508,536	Logic Z: Input	External movement	1.001	1 bit	X		X		
509,537	Logic Z: Input	External brightness	9.004	2 bytes	X		X	X	X
510,538	Logic Z: Input	Brightness threshold lower	9.004	2 bytes	X		X	X	X
511,539	Logic Y: Input	Brightness threshold upper	9.004	2 bytes	X		X	X	X
512,540	Logic Y: Input	External temperature	9.001	2 bytes	X		X		
513,541	Logic Y: Input	Temperature threshold lower	9.001	2 bytes	X		X	X	X
514,542	Logic Y: Input	Temperature threshold upper	9.001	2 bytes	X		X	X	X
515,543	Logic Y: Input: 1	External input	9.001	2 bytes	X		X		X
516,544	Logic Y: Input: 2	External input	1.001	1 bit	X		X		X
517,545	Logic Y: Input: 3	External input	1.001	1 bit	X		X		X
518,546	Logic Y: Output	Result status	1.002	1 bit	X			X	
519,547	Logic Y: Output: 1	Switching...Threshold	1.001	1 bit	X	X		X	
520,548	Logic Y: Output: 1	Delay time on TRUE state	7.005	2 bytes	X		X	X	X
521,549	Logic Y: Output: 1	Delay time on FALSE state	7.005	2 bytes	X		X	X	X
522,550	Logic Y: Output: 2	Switching...Threshold	5.001	1 byte	X	X		X	
523,551	Logic Y: Output: 2	Delay time on TRUE state	7.005	2 bytes	X		X	X	X
524,552	Logic Y: Output: 2	Delay time on FALSE state	7.005	2 bytes	X		X	X	X
525,553	Logic Y: Output: 3	Switching...Threshold	1.008	1 bit	X	X		X	
526,554	Logic Y: Output: 3	Delay time on TRUE state	7.005	2 bytes	X		X	X	X
527,555	Logic Y: Output: 3	Delay time on FALSE state	7.005	2 bytes	X		X	X	X
528,556	Logic Y: Output: 4	Switching...Threshold	1.005	1 bit	X	X		X	
529,557	Logic Y: Output: 4	Delay time on TRUE state	7.005	2 bytes	X		X	X	X
530,558	Logic Y: Output: 4	Delay time on FALSE state	7.005	2 bytes	X		X	X	X
531,559	Logic Y: Output: 5	Switching...Threshold	1.010	1 bit	X	X		X	
532,560	Logic Y: Output: 5	Delay time on TRUE state	7.005	2 bytes	X		X	X	X
533,561	Logic Y: Output: 5	Delay time on FALSE state	7.005	2 bytes	X		X	X	X

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 KNX Binary Input.

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)

4.2. Inputs

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. In this section, digital and analog input objects are described only for one channel due to their identity.

X: 1…12 / **Y:** 6…12

Object Number	Object Name	Function	Type	Flags
2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57	**Input X**	**Block**	**1 bit**	**CW**

This object is used to lock the universal interface channel. It becomes visible when the "use universal interface 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 universal interface 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)

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Switch function**	**Switch**	**1 bit**	**CT / 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)

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Switch function**	**Switch - long**	**1 bit**	**CT / 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)

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Switch/Dim function**	**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)

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Switch/Dim function**	**Dimming**	**1 bit**	**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 the actuation with START-STOP-DIMMING.

DPT: 3.007 (Dimming)

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Shutter function**	**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)

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Shutter function**	**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)

5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	**Input X: Shutter function**	**Upper limit operation**	**1 bit**	**CW**

This communication object changes in functionality depending on the selected input function. According to the input configuration on the ETS parameter page, the object usage changes. If the shutter function is selected, ‘0’ is no upper limit operation, and ‘1’ upper-end operation.

DPT: 1.002 (boolean)

6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61	**Input X: Shutter function**	**Lower limit operation**	**1 bit**	**CW**

This object is used for the shutter actuator indicate 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, and ‘1’ lower end operation.

DPT: 1.002 (boolean)

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Valued/Forced Op**	**Forced operation**	**2 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

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Control Scene**	**8-bit Scene**	**1 byte**	**CT**

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

DPT: 18.001 (scene control)

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Control Scene**	**Store Scene**	**1 bit**	**CT**

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)

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: RGB control**	**Red colour / RGB colour**	**1 byte / 3 bytes**	**CT / 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(0…255)) / 232.600 (RGB value 3x(0…255))

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: RGB control**	**Green colour**	**1 byte**	**CT**

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

DPT: 5.010 (counter pulses(0…255))

5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	**Input X: RGB control**	**Blue colour**	**1 byte**	**CT**

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

DPT: 5.010 (counter pulses(0…255))

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Mode Selection**	**Mode Selection**	**1 byte**	**CT**

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

Note: There can be up to 4 different HVAC states (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)

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Mode Selection**	**HVAC-Mode State**	**1 byte**	**CT**

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)

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Command Sequence**	**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

3, 8, 13, 18, 23, 28, 33, 38, 43, 58, 53, 58	**Input X: Command Sequence**	**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

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Command Sequence**	**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

5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	**Input X: Command Sequence**	**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

6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61	**Input : Command Sequence**	**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

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: Counter**	**Counter Value**	**1 byte / 2 bytes/ 4 bytes**	**CRT**

This object keeps the current value of the press counter.

DPT: According to parameter selection

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: Counter**	**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

5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	**Input X: Counter**	**Overflow Value**	**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(0…255))

3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58	**Input X: RGBW control**	**Red colour / Percent Value (RGBW)**	**1 byte / 6 bytes**	**CRT / CT**

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: 5.010 (counter pulses(0…255)) / 251.600 (RGBW value 4x(0..100%))

4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59	**Input X: RGBW control**	**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(0…255))

5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	**Input X: RGBW control**	**Blue colour**	**1 byte**	**CT**

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

DPT: 5.010 (counter pulses(0…255))

6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61	**Input X: RGBW control**	**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(0…255))

33, 38, 43, 48, 53, 58	**Input Y - Analog**	**Temperature**	**1 byte**	**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))

4.3. 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.

X: 1…6

Object Number	Object Name	Function	Type	Flags
62, 65, 68, 71, 74, 77	**Temperature X**	**Alarm**	**1 bit**	**CRT**

This object is used to send the alarm temperature value calculated by the KNX Binary Input 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)

63, 66, 69, 72, 75, 78	**Temperature X**	**Actual Temperature**	**2 bytes**	**CRT**

This object is used to send the actual temperature value calculated by the KNX Binary Input 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))

64, 67, 70, 73, 76, 79	**Temperature X**	**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 KNX Binary Input.

DPT: 9.001 (temperature (°C))

4.4. 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.

X: 1 … 6

Object Number	Object Name	Function	Type	Flags
80, 151, 222, 293, 364, 435	Thermostat X	Thermostat Disabling	1 bit	CW / CRT*

This object is used to set the KNX Binary Input 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 KNX Binary Input thermostat will continue working.

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

DPT: 1.003 (enable)

81, 152, 223, 294, 365, 436	Thermostat X	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)

84, 155, 226, 297, 368, 439	Thermostat X	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)

85, 156, 227, 298, 369, 440	Thermostat X	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)

86, 157, 228, 299, 370, 441	**Thermostat X**	**Thermostat Operation Mode Status/Feedback**	**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)

87, 158, 229, 300, 371, 442	**Thermostat X**	**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)

88, 159, 230, 301, 372, 443	**Thermostat X**	**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)

89, 160, 231, 302, 373, 444	**Thermostat X**	**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)

90, 161, 232, 303, 374, 445	**Thermostat X**	**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)

91, 162, 233, 304, 375, 446	Thermostat X	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)

92, 163, 234, 305, 376, 447	Thermostat X	Thermostat Heating/Cooling Status/Feedback	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)

93, 164, 235, 306, 377, 448	Thermostat X	Thermostat Heating Control Disabling	1 bit	CW

This object activates or deactivates the heating system.

DPT: 1.001 (switch)

94, 135, 236, 307, 378, 449	Thermostat X	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)

95, 136, 237, 308, 379, 450	Thermostat X	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)

96, 137, 238, 309, 380, 451	Thermostat X	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)

97, 138, 239, 310, 381, 452	**Thermostat X**	**Thermostat Cooling Control Disabling**	**1 bit**	**CW**

This object activates or deactivates the cooling system.

DPT: 1.001 (switch)

98, 139, 240, 311, 382, 453	**Thermostat X**	**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: 1.001 (switch)

99, 140, 241, 312, 383, 454	**Thermostat X**	**Cooling Value**	**1 bit / 1 byte**	**CRT**

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

DPT: 1.001 (switch) / 5.001 (percentage (0...255%))

100, 141, 242, 313, 383, 455	**Thermostat X**	**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)

101, 142, 243, 314, 384, 456	**Thermostat X**	**Additional Heating Control Disabling**	**1 bit**	**CW**

This object activates or deactivates the additional heating system.

DPT: 1.001 (switch)

102, 143, 244, 315, 385, 457	**Thermostat X**	**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)

103, 144, 245, 316, 386, 458	**Thermostat X**	**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)

104, 145, 246, 317, 387, 459	**Thermostat X**	**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)

105, 146, 247, 318, 388, 460	**Thermostat X**	**Thermostat Additional Cooling Control Disabling**	**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: 1.003 (enable)

106, 147, 248, 319, 389, 461	**Thermostat X**	**Thermostat Additional Cooling Control Running**	**1 bit**	**CRT**

This object activates or deactivates the additional cooling system.

DPT: 1.002 (Boolean)

107, 148, 249, 320, 390, 462	**Thermostat X**	**Thermostat Additional Cooling Value**	**1 bit / 1 byte**	**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: 1.001 (switch) / 5.001 (percentage (0..100%))

108, 149, 250, 321, 391, 463	**Thermostat X**	**Thermostat Additional Cooling Value Request**	**1 bit**	**CW**

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

DPT: 1.016 (acknowledge)

109, 150, 251, 322, 392, 464	**Thermostat X**	**Room Temperature Output (C°)** **Room Temperature Input (C°)** **Room Temperature Output (F°)** **Room Temperature Input (F°)**	**2 bytes**	**CRW / 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: 9.001 (temperature (C°)) / 9.027 (temperature (C°))

110, 181, 252, 323, 394, 465	**Thermostat X**	**Thermostat Actual Setpoint Output**	**2 bytes**	**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)) / 9.002 (temperature difference (K)) / 9.027 (temperature difference (K))

111, 182, 253, 324, 395, 466	**Thermostat X**	**Thermostat Manual Setpoint Input**	**2 bytes**	**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)) / 9.002 (temperature difference (K)) / 9.027 (temperature difference (K))

112, 183, 254, 325, 396, 467	**Thermostat X**	**Thermostat Manual Setpoint Reset**	**1 bit**	**CW**

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

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

DPT: 1.015 (reset)

113, 184, 255, 326, 397, 468	**Thermostat X**	**Thermostat Heating Comfort Setpoint Temperature**	**2 bytes**	**CW**

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

114, 185, 256, 327, 398, 469	**Thermostat X**	**Thermostat Heating Standby Setpoint Temperature**	**2 bytes**	**CW**

The setpoint temperature that is desired to configure manually can be reset with this object.
DPT: 9.001 (temperature (°C)) / 9.027 (temperature difference (K))

115, 186, 257, 328, 399, 470	**Thermostat X**	**Thermostat Heating Economy 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))

116, 187, 258, 329, 400, 471	**Thermostat X**	**Thermostat Heating Protection 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))

117, 188, 259, 330, 401, 472	**Thermostat X**	**Thermostat Cooling Comfort 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))

118, 189, 260, 331, 402, 473	**Thermostat X**	**Thermostat Cooling Standby 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))

119, 190, 261, 332, 403, 474	**Thermostat X**	**Thermostat Cooling Economy 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))

120, 191, 262, 333, 404, 475	**Thermostat X**	**Thermostat Cooling Protection 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))

121, 192, 263, 334, 405, 476	**Thermostat X**	**Thermostat Fan Controller Disable**	**1 bit**	**CW**

The setpoint temperature value for cooling economy mode is configured with this object.
DPT: 1.003 (enable)

122, 193, 264, 335, 406, 477	**Thermostat X**	**Thermostat Fan Controller Status**	**1 bit**	**CRT**

The setpoint temperature value for cooling protection mode is configured with this object.
DPT: 1.003 (enable)

123, 194, 265, 336, 407, 478	**Thermostat X**	**Fan Controller Working Mode**	**1 bit**	**CW**

This object is used to set the KNX Binary Input 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 KNX Binary Input fan controller will continue working.

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

DPT: 1.001 (switch)

124, 195, 266, 337, 408, 479	**Thermostat X**	**Fan Controller Working Mode 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: 1.001 (switch)

125, 196, 267, 338, 409, 480	**Thermostat X**	**Fan Controller Proportional Output**	**1 byte**	**CRT**

This object is used to switch over to automatic or manual fan speed control mode.
DPT: 5.001 (percentage (0..100%))

126, 197, 268, 339, 410, 481	**Thermostat X**	**Fan Controller Manual Step**	**1 bit**	**CW**

This object indicates the manual / automatic fan operating mode with 1 bit value.
DPT: 1.007 (up/down)

127, 198, 269, 340, 411, 482	**Thermostat X**	**Fan Controller Manual Up/Down** **Fan Controller Manual Stage**	**1 bit /** **1 byte**	**CW**

This object is used to send the output value of the fan proportional controller.
DPT: 1.008 (up / down) / 5.100 (fan stage (0..255))

128, 199, 270, 341, 412, 483	**Thermostat X**	**Fan Controller Speed**	**1 byte**	**CRT**

This object is used to increase or decrease the fan speed
DPT: 5.001 ((percentage (0..100%) / 5.100 (fan stage (0...255)

129, 200, 271, 342, 413, 484	**Thermostat X**	**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))

130, 201, 272, 343, 414, 485	**Thermostat X**	**Fan Level 1**	**1 bit**	**CRT**

This object indicates the Fan Level 1 value with a 1-bit value.
DPT: 1.001 (switch)

131, 202, 273, 344, 415, 486	**Thermostat X**	**Fan Level 2**	**1 bit**	**CRT**

This object indicates the Fan Level 2 value with a 1-bit value.
DPT: 1.001 (switch)

132, 203, 274, 345, 416, 487	**Thermostat X**	**Fan Level 3**	**1 bit**	**CRT**

This object indicates the Fan Level 3 value with a 1-bit value.
DPT: 1.001 (switch)

133, 204, 275, 346, 417, 488	**Thermostat X**	**Fan Level 4**	**1 bit**	**CRT**

This object indicates the Fan Level 4 status with a 1-bit value.
DPT: 1.001 (switch)

134, 205, 276, 347, 418, 489	**Thermostat X**	**Fan Level 5**	**1 bit**	**CRT**

This object indicates the Fan Level 5 status with a 1-bit value.
DPT: 1.001 (switch)

135, 206, 277, 348, 419, 490	**Thermostat X**	**Fan Level 1 Feedback Input**	**1 bit**	**CWU**

This object indicates the Fan Level 1 status with a 1-bit value.
DPT: 1.001 (switch)

136, 207, 278, 349, 420, 491	**Thermostat X**	**Fan Level 2 Feedback Input**	**1 bit**	**CWU**

This object indicates the Fan Level 2 status with a 1-bit value.
DPT: 1.001 (switch)

137, 208, 279, 350, 421, 492	**Thermostat X**	**Fan Level 3 Feedback Input**	**1 bit**	**CWU**

This object indicates the Fan Level 3 status with a 1-bit value.
DPT: 1.001 (switch)

138, 209, 280, 351, 422, 493	**Thermostat X**	**Fan Level 4 Feedback Input**	**1 bit**	**CWU**

This object indicates the Fan Level 4 status with a 1-bit value.
DPT: 1.001 (switch)

139, 210, 281, 352, 423, 494	**Thermostat X**	**Fan Level 5 Feedback Input**	**1 bit**	**CWU**

This object indicates the Fan Level 4 status with a 1-bit value.
DPT: 1.001 (switch)

146, 217, 288, 359, 430, 501	**Thermostat X**	**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))

147, 218, 289, 360, 431, 502	**Thermostat X**	**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))

148, 219, 290, 361, 432, 503	**Thermostat X**	**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))

149, 220, 291, 362, 433, 504	**Thermostat X**	**Temperature Limit Additional Cooling Source**	**1 bit**	**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 exceeded.

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

150, 221, 292, 363, 434, 505	**Thermostat X**	**Time**	**3 bytes**	**CW**

This object is used to set date and time. Date and time are used thermostat weekly program. If weekly program is active but any telegram hasn't received over "Time" object yet, the weekly program doesn't run.

DPT: 10.001 (time of day)

4.5. 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 12 identical logic channels in the KNX Binary Input, so only one logical channel is described here. The X values can be between 1…2 and Y values also can be 1…5. Please do not forget to take this into account.

Object Number	Object Name	Function	Type	Flags
506 , 534	**Logic X:**	**Lock function**	**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)

507, 535	**Logic X:**	**Feedback of block**	**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 status value will be sent from this object.

DPT: 1.003 (enable)

508, 536	**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)

509, 537	**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)

510, 538	**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)

511, 539	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)

512, 540	Logic X: Input	External temperature	2 bytes	CWTU

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))

513, 541	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))

514, 542	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))

515, 543	Logic X: Input:1	External input-1	1 bit / 1 byte / 2 bytes/ 4 bytes	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 are only ‘1’ or ‘0’ values for calculating the input status. But for other inputs (such as 1 byte, etc.) the received value is compared to the external input value parameter.

DPT: According to parameter selection, DPT changes.

516, 544	**Logic X: Input:2**	**External input-2**	**1 bit / 1 byte / 2 bytes/ 4 bytes**	**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 are only '1' or '0' values for calculating the input status. But for other inputs (such as 1 byte, etc.) the received value is compared to the external input value parameter.

DPT: According to parameter selection, DPT changes.

517, 545	**Logic X: Input:3**	**External input-3**	**1 bit / 1 byte / 2 bytes/ 4 bytes**	**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 are only '1' or '0' values for calculating the input status. But for other inputs (such as 1 byte, etc.) the received value is compared to the external input value parameter.

DPT: According to parameter selection, DPT changes.

518, 546	**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)

519, 522, 525, 528, 531	**Logic 1: Output: Y**	**Switching…Threshold**	**1 bit / 1 byte / 2 bytes/ 14bytes**	**CRT**

This object is used to send the related output object's value to the KNX bus line. When the logic function block's status changes, the sending value also can be configured separately. In addition, according to the output type, the object's value type will be changed.

DPT: According to parameter selection, DPT changes.

520, 523, 526, 529, 532	**Logic 1: Output: Y**	**Delay time on the TRUE state**	**2 bytes**	**CWTU**

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))

521, 524, 527, 530, 533	**Logic 1: Output: Y**	**Delay time on FALSE state**	**2 bytes**	**CWTU**

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))

547, 550, 553, 556, 559	**Logic 2: Output: Y**	**Switching…Threshold**	**1 bit / 1 byte / 2 bytes/ 14bytes**	**CRT**

This object is used to send the related output object's value to the KNX bus line. When the logic function block's status changes, the sending value also can be configured separately. In addition, according to the output type, the object's value type will be changed.

DPT: According to parameter selection, DPT changes.

548, 551, 554, 557, 560	**Logic 2: Output: Y**	**Delay time on TRUE state**	**2 bytes**	**CWTU**

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))

549, 552, 555, 558, 561	**Logic 2: Output: Y**	**Delay time on FALSE State**	**2 bytes**	**CWTU**

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 documentation support via our website 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 Center No:95/1 Kartal/İstanbul
Tel: +90 (216) 326 26 40 Fax: +90 (216) 324 25 03
Web adress: http://www.interratechnology.com